Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Αξιοποιώντας τα Δημόσια Ανοικτά Δεδομένα της Πυροσβεστικής: η εφαρμογή greekfires.gr** **Enabling exploitation of Greek Fire Service's Public Open Data: the greekfires.gr application**
Ονοματεπώνυμο Φοιτητή	**Γεώργιος Τζαλαβράς**
Πατρώνυμο	**Μιχαήλ**
Αριθμός Μητρώου	**ΜΠΠΛ/ 11049**
Επιβλέπων	**Δημήτριος Βέργαδος, Επίκουρος Καθηγητής**

Ημερομηνία Παράδοσης **Σεπτέμβριος 2015**

Τριμελής Εξεταστική Επιτροπή

(υπογραφή) (υπογραφή) (υπογραφή)

Δημήτριος Βέργαδος
Επίκουρος Καθηγητής

Πίνακας Περιεχομένων

ΠΕΡΙΛΗΨΗ

Από τα δημόσια ανοιχτά δεδομένα της Πυροσβεστικής υπηρεσίας για τις δασικές πυρκαγιές της Ελλάδος, προέκυψε η ανάγκη δημιουργίας ενός ιστοτόπου (greekfires.gr) που θα παρουσιάζει το σύνολο των γεωγραφικών σημείων των πυρκαγιών, καθώς και την ημερολογιακή εξέλιξή τους. Στον ιστότοπο αυτό, παρέχεται επιπλέον η δυνατότητα ενημέρωσης ενός μεγάλου συνόλου πληροφοριών ανά πυρκαγιά (λ.χ. καμένη έκταση, υπηρεσιακό προσωπικό εμπλοκής στην κατάσβεση, πυροσβεστικά μέσα κ.α.).

Επιπρόσθετα, για την αποτελεσματικότερη διαχείριση του συνόλου της πληροφορίας αυτής, δημιουργήθηκε μία back-end εφαρμογή, όπου ο χρήστης μπορεί να εισάγει, να τροποποιεί και να επεξεργάζεται όλη την παρεχόμενη πληροφορία από τα ανοικτά δεδομένα σε μία λειτουργική βάση δεδομένων. Για το σχήμα της βάσης δεδομένων χρησιμοποιήθηκε η διοικητική διαίρεση της Ελλάδας βάσει του προγράμματος «Καλλικράτης».

Στόχος της παρούσας μεταπτυχιακής διπλωματικής εργασίας είναι: α) η αξιοποίηση των δημοσίων ανοικτών δεδομένων της Πυροσβεστικής Υπηρεσίας, β) η εύγλυπτη γεωσυσχετισμένη αποτύπωσή τους βάσει της χρονικής εξέλιξης των πυρκαγιών, γ) η στατιστική ανάλυση των περιστατικών πυρκαγιών με ενιαίες ερωτήσεις, και δ) η παροχή ενός μηχανισμού αποθήκευσης και ομογενοποίησης της παρεχόμενης πληροφορίας.

ABSTRACT

Inspired by the public and open data provided by the Greek Fire Service, we created the publicly available website greekfires.gr, where the citizen can find information in respect to fire incidents and their chronological evolution in Greece. The site also provides detailed information such as burnt area, staff /firefighting resources and others.

In addition, a back-end application capable of handling all necessary information was implemented, along with a relational schema according to the decentralized administration program “Kallikrates”.

During this dissertation we aim at: a) showing the benefits of using the public open data provided by the Greek Fire Service, b) presenting fire incidents in respect to their geo-location over time, c) illustrating statistical results, and d) provide a back-end application capable of storing and representing data from fire incidents.

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

Αντικείμενο της συγκεκριμένης εργασίας είναι η δημιουργία μιας διαδικτυακής εφαρμογής για την πληροφόρηση των πολιτών σχετικά με περιστατικά δασικών πυρκαγιών της Ελλάδος, καθώς και η δημιουργία μίας back-end εφαρμογής για την εισαγωγή και επεξεργασία της σχετικής πληροφορίας. Το σύνολο της πληροφορίας προέρχεται από τα δημόσια ανοιχτά δεδομένα που προσφέρονται από την Πυροσβεστική, τα οποία δίνονται μέσω ανοικτών προτύπων στον σύνδεσμο «Δημόσια Δεδομένα» του ιστοτόπου της Πυροσβεστικής Υπηρεσίας (www.fireservice.gr).

Τα δημόσια ανοιχτά δεδομένα είναι τα στοιχεία και οι πληροφορίες που παράγουν και διαθέτουν οι φορείς της δημόσιας διοίκησης και τα οποία είναι ανοιχτά σε όλους τους πολίτες. Σύμφωνα με τον Νόμο 4305/2014 επαναπροσδιορίζεται η ανοικτή διάθεση και περαιτέρω χρήση εγγράφων, πληροφοριών και δεδομένων του δημόσιου τομέα, με την προσαρμογή της εθνικής νομοθεσίας στις διατάξεις της Οδηγίας 2013/37/ΕΕ του Ευρωπαϊκού Κοινοβουλίου, για την περαιτέρω ενίσχυση της διαφάνειας [1].

Στο front-end μέρος, σκοπός της εργασίας είναι η αποτύπωση επάνω σε χρονολογικό χάρτη (Google Maps) των σημείων των πυρκαγιών βάσει της ημερομηνίας έναρξης και κατάσβεσής τους. Επιπλέον, προσφέρεται στον χρήστη η δυνατότητα να ενημερωθεί με αναλυτικές πληροφορίες για κάθε περιστατικό πυρκαγιάς, όπως αυτές προκύπτουν από το αρχείο περιστατικών της Πυροσβεστικής. Για την περαιτέρω ενημέρωση του χρήστη, χρησιμοποιούνται προγραμματιστικά εργαλεία προκειμένου να παρουσιαστούν στατιστικά στοιχεία τα οποία προκύπτουν από το σύνολο των περιστατικών. Τέλος, παρέχεται η δυνατότητα της επικοινωνίας του χρήστη με την εφαρμογή.

Στο back-end μέρος τώρα, ο σκοπός της εργασίας είναι η δημιουργία ενός συστήματος, στο οποίο ο διαχειριστής θα μπορεί να συνδέεται και να εισάγει, να επεξεργάζεται και να τροποποιεί τα δεδομένα των περιστατικών, καθώς και όλη την σχετική πληροφορία που σχετίζεται, όπως λ.χ. τα πυροσβεστικά μέσα και το προσωπικό που παίρνει μέρος στις κατασβέσεις, οι τύποι και τα στρέμματα των καμένων δασικών εκτάσεων, κ.α. Στο σύστημα αυτό δίνεται η δυνατότητα εισαγωγής της πληροφορίας μέσω ανοικτών προτύπων (π.χ. CSV). Αντιστρόφως, παρέχεται η δυνατότητα στον χρήστη να εξάγει σε ανοικτά πρότυπα το σύνολο της αποθηκευμένης πληροφορίας για περαιτέρω χρήση. Επιπλέον, παρέχεται η δυνατότητα ο χρήστης μέσω ενός plugin που χρησιμοποίει Google Maps, να εισάγει το σημείο της πυρκαγιάς επάνω στον χάρτη.

Για την δημιουργία των υπηρεσιών χρησιμοποιούνται σύγχρονες web programming τεχνολογίες, όπως είναι το Bootstrap Framework (HTML, CSS, jQuery) και το Laravel PHP Framework (PHP, MYSQL).

Η δομή της βάσης δεδομένων δημιουργήθηκε βάσει των δεδομένων που παρέχονται δημόσια από την Πυροσβεστική, εμπλουτισμένη με στοιχεία του σχεδίου «Καλλικράτης» για το σύνολο των περιφερειών, νομών και δήμων της Ελλάδος. Αυτός ο εμπλουτισμός συνιστά και βασική επαύξηση των ανοικτών δεδομένων της Πυροσβεστικής. Με αυτήν την συνεισφορά μας, τα δεδομένα πλέον θα είναι πιο αξιόπιστα όσον αφορά την γεωγραφική συσχέτισή τους. Επίσης, δημιουργήθηκαν οι δομές διασύνδεσης των 13 περιφερειών της χώρας, με τους αντίστοιχους νομούς και δήμους (βλ., «Κωδικολόγιο Καλλικράτειων Δήμων και Κοινοτήτων» ο κατάλογος του οποίου δίνεται από την σελίδα του Υπουργείου Εσωτερικών) [2].

Στόχος της παρούσας μεταπτυχιακής διπλωματικής εργασίας είναι: α) η αξιοποίηση των δημοσίων ανοικτών δεδομένων της Πυροσβεστικής Υπηρεσίας, β) η εύγλυπτη γεωσυσχετισμένη αποτύπωσή τους βάσει της χρονικής εξέλιξης των πυρκαγιών, γ) η στατιστική ανάλυση των περιστατικών των πυρκαγιών με ενιαίες ερωτήσεις, και δ) η παροχή ενός μηχανισμού αποθήκευσης και ομογενοποίησης της παρεχόμενης πληροφορίας.

Στο δεύτερο κεφάλαιο δίνεται ο ορισμός των δημόσιων ανοικτών δεδομένων και παρουσιάζονται τα πρότυπα βάσει των οποίων παρουσιάζονται τα δεδομένα αυτά. Επιπλέον, γίνεται η ανάλυση του μοντέλου 5-star Open Data, το οποίο κατηγοριοποιεί σε πέντε βαθμίδες την «ανοικτότητα» των δεδομένων. Παράλληλα, αναλύεται επίσης και το μοντέλο Linked Open Data (LOD), το οποίο συνίσταται από διασυνδεδεμένα δεδομένα της πέμπτης βαθμίδας του μοντέλου 5-star Open Data. Επίσης, παρέχονται παραδείγματα εφαρμογών και πρωτοβουλιών, οι οποίες παρουσιάζουν διάφορα δημόσια δεδομένα. Τέλος, πραγματοποιείται και η ανάλυση των δημόσιων δεδομένων της Πυροσβεστικής Υπηρεσίας με τα οποία ασχολείται η συγκεκριμένη εργασία.

Στο τρίτο κεφάλαιο αναλύεται το προτεινόμενο σύστημα σε επίπεδο αρχιτεκτονικής και τεχνολογιών που χρησιμοποιήθηκαν, καθώς και η βάση δεδομένων που το στηρίζει. Αρχικά πραγματοποιείται η ανάλυση των τεχνολογιών σε επίπεδο client-side (front-end) και έπειτα σε επίπεδο server-side (back-end). Τέλος, παρουσιάζεται η δομή και οι συσχετίσεις όλων των πεδίων και των ιδιοτήτων αυτών (σχήμα βάσης δεδομένων) μέσω της παρουσίασης του σχεσιακού μοντέλου και του μοντέλου οντοτήτων-συσχετίσεων.

Στο τέταρτο κεφάλαιο, παρουσιάζεται ο τρόπος λειτουργίας της εφαρμογής greekfires.gr τόσο σε front-end όσο και σε back-end επίπεδο, ενώ αναλύονται ο σκοπός της εφαρμογής με τα οφέλη τα οποία προκύπτουν από την δημιουργία της.

Τέλος, το πέμπτο και τελευταίο κεφάλαιο παρουσιάζει συνοπτικά τον σκοπό της εργασίας μας, καθώς και τα οφέλη που προκύπτουν τόσο για τον απλό χρήστη/πολίτη της εφαρμογής greekfires.gr, όσο και τον χρήστη του back-end. Επιπρόσθετα, αναφέρονται τα συμπεράσματα και οι πιθανές επεκτάσεις σχετικά με μελλοντική έρευνα.

ΚΕΦΑΛΑΙΟ 2: ΑΝΟΙΚΤΑ ΔΕΔΟΜΕΝΑ

2.1 ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό, δίνεται αρχικά ο ορισμός των ανοιχτών δεδομένων και κατόπιν καταγράφονται τα πρότυπα με τα οποία μπορούν να αναπαρασταθούν τα δεδομένα αυτά. Επίσης, παρουσιάζεται το 5-star Open Data Model, σύμφωνα με το οποίο πραγματοποιείται μία κατηγοριοποίηση της «ανοικτότητας» των δεδομένων σε πέντε βαθμίδες. Τέλος, παρέχονται παραδείγματα Δημοσίων Ανοικτών Δεδομένων από Ευρώπη και Ελλάδα, ενώ αναλύονται τα δεδομένα που παρέχονται από την Πυροσβεστική Υπηρεσία και είναι αντικείμενο χρήσης της παρούσας διπλωματικής εργασίας.

2.2 ΟΡΙΣΜΟΣ

Για την ανάλυση των ανοιχτών δεδομένων με τα οποία ασχολείται η εργασία, είναι προαπαιτούμενο να οριστεί αρχικά ο όρος «Ανοιχτά Δεδομένα». Έτσι λοιπόν, ως Ανοιχτά Δεδομένα ορίζονται τα δεδομένα στα οποία ο καθένας μπορεί να έχει ελεύθερη πρόσβαση με σκοπό να τα χρησιμοποιήσει, να τα επεξεργαστεί και να τα παρέχει στον οποιοδήποτε, προκειμένου να χρησιμοποιηθούν για οποιοδήποτε σκοπό. Ο μόνος όρος είναι κατά την χρησιμοποίηση των δεδομένων αυτών να γίνεται αναφορά στους δημιουργούς και να διατίθενται, με τη σειρά τους, υπό τους ίδιους όρους [3].

Στην Ελλάδα έπειτα από τον νόμο 4305/2014, τα ανοιχτά κυβερνητικά δεδομένα (Open Government Data) παρέχονται δημόσια για τους εξής λόγους:

Α) Ενίσχυση της διαφάνειας, προκειμένου ο κάθε πολίτης να έχει πρόσβαση στα δημόσια ανοιχτά δεδομένα με σκοπό την ενημέρωσή του αλλά και την κοινή χρήση και επαναχρησιμοποίηση των δεδομένων αυτών.

Β) Απελευθέρωση της κοινωνικής και εμπορικής αξίας, καθώς με το άνοιγμα των δεδομένων, η κυβέρνηση μπορεί να βοηθήσει να προωθηθεί η δημιουργία καινοτόμων επιχειρήσεων και των υπηρεσιών που παρέχουν κοινωνική και εμπορική αξία.

Γ) Συμμετοχική διακυβέρνηση, καθώς οι πολίτες έχουν τη δυνατότητα να είναι πολύ πιο άμεσα ενημερωμένοι και να συμμετέχουν στη διαδικασία λήψης αποφάσεων.

2.3 ΕΙΔΗ ΑΝΟΙΚΤΩΝ ΠΡΟΤΥΠΩΝ

Τα ανοικτά δεδομένα μπορούν να αναπαρασταθούν από ανοικτά πρότυπα (open file formats), τα οποία δεν επιβαρύνονται με οποιαδήποτε πνευματικά δικαιώματα, διπλώματα ευρεσιτεχνίας, εμπορικά σήματα ή άλλους περιορισμούς [4]. Επίσης, ο καθένας μπορεί να τα χρησιμοποιήσει χωρίς χρηματικό κόστος και για οποιονδήποτε επιθυμητό σκοπό. Τα ανοικτά πρότυπα μπορούν να αναπαρασταθούν με πολλούς τύπους αρχείων, όπως αρχεία απλού κειμένου (plain text),csv, xml, json και rdf.

2.3.1 Plain text

Το απλό κείμενο [5] (plain text) είναι τα περιεχόμενα ενός απλού σειριακού αρχείου, όταν αυτά διαβάζονται σαν κείμενο, χωρίς να είναι αναγκαία η επεξεργασία για την εμφάνισή του, σε αντίθεση με το κείμενο μορφοποίησης (formatted text). Η κωδικοποίηση ενός plain text αρχείου, μπορεί να είναι σε ASCII ή με βάση το πρότυπο Unicode. Τα αρχεία απλού κειμένου μπορούν να ανοιχτούν, να διαβαστούν και να τροποποιηθούν από τους περισσότερους διορθωτές κειμένου, όπως είναι το Notepad ή το Sublime Text. Ένα απλό κείμενο χρησιμοποιείται συχνά για αρχεία ρυθμίσεων (configuration files), τα οποία διαβάζονται κατά την εκκίνηση ενός προγράμματος και περιέχουν τις αποθηκευμένες ρυθμίσεις. Η κατάληξη που πρέπει να έχει ένα plain text αρχείο είναι η .txt.

2.3.2 CSV

Ένα αρχείο csv [6] περιλαμβάνει ένα σύνολο εγγραφών οι οποίες διαχωρίζονται με το κόμμα. Το αρχείο csv αποθηκεύει τα δεδομένα σαν μία μορφή πίνακα, στον οποίο κάθε γραμμή του πίνακα αποτελείται από τις στήλες, οι οποίες διαχωρίζονται με το κόμμα. Τα αρχεία csv μπορούν να επεξεργαστούν από διάφορες spread-sheet εφαρμογές, όπως είναι το OpenOffice και το Microsoft Excel. Η κατάληξη που πρέπει να έχει ένα αρχείο csv είναι η .csv. Τα csv αρχεία χρησιμοποιούνται από διάφορες προγραμματιστικές εφαρμογές, με σκοπό την εισαγωγή και εξαγωγή των εγγραφών που περιλαμβάνουν. Για παράδειγμα, μέσω του phpMyAdmin, για τις εγγραφές των πινάκων μίας βάσης μπορεί να γίνει Import καθώς και Export αυτών, με την χρήση ενός csv αρχείου.

Παράδειγμα μορφής ενός csv αρχείου είναι το παρακάτω:

"1","ΑΙΤΩΛΟΑΚΑΡΝΑΝΙΑΣ","4"

"2","ΑΝΑΤΟΛΙΚΗΣ ΑΤΤΙΚΗΣ","2"

"3","ΑΝΔΡΟΥ","11"

"4","ΑΡΓΟΛΙΔΑΣ","12"

"5","ΑΡΚΑΔΙΑΣ","12"

Στο παραπάνω παράδειγμα, κάθε γραμμή του πίνακα αποτελείται από τρείς στήλες οι οποίες διαχωρίζονται με κόμμα και των οποίων οι τιμές περιλαμβάνονται ανάμεσα σε double quotes.

2.3.3 XML

Η XML [7] (Extensible Markup Language) είναι μία γλώσσα επισήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Η XML σχεδιάστηκε δίνοντας έμφαση στην απλότητα, τη γενικότητα και τη χρησιμότητα στο διαδίκτυο. Σε αντίθεση με την HTML, οι χρήστες μπορούν να δημιουργήσουν ετικέτες χρησιμοποιώντας δική τους ονοματολογία. Επίσης, το περιεχόμενο των ετικετών περιλαμβάνει κείμενο σε plain text μορφή αλλά και κάποιες ιδιότητες οι οποίες ορίζονται μέσα στην ετικέτα. Η κατάληξη που έχει ένα αρχείο xml είναι η .xml.

Η σύνταξη και η δομή της XML δίνεται από το παρακάτω παράδειγμα:

<?xml version="1.0" encoding='UTF-8'?>

<staff type='firemen'>

<name>ΠΥΡΟΣ. ΣΩΜΑ</name>

</ staff >

Εδώ, η ετικέτα staff ορίζει το προσωπικό με το όνομα ΠΥΡΟΣ. ΣΩΜΑ το οποίο περιέχεται στην ετικέτα <name> και του οποίου ο τύπος firemen ορίζεται με την ιδιότητα type στην ετικέτα staff.

2.3.4 JSON

Το JSON [8] (JavaScript Object Notation) είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων. Η δομή του προσφέρει την δυνατότητα σε επίπεδο ανθρώπου να είναι πιο κατανοητή η ανάγνωση αλλά και η εγγραφή των στοιχείων από τα οποία αποτελείται. Επιπλέον, το JSON αρχείο είναι πιο εύκολο για τις μηχανές να το αναλύσουν (parse) και να το παράγουν (generate). Είναι βασισμένο πάνω σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript, ενώ είναι ένα πρότυπο κειμένου το οποίο είναι τελείως ανεξάρτητο από τις γλώσσες προγραμματισμού, αν και χρησιμοποιεί πρακτικές οι οποίες είναι γνωστές στους προγραμματιστές των γλωσσών C, C++, C#, Java, JavaScript, Perl, Python, και άλλων. Οι παραπάνω ιδιότητες λοιπόν κάνουν το JSON μια ιδανική γλώσσα προγραμματισμού ανταλλαγής δεδομένων.

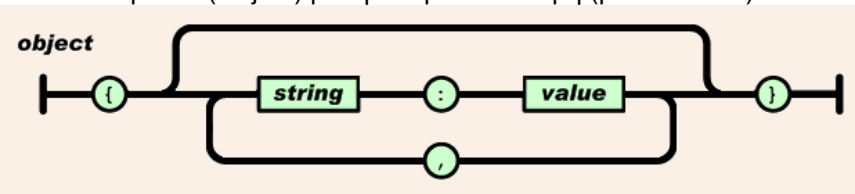
Το JSON είναι χτισμένο σε δύο δομές, τις οποίες όλες οι μοντέρνες γλώσσες προγραμματισμού τις υποστηρίζουν:

Α) Μια δομή συλλογής από ζευγάρια ονομάτων/τιμών. Σε διάφορες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένα object, μία καταχώριση, μία δομή, ένα λεξικό, μία λίστα κλειδιών.

Β) Μία δομή ταξινομημένης λίστας τιμών. Στις περισσότερες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένας πίνακας (array), ένα διάνυσμα, μία λίστα ή μία ακολουθία.

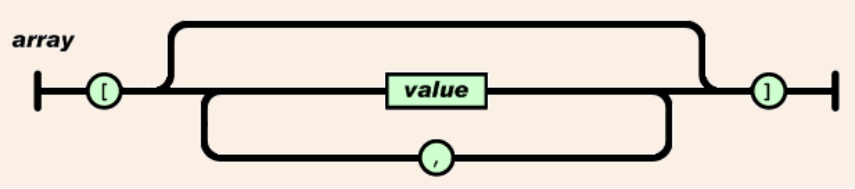
Οι δομές του JSON αναπαρίστανται μέσω τριών τύπων δεδομένων:

1. Του αντικειμένου (Object) με την παρακάτω δομή (βλ. Εικόνα 1):



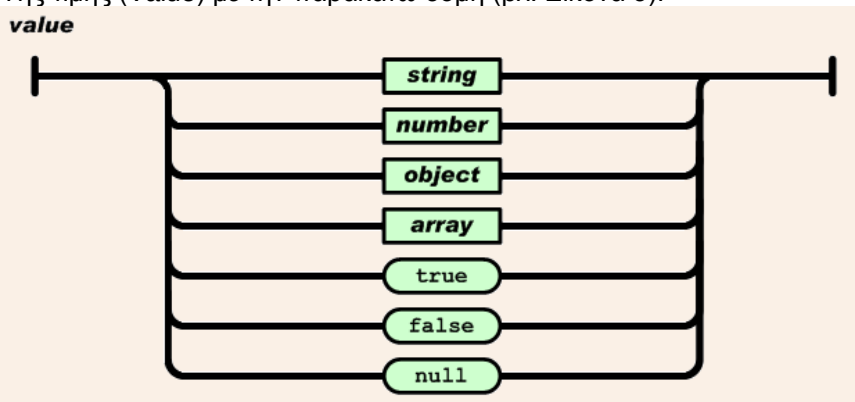
Εικόνα 1 - JSON Object

2. Του πίνακα (Array) με την παρακάτω δομή (βλ. Εικόνα 2):



Εικόνα 2 – JSON Array

3. Της τιμής (Value) με την παρακάτω δομή (βλ. Εικόνα 3):



Εικόνα 3 – JSON Value

Παράδειγμα αρχείου JSON το οποίο ακολουθεί τους παραπάνω κανόνες είναι το εξής:

[

{"title":"Ν. ΡΟΔΟΠΗΣ","options" : { "tags": "['61']" , "infoHtml": " <h5 class=nomargin>Δ. ΚΟΜΟΤΗΝΗΣ</h5><h6>02/01/2014 17:55:00</h6><h6>02/01/2014 21:40:00</h6>Περισσότερα+ " },"start":"2014-01-02T17:55:00+00:00","end":"2014- 01-02T21:40:00+00:00","point":{"lat":41.118469,"lon":25.3964763}},

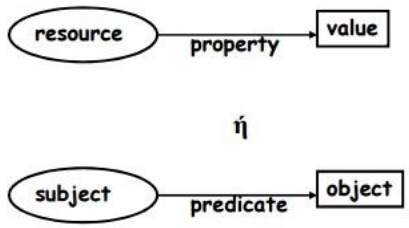
{"title":"Ν. ΡΟΔΟΠΗΣ","options" : { "tags": "['61']" , "infoHtml": " <h5 class=nomargin>Δ. ΚΟΜΟΤΗΝΗΣ</h5><h6>08/01/2014 12:49:00</h6><h6>08/01/2014 13:25:00</h6>Περισσότερα+ " },"start":"2014-01-08T12:49:00+00:00","end":"2014-01-08T13:25:00+00:00","point":{"lat":41.118469,"lon":25.3964763}}

]

Σε αυτό ορίζεται ένας πίνακας με δύο εγγραφές, στον οποίο υπάρχουν τα Objects (title, options, start, end, point), με τις τιμές τους, εκ των οποίων κάποια περιλαμβάνουν ως τιμές μία ακολουθία από επιμέρους σύνολα Objects με τις τιμές τους, όπως για παράδειγμα είναι το Object “point” το οποίο έχει ως τιμή ένα σύνολο από δύο Objects, των lat και lon, με τιμές τύπου number.

2.3.5 RDF

Το RDF [9] (Resource Description Framework) είναι ένα μοντέλο δεδομένων τα οποία αναπαρίστανται μέσω ενός κατευθυνόμενου γράφου ετικετών. Αν και η XML μπορεί να παρέχει την σύνταξη της RDF, η RDF διαφέρει από την XML, καθώς στην XML η αναπαράσταση των δεδομένων ακολουθεί μία δενδροειδή μορφή και επιπλέον υπάρχει η δυνατότητα τα RDF δεδομένα να μην εμφανίζονται σε μορφή XML. Μία δήλωση RDF αποτελείται από ένα σύνολο επιμέρους δηλώσεων. Κάθε δήλωση αποτελείται από τους πόρους (resources) που ορίζουν το υποκείμενο, οι οποίοι έχουν κάποιες ιδιότητες (properties) που ορίζουν το κατηγόρημα, οι τιμές (values) των οποίων ορίζουν το αντικείμενο. Η παραπάνω δήλωση αποτελεί την "τριπλέτα" της RDF (βλ. Εικόνα 4).



Εικόνα 4 – Τριπλέτα της RDF

Το RDF χρησιμοποιεί αναφορές σε URI για την ταυτοποίηση υποκειμένων, κατηγορημάτων και αντικειμένων. Για παράδειγμα, το URI http://www.w3.org/People/EM/contact#me της ιστοσελίδας W3C, αποτελεί ένα υποκείμενο βάση του οποίου περιγράφεται το άτομο Eric Miller, του οποίου οι προσωπικές πληροφορίες πηγάζουν από το συγκεκριμένο URI.

Η αναπαράσταση του RDF γίνεται μέσω του μοντέλου γράφων (graph model), της σημειογραφίας τριάδων (triples notation), καθώς και του εγγράφου RDF/XML.

Έτσι, ως γράφος το παραπάνω URI έχει την παρακάτω μορφή της εικόνας (Εικόνα 5):

Εικόνα 5 – Γράφος RDF

Μία τριπλέτα του γράφου είναι το υποκείμενο URI http://www.w3.org/People/EM/contact#, με την ιδιότητα 'τίτλος' να έχει την τιμή 'Dr'.

Σε αναπαράσταση σημειογραφίας τριάδων, το παραπάνω παράδειγμα αναπαριστάται ως εξής:

<http://www.w3.org/People/EM/contact#me>
<http://www.w3.org/2000/10/swap/pim/contact#fullName> "Eric Miller" .

<http://www.w3.org/People/EM/contact#me>
<http://www.w3.org/2000/10/swap/pim/contact#mailbox> <mailto:e.miller123(at)example> .

<http://www.w3.org/People/EM/contact#me> <http://www.w3.org/2000/10/swap/pim/contact#personalTitle> "Dr." .

<http://www.w3.org/People/EM/contact#me> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://www.w3.org/2000/10/swap/pim/contact#Person> .

Τέλος, ως έγγραφο RDF/XML το παραπάνω παράδειγμα έχει ως εξής:

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDFxmlns:contact="http://www.w3.org/2000/10/swap/pim/contact#"
xmlns:eric="http://www.w3.org/People/EM/contact#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:mailbox rdf:resource="mailto:e.miller123(at)example"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:personalTitle>Dr.</contact:personalTitle>
  </rdf:Description>
  <rdf:Description rdf:about="http://www.w3.org/People/EM/contact#me">
    <rdf:type rdf:resource="http://www.w3.org/2000/10/swap/pim/contact#Person"/>
  </rdf:Description>
</rdf:RDF>
```

2.4 ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ "ΑΝΟΙΚΤΟΤΗΤΑΣ" ΚΑΙ LINK OPEN DATA CLOUD

Στην προηγούμενη ενότητα έγινε η ανάλυση των τύπων των αρχείων με βάση τα οποία μπορούν να αναπαρασταθούν τα ανοιχτά δεδομένα. Στην ενότητα αυτή, γίνεται η ανάλυση του μοντέλου 5-star Open Data [10] που προτάθηκε από τον Tim Berners-Lee και το οποίο κατηγοριοποιεί τα ανοιχτά δεδομένα ανάλογα με τον τρόπο με τον οποίο αυτά παρέχονται και παρουσιάζονται (π.χ. csv, excel, rdf,.κ.λπ.).

Με τον όρο Linked Open Data [11], αναφερόμαστε στα ανοιχτά δεδομένα τα οποία με την χρήση του Web μπορούν να δημοσιοποιηθούν και να συνδεθούν με άλλα ανοιχτά δεδομένα, με τα οποία δεν υπήρχε καμία σύνδεση πρωτύτερα. Η σύνδεση των δεδομένων αυτών γίνεται με βάση τα URI και της RDF.

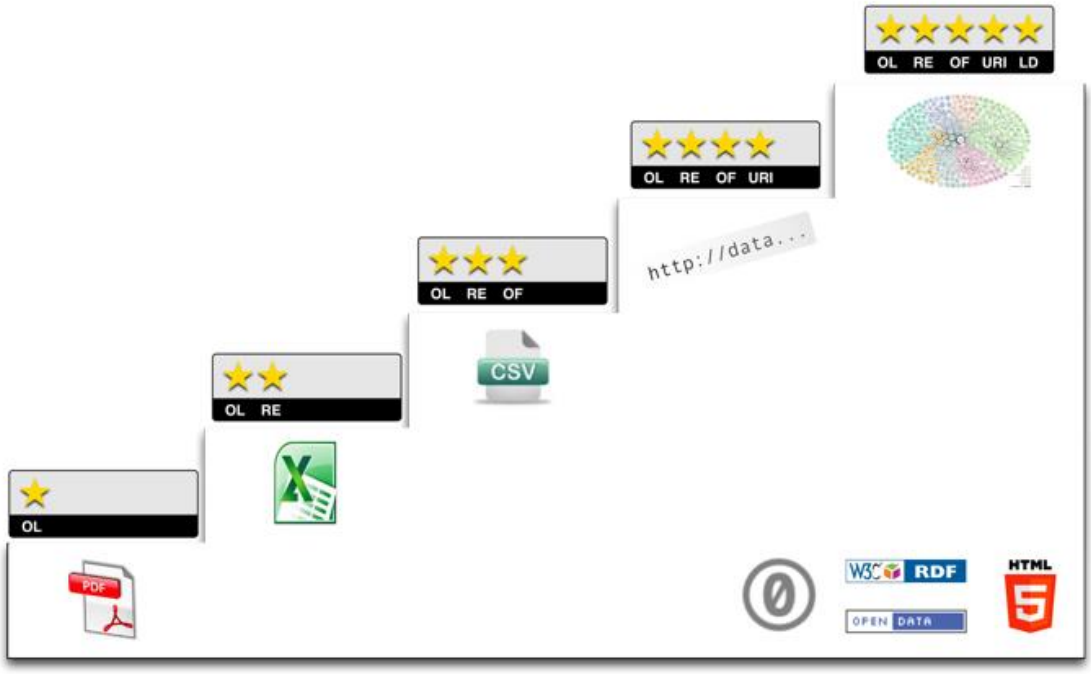
Όπως υποδεικνύει και το όνομα του μοντέλου 5-star Open Data [10], τα ανοιχτά δεδομένα μπορούν να κατηγοριοποιηθούν σε 5 βαθμίδες, με τα δεδομένα τα οποία βρίσκονται στην $5^{η}$ και τελευταία βαθμίδα, να αποτελούν τα δεδομένα τα οποία παρουσιάζονται στο μοντέλο του Link Open Data Cloud.

Η ανάλυση του μοντέλου 5-star Open Data, γίνεται με το παράδειγμα μίας πυρκαγιάς με τις πληροφορίες της Πυροσβεστικής Υπηρεσίας, του Νομού, της Ημερομηνίας και της Ώρας.

Για παράδειγμα η πυρκαγιά με τις εξής πληροφορίες:

Πυροσβεστική Υπηρεσία	Νομός	Ημερομηνία	Ώρα
1ος Π.Σ. ΑΘΗΝΩΝ	ΑΤΤΙΚΗΣ	19/6/2014	21:26

Οι πέντε βαθμίδες του 5-star Open Data παρουσιάζονται από την παρακάτω εικόνα (Εικόνα 6):



Εικόνα 6 – 5 Star Open Data

Πρώτη Βαθμίδα★: Δημοσιοποίηση των δεδομένων στο διαδίκτυο σε οποιαδήποτε μορφή.

Στην συγκεκριμένη βαθμίδα, η πληροφορία είναι κατανοητή μόνο στον άνθρωπο και δεν είναι δυνατή η αναζήτησή της από υπολογιστές, ούτε η αυτόματη κατηγοριοποίηση της πληροφορίας αυτής. Για παράδειγμα, ο παραπάνω πίνακας με τις πληροφορίες της πυρκαγιάς μπορεί να υπάρχει σε ένα αρχείο PDF.

Δεύτερη Βαθμίδα★★: Δημοσιοποίηση σε δομημένη μορφή (για παράδειγμα excel).

Η πληροφορία είναι προσβάσιμη σε υπολογιστές και καθίσταται δυνατή η αναζήτηση με βάση συγκεκριμένες λέξεις, όπως για παράδειγμα η λέξη ΑΤΤΙΚΗΣ. Η πληροφορία παραμένει όμως κατανοητή μόνο στον τελικό χρήστη.

Τρίτη Βαθμίδα★★★: Δημοσιοποίηση σε ανοιχτή μορφή (π.χ. σε CSV).

Η πληροφορία είναι προσβάσιμη σε όλους. Το πρότυπο είναι γνωστό και κάθε υπολογιστής μπορεί να αναζητήσει και να εξάγει μία πληροφορία από τα αρχεία. Όπως και στην προηγούμενη περίπτωση, μόνο ο τελικός χρήστης κατανοεί την σημασία της πληροφορίας. Στην βαθμίδα αυτή ο παραπάνω πίνακας μπορεί να πάρει την μορφή του παρακάτω CSV αρχείου:

Πυροσβεστική Υπηρεσία, Νομός, Ημερομηνία, Ώρα
1ος Π.Σ. ΑΘΗΝΩΝ, ΑΤΤΙΚΗΣ, 19/06/2014, 21:26

Τέταρτη Βαθμίδα★★★★: Χρησιμοποίηση URI για τον χαρακτηρισμό των αντικειμένων.

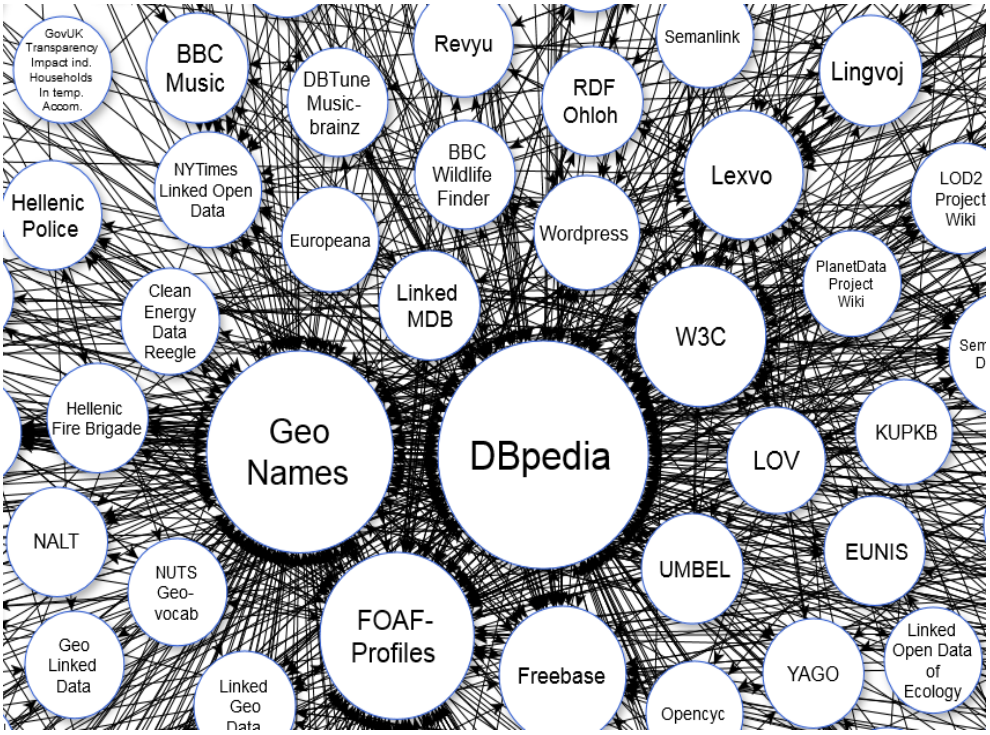
Η πληροφορία είναι προσβάσιμη από κάθε υπολογιστή. Ο χρήστης μπορεί να δει την πληροφορία μέσω URI links. Ο υπολογιστής στον οποίο είναι αποθηκευμένες οι

πληροφορίες μπορεί επιπλέον να προχωρήσει σε πιο σύνθετες αναζητήσεις με βάση την πληροφορία αυτή. Έτσι, οι πληροφορίες της πυρκαγιάς θα μπορούν να αναπαρασταθούν με τα εξής URI για παράδειγμα:

http://www.greekfires.gr/.../1ος-Π.Σ-ΑΘΗΝΩΝ
http://www.greekfires.gr/.../ΑΤΤΙΚΗΣ
http://www.greekfires.gr/.../19/6/2014
http://www.greekfires.gr/.../21:26

Πέμπτη Βαθμίδα★★★★★: Σύνδεση δεδομένων με άλλα δεδομένα.

Η πληροφορία δεν είναι πλέον προσβάσιμη και κατανοητή από όλους. Μόνο ο υπολογιστής στο διαδίκτυο μπορεί να κατανοήσει ότι η οντότητα για παράδειγμα της Ημερομηνίας ή του Νομού αντιστοιχεί σε μία συγκεκριμένη πληροφορία με βάση την οποία μπορεί να κάνει αναζήτηση άλλων πληροφοριών. Έτσι, τα δεδομένα της Πυροσβεστικής μπορούν να συνδεθούν με άλλα δεδομένα και αυτό να οδηγήσει στο μοντέλο του Linked Open Data, μέρος του οποίου παρουσιάζεται από την παρακάτω εικόνα (Εικόνα 7):



Εικόνα 7 - Linked Open Data Cloud

2.5 ΔΗΜΟΣΙΑ ΑΝΟΙΚΤΑ ΔΕΔΟΜΕΝΑ

Στις προηγούμενες ενότητες ορίστηκαν τα ανοικτά δεδομένα, ενώ έγινε και η ανάλυση των μοντέλων μέσω των οποίων αυτού του είδους τα δεδομένα μπορούν να παρουσιαστούν και να επεξεργαστούν. Στην ενότητα αυτή, δίνονται παραδείγματα ανοικτών δεδομένων τα οποία προσφέρονται δημόσια σε πολίτες, φορείς, ερευνητές κ.α., από οργανισμούς της Ευρώπης αλλά και της Ελλάδος.

2.5.1 Η πολιτική της Ευρώπης

Η Ευρωπαϊκή Επιτροπή από τον Δεκέμβριο του 2011 έχει πάρει πρωτοβουλίες προκειμένου να υποστηρίξει την διάδοση και χρησιμοποίηση όλο και περισσότερων δημόσιων ανοιχτών δεδομένων

από τους φορείς των κρατών μελών της Ευρωπαϊκής Ένωσης [12]. Οι λόγοι για τους οποίους υποστηρίζει το άνοιγμα των δημόσιων δεδομένων είναι οικονομικοί, κοινωνικοί και εκπαιδευτικοί.

Τα μέτρα τα οποία πήρε η Ευρωπαϊκή Επιτροπή στηρίζονται σε τρείς τομείς:

Α) Στην αναπροσαρμογή του νομοθετικού πλαισίου, η οποία αφορά την επαναχρησιμοποίηση των δεδομένων στα πλαίσια της Ευρώπης.

Β) Στην κινητοποίηση των φορών να χρηματοδοτήσουν το άνοιγμα των δημόσιων δεδομένων, μέσω της κατασκευής Ευρωπαϊκών data-portals.

Γ) Στην διευκόλυνση του συντονισμού και της προσφοράς εμπειρίας στην χρήση δημόσιων δεδομένων μεταξύ των κρατών μελών.

Στόχος των παραπάνω μέτρων είναι με το άνοιγμα των δημόσιων δεδομένων:

1^{ov} Να επιτευχθεί οικονομική ανάπτυξη μέσω των εταιριών οι οποίες θα αξιοποιήσουν τα δημόσια δεδομένα.

2^{ov} Να χρησιμοποιηθούν τα δημόσια δεδομένα στον τομέα της έρευνας από εκπαιδευτικούς φορείς.

3^{ov} Να ενισχυθεί η διαφάνεια στα κράτη μέλη καθώς και η καλύτερη ενημέρωση των πολιτών.

Στα πλαίσια λοιπόν της Ευρώπης έχουν δημιουργηθεί διάφορα data-portals, όπως είναι τα παρακάτω:

publicdata.eu [13]: Πρόκειται για μία ιστοσελίδα η οποία δίνει την δυνατότητα πρόσβασης σε ανοιχτά δεδομένα τα οποία πηγάζουν από τοπικούς και εθνικούς τομείς των χωρών της Ευρώπης. Τα δεδομένα τα οποία προσφέρονται σε αυτή την σελίδα προέρχονται από πολλούς τομείς (τομείς υγείας, εκπαίδευσης, οικονομίας, μεταφορών κ.ά.), όπως είναι για παράδειγμα τα ετήσια δεδομένα της χρησιμοποίησης λεωφορείων στην Αγγλία ή τα δεδομένα της αύξησης της ατμοσφαιρικής ρύπανσης στην Γαλλία.

opendata.paris.fr [14]: Ιστοσελίδα η οποία προσφέρει δημόσια δεδομένα σχετικά με την πόλη του Παρισιού, όπως είναι για παράδειγμα τα δεδομένα που προσφέρουν γεωγραφική πληροφορία σχετικά με την φύτευση δέντρων στο Παρίσι ή τα δεδομένα που ενημερώνουν τους πολίτες σχετικά με την οικονομική κατάσταση του δήμου.

data.gouv.fr/fr/ [15]: Ιστοσελίδα η οποία προσφέρει δημόσια δεδομένα του Γαλλικού κράτους σε πολλούς τομείς, όπως για παράδειγμα δεδομένα σχετικά με τον πληθυσμό από το 1962 έως το 2013 ή των τροχαίων ατυχημάτων από το 2006 έως το 2011. Στην σελίδα αυτή τα δεδομένα παρέχονται από κρατικούς αλλά και μη κρατικούς τομείς της Γαλλίας.

dati.piemonte.it [16]: Ιστοσελίδα η οποία προσφέρει δημόσια δεδομένα σχετικά με τις περιοχές της Ιταλίας, ενώ δίνεται η δυνατότητα τα δεδομένα αυτά να προσφέρονται σε RDF μορφή.

data.overheid.nl [17]: Ιστοσελίδα η οποία προσφέρει δημόσια δεδομένα του Ολλανδικού κράτους.

data.gov.uk [18]: Ιστοσελίδα η οποία προσφέρει δημόσια δεδομένα της Αγγλίας, ενώ προσφέρει και γεωγραφική πληροφορία των δεδομένων, όπως κάνει και η ιστοσελίδα του Γαλλικού κράτους.

2.5.2 Κατάσταση στην Ελλάδα

Όπως γίνεται και στην υπόλοιπη Ευρώπη, έτσι και στην Ελλάδα έχουν δημιουργηθεί υπηρεσίες και πρωτοβουλίες, οι οποίες προσφέρουν ανοιχτά τα δημόσια δεδομένα.

Παραδείγματα τέτοιων πρωτοβουλιών είναι:

geodata.gov.gr [19]: Το geodata.gov.gr συμπεριλαμβάνεται στις δράσεις της Ελληνικής Κυβέρνησης για το άνοιγμα των δημόσιων δεδομένων. Δημιουργήθηκε και αναπτύχθηκε από το Ινστιτούτο Πληροφοριακών Συστημάτων του Ερευνητικού Κέντρου «Αθήνα» και προσφέρει δημόσια δεδομένα σε πολλούς τομείς, όπως είναι τα δημόσια δεδομένα τα οποία δίνονται από τους δήμους ή τα δημόσια δεδομένα τα οποία παρέχονται από άλλους τομείς, όπως είναι για παράδειγμα τα δεδομένα σχετικά με τα υπόγεια ύδατα. Τα δεδομένα αυτά προσφέρονται σε διάφορες μορφές (xls, kml, κ.ά.), ενώ παρέχεται η δυνατότητα αποτύπωσης αυτών των δεδομένων επάνω σε χάρτες.

opengov.gr [20]: Το Υπουργείο Εσωτερικών και Διοικητικής Ανασυγκρότησης στα πλαίσια της διαφάνειας δημιούργησε την συγκεκριμένη ιστοσελίδα, η οποία παρέχει τα δημόσια δεδομένα που σχετίζονται με τις αποφάσεις όλων των Υπουργείων. Έτσι, παρουσιάζονται όλες οι αποφάσεις στους πολίτες και δίνεται η δυνατότητα αποθήκευσης αυτών σε μορφή pdf.

lorax.gr [21]: Πρόκειται για μία ιστοσελίδα η οποία δημιουργήθηκε στα πλαίσια του διαγωνισμού Open Data Hacathlon 2014 και η οποία προσφέρει δεδομένα σε κάποιους τομείς (Δασικές Πυρκαγιές 2000-2012, Ανήλικοι Παραβάτες, κ.ά.), μέσω γραφικής αναπαράστασης και ευρετηρίων.

astynomia.gr [22]: Η Ελληνική Αστυνομία στην ιστοσελίδα της προσφέρει δημόσια δεδομένα τα οποία σχετίζονται με τους τομείς δράσης της, όπως είναι για παράδειγμα τα τροχαία ατυχήματα και οι κλοπές. Η διάθεση των δεδομένων αυτών γίνεται μέσω στατιστικών δεδομένων σε μορφή εικόνων, καθώς και σε μορφή λίστας αρχείων pdf.

fireservice.gr [23]: Όπως η αστυνομία έτσι και η Πυροσβεστική Υπηρεσία στην ιστοσελίδα της στον παραπάνω σύνδεσμο προσφέρει τα δημόσια δεδομένα σχετικά με τους τομείς δράσης της σε μορφές xls, xlsx και csv. Τα δεδομένα αυτά προσφέρουν πληροφορίες σχετικά με τις δασικές πυρκαγιές των ετών 2000-2012, 2013, 2014, καθώς και τα αστικά συμβάντα του έτους 2014. Επιπλέον, προσφέρει δεδομένα τα οποία σχετίζονται με τον ηλεκτρονικό εξοπλισμό της, τον κατάλογο της βιβλιοθήκης της, το πυροσβεστικό υλικό και τα αντικείμενα της εκπαίδευσής της.

2.6 ΤΑ ΑΝΟΙΚΤΑ ΔΕΔΟΜΕΝΑ ΤΗΣ ΠΑΡΟΥΣΑΣ ΕΡΓΑΣΙΑΣ

Τα ανοικτά δεδομένα με τα οποία ασχολείται η συγκεκριμένη διπλωματική εργασία, προέρχονται από τα δεδομένα τα οποία προσφέρει η Πυροσβεστική Υπηρεσία στην ιστοσελίδα της, σχετικά με τις Δασικές Πυρκαγιές του έτους 2014. Τα δεδομένα αυτά προέρχονται από το αρχείο excel (Dasikes_pyrkagies_2014_el_gr), και περιλαμβάνουν το σύνολο των δασικών πυρκαγιών με τις πληροφορίες της καμένης έκτασης της κάθε πυρκαγιάς, καθώς και το σύνολο και τύπο των μέσων και του προσωπικού το οποίο πήρε μέρος στην πυρόσβεση. Καθώς στην συγκεκριμένη εργασία παρέχεται η δυνατότητα το αρχείο excel έπειτα από κατάλληλη επεξεργασία να μετατραπεί σε αρχείο csv, καθίσταται σαφές ότι πραγματευόμαστε αρχεία δεύτερης και τρίτης κατηγορίας βάσει του 5-Star Open Data Model (αρχεία excel και csv αντίστοιχα).

Έτσι, το αρχείο αυτό περιλαμβάνει έναν πίνακα με πεδία τα οποία χωρίζονται σε τέσσερις κατηγορίες:

1) Η πρώτη κατηγορία περιλαμβάνει την γεωγραφική περιοχή και την χρονική στιγμή της πυρκαγιάς, καθώς και την υπηρεσία η οποία ανέλαβε την κατάσβεση της πυρκαγιάς. Σε αυτήν την κατηγορία οι παραπάνω πληροφορίες παρέχονται από τις ακόλουθες στήλες του πίνακα: *Υπηρεσία, Νομός, Ημερ/νία Έναρξης, Ώρα Έναρξης, Ημερ/νία Κατάσβεσης, Ώρα Κατάσβεσης, Δασαρχείο, Δήμος, Περιοχή, Διεύθυνση*.

2) Η δεύτερη κατηγορία περιλαμβάνει την καμένη έκταση της πυρκαγιάς ανάλογα με τον τύπο της έκτασης και οι συγκεκριμένες πληροφορίες παρέχονται από τις ακόλουθες στήλες του πίνακα: *Δάση, Δασική Έκταση, Άλση, Χορτ/κές Εκτάσεις, Καλάμια – Βάλτοι, Γεωργικές Εκτάσεις, Υπολείμματα Καλλιεργειών, Σκουπιδότοποι*.

3) Η τρίτη κατηγορία περιλαμβάνει τον αριθμό του προσωπικού ανάλογα με τον τύπο ο οποίος πήρε μέρος κατά την διάρκεια της πυρόσβεσης. Οι πληροφορίες αυτές δίνονται από τις ακόλουθες στήλες του πίνακα : *ΠΥΡΟΣ. ΣΩΜΑ, ΠΕΖΟΠΟΡΑ ΤΜΗΜΑΤΑ, ΕΘΕΛΟΝΤΕΣ, ΣΤΡΑΤΟΣ, ΑΛΛΕΣ ΔΥΝΑΜΕΙΣ*.

4) Η τέταρτη και τελευταία κατηγορία, περιλαμβάνει τον αριθμό των οχημάτων και των εναέριων μέσων τα οποία πήραν μέρος κατά την διάρκεια της πυρόσβεσης της πυρκαγιάς. Οι πληροφορίες αυτές παρέχονται από τις ακόλουθες στήλες του πίνακα: *ΠΥΡΟΣ. ΟΧΗΜ., ΟΧΗΜ. ΟΤΑ, ΒΥΤΙΟΦΟΡΑ, ΜΗΧΑΝΗΜΑΤΑ, ΕΛΙΚΟΠΤΕΡΑ, Α/Φ CL415, Α/Φ CL215, Α/Φ PZL, Α/Φ GRU*.

Από την επεξεργασία των δεδομένων της Πυροσβεστικής Υπηρεσίας, προέκυψε ότι υπάρχουν κάποιες ελλείψεις και λάθη σε ότι αφορά την πρώτη κατηγορία δεδομένων η οποία αναλύθηκε παραπάνω. Στις υπόλοιπες κατηγορίες (καμένη έκταση, προσωπικό, μέσα πυρόσβεσης), δεν υπάρχουν πεδία κενά και ο αριθμός αυτών είναι σωστά συμπληρωμένος.

Πιο συγκεκριμένα στην πρώτη κατηγορία, υπάρχουν πολλά λάθη στην εισαγωγή των ημερομηνιών κατάσβεσης μίας πυρκαγιάς, καθώς εμφανίζονται πολλές ημερομηνίες από αυτές να είναι προγενέστερες των ημερομηνιών έναρξης των πυρκαγιών. Επίσης, υπάρχουν πολλές ημερομηνίες κατάσβεσης πυρκαγιών, στις οποίες υπάρχει στην θέση του μήνα η ημέρα, με αποτέλεσμα ο χρόνος διάρκειας των πυρκαγιών να είναι πολύ μεγάλος (διάρκεια πάνω από 2 μήνες).

Μάλιστα, στην πυρκαγιά με τα εξής στοιχεία: *Ημ/νία Έναρξης: 2014-01-05, Ημ/νία Κατάσβεσης:18:40:00 2015-01-05 20:00:00, Νομός: ΜΕΣΟΛΟΓΓΙΟΥ, Διεύθυνση: Δ.Κ ΕΥΗΝΟΧΩΡΙΟΥ*, παρατηρείται ότι έχει γίνει λανθασμένη εισαγωγή έτους στην ημερομηνία κατάσβεσης (2015 αντί 2014), με αποτέλεσμα η πυρκαγιά να έχει διαρκέσει έναν ολόκληρο χρόνο. Για την συγκεκριμένη πυρκαγιά έγινε διόρθωση της ημερομηνία αυτής.

Επίσης, σε πολλές πυρκαγιές η ημερομηνία κατάσβεσης δεν έχει εισαχθεί. Για τον λόγο αυτό, δεν υπάρχει κάποια εξήγηση στο αρχείο excel. Ακολουθώντας λοιπόν αυτή την πιθανότατα παράβλεψη, στην συγκεκριμένη εργασία έχει προβλεφθεί να μπορεί να γίνει η εισαγωγή μίας πυρκαγιάς χωρίς να έχει ακριβή ημερομηνία κατάσβεσης. Για τον λόγο αυτό στον χάρτη των πυρκαγιών, η συγκεκριμένη πυρκαγιά εκλαμβάνεται ως σημείο, χωρίς να έχει χρονική διάρκεια.

Τέλος, παρατηρείται ότι σε πολλές πυρκαγιές οι τιμές στα πεδία *Δασαρχείο, Περιοχή* είναι κενές και ότι οι τιμές στα πεδία *Νομός, Δήμος*, να μην έχουν γραφτεί, όπως αυτές υπάρχουν στην λίστα Νομών-Δήμων του Νομού Καλλικράτης.

ΚΕΦΑΛΑΙΟ 3: ΠΡΟΤΕΙΝΟΜΕΝΟ ΣΥΣΤΗΜΑ

3.1 ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό γίνεται η ανάλυση του προτεινόμενου συστήματος σε επίπεδο αρχιτεκτονικής και τεχνολογιών οι οποίες χρησιμοποιήθηκαν, καθώς και η ανάλυση της Βάσης Δεδομένων, η δομή της οποίας προέκυψε από τις ανάγκες του συστήματος. Αρχικά γίνεται η ανάλυση των τεχνολογιών σε επίπεδο Client-side (Front-end) και έπειτα σε επίπεδο Server-side (Back-end). Τέλος, γίνεται η περιγραφή των βημάτων από τα οποία προέκυψε η τελική βάση δεδομένων (greek_fires), καθώς και η σύνδεση των πινάκων αυτής μέσω της παρουσίασης του Μοντέλου Οντοτήτων-Συσχετίσεων και του Σχεσιακού μοντέλου.

3.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ - ΤΕΧΝΟΛΟΓΙΕΣ

Η αρχιτεκτονική του συστήματος βασίζεται σε δύο μέρη:

A) Στο Client-side (Front-end)

B) Στο Server-side (Back-end)

Το Client-side είναι το μέρος του συστήματος με το οποίο αλληλεπιδρά ο κάθε χρήστης. Αποτελεί επομένως την κυρίως web σελίδα του συστήματος η οποία είναι ορατή από όλους τους χρήστες αυτής. Αυτή βασίζεται σε τεχνολογίες κυρίως Front-end (HTML, CSS, JS).

Το Server-side τώρα, είναι το μέρος του συστήματος με το οποίο αλληλεπιδρά ο διαχειριστής (admin) του συστήματος και φέρνει σε επαφή τον διαχειριστή με την βάση δεδομένων. Το μέρος αυτό χρησιμοποιεί τις Back-end τεχνολογίες (PHP, MYSQL).

3.2.1 Client-side Τεχνολογίες

A. Bootstrap (Front-end - Responsive Web Development)

Το Bootstrap είναι το πιο δημοφιλές HTML, CSS, και JS framework για την ανάπτυξη responsive ιστοσελίδων στο διαδίκτυο [24]. Με τον όρο responsive εννοούμε την δημιουργία μίας «έξυπνης» ιστοσελίδας η οποία θα προσαρμόζει το μέγεθος και τα βασικά χαρακτηριστικά της (μενού, εικόνες, κείμενο) ανάλογα με τις διαστάσεις της οθόνης της συσκευής του χρήστη (υπολογιστής, κινητό, tablet). Η τεχνική του responsive web design πλέον αρχίζει και χρησιμοποιείται ευρέως, διότι είναι μια οικονομική λύση για αυτούς που θέλουν η ιστοσελίδα τους να είναι συμβατή σε κάθε browser και σε κάθε συσκευή. Έτσι, μπορεί να επιτευχθεί η αύξηση των επισκέψεων της σελίδας, αλλά και η εμπειρία πλοήγησης των επισκεπτών αυτής.

Το Bootstrap είναι μια συλλογή εργαλείων ανοιχτού κώδικα για τη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Περιέχει HTML και CSS για τις μορφές τυπογραφίας, κουμπιά πλοήγησης, καθώς και προαιρετικές επεκτάσεις JavaScript. Στηρίζει το Front-end Web Development, δηλαδή αυτό που βλέπει και αλληλεπιδρά ο χρήστης της σελίδας. Επίσης, το Bootstrap είναι συμβατό με όλους τους φυλλομετρητές (browsers).

Το Bootstrap αναπτύχθηκε από τον Mark Otto και τον Jacob Thornton στο Twitter ως ένα πλαίσιο για την ενθάρρυνση της συνέπειας στα εσωτερικά εργαλεία. Τον Αύγουστο του 2011 κυκλοφόρησε το Twitter Bootstrap ως λογισμικό ανοιχτού κώδικα. Τον Φεβρουάριο του 2012, ήταν το πιο δημοφιλές έργο ανάπτυξης στο GitHub (ιστοσελίδα «κοινωνικής δικτύωσης» και ανάπτυξης κώδικα).

Για να χρησιμοποιηθεί το Bootstrap σε μια σελίδα HTML, πρέπει να γίνει λήψη και εγκατάσταση του αρχείου CSS (bootstrap.min.css) και της βιβλιοθήκης jQuery (bootstrap.min.js) του Bootstrap, από την σελίδα του Bootstrap (http://getbootstrap.com).

Σύστημα πλέγματος (Grid System) και ανταποκρίσιμος σχεδιασμός (responsive design)

Το Bootstrap παρέχει ένα πολύ ευέλικτο σύστημα grid [25]. Με την responsive συμπεριφορά του προσαρμόζεται αναλόγως στο πλάτος της οθόνης.

Υπάρχουν 4 επιπέδα grid columns (στήλες πλέγματος): extra-small, small, medium, large τα οποία χρησιμοποιούνται για συσκευές με πλάτος <768px, ≥768px, ≥992px, και ≥1200px αντίστοιχα.

Κανόνες:

- Το πλέγμα (γραμμές και στήλες) ορίζεται εσωτερικά ενός div.container.
- Οι γραμμές div.row περιέχουν (εσωτερικά) στήλες .col-*-*, και το άθροισμα των στηλών πρέπει να είναι έως 12.
- Οι στήλες ορίζονται ως col-επίπεδο-μέγεθος, π.χ. .col-md-6 στήλη μεγέθους 6 (50% οθόνης αφού 12 είναι το max της γραμμής) σε οθόνες μεσαίου μεγέθους ≥992px.

HTML και κλάσεις CSS του Bootstrap

Το Bootstrap χρησιμοποιεί ένα μεγάλο πλήθος κλάσεων, οι οποίες εισάγονται στις ετικέτες HTML και ορίζουν την συμπεριφορά, τα χρώματα και τα μεγέθη των ετικετών. Επίσης, ορίζουν και την θέση που έχουν οι ετικέτες επάνω στην σελίδα, ανάλογα με το μέγεθος της οθόνης [25].

HTML και κλάσεις Header

Για την δημιουργία ενός header στην σελίδα, το Bootstrap χρησιμοποιεί τον παρακάτω κώδικα:

```
<nav class="navbar navbar-default navbar-fixed-top affix-top" id="navbar-fires">
    <div class="container">
      <div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
      data-target="#navbar" aria-expanded="false" aria-controls="navbar">
          <span class="sr-only">Toggle navigation</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand page-scroll" href="#intro">GreekFires</a>
      </div>
      <div id="navbar" class="navbar-collapse collapse">

        <ul class="nav navbar-nav navbar-right">
          <li><a href="#mapsection" class="page-scroll">Χάρτης</a></li>
          <li><a href="#statistics" class="page-scroll">Στατιστικά</a></li>
          <li><a href="#contact"    class="page-scroll">Επικοινωνία</a></li>
        </ul>
      </div>
    </div>
  </nav>
```

Με την ετικέτα nav ορίζεται το header το οποίο για να είναι πάντα στατικό και ορατό στο πάνω μέρος της σελίδας, χρησιμοποιεί την κλάση navbar-fixed-top.

Το κομμάτι του παρακάτω κώδικα, χρησιμοποιείται για να εμφανίζεται με την χρήση του κουμπιού το μενού στις μικρές συσκευές.

```
<div class="navbar-header">
        <button type="button" class="navbar-toggle collapsed" data-toggle="collapse"
       data-target="#navbar" aria-expanded="false" aria-controls="navbar">
          <span class="sr-only">Toggle navigation</span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        <a class="navbar-brand page-scroll" href="#intro">GreekFires</a>
      </div>

Η λίστα των συνδέσμων στο μενού τοποθετείται ως εξής:
<ul class="nav navbar-nav navbar-right">
          <li><a href="#mapsection" class="page-scroll">Χάρτης</a></li>
          <li><a href="#statistics" class="page-scroll">Στατιστικά</a></li>
          <li><a href="#contact"    class="page-scroll">Επικοινωνία</a></li>
</ul>
```

Πρόκειται για μία λίστα στοιχείων στην οποία χρησιμοποιείται η κλάση .navbar-right προκειμένου να τοποθετηθούν τα στοιχεία στα δεξιά.

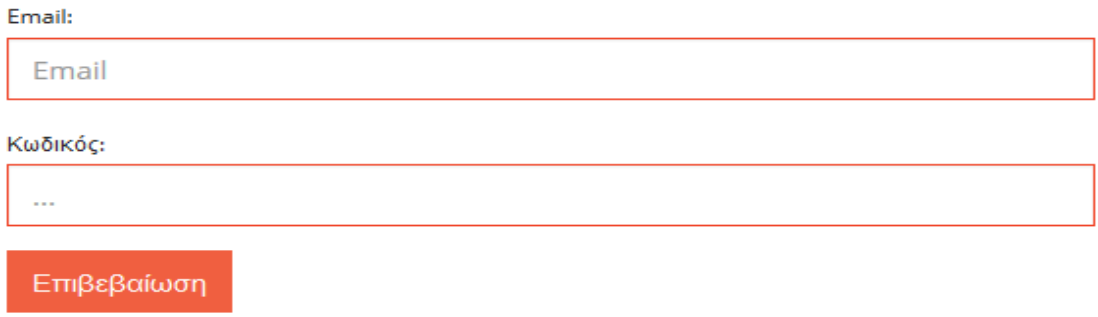
HTML και κλάσεις Φόρμας

Μία φόρμα ορίζεται στο Bootstrap ως εξής:

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1"
          placeholder="Enter  email">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1"
          placeholder="Password">
  </div>
  <div class="form-group has-error">
  <label class="control-label" for="inputError1">Input with error</label>
  <input type="text" class="form-control" id="inputError1">
</div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

Η κλάση .form-group που περιέχεται σε ένα div της φόρμας περιέχει τους τύπους των input (text, textarea, password, checkboxes, radio-buttons κ.ά.) καθώς και τις ετικέτες τους (label). Η ιδιότητα for του label και το id του input πρέπει πάντα να έχουν το ίδιο όνομα, καθώς στην επιβεβαίωση των στοιχείων της φόρμας χρησιμοποιώντας το κουμπί button τύπου submit, εάν κάποιο input δεν έχει τιμή (είναι κενό), θα χρησιμοποιηθεί η κλάση .has-error. Η συγκεκριμένη κλάση εισάγεται στο div της κλάσης form-group και με αυτόν τον τρόπο επισημαίνει με κατάλληλο χρώμα την έλλειψη των στοιχείων της φόρμας κατά την επιβεβαίωση.

Η κλάση .form-control χρησιμοποιείται σε input ετικέτες και ορίζει το στυλ που θα έχουν τα input πεδία. Το αποτέλεσμα μίας επιβεβαίωσης μίας φόρμας με έλλειψη συμπληρωμένων στοιχείων, απεικονίζεται από την παρακάτω εικόνα (Εικόνα 8):



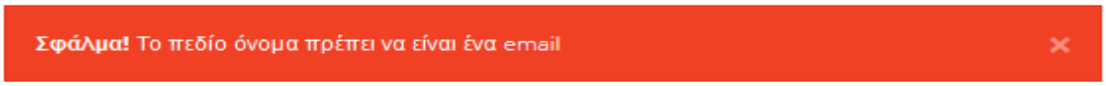
Εικόνα 8 – Επιβεβαίωση φόρμας Bootstrap και έλλειψη συμπληρωμένων στοιχείων

HTML και κλάσεις Alert

Το Bootstrap μπορεί να εισάγει μηνύματα λάθους ή επιβεβαίωσης χρησιμοποιώντας την κλάση .alert ,όπως δείχνει ο παρακάτω κώδικας:

```
<div class="alert alert-danger alert-dismissible" role="alert">
    <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>
    <strong>Σφάλμα!</strong> Το πεδίο όνομα πρέπει να είναι ένα email
</div>
```

Η κλάση .alert-danger χρησιμοποιείται για να εισάγει ένα χρώμα ως φόντο στο μήνυμα λάθους (κόκκινο χρώμα) ενώ με την κλάση .alert-dismissible μπορεί να κλείσει το Alert μήνυμα (βλ. Εικόνα 9).



Εικόνα 9 – Bootstrap Alert

HTML και κλάσεις Πίνακα (Table)

Ο κώδικας του πίνακα στο Bootstrap είναι ίδιος με τους πίνακες στην HTML. Αυτό που αλλάζει είναι ότι στην ετικέτα <table> μπορούν να εισαχθούν κατάλληλες κλάσεις που ορίζουν την μορφή που θα έχει ο πίνακας.

```
<table class="table table-striped table-hover" id="fire_departments">
    <thead>
        <tr>
            <th>Όνομα</th>
            <th>Περιφέρεια</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>ΑΝΑΤΟΛΙΚΗΣ ΑΤΤΙΚΗΣ</td>
            <td>ΑΤΤΙΚΗΣ</td>
    </tr>
    </tbody>
</table>
```

Η κλάση .table-striped (βλ. Εικόνα 10) δημιουργεί πίνακα με γραμμές οι οποίες εναλλάξ έχουν διαφορετικό στυλ, προκειμένου η κάθε γραμμή να ξεχωρίζει από την επόμενη.

Όνομα	Περιφέρεια
ΑΙΤΩΛΟΑΚΑΡΝΑΝΙΑΣ	ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ
ΑΝΑΤΟΛΙΚΗΣ ΑΤΤΙΚΗΣ	ΑΤΤΙΚΗΣ
ΑΝΔΡΟΥ	ΝΟΤΙΟΥ ΑΙΓΑΙΟΥ
ΑΡΓΟΛΙΔΑΣ	ΠΕΛΟΠΟΝΝΗΣΟΥ

Εικόνα 10 – Κλάση Table Striped

Η κλάση .table-hover (βλ. Εικόνα 11) βοηθάει στο να αλλάζει το φόντο της γραμμής από την οποία περνάει ο κέρσορας.

Όνομα	Περιφέρεια
ΑΙΤΩΛΟΑΚΑΡΝΑΝΙΑΣ	ΔΥΤΙΚΗΣ ΕΛΛΑΔΑΣ
ΑΝΑΤΟΛΙΚΗΣ ΑΤΤΙΚΗΣ	ΑΤΤΙΚΗΣ
ΑΝΔΡΟΥ	ΝΟΤΙΟΥ ΑΙΓΑΙΟΥ
ΑΡΓΟΛΙΔΑΣ	ΠΕΛΟΠΟΝΝΗΣΟΥ

Εικόνα 11 – Κλάση Table Hover

HTML και κλάσεις Λίστας

Ο κώδικας μίας λίστας HTML είναι ο εξής:

```
<ul class="list-unstyled list-inline">
  <li>...</li>
</ul>
```

Η κλάση list-unstyled δημιουργεί μία λίστα χωρίς bullets. Με την κλάση list-inline τα στοιχεία της λίστας τοποθετούνται στην ίδια γραμμή.

Κλάσεις text και pull Bootstrap

Για την στοίχιση κειμένου μπορεί να χρησιμοποιηθεί η κλάση text ακολουθούμενη την φορά που θέλουμε να έχει το κείμενο:

.text-left => text-align:left

.text-right => text-align:right

.text-center => text-align:center

Για την στοίχιση κάποιου div μπορεί να χρησιμοποιηθεί η κλάση pull ακολουθούμενη την φορά που θέλουμε να έχει το div:

.pull-left => float:left

.pull-right => float:right

Bootstrap Modal

Για την χρήση ενός pop-up παραθύρου χρησιμοποιούμε το Bootstrap Modal (βλ. Εικόνα 12). Ο κώδικας HTML είναι ο εξής:

```
<div class="modal fade" id="succes_modal">
        <div class="modal-dialog">
          <div class="modal-content">
           <div class="modal-header">
              <h4 class="modal-title text-center text-success"><big><span class="glyphicon
        glyphicon-ok"></span></big></h4>
            </div>
```

```
        <div class="modal-body text-center">
          <p>Οι αλλαγές σας αποθηκεύτηκαν με επιτυχία</p>
        </div>
        <div class="modal-footer text-center">
          <button type="button" class="btn btn-primary"
      data-dismiss="modal">Κλείσιμο</button>
        </div>
      </div>
    </div>
  </div>
```

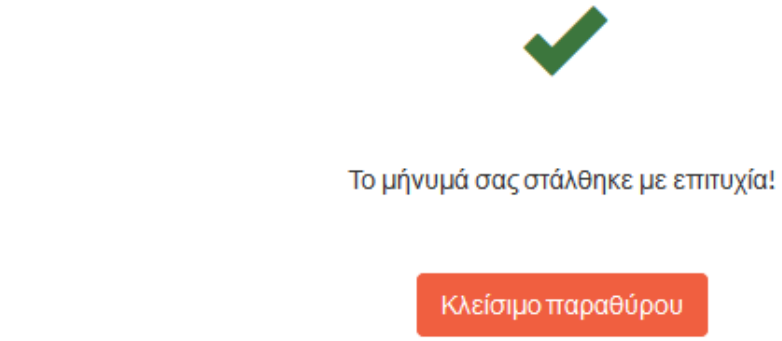
Το modal αποτελείται από τρεις κλάσεις:

a) .modal-header , στην οποία υπάρχει ένας τίτλος
b) .modal-body, στην οποία υπάρχει το περιεχόμενο
c) .modal-footer, στην οποία εισάγεται ένα κουμπί που κλείνει το modal χρησιμοποιώντας την ιδιότητα data-dismiss="modal".

Για να μπορέσει το modal να εμφανιστεί θα πρέπει να καλεστεί η παρακάτω jQuery εντολή:

```
$('#succes_modal).modal();
```

(Όπου succes_modal το id του modal)



Εικόνα 12 – Bootstrap Modal

Bootstrap Popover

Ένα popover είναι ένα παράθυρο φούσκας που ανοίγει δίπλα από ένα κουμπί όταν αυτό πατηθεί (βλ. Εικόνα 13).

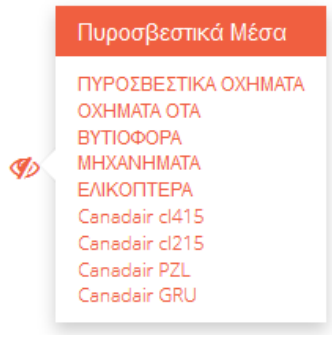
```
<button type="button" id="staff_popover" class="btn btn-lg btn-danger btn-popover" data-toggle="popover" title="Προσωπικό" data-content="…"><i class="fa fa-eye"></i></button>
```

Στην ιδιότητα 'data-content' εισάγεται το περιεχόμενο που απεικονίζεται όταν το popover ενεργοποιηθεί.

Για την ενεργοποίηση του popover πρέπει να εισαχθεί η ακόλουθη jQuery εντολή:

```
$('[data-toggle="popover"]').popover({html: true});
```

Η ιδιότητα 'html' έχει τιμή 'true', προκειμένου να μπορέσει το popover να αναπαραστήσει κώδικα HTML.



Εικόνα 13 – Bootstrap Popover

Bootstrap Tooltip

Όπως το popover ένα tooltip είναι ένα παράθυρο φούσκας που ανοίγει δίπλα από ένα κουμπί όταν ο κέρσορας περάσει από πάνω του (βλ. Εικόνα 14).

```
<a href="#" class="modify" data-toggle="tooltip" data-placement="top" title="Επεξεργασία"><i class="fa fa-pencil-square-o"></i></a>
```

Στην ιδιότητα 'title' εισάγεται το περιεχόμενο που απεικονίζεται όταν το tooltip ενεργοποιηθεί (εδώ Επεξεργασία). Με την ιδιότητα 'data-placement' καθορίζεται η θέση στην οποία θα απεικονιστεί το tooltip (εδώ top).

Για την ενεργοποίηση του tooltip πρέπει να εισαχθεί η ακόλουθη jQuery εντολή:

```
$('[data-toggle="tooltip"]').tooltip();
```

Εικόνα 14 – Bootstrap Tooltip

Media Queries

Το Bootstrap χρησιμοποιεί τα media queries στα αρχεία CSS, προκειμένου κάποιοι κανόνες CSS να αλλάζουν ανάλογα με τις απαιτήσεις της ανάλυσης της οθόνης (tablet-κινητό-μεγάλη οθόνη υπολογιστή) [27].

Η γραφή ενός media query έχει ως εξής:

```
@media(συνθήκη) {
    Κανόνες CSS
}
```

, όπου η συνθήκη ορίζει το εύρος ή την τιμή του μεγέθους της ανάλυσης της οθόνης.

Παράδειγμα:

```
@media(width:768px) {
  div#mapcontainer {
    height: 600px;
  }
}
```

Εδώ, το div με id mapcontainer, έχει ύψος 600px όταν η συσκευή έχει πλάτος 768px. Όταν δεν έχει η συσκευή πλάτος 768px, θα εφαρμοστεί ο κανόνας CSS ο οποίος είναι έξω από το κομμάτι των εντολών CSS που ορίζει το συγκεκριμένο media query.

B. Blade Laravel Template

Πριν την ανάλυση του php framework Laravel, είναι απαραίτητη η κατανόηση της γραφής Blade την οποία χρησιμοποιεί το Laravel [28]. Αυτό επειδή η συγκεκριμένη γραφή αναπαριστά με διαφορετικό τρόπο τις HTML ετικέτες, καθώς και τον κώδικα PHP ο οποίος χρησιμοποιείται σε ένα αρχείο που περιέχει HTML κώδικα.

Κάθε αρχείο προκειμένου να μπορέσει να αναπαραστήσει τον κώδικα Blade, θα πρέπει να αποθηκευτεί με την κατάληξη .blade.php

Στην γραφή Blade χρησιμοποιούνται οι αγκύλες {{ }} οι οποίες είναι ισοδύναμες όπως ο κώδικας <?php echo “”; ?> στην php.

Blade HTML ετικέτες

Για την καλύτερη κατανόηση της απεικόνισης των ετικετών HTML, δίνεται ο κώδικας HTML και ο αντίστοιχος που προκύπτει με την Blade γραφή [29].

Επάνω είναι ο κώδικας HTML ενώ κάτω ο αντίστοιχος σε Blade.

Ετικέτα style:

```
<link type="text/css" rel="stylesheet" href="css/bootstrap-select.min.css'">
{{HTML::style('css/bootstrap-select.min.css')}}
```

Ετικέτα script:

```
<script src="js/bootstrap.min.js"></script>
{{HTML::script('js/bootstrap.min.js')}}
```

Ετικέτα input:

```
<input name="latitude" type="text" value="" class=”form-control” id=”latitude”>
{{Form::text('latitude','', array('id' => 'latitude', 'class' => 'form-control'))}}
```

Ετικέτα input hidden:

```
<input name="id_force" type="hidden" value="1" id=”id_force”>
{{Form::hidden('id_force', '1', array('id' => 'id_force'))}}
```

Ετικέτα textarea:

```
<textarea class= “form-control" name="message" cols="30" rows="8" placeholder="'To μήνυμά σας..." id=”message”></textarea>
{{ Form:: textarea ('message', '', array('class' => 'form-control', 'placeholder' => 'To μήνυμά σας...', 'size' => '30x8' , 'id' => 'message'))}}
```

Ετικέτα password:

```
<input class="'form-control" name="password" type="password" placeholder="…” id=”password”>
{{Form::password('password', array('id' => 'password', 'class' => 'form-control','placeholder' => '...'))}}
```

Ετικέτα form:

```
<form method="POST" action="'fires/export"></form>
{{ Form::open(array('url' => 'fires/export',  'method' => 'post')) }} {{Form::close()}}
```

Κουμπί form submit:

```
<input class="btn btn-success" type="submit" value="Αποστολή Μηνύματος" id="contact-submit">
{{ Form::submit('Αποστολή Μηνύματος', array('class' => 'btn btn-success', 'id' => 'contact-submit')) }}
```

Link a:

```
<a href="fire/insert">
<a href="{{URL::to ('fire/insert')}}"
```

Επαναληπτική διαδικασία foreach:

Η επαναληπτική διαδικασία ανοίγει με την εντολή @foreach και τελειώνει με την εντολή @endforeach. Μέσα στην παρένθεση της foreach εισάγεται η συνθήκη.

Για παράδειγμα ο παρακάτω κώδικας σε Blade:

```
@foreach ($regions as $region)
    {{$region->name}}
@endforeach
```

Αντιστοιχεί με τον ακόλουθο στην απλή php:

```
<?php
foreach ($regions as $region) {
    echo $region->name;
} ?>
```

Συνθήκη if – else:

Η συνθήκη if-else ανοίγει με την εντολή @if και τελειώνει με την εντολή @endif. Μέσα στην παρένθεση της if εισάγεται η συνθήκη.

Για παράδειγμα ο παρακάτω κώδικας σε Blade:

```
@if (isset($section) && $section == 'edit')
    <p class="lead">Επεξεργασία πυρκαγιάς</p>
@else
    <p class="lead">Εισαγωγή πυρκαγιάς</p>
@endif
```

Αντιστοιχεί με τον ακόλουθο στην απλή php:

```
<?php
if (isset($section) && $section == 'edit') { ?>
    <p class="lead">Επεξεργασία πυρκαγιάς</p>
<?php } else { ?>
    <p class="lead">Εισαγωγή πυρκαγιάς</p>
<?php } ?>
```

Εάν υπάρχουν περισσότερες συνθήκες τότε η συνθήκη if γίνεται:

```
@if (συνθήκη)
@elseif (συνθήκη)
@elseif (συνθήκη)
@else
@endif
```

Yield – Section - Include

Το blade template επιτρέπει να εισάγονται ίδιου τύπου κομμάτια κώδικα σε κάποια σημεία της σελίδας. Για να επιτευχθεί αυτό χρειάζεται να υπάρχει μία σελίδα η οποία αποτελεί το layout master και χρησιμοποιείται ως 'πατέρας' πάνω στην οποία αναφέρονται οι διάφορες σελίδες ως 'παιδιά' και οι οποίες ακολουθούν την ίδια δομή.

Με την εντολή yield ορίζεται το σημείο στον κώδικα της σελίδας HTML μέσα στον οποίο υπάρχει κάποιος τύπος κώδικα, όπως για παράδειγμα ο κώδικας που ορίζει τον τίτλο της σελίδας ή τα μονοπάτια των αρχείων CSS και JS. Οι εντολές yield εισάγονται στην σελίδα που χρησιμοποιείται ως 'πατέρας' (master).

Οι εντολές section εισάγονται στις σελίδες 'παιδιά' οι οποίες χρησιμοποιούν το master template. Αυτές αναφέρονται στις αντίστοιχες εντολές yield που έχουν εισαχθεί. Μία εντολή section ανοίγει ως @section και κλείνει ως @stop. Ανάμεσα στις @section και @stop εισάγεται ο κώδικας.

Για παράδειγμα, μέσα στην ετικέτα title κάθε σελίδας θέλουμε να εισάγουμε το όνομα της σελίδας. Στην σελίδα master χρησιμοποιείται η εντολή @yield('title'). Το όνομα title είναι αυτό που επιλέχτηκε. Θα μπορούσε να ήτανε κάποιο άλλο, όπως titleHead. Επίσης, με τον ίδιο τρόπο χρησιμοποιούνται οι εντολές @yield('extraStyle'), @yield('content'), @yield('extraModal') και @yield('extraScript').

Για την καλύτερη κατανόηση χρησιμοποιείται ο παρακάτω πίνακας παραδειγμάτων (Πίνακας 1), ο οποίος αποτελείται αριστερά από τις εντολές yield που ορίζονται και δεξιά το πώς αυτές οι εντολές αποτυπώνονται από τις αντίστοιχες section εντολές.

@yield('title')	*@section('title')* *'GrrekFires'* *@stop*
@yield('extraStyle')	*@section('title')* *{{HTML::style('css/bootstrap.min.css')}}* *@stop*
@yield('content')	*@section('content')* * …. * *@stop*
@yield('extraModal')	*@section('extraModal')* * * *…* * * *@stop*
@yield('extraScript')	*@section('extraScript')* *{{HTML::script('js/jquery.form.min.js')}}* *@stop*

Πίνακας 1 – Yield και Section εντολές

Η εντολή include αρχίζει και αυτή με το σύμβολο @ και ορίζει το μονοπάτι του αρχείου PHP το οποίο καλείται. Για παράδειγμα η εντολή @include('admin.shared.header'), θα εισάγει στην σελίδα το κομμάτι του κώδικα το οποίο υπάρχει στο μονοπάτι admin/shared/header (στο laravel τα μονοπάτια εισάγονται με τελείες).

3.2.2 Server-side Τεχνολογίες

A. Laravel (MVC PHP Framework)

Το Laravel είναι ένα μοντέρνο PHP framework το οποίο χρησιμοποιείται για την κατασκευή διαδικτυακών εφαρμογών. Το laravel δημιουργήθηκε από τον Taylor Otwell τον Ιούνιο του 2011, ενώ τον Μάιο του 2014 κυκλοφόρησε η έκδοση 4.2, η οποία χρησιμοποιείται σε αυτή την εργασία. Η τελευταία έκδοση 5.1 κυκλοφόρησε τον Ιούνιο του 2015 [30].

Η αρχιτεκτονική του Laravel βασίζεται στο MVC design pattern, ενώ αυτό που το κάνει να ξεχωρίζει από τα υπόλοιπα framework είναι το απλό και διαισθητικό συντακτικό, το οποίο επιτρέπει την αποτύπωση πολλών αποτελεσμάτων με λίγες γραμμές κώδικα.

Για την εγκατάσταση του laravel χρειάζεται να έχει αρχικά εγκατασταθεί ένας Composer [31]. Ο Composer είναι μία εφαρμογή η οποία αναλαμβάνει να κατεβάσει και να ενημερώσει τα διάφορα πακέτα PHP που χρησιμοποιούνται στην εφαρμογή. Το ίδιο το Laravel είναι ένα PHP πακέτο το οποίο στηρίζεται με τη σειρά του σε άλλα πακέτα. Επίσης, η εφαρμογή Laravel απαιτεί ο server να υποστηρίζει μία έκδοση PHP >= 5.4

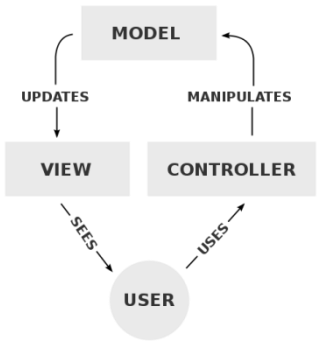
Μετά την εγκατάσταση του composer το laravel εγκαθίσταται τοπικά στον server που χρησιμοποιείται μέσω ενός command line με την εκτέλεση της εντολής:

```
composer create-project laravel/laravel 4.2 --prefer-dist
```

Για την εργασία ο φάκελος laravel που δημιουργήθηκε μετονομάστηκε σε Fires που είναι το όνομα του project.

Model–view–controller (MVC)

Το Model–view–controller (σε συντομογραφία αναφέρεται ως MVC) είναι ένα μοντέλο αρχιτεκτονικής λογισμικού, το οποίο χρησιμοποιείται για την δημιουργία περιβαλλόντων αλληλεπίδρασης χρήστη [32]. Στο μοντέλο αυτό η εφαρμογή διαιρείται σε τρία διασυνδεδεμένα μέρη, ώστε να διαχωριστεί η παρουσίαση της πληροφορίας στον χρήση από την μορφή που έχει αποθηκευτεί στο σύστημα. Το κύριο μέρος του μοντέλου είναι το αντικείμενο *Model* το οποίο διαχειρίζεται την ανάκτηση/αποθήκευση των δεδομένων στο σύστημα. Το αντικείμενο *View* χρησιμοποιείται μόνο για να παρουσιάζεται η πληροφορία στον χρήστη (π.χ. με γραφικό τρόπο μέσω των σελίδων). Το τρίτο μέρος είναι ο *Controller* ο οποίος δέχεται την είσοδο και στέλνει εντολές στο αντικείμενο *Model* και στο *View,* όπως απεικονίζει η παρακάτω εικόνα (Εικόνα 15):



Εικόνα 15 – MVC (Model-View-Controler)

Models

Τα Models είναι για το Laravel οι κλάσεις πάνω στις οποίες δημιουργούνται τα αντικείμενα.

Στην σύνταξη της κλασικής PHP 5, εκτός Laravel, για να ορίσουμε μία κλάση και εν συνεχεία να την καλέσουμε, ορίζουμε ένα αρχείο php το όνομα του οποίου αποτελεί το όνομα της κλάσης, π.χ. Fire.php. Σε αυτό το αρχείο εισάγουμε το εξής κομμάτι κώδικα:

```
<?php
    class Fire {
        Συναρτήσεις…
    }
?>
```

Στο κομμάτι των συναρτήσεων, έχουμε τις συναρτήσεις εκείνες τις οποίες μπορεί να χρησιμοποιήσει ένα αντικείμενο της κλάσης Fire όταν αυτό δημιουργηθεί. Οι συναρτήσεις αυτές συνδέουν την βάση δεδομένων με το αντικείμενο της κλάσης.

Το αντικείμενο με όνομα fire της κλάσης Fire είναι μία μεταβλητή η οποία ορίζεται ως εξής:

```
    $fire = new Fire();
```

Έστω η συνάρτηση updateFire($id) της κλάσης Fire. Για να χρησιμοποιηθεί η συγκεκριμένη συνάρτηση, χρησιμοποιείται ένας pointer (βέλος ->) που δείχνει σε αυτή με το αντικείμενο της κλάσης, δηλαδή:

```
    $fire->updateFire ($id);
```

Η παραπάνω διαδικασία χρησιμοποιείται στην γραφή της PHP 5. Για το Laravel, οι κλάσεις που αποτελούν τα Models, εισάγονται στον φάκελο models. Αντιστοίχως, το όνομα του αρχείου είναι το ίδιο με το όνομα του μοντέλου. Στο μοντέλο ορίζεται ο πίνακας της βάσης δεδομένων πάνω στον οποίο αναφέρεται, καθώς και οι διάφορες συναρτήσεις.

Για παράδειγμα, το μοντέλο BurntArea έχει ως εξής:

```
<?php
1.  use Illuminate\Database\Eloquent\SoftDeletingTrait;
2.  class BurntArea extends Eloquent {
3.      protected $table = 'burnt_area';
4.      public $timestamps = true;
5.      use SoftDeletingTrait;
6.      protected $dates = ['deleted_at'];
7.      public function fire()
8.      {
9.          return $this->belongsTo('Fire', 'id_fire');
10.     }
11. }
```

Στην γραμμή 1 ορίζεται ότι το μοντέλο χρησιμοποίει τις μεθόδους SoftDeleting για να εισαχθεί στον πίνακα burnt_area η ημερομηνία διαγραφής (στήλη deleted_at) κατά την διαγραφή μία εγγραφής.

Στην γραμμή 2 ορίζεται το όνομα του μοντέλου και ότι αυτό αναφέρεται στο μοντέλο Eloquent. Το μοντέλο Eloquent αποτελεί μοντέλο του Laravel, το οποίο χρησιμοποιείται για πιο εύκολη γραφή ερωτημάτων στην βάση δεδομένων, με αναφορά στους πίνακες της βάσης δεδομένων μέσω των μοντέλων. Η ανάλυσή του θα γίνει παρακάτω.

Στην γραμμή 3 ορίζεται στην μεταβλητή table το όνομα του πίνακα στον οποίο αναφέρεται το μοντέλο (εδώ 'burnt_area').

Στις γραμμές 4-6 ορίζονται κανόνες και μεταβλητές οι οποίες χρησιμοποιούνται για τις datetime στήλες created_at, updated_at, deleted_at του πίνακα.

Στις γραμμές 7-9 ορίζεται μία συνάρτηση του μοντέλου την οποία το αντικείμενο του μοντέλου $burtn_area μπορεί να την χρησιμοποιήσει ως εξής:

$burtn_area->fire();

Routing

Το routing γίνεται μέσω του αρχείου routes.php, το οποίο ορίζεται στον φάκελο app του Laravel (app/routes.php). Αυτό περιέχει τα μονοπάτια των Url με βάση τα οποία θα εκτελεστούν οι συναρτήσεις, οι οποίες ορίζονται στα αρχεία των Controllers και οι οποίες εμφανίζουν τις σελίδες (views) ή επιστρέφουν κάποια αποτελέσματα. Εάν ο route που δίνεται με βάση το URL της σελίδας δεν υπάρχει, τότε εμφανίζεται η σελίδα λάθους.

Υπάρχουν δύο τύποι routing, ο post και ο get. Ο post χρησιμοποιείται όταν κάποιες παράμετροι-τιμές δίνονται στην συνάρτηση ενώ ο get για να εμφανίσει την πληροφορία στις σελίδες.

Παράδειγμα post :

```
Route::post('get/fire/informations/by/id',array(
                    'uses'=>'HomeController@getFireInformationsById')
        );
```

Παράδειγμα get :

```
Route::get('/admin', array('uses' => 'AdminController@loginCreate'));
```

Επιπλέον, υπάρχει η δυνατότητα ονομασίας ενός route και αυτό βοηθάει για να χρησιμοποιηθεί το όνομα αυτό από κάποια συνθήκη.

Παράδειγμα route ονόματος:

```
Route::get('fires/list', array('as' => 'fires_list', 'uses' => 'FireController@firesListCreate'));
```

Ο route με όνομα fires_list χρησιμοποιείται από την συνάρτηση currentRouteName, προκειμένου να εισάγει την κλάση active στο μέλος της λίστας της σελίδας την οποία επισκέπτεται ο χρήστης.

Για παράδειγμα:

```
<li class="{{ Route::currentRouteName() == 'fires_list' ? 'active' : ''; }}">
    <a href="{{URL::to('fires/list')}}">Λίστα πυρκαγιών</a>
</li>
```

Controllers

Οι Controllers είναι τα αρχεία php τα οποία υπάρχουν στον φάκελο app/controllers, των οποίων οι συναρτήσεις ορίζονται στο αρχείο routes. Περιλαμβάνουν το σύνολο των συναρτήσεων οι οποίες είτε δέχονται κάποιες εισόδους (τιμές) από routes τύπου post προκειμένου να εξάγουν κάποιο αποτέλεσμα, είτε χρησιμοποιούνται για την εμφάνιση της σελίδας (view) μέσω των routes τύπου get.

Για να δημιουργηθεί ένας Controller, π.χ. ο FireController, χρειάζεται μέσω ενός command prompt να εισαχθεί η ακόλουθη εντολή:

```
php artisan controller:make FireController
```

Όπως τα αρχεία Models αναφέρονται στο Eloquent Model έτσι τα αρχεία Controllers αναφέρονται στον BaseController Controller.

Ένας Controller έχει την εξής δομή:

```
<?php
class FireController extends BaseController {
    Σύνολο συναρτήσεων…
}
?>
```

Για την καλύτερη κατανόηση της σύνδεσης Model-View-Controller (MVC) στο Laravel, δίνεται το παρακάτω παράδειγμα το οποίο εμφανίζει την σελίδα της λίστας των πυρκαγιών στον admin.

1. Στο αρχείο routes.php εισάγεται η παρακάτω εντολή:

```
Route::get('fires/list', array('as' => 'fires_list', 'uses' => 'FireController@firesListCreate'));
```

Ο συγκεκριμένος route είναι τύπου get και για να εμφανίσει την σελίδα χρησιμοποιεί την συνάρτηση firesListCreate του Controller FireController.

2. Στο αρχείο FireController.php υπάρχει η συνάρτηση firesListCreate:

```
public function firesListCreate() {
        $fires = Fire::all();
        $data  = array('title' => 'Πυρκαγιές', 'fires' => $fires);
        $this->layout->content = View::make('admin.fires.list')->with($data);
}
```

Η εντολή $fires = Fire::all(); είναι ένα query ερώτημα το οποίο γίνεται μέσω του μοντέλου Fire και το οποίο επιστρέφει όλες τις εγγραφές του πίνακα fires.

Με την εντολή View::make('admin.fires.list')->with($data), εμφανίζεται η σελίδα στον χρήστη η οποία βρίσκεται στο μονοπάτι 'admin.fires.list' και η οποία περιέχει τα δεδομένα που ορίζονται στον πίνακα $data.

Κάθε σελίδα (view) βρίσκεται στον φάκελο app/views. Η συγκεκριμένη σελίδα list βρίσκεται στο αρχείο list.blade.php, το οποίο είναι στο μονοπάτι views/admin/fires/.

Form Validation

Οι συναρτήσεις των Contollers τύπου post είναι αυτές που δέχονται κάποιες εισόδους (Input) όταν υποβάλλεται μια φόρμα. Τα Input αυτά έχουν κάποιους τύπους:

- Τύπος get
  Ορίζεται ως εξής : Input::get('name');
- Τύπος has
  Ορίζεται ως εξής : Input::has ('name') και χρησιμοποιείται για κάποια συνθήκη, για παράδειγμα:

```
if (Input::has('name'))
{
  //
}
```

- Τύπος all
  Ορίζεται ως εξής : Input::all();
  Περιλαμβάνει το σύνολο των Input οι οποίες υποβλήθηκαν από την φόρμα.
- Τύπος old
  Ορίζεται ως εξής : Input::old ('name') και αποτελεί την τιμή η οποία υποβλήθηκε από την φόρμα.

- Τύπος file
  Ορίζεται ως εξής : Input::file('document') και δείχνει ότι ένα Input είναι τύπου αρχείο.

Όταν λοιπόν υποβάλλεται μία φόρμα, τότε γίνεται έλεγχος (Validation) για το εάν όλες οι τιμές των Input της φόρμας έχουν συμπληρωθεί. Για την καλύτερη κατανόηση της λειτουργίας του Validation δίνεται το ακόλουθο παράδειγμα:

Έστω η φόρμα που υπάρχει στην σελίδα (view) forces/list.

```
{{ Form::open(array('url' => 'force/insert',  'method' => 'post')) }}
```

Το url της φόρμας είναι το 'force/insert' και είναι τύπου post. Επομένως, ο route με url το 'force/insert' θα είναι και αυτός τύπου post και είναι ο εξής:

```
Route::post('force/insert', array('uses' => 'ForceController@forceInsertStore'));
```

Στην φόρμα υπάρχουν τα εξής τρία Inputs:

```
a)  {{Form::text('name', Input::old('name'), array('id' => 'name', 'class' => 'form-control'))}}
```

```
b) {{Form::text('type', Input::old('type'), array('id' => 'type', 'class' => 'form-control', 'placeholder' => 'π.χ.: other_forces'))}}
c) <select class="form-control selectpicker" title='Επιλέξτε μέσο' id="id_force_type" name="id_force_type">
        <option class="hide" value="0"></option>
        @foreach ($forces_types as $forces_type)
            <option value="{{$forces_type->id}}">{{$forces_type->name}}</option>
        @endforeach
    </select>
```

Όταν η φόρμα υποβληθεί μέσω του κουμπιού τύπου submit, τότε ο route καλεί την συνάρτηση forceInsertStore του Controller 'ForceController'.

Η συνάρτηση αυτή έχει ως εξής:

```
    public function forceInsertStore()
     {
       $rules = array(
         'name'          => array('required'),
         'type'          => array('required'),
         'id_force_type' => array('required')
       );
       $validation = Validator::make(Input::all(), $rules);

       if ($validation->fails())
       {
          return Redirect::to('forces/list')->withErrors($validation)->withInput()->with(array('error'=>'insert'));
       }
       else
        {
          if (Input::has('id_force_type') && Input::get('id_force_type') == 0)
            return Redirect::to('forces/list')->withErrors($validation)->withInput()->with(array('error'=>'insert', 'force_type_empty' => true));

          $force               = new Force();
          $force->name         = Input::get('name');
          $force->type         = Input::get('type');
          $force->id_force_type = Input::get('id_force_type');
          $force->created_at    = date("Y-m-d H:i:s");
          $force->updated_at    = date("Y-m-d H:i:s");
          $force->save();

          return Redirect::to('/forces/list')->with(array('success'=> true));
       }
    }
```

Στην συνάρτηση αυτή γίνεται έλεγχος για το εάν έχουν σωστά υποβληθεί οι τιμές των Input.

Ορίζεται αρχικά μία μεταβλητή η $rules, η οποία είναι ένας πίνακας με τιμές τα ονόματα των Input (name, type, id_force_type). Οι τιμές αυτές είναι τύπου 'required', καθώς είναι υποχρεωτικές για την εισαγωγή ενός μέσου.

Εν συνεχεία ορίζεται η μεταβλητή $validation = Validator::make(Input::all(), $rules). Με το μοντέλο Validator το Laravel παίρνει όλα τα Input με τις τιμές τους (Input::all(), $rules) και με την συνθήκη: if ($validation->fails()), ελέγχει για το εάν έχουν υποβληθεί σωστά τα Input της φόρμας.

Εάν η συνθήκη ισχύει, τότε μέσω της εντολής: return Redirect::to('forces/list')->withErrors($validation)->withInput()->with(array('error'=>'insert')); , εφαρμόζει τον route που επιστρέφει στην σελίδα (view) forces/list, με τα λάθη που προέκυψαν μέσω της συνάρτησης withErrors($validation) και έχοντας τις τιμές των Input μέσω της συνάρτησης withInput(). Στην

συνάρτηση with(array('error'=>'insert')), έχει εισαχθεί μία μεταβλητή η error η οποία έχει τιμή insert και η οποία χρησιμοποιείται για την προβολή κατάλληλου μηνύματος λάθους.

Αντιθέτως, εάν η συνθήκη δεν ισχύει τότε η συνάρτηση δημιουργεί μία νέα μεταβλητή $force του μοντέλου Force, η οποία εισάγει με την βοήθεια των pointers τις τιμές των Input στις αντίστοιχες στήλες του πίνακα forces, ο οποίος ορίζεται στο μοντέλο Force. Μετά από αυτή την διαδικασία, εκτελείται η εντολή: return Redirect::to('/forces/list')->with(array('success'=> true)), η οποία εφαρμόζει τον route που επιστρέφει στην σελίδα (view) forces/list, με την μεταβλητή success να έχει την τιμή true για την προβολή κατάλληλου μηνύματος.

Β. Laravel και Βάση Δεδομένων

Το Laravel δημιουργεί την σύνδεση με την βάση δεδομένων όπως την ορίζει στο αρχείο database.php, το οποίο βρίσκεται στο μονοπάτι app/config. Στο αρχείο αυτό μπορούν να οριστούν ο τύπος της βάσης δεδομένων, για παράδειγμα mysql, καθώς και το όνομα και τα στοιχεία της βάσης (όνομα, κωδικός, κωδικοποίηση κ.ά.).

Στην εργασία αυτή το αρχείο αυτό έχει ως εξής:

```
'mysql' => array(
        'driver'    => 'mysql',
        'host'      => 'localhost',
        'database'  => 'greek_fires',
        'username'  => 'root',
        'password'  => '',
                'charset'   => 'utf8',
                'collation' => 'utf8_unicode_ci',
                'prefix'    => '',
        )
```

Τα queries (ερωτήματα) στην βάση δεδομένων μπορούν να γίνουν με δύο τρόπους στο Laravel:

Α) Χρησιμοποιώντας το μοντέλο DB στο οποίο ορίζεται το όνομα του πίνακα στον οποίο γίνονται τα ερωτήματα

Παράδειγμα:

```
$users = DB::table('users')->get();
```

Η μεταβλητή $users παίρνει με την μέθοδο get όλες τις εγγραφές του πίνακα users.

Β) Χρησιμοποιώντας το όνομα του μοντέλου το οποίο περιλαμβάνει το όνομα του πίνακα και το οποίο χρησιμοποίει το μοντέλο Eloquent.

Το παραπάνω παράδειγμα με αυτόν τον τρόπο γίνεται:

```
$users = User::all();
```

Η μεταβλητή $users παίρνει με την μέθοδο all όλες τις εγγραφές του πίνακα users, ο οποίος ορίζεται στο μοντέλο User.

Με τον Β τρόπο, επιτυγχάνεται η γραφή λιγότερου και πιο ευδιάκριτου κώδικα, καθώς ορίζει το μοντέλο στο οποίο γίνονται τα ερωτήματα. Στην συγκεκριμένη εργασία γίνεται η ανάλυση των ερωτημάτων του Β τρόπου χρησιμοποιώντας το μοντέλο Eloquent.

Γ. Eloquent και Queries (Ερωτήματα) στην Βάση Δεδομένων [33]

Select Query

Με το ερώτημα select γίνεται η επιλογή όλων ή κάποιων εγγραφών από τον πίνακα.

Μέθοδος all

Με την μέθοδο all γίνεται η επιλογή όλων των εγγραφών του πίνακα.

Για παράδειγμα, η επιλογή όλων των εγγραφών του πίνακα ‘regions’, ο οποίος ορίζεται στο μοντέλο ‘Region’, γίνεται με την παρακάτω εντολή:

```
$regions = Region::all();
```

Αυτή η εντολή στην Mysql αντιστοιχεί στο ερώτημα: SELECT * FROM regions.

Μέθοδος get

Με την μέθοδο get γίνεται η επιλογή των εγγραφών του πίνακα, όπως ορίζεται στις συνθήκες WHERE που περιλαμβάνει.

Για παράδειγμα, για να επιλεγούν τα πυροσβεστικά τμήματα μίας περιφέρειας μέσω του id της περιφέρειας (id_region), εκτελείται το ακόλουθο ερώτημα:

```
$fire_departments = FireDepartment::where('id_region', '=', $region->id)->get();
```

Αυτή η εντολή στην Mysql αντιστοιχεί στο ερώτημα:

```
SELECT * FROM fire_departments WHERE id_region=:id_region.
```

Με την συγκεκριμένη μέθοδο, μπορεί να γίνει συνδυασμός μεθόδων (take, ordeBy, groupBy) με σκοπό την επιλογή εγγραφών όπως ορίζουν αυτές οι μέθοδοι.

Για παράδειγμα, για την επιλογή των 10 πυρκαγιών με χρονολογική σειρά την ημερομηνία έναρξής τους, εκτελείται το παρακάτω ερώτημα:

```
$latest_fires = Fire::select('fires.*')->orderBy('start_date','desc')->take(10)->get();
```

Μέθοδος first

Με την μέθοδο first γίνεται η επιλογή μίας μόνο εγγραφής του πίνακα, όπως ορίζεται στις συνθήκες WHERE που περιλαμβάνει.

Για παράδειγμα, για να επιλεγεί ο Νομός στον οποίο το όνομα ορίζεται στην μεταβλητή $prefecture, εκτελείται το παρακάτω ερώτημα:

```
$prefecture_exists  = Prefecture::where('name', 'like', $prefecture)->first();
```

Αυτή η εντολή στην Mysql αντιστοιχεί στο ερώτημα:

```
SELECT * FROM prefectures WHERE name LIKE $prefecture.
```

Μέθοδος find

Με την μέθοδο find γίνεται η επιλογή μίας μόνο εγγραφής του πίνακα, όπως ορίζεται στην συνθήκη της μεθόδου.

Για παράδειγμα, για να επιλεγεί η πυρκαγιά στην οποία το id έχει δοθεί από μία Input τιμή, την Input::get('id_fire'), εκτελείται το παρακάτω ερώτημα:

```
$fire = Fire::find(Input::get('id_fire'));
```

Αυτή η εντολή στην Mysql αντιστοιχεί στο ερώτημα:

```
SELECT * FROM fires WHERE id=:Input::get('id_fire').
```

Μέθοδος count

Με την μέθοδο count επιστρέφεται ο αριθμός των εγγραφών του πίνακα, όπως ορίζεται στις συνθήκες WHERE τις οποίες ενδέχεται να περιλαμβάνει.

Για παράδειγμα, για να επιλεγεί ο αριθμός των πυρκαγιών, εκτελείται το ακόλουθο ερώτημα:

```
$count_fires = Fire::count();
```

Αυτή η εντολή στην Mysql αντιστοιχεί στο ερώτημα:

```
SELECT COUNT(*) as count FROM fires.
```

Insert Query

Για την εισαγωγή μίας νέας εγγραφής σε έναν πίνακα, πρέπει πρώτα να δημιουργηθεί ένα νέο μοντέλο, στο οποίο ο πίνακας αυτός ορίζεται. Η αποθήκευση αυτή γίνεται μέσω της μεθόδου save().

Για παράδειγμα, για να γίνει μία νέα εισαγωγή στον πίνακα staff, εκτελείται ο ακόλουθος κώδικας:

```
$staff              = new Staff();
$staff->name        = Input::get('name');
$staff->type        = Input::get('type');
$staff->created_at  = date("Y-m-d H:i:s");
$staff->updated_at  = date("Y-m-d H:i:s");
$staff->save();
```

Αρχικά, δημιουργείται ένα νέο αντικείμενο ($staff) του μοντέλου Staff με την εντολή:

```
$staff = new Staff();
```

Εν συνεχεία, εισάγονται οι τιμές στις στήλες του πίνακα (name, type, created_at, deleted_at). Ο pointer staff δείχνει κάθε φορά το όνομα της στήλης.

Τέλος, ο pointer staff δείχνει στην μέθοδο save(), με την οποία γίνεται η αποθήκευση της νέας εγγραφής στον πίνακα staff.

Update Query

Για την ανανέωση των τιμών των στηλών μίας εγγραφής σε έναν πίνακα, πρέπει πρώτα να γίνει ο εντοπισμός της συγκεκριμένης εγγραφής με την μέθοδο find(). Εν συνεχεία, εκτελείται η ίδια διαδικασία με το Insert ερώτημα και η αποθήκευση γίνεται και αυτή με την χρησιμοποίηση της μεθόδου save().

Για παράδειγμα, για να γίνει η ανανέωση μίας εγγραφής στον πίνακα staff, εκτελείται ο ακόλουθος κώδικας:

```
$staff              = Staff::find(Input::get('staff_id'));
$staff->name        = Input::get('name');
$staff->type        = Input::get('type');
$staff->created_at  = date("Y-m-d H:i:s");
$staff->updated_at  = date("Y-m-d H:i:s");
$staff->save();
```

Αρχικά, με την μέθοδο find() εντοπίζεται η εγγραφή με την τιμή η οποία δόθηκε στην Input::get('staff_id').

Εν συνεχεία, εισάγονται οι τιμές στις στήλες του πίνακα (name, type, created_at, deleted_at). Ο pointer staff δείχνει κάθε φορά το όνομα της στήλης.

Τέλος, ο pointer staff δείχνει στην μέθοδο save(), με την οποία γίνεται η αποθήκευση και επομένως η ανανέωση της εγγραφής στον πίνακα staff.

Delete Query

Για την διαγραφή μίας εγγραφής από έναν πίνακα, πρέπει πρώτα να γίνει ο εντοπισμός της συγκεκριμένης εγγραφής με την μέθοδο find(). Εν συνεχεία, εκτελείται η διαδικασία της διαγραφής με την χρησιμοποίηση της μεθόδου delete() ή της μεθόδου forceDelete().

Μέθοδος delete

Με την μέθοδο delete(), η εγγραφή δεν διαγράφεται από τον πίνακα αλλά ανανεώνεται η στήλη deleted_at με την ημερομηνία που έγινε η διαγραφή. Έτσι, η εγγραφή συνεχίζει να υπάρχει στον πίνακα αλλά δεν χρησιμοποιείται από τις μεθόδους get, first, find, count.

Για παράδειγμα, για να γίνει η διαγραφή της εγγραφής του πίνακα staff, η οποία αντιστοιχεί στο id που δίνεται από την Input, εκτελείται ο ακόλουθος κώδικας:

```
$staff = Staff::find(Input::get('detete_staff_id'));
$staff->delete();
```

Μέθοδος forceDelete

Με την μέθοδο forceDelete(), η εγγραφή διαγράφεται εξολοκλήρου από τον πίνακα.

Για παράδειγμα, για να γίνει η εξολοκλήρου διαγραφή της εγγραφής του πίνακα staff, η οποία αντιστοιχεί στο id που δίνεται από την Input, εκτελείται ο ακόλουθος κώδικας:

```
$staff = Staff::find(Input::get('detete_staff_id'));
$staff-> forceDelete();
```

Join Query

Ένα join query γίνεται με την μέθοδο join, στην οποία ορίζεται ο πίνακας και τα id των πινάκων στα οποία γίνεται το join ερώτημα. Για παράδειγμα, για να επιλεγούν οι 12 μεγαλύτερες πυρκαγιές ανά νομό εκτελείται το παρακάτω ερώτημα:

```
$fires_by_prefecture = Fire::select(DB::raw('prefectures.name, count(id_prefecture) as count_prefectures'))
->join('prefectures', 'prefectures.id', '=', 'fires.id_prefecture')
->groupBy('fires.id_prefecture')
->orderBy('count_prefectures', 'desc')
->take(12)
->get();
```

Δ. Eloquent Model Relationships

Τα Eloquent Model Relationships χρησιμοποιούνται για την απλούστερη γραφή κώδικα, καθώς εφαρμόζουν τα select και join ερωτήματα μέσω μεθόδων που χρησιμοποιούν, ανάλογα της σχέσης που έχουν δύο μοντέλα μεταξύ τους [34].

Σχέση One To One

Η σχέση αυτή προκύπτει όταν υπάρχει μία σχέση δύο μοντέλων ένα προς ένα (one-to-one relationship). Περιλαμβάνει τις εξής δύο σχέσεις:

Συνάρτηση hasOne

Για παράδειγμα, ανάμεσα στο μοντέλο Fire και BurntArea. Αυτή προκύπτει καθώς μία πυρκαγιά έχει μία καμένη έκταση. Έτσι, δημιουργείται μία σχέση η οποία ορίζεται από την ακόλουθη συνάρτηση burnt_area και η οποία εισάγεται στο μοντέλο Fire.

```
public function burnt_area() {
    return $this->hasOne('BurntArea', 'id_fire');
}
```

Συνάρτηση belongsTo

Είναι η αντίστροφη της σχέσης hasOne . Για παράδειγμα, ανάμεσα στο μοντέλο Fire και BurntArea. Αυτή προκύπτει καθώς μία καμένη έκταση ανήκει σε μία πυρκαγιά. Έτσι, δημιουργείται μία σχέση η οποία ορίζεται από την ακόλουθη συνάρτηση fire και η οποία εισάγεται στο μοντέλο BurntArea.

```
public function fire() {
    return $this->belongsTo('Fire', 'id_fire');
}
```

Σχέση One To Many

Η σχέση αυτή προκύπτει όταν υπάρχει μία σχέση δύο μοντέλων ένα προς πολλά.

Συνάρτηση hasMany

Για παράδειγμα, ανάμεσα στο μοντέλο Prefecture και Municipality. Ένας νομός μπορεί να έχει πολλούς δήμους. Έτσι, δημιουργείται μία σχέση η οποία ορίζεται από την ακόλουθη συνάρτηση municipalities και η οποία εισάγεται στο μοντέλο Prefecture.

```
public function municipalities(){
    return $this->hasMany('Municipality');
}
```

Η αντίστροφη της σχέσης hasMany είναι η belongsTo. Έτσι, για το μοντέλο Municipality η συνάρτηση prefecture ορίζει την σχέση belongsTo.

```
public function prefecture() {
    return $this->belongsTo('Prefecture', 'id_prefecture');
}
```

Σχέση Many To Many (Συνάρτηση belongsToMany)

Η σχέση αυτή προκύπτει όταν υπάρχει μία σχέση πολλά προς πολλά ανάμεσα στα δύο μοντέλα.

Για παράδειγμα, ανάμεσα στα μοντέλα Fire και Force. Έτσι δημιουργείται ο πίνακας fires_forces και η σχέση fires χρησιμοποίει την συνάρτηση belongsToMany στο μοντέλο Force:

```
public function fires(){
return $this->belongsToMany('Fire',  'fires_forces', 'id_force', 'id_fire')->withPivot('number');
}
```

Πίνακες Pivot

Μία σχέση Many To Many εκτελείται ανάμεσα σε δύο μοντέλα τα οποία ορίζουν έναν πίνακα ο οποίος έχει δημιουργηθεί από την σχέση Πολλά προς Πολλά των δύο πινάκων. Για παράδειγμα ανάμεσα στα μοντέλα Staff και Fire με πίνακες staff και fires αντίστοιχα, έχει προκύψει ο πίνακας fires_staff, ο οποίος ορίζεται στο μοντέλο FireStaff. Ο πίνακας fires_staff έχει ως στήλες τα κλειδιά των δύο πινάκων fires και staff (id_fire, id_staff), αλλά και την στήλη number, η οποία υποδηλώνει των αριθμό του προσωπικού που παίρνει μέρος στην πυρκαγιά. Η στήλη αυτή δηλώνεται με την μέθοδο withPivot ως εξής: withPivot('number') και χρησιμοποιείται πάνω στην μέθοδο belongsToMany. Με την χρήση των pivot πινάκων επιτυγχάνεται η πιο εύκολη εφαρμογή ερωτημάτων (queries) πάνω στον πίνακα fires_staff, χρησιμοποιώντας τις μεθόδους detach (για διαγραφή εγγραφής), pivot (για επιλογή εγγραφής), updateExistingPivot (για ανανέωση εγγραφής).

Συνάρτηση detach

Η συνάρτηση detach χρησιμοποείται για να γίνει η διαγραφή μίας εγγραφής από έναν πίνακα μίας σχέσης Many To Many. Για παράδειγμα, για να διαγραφούν οι εγγραφές του πίνακα fires_staff με βάση το id_fire, αρχικά επιλέγονται οι εγγραφές αυτές με την μέθοδο staff() ως εξής:

```
    $fire_staff  = $fire->staff()->get();
```

Η μεταβλητή $fire_staff περιλαμβάνει όλες τις εγγραφές του πίνακα fire_staff με βάση το id_fire. Για την διαγραφή αυτών χρησιμοποιείται μία επαναληπτική διαδικασία, η οποία χρησιμοποιεί την μέθοδο detach. Η μέθοδος detach εκτελείται στην μέθοδο staff του μοντέλου Fire.

```
foreach ($fire_staff as $staff) {
        $fire->staff()->detach($staff->id);
}
```

Συνάρτηση updateExistingPivot

Με την συνάρτηση updateExistingPivot επιτυγχάνεται η ανανέωση της στήλης number με βάση το id_fire και το id_staff του πίνακα fires_staff.

Η συνάρτηση αυτή παίρνει ως παραμέτρους το id_staff με την μεταβλητή $staff_id και τον αριθμό (number) η τιμή του οποίου θα ανανεωθεί στην στήλη number του πίνακα fires_staff.

Η διαδικασία αυτή έχει ως εξής:

```
        $staff_ids = Input::get('staff_ids');
        foreach ($staff_ids as $staff_id) {
                $fire->staff()->updateExistingPivot($staff_id, array('number' =>
Input::get('staff_type_'.$staff_id)), false);
        }
```

Συνάρτηση pivot

Η συνάρτηση pivot χρησιμοποιείται για να επιλεγεί η στήλη number μίας εγγραφής του πίνακα fires_staff με βάση το id_staff και το id_fire.

Για παράδειγμα, ο παρακάτω κώδικας θα εκτυπώσει των αριθμό των πυροσβεστικών μέσων με βάση το id της πυρκαγιάς (id_fire) και το id του προσωπικού (id_staff) του πίνακα fires_staff.

```
$fire_staff  = $fire->staff()->get();
foreach ($fire_staff as $staff) {
        echo $staff->pivot->number;
}
```

Has Many Through

Η συγκεκριμένη σχέση εξυπηρετεί δύο μοντέλα τα οποία μπορούν να συνδέονται μέσω ενός τρίτου. Για παράδειγμα, το μοντέλο Region συνδέεται με το μοντέλο Municipality, καθώς μπορεί να έχει πολλούς δήμους το μοντέλο Municipality μέσω του μοντέλου Prefecture, το οποίο ανήκει στο μοντέλο Region. Δηλαδή, το id του πίνακα regions υπάρχει ως foreign key στον πίνακα prefectures (id_region), του οποίου το id υπάρχει στον πίνακα municipalitities ως foreign key (id_prefecture).

Η σχέση αυτή αποτυπώνεται από την συνάρτηση municipalities στο μοντέλο Region, η οποία χρησιμοποίει την συνάρτηση hasManyThrough.

```
public function municipalities() {
 return $this->hasManyThrough('Municipality', 'Prefecture', 'id_region', 'id_prefecture');
}
```

E. Laravel Schema Designer (LaravelSd)

Το σχεσιακό μοντέλο της βάσης δεδομένων κατασκευάστηκε από το laravelsd [35]. Το Laravel Schema που δημιουργήθηκε περιλαμβάνει τους πίνακες με κάποιες τιμές που υπάρχουν στις στήλες τους, καθώς και το πώς αυτοί συνδέονται μεταξύ τους.

Εισαγωγή – Επεξεργασία Πίνακα

Αρχικά, κατασκευάζεται η βάση δεδομένων δίνοντας το όνομα του project στο laravelsd.com και εν συνεχεία εισάγονται οι πίνακες.

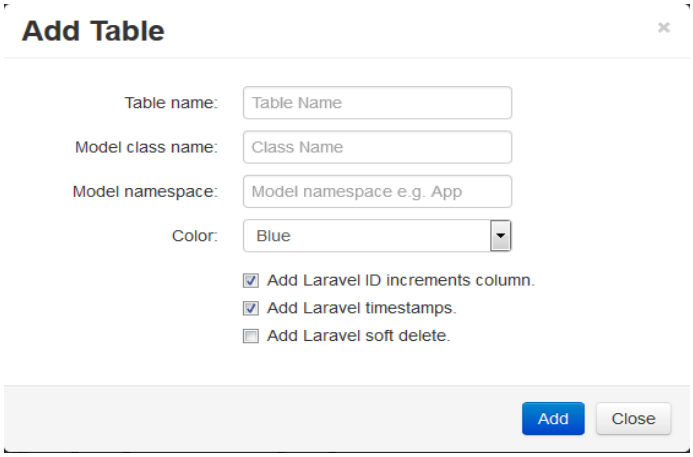
Για την κατασκευή ενός πίνακα στο laravelsd, πατώντας το κουμπί ADD TABLE εμφανίζεται ένα popup (βλ. Εικόνα 16) στο οποίο:

α) εισάγεται το όνομα του πίνακα

β) εισάγεται το μοντέλο (Model) στο οποίο ο πίνακας αναφέρεται

γ) εισάγεται ένα χρώμα (απλά για την απεικόνιση στο laravelsd)

δ) επιλέγεται από τα checkboxes για το εάν ο πίνακας έχει id increment ως Primary Key, για το εάν περιλαμβάνει timestamps (created_at - updated_at στήλες) , καθώς και για το εάν δημιουργείται η στήλη deleted_at με την επιλογή της φράσης Add Laravel soft delete.



Εικόνα 16 – Laravel Schema Designer (Εισαγωγή νέου πίνακα).

Αφού δημιουργηθεί ο πίνακας, μπορεί να γίνει επεξεργασία των στηλών που έχουν εισαχθεί καθώς και η δημιουργία σχέσης (Relationship) μεταξύ αυτού του πίνακα και κάποιου άλλου. Στην ουσία, χρησιμοποιείται το Model του πίνακα το οποίο θα έχει κάποια σχέση με κάποιο άλλο Model (πίνακα). Επίσης, μπορούν να εισαχθούν τιμές στις στήλες του πίνακα οι οποίες αποτελούν τα λεγόμενα Seedings. Όλες οι παραπάνω διαδικασίες μπορούν να γίνουν από τα ανάλογα κουμπιά που υπάρχουν δίπλα στο όνομα του πίνακα (βλ. Εικόνα 17).

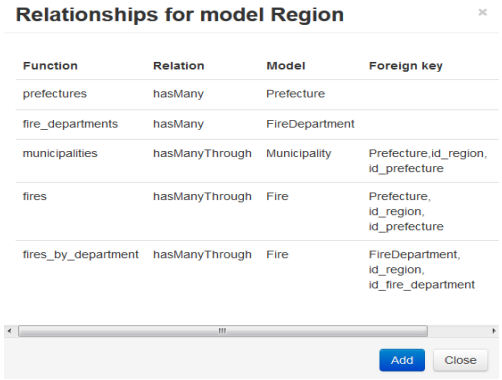
Εικόνα 17 – Κουμπιά Εντολών Πίνακα

Τα κουμπιά από αριστερά προς τα δεξιά είναι τα εξής:

Make relationship – Add seeding – Save Table as template – Edit table – Delete table

Εισαγωγή Σχέσης (Relationship)

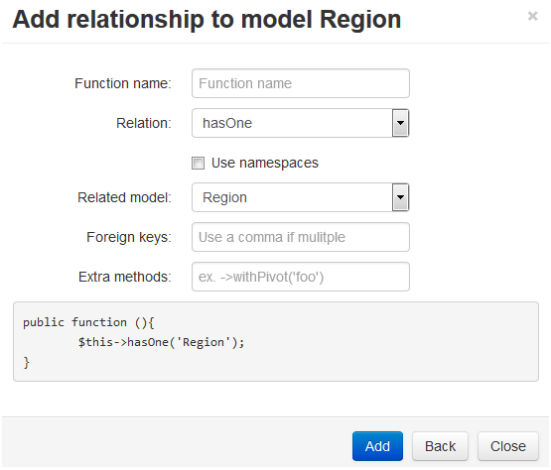
Η εισαγωγή μίας σχέσης μπορεί να γίνει πατώντας το πρώτο κουμπί με τα βελάκια (Make Relationship). Αρχικά ανοίγει ένα popup το οποίο περιλαμβάνει την λίστα όλων των σχέσεων του μοντέλου, όπως δείχνει η παρακάτω εικόνα (Εικόνα 18):



Εικόνα 18 – Model Relationships

Με την επιλογή Add εμφανίζεται ένα νέο παράθυρο στο οποίο εισάγεται το όνομα και ο τύπος της σχέσης, καθώς και εάν από αυτήν θα δημιουργηθούν foreign keys (βλ. Εικόνα 19). Η εισαγωγή Extra methods χρειάζεται για σχέσεις πινάκων Πολλά προς Πολλά, και εισάγεται στην περίπτωση δημιουργίας νέας στήλης στον πίνακα που δημιουργείται από την σχέση.

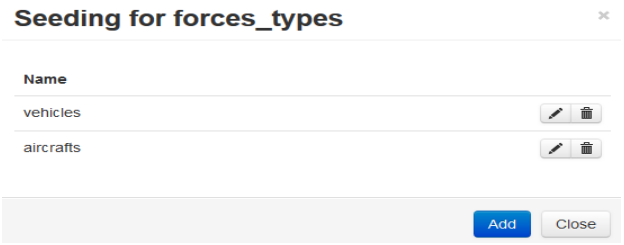
Για παράδειγμα, από την σχέση των πινάκων fires και staff δημιουργείται ο πίνακας fires_staff, που με την εισαγωγή της μεθόδου ->withPivot('number'), μπορεί να χρησιμοποιείται η στήλη number του πίνακα fires_staff.



Εικόνα 19 – Εισαγωγή Σχέσης

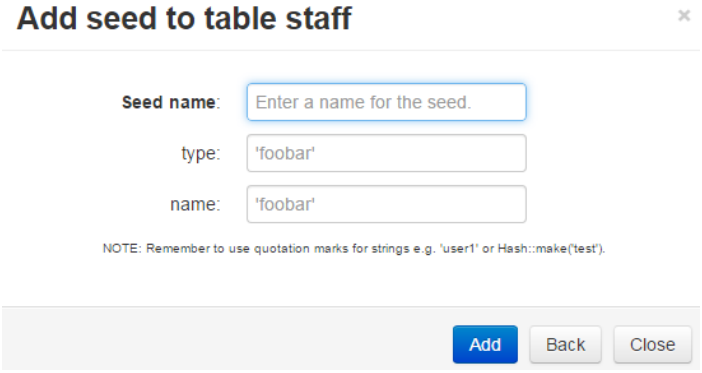
Εισαγωγή Seeding

Η εισαγωγή ενός Seeding μπορεί να γίνει με την επιλογή του κουμπιού Add Seeding. Εάν υπάρχουν ήδη κάποια seedings τότε θα εμφανιστεί η λίστα με αυτά, όπως δείχνει η παρακάτω εικόνα (Εικόνα 20):



Εικόνα 20 – Λίστα Seedings

Με την επιλογή του κουμπιού Add, μπορούν να εισαχθούν οι τιμές στις στήλες του πίνακα, όπως δείχνει η παρακάτω εικόνα (Εικόνα 21):



Εικόνα 21 – Εισαγωγή Seeding

Στην συγκεκριμένη εργασία, seedings έχουν χρησιμοποιηθεί για τους πίνακες forces, forces_types, staff και users.

Export Laravel Schema

Με το laravelsd επιτυγχάνεται η δημιουργία των πινάκων (migrations), των τιμών των πινάκων (seedings), καθώς και η δημιουργία των μοντέλων (models) με τις σχέσεις τους και η δημιουργία των αντίστοιχων controllers.

Με το κουμπί Export All, γίνεται η δημιουργία ενός αρχείου zip, το οποίο περιλαμβάνει τους εξής φακέλους:

Α) migrations

Β) seedings

Γ) models

Δ) controllers

Το περιεχόμενο αυτών των φακέλων αντιγράφεται στους αντίστοιχους φακέλους του Laravel. Οι φάκελοι migrations και seedings βρίσκονται στο μονοπάτι app/database.

Migrations και Seedings

Έπειτα από την αντιγραφή των φακέλων migrations και seedings στο Laravel, απαιτείται να εκτελεστεί η ακόλουθη εντολή από ένα command prompt, προκειμένου να αναγνωριστεί το περιεχόμενο αυτών από το Laravel:

```
php artisan migrate:refresh –seed
```

Η εντολή αυτή δημιουργεί τον πίνακα migrations στην βάση δεδομένων, με βάση τον οποίο κάθε φορά που η παραπάνω εντολή εκτελείται, ελέγχεται εάν έχει αλλάξει κάποιο αρχείο στους φακέλους migrations και seedings, προκειμένου να γίνει ανανέωση αυτών.

3.3 ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Το αρχείο της πυροσβεστικής σε μορφή excel περιλαμβάνει τις ακόλουθες στήλες:

Υπηρεσία, Νομός, Ημερ/νία Έναρξης, Ώρα Έναρξης, Ημερ/νία Κατάσβεσης, Ώρα Κατάσβεσης, Δασαρχείο, Δήμος, Περιοχή, Διεύθυνση, Δάση, Δασική Έκταση, Άλση, Χορτ/κές Εκτάσεις, Καλάμια – Βάλτοι, Γεωργικές Εκτάσεις, Υπολείμματα Καλλιεργειών, Σκουπιδότοποι, ΠΥΡΟΣ. ΣΩΜΑ, ΠΕΖΟΠΟΡΑ ΤΜΗΜΑΤΑ, ΕΘΕΛΟΝΤΕΣ, ΣΤΡΑΤΟΣ, ΑΛΛΕΣ ΔΥΝΑΜΕΙΣ, ΠΥΡΟΣ. ΟΧΗΜ., ΟΧΗΜ. ΟΤΑ, ΒΥΤΙΟΦΟΡΑ, ΜΗΧΑΝΗΜΑΤΑ, ΕΛΙΚΟΠΤΕΡΑ, Α/Φ CL415, Α/Φ CL215, Α/Φ PZL, Α/Φ GRU.

Άμεσα προέκυψε η ανάγκη να χωριστούν αυτές οι στήλες και επομένως να δημιουργηθούν κάποιοι πίνακες.

Η διαίρεση αυτή έγινε σε 4 επίπεδα:

Α) Το πρώτο περιλαμβάνει όλες τις χρονικές και τοπικές πληροφορίες της φωτιάς (Υπηρεσία, Νομός, Ημερ/νία Έναρξης, Ώρα Έναρξης, Ημερ/νία Κατάσβεσης, Ώρα Κατάσβεσης, Δασαρχείο, Δήμος, Περιοχή, Διεύθυνση)

Β) Το δεύτερο περιλαμβάνει όλη την καμένη έκταση της φωτιάς (Δάση, Δασική Έκταση, Άλση, Χορτ/κές Εκτάσεις, Καλάμια – Βάλτοι, Γεωργικές Εκτάσεις, Υπολείμματα Καλλιεργειών, Σκουπιδότοποι)

Γ) Το τρίτο περιλαμβάνει το προσωπικό που παίρνει μέρος στην πυρόσβεση της φωτιάς (ΠΥΡΟΣ. ΣΩΜΑ, ΠΕΖΟΠΟΡΑ ΤΜΗΜΑΤΑ, ΕΘΕΛΟΝΤΕΣ, ΣΤΡΑΤΟΣ, ΑΛΛΕΣ ΔΥΝΑΜΕΙΣ)

Δ) Το τέταρτο περιλαμβάνει τα μέσα που παίρνουν μέρος στην πυρόσβεση της φωτιάς και χωρίζονται σε δύο κατηγορίες σε οχήματα και εναέρια μέσα (ΠΥΡΟΣ. ΟΧΗΜ., ΟΧΗΜ. ΟΤΑ, ΒΥΤΙΟΦΟΡΑ, ΜΗΧΑΝΗΜΑΤΑ, ΕΛΙΚΟΠΤΕΡΑ, Α/Φ CL415, Α/Φ CL215, Α/Φ PZL, Α/Φ GRU).

Από το πρώτο επίπεδο προέκυψαν οι εξής πίνακες:

Α.1) fire_departments (λίστα υπηρεσιών)

Α.2) prefectures (λίστα νομών)

Α.3) municipalities (λίστα δήμων)

Α.4) regions (λίστα περιφερειών)

Α.5) fires

Ο πίνακας Α.5 regions δημιουργήθηκε από την ανάγκη να αποτυπωθούν τα πυροσβεστικά τμήματα σε σχέση με τις 13 περιφέρειες που ανήκουν, όπως αυτό προκύπτει από την κατηγοριοποίηση αυτών από το site της πυροσβεστικής. Επιπλέον, αυτό βοήθησε και στην αποτύπωση των νομών με βάση την περιφέρεια στην οποία ανήκουν.

Από το δεύτερο επίπεδο προέκυψε ο πίνακας:

Β1) burnt_area (καμένη έκταση της κάθε πυρκαγιάς)

Από το τρίτο επίπεδο προέκυψαν οι πίνακες:

Γ1) staff

Γ2) fires_staff

Καθώς πολλοί τύποι προσωπικού ανήκουν σε περισσότερες από μία πυρκαγιές, δημιουργήθηκε ο πίνακας fires_staff.

Από το τέταρτο επίπεδο προέκυψαν οι πίνακες:

Δ1) forces

Δ2) fires_forces

Δ3) forces_types

Καθώς πολλοί τύποι μέσων ανήκουν σε περισσότερες από μία πυρκαγιές, δημιουργήθηκε ο πίνακας fires_forces. Ο πίνακας forces_types δημιουργήθηκε από την ανάγκη να κατηγοριοποιηθούν τα μέσα αυτά ανάλογα με τον τύπο τους (όχημα ή εναέριο).

3.3.1 ΔΗΜΙΟΥΡΓΙΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Πριν την τελική κατασκευή της βάσης δεδομένων, χρειάστηκε να δημιουργηθεί μία βοηθητική βάση δεδομένων, η basic_fires,από την οποία θα μπορούσαν να προκύψουν τα διάφορα ξένα κλειδιά (foreign keys). Για παράδειγμα, παίρνοντας το σύνολο των νομών και των δήμων, για να μπορέσει να υπάρξει το κλειδί του νομού στην στήλη id_prefecture στον πίνακα των δήμων (municipalities), χρειάστηκε να εισαχθεί αρχικά στον πίνακα municipalities στην στήλη prefecture_name το όνομα του νομού. Αυτό έγινε με την χρήση βοηθητικού αρχείου csv το οποίο προέκυψε από την λίστα των νομών, έπειτα από import αυτού στην βάση δεδομένων χρησιμοποιώντας το phpmyadmin.

Ο κατάλογος νομών-δήμων προέκυψε από το αρχείο σε μορφή excel που βρίσκεται στην σελίδα του Υπουργείου Εσωτερικών στην ετικέτα «Κωδικολόγιο Καλλικράτειων Δήμων και Κοινοτήτων».

Έτσι, αφού αρχικά τοποθετήθηκε στον πίνακα municipalities στην στήλη prefecture_name το όνομα του νομού, έπειτα εκτελέστηκε το ακόλουθο ερώτημα το οποίο εισήγαγε το id_prefecture στον πίνακα municipalities:

```
UPDATE municipalities
   JOIN prefectures ON municipalities.prefecture_name = prefectures.name
SET municipalities.id_prefecture = prefectures.id
```

Μετά την επιτυχή εκτέλεση του παραπάνω ερωτήματος, έγινε η εκτέλεση του ερωτήματος:

```
ALTER TABLE `municipalities` DROP `region_name`;
```

, προκειμένου να διαγραφεί η στήλη prefecture_name από τον πίνακα municipalities.

Με ανάλογο τρόπο έγινε και η συμπλήρωση του πίνακα των νομών (prefectures) με βάση τις περιφέρειες (regions). Επίσης, ο πίνακας fire_departments συμπληρώθηκε με την ίδια διαδικασία, αφού πρώτα έγινε η εισαγωγή των στοιχείων των πυροσβεστικών τμημάτων ανά περιφέρεια, όπως αυτά προέκυψαν από την σελίδα της πυροσβεστικής υπηρεσίας.

Έπειτα από την παραπάνω διαδικασία δημιουργήθηκε η τελική βάση δεδομένων greek_fires, της οποίας οι στήλες των πινάκων regions, prefectures, municipalities και fire_departments συμπληρώθηκαν από export της αρχικής βάσης basic_fires και import στην τελική βάση greek_fires.

3.3.2 ΠΙΝΑΚΕΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

Σε όλους τους πίνακες, εκτός από τους πίνακες fires_staff και fires_forces οι οποίοι χρησιμοποιούν ως id δύο ξένια κλειδιά, χρησιμοποιούνται οι στήλες:

- created_at (ημερομηνία εισαγωγής στον πίνακα με την εντολή INSERT).
- updated_at (ημερομηνία νέας ενημέρωσης της συγκεκριμένης εγγραφής του πίνακα με την εντολή UPDATE ή ημερομηνία της πρώτης εισαγωγής με την εντολή INSERT).
- deleted_at (ημερομηνία διαγραφής της εγγραφής του πίνακα με την εντολή DELETE).

Πίνακας burnt_area

Στήλες πίνακα:

id, created_at, updated_at, deleted_at, forest, woodland, grove, grass_land, reed_marches, farmland, cultivation_waste, dumping_ground, id_fire

Μια πυρκαγιά μπορεί να έχει μία καμένη περιοχή και αντιστοίχως μία καμένη περιοχή μπορεί να ανήκει σε μία πυρκαγιά (βλ. Πίνακας 2). Ο πίνακας burnt_area παίρνει ως foreign key το id_fire.

Forest	Δάση
Woodland	Δασική Έκταση
Grove	Άλση
grass_land	Χορτ/κές Εκτάσεις
reed_marches	Καλάμια – Βάλτοι
Farmland	Γεωργικές Εκτάσεις
cultivation_waste	Υπολείμματα Καλλιεργειών
dumping_ground	Σκουπιδότοποι

Πίνακας 2 – Αντιστοίχηση στηλών πίνακα burn_area με τις στήλες 'καμένη έκταση' του excel

Πίνακας fire_departments

Στήλες πίνακα:

id, created_at, updated_at, deleted_at, name, address, city, postal_code, phone, latitude, longitude, id_region

Μια πυροσβεστική υπηρεσία ανήκει σε μία περιφέρεια ενώ μία περιφέρεια μπορεί να έχει πολλές πυροσβεστικές υπηρεσίες. Ο πίνακας fire_departments παίρνει ως foreign key το id_region.

Οι στήλες address, city, postal_code, phone, latitude και longitude, έχουν εισαχθεί για τυχόν περαιτέρω πληροφορία και δεν χρησιμοποιούνται σε αυτή την εργασία.

Πίνακας regions

Στήλες πίνακα:

id, created_at, updated_at, deleted_at, name

Ο πίνακας regions περιλαμβάνει τις περιφέρειες και το κλειδί του χρησιμοποιείται από άλλους πίνακες ως foreign key.

Πίνακας prefectures

Στήλες πίνακα:

id , created_at, updated_at, deleted_at, name, id_region

Ένας νομός ανήκει σε μία περιφέρεια ενώ μία περιφέρεια μπορεί να έχει πολλούς νομούς. Ο πίνακας prefectures παίρνει ως foreign key το id_region.

Πίνακας municipalities

Στήλες πίνακα:

id, created_at, updated_at, deleted_at, name, code, id_prefecture

Ένας δήμος ανήκει σε έναν νομό ενώ ένας νομός μπορεί να έχει πολλούς δήμους. Ο πίνακας municipalities παίρνει ως foreign key το id_prefecture.

Πίνακας fires

Στήλες πίνακα:

id, created_at, updated_at, deleted_at, id_fire_department, id_prefecture , id_municipality, start_date, end_date, forestry, area, address, latitude, longitude, gmap_status

Μία πυρκαγιά διαχειρίζεται από μία πυροσβεστική υπηρεσία η οποία μπορεί να διαχειρίζεται πολλές πυρκαγιές. Επίσης, ένας νομός έχει πολλές πυρκαγιές ενώ μία πυρκαγιά εκδηλώνεται σε έναν νομό. Τέλος, η πυρκαγιά εκδηλώνεται σε έναν δήμο στον οποίο μπορούν να υπάρξουν πολλές πυρκαγιές.

Από τα παραπάνω, ο πίνακας fires παίρνει ως foreign keys τα εξής: id_fire_department, id_prefecture, id_municipality.

Επιπλέον, η στήλη forestry αντιστοιχεί στο Δασαρχείο, η στήλη area αντιστοιχεί στην Περιοχή, η στήλη address αντιστοιχεί στην Διεύθυνση ενώ οι στήλες latitude και longitude αποτελούν τις γεωγραφικές συντεταγμένες που έχει η πυρκαγιά επάνω στον χάρτη.

Τέλος, η στήλη gmap_status ενημερώνεται κατά την εισαγωγή των πυρκαγιών από ένα αρχείο CSV και απεικονίζει το αποτέλεσμα της μεθόδου εύρεσης των συντεταγμένων των πυρκαγιών από τους χάρτες GoogleMaps. Η διαδικασία αυτή αναλύεται εκτενέστερα παρακάτω.

Πίνακας staff

Στήλες πίνακα:

id, created_at, updated_at, deleted_at, type, name

Ο πίνακας staff περιλαμβάνει τους τύπους προσωπικού (ΠΥΡΟΣ. ΣΩΜΑ, ΠΕΖΟΠΟΡΑ ΤΜΗΜΑΤΑ, ΕΘΕΛΟΝΤΕΣ, ΣΤΡΑΤΟΣ, ΑΛΛΕΣ ΔΥΝΑΜΕΙΣ) και το κλειδί του χρησιμοποιείται από άλλους πίνακες ως foreign key.

Πίνακας fires_staff

Στήλες πίνακα:

id_fire, id_staff, number

Σε μία πυρκαγιά μπορούν να υπάρξουν πολλοί τύποι προσωπικού, οι οποίοι μετέχουν σε πολλές πυρκαγιές. Γι' αυτό τον λόγο δημιουργείται από τους πίνακες fires και staff ο πίνακας fires_staff, οποίος έχει ως κλειδί τα foreign keys των πινάκων fires και staff (id_fire, id_staff). Επίσης, για κάθε τύπο προσωπικού ο οποίος καταγράφεται σε μία πυρκαγιά, υπάρχει και ένας αριθμός. Έτσι, προκύπτει η στήλη number στον πίνακα fires_staff.

Πίνακας forces

Στήλες πίνακα:

id, created_at, updated_at, deleted_at, type, name, id_force_type

Ο πίνακας forces περιλαμβάνει τους τύπους πυροσβεστικών μέσων (ΠΥΡΟΣ. ΟΧΗΜ., ΟΧΗΜ. ΟΤΑ, ΒΥΤΙΟΦΟΡΑ, ΜΗΧΑΝΗΜΑΤΑ, ΕΛΙΚΟΠΤΕΡΑ, Α/Φ CL415, Α/Φ CL215, Α/Φ PZL, Α/Φ GRU) και το κλειδί του χρησιμοποιείται από άλλους πίνακες ως foreign key.

Πίνακας fires_forces

Στήλες πίνακα:

id_fire, id_force, number

Σε μία πυρκαγιά μπορούν να υπάρξουν πολλοί τύποι πυροσβεστικών μέσων, οι οποίοι μετέχουν σε πολλές πυρκαγιές. Γι' αυτό τον λόγο δημιουργείται από τους πίνακες fires και forces ο πίνακας fires_forces, οποίος έχει ως κλειδί τα foreign keys των πινάκων fires και forces (id_fire, id_force). Επίσης, για κάθε τύπο πυροσβεστικού μέσου ο οποίος καταγράφεται σε μία πυρκαγιά, υπάρχει και ένας αριθμός. Έτσι, προκύπτει η στήλη number στον πίνακα fires_ forces.

Πίνακας forces_types

Στήλες πίνακα:

id, created_at, updated_at, deleted_at, type, name

Κάθε πυροσβεστικό μέσο χαρακτηρίζεται από έναν τύπο (ΕΝΑΕΡΙΑ ή ΟΧΗΜΑΤΑ). Έτσι δημιουργήθηκε ο πίνακας forces_types που αντιπροσωπεύει αυτούς τους δύο τύπους και που το κλειδί του χρησιμοποιείται από τον πίνακα forces ως foreign key (id_force_type).

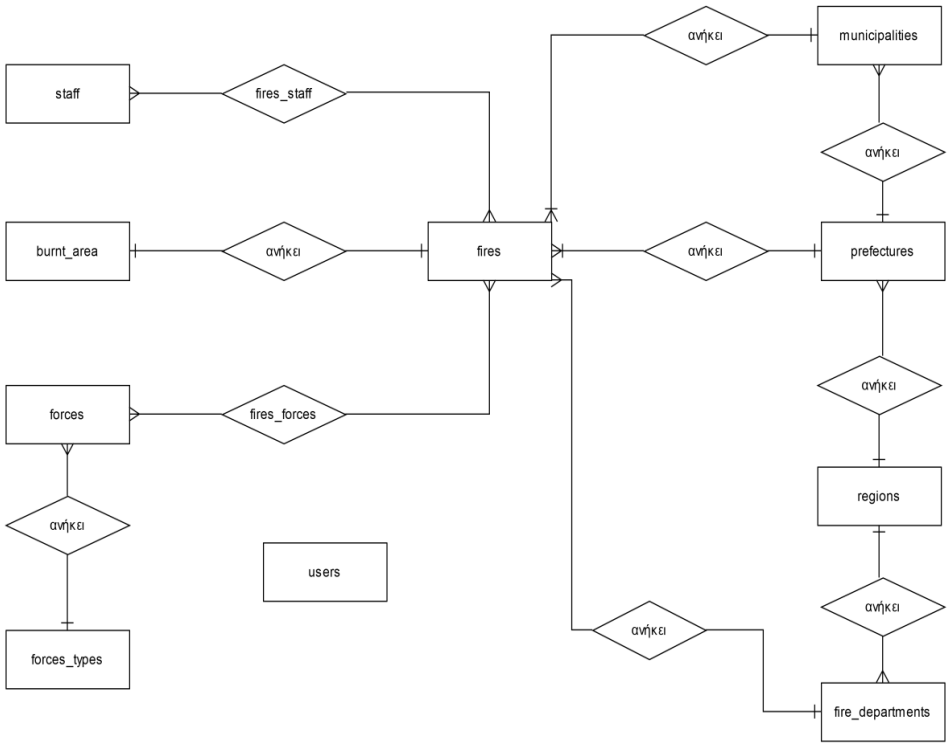
Πίνακας users

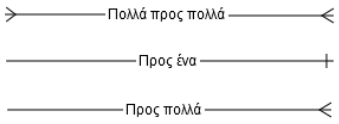
Στήλες πίνακα:

id, created_at, updated_at, deleted_at, name, last_name, email, password, remember_token

Ο πίνακας users χρησιμοποιείται για την σύνδεση στο σύστημα του Admin. Η στήλη remember_token χρησιμοποιείται από το Laravel PHP Framework και είναι απαραίτητη για την επεξεργασία του λογαριασμού του Admin. Χρησιμοποιείται για περαιτέρω ασφάλεια του λογαριασμού καθώς ανανεώνεται κατά την είσοδο και έξοδο του Admin από το σύστημα, προκειμένου να αποφευχθεί τυχόν παραβίαση του λογαριασμού.

3.3.3 ΤΟ ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ ΤΗΣ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

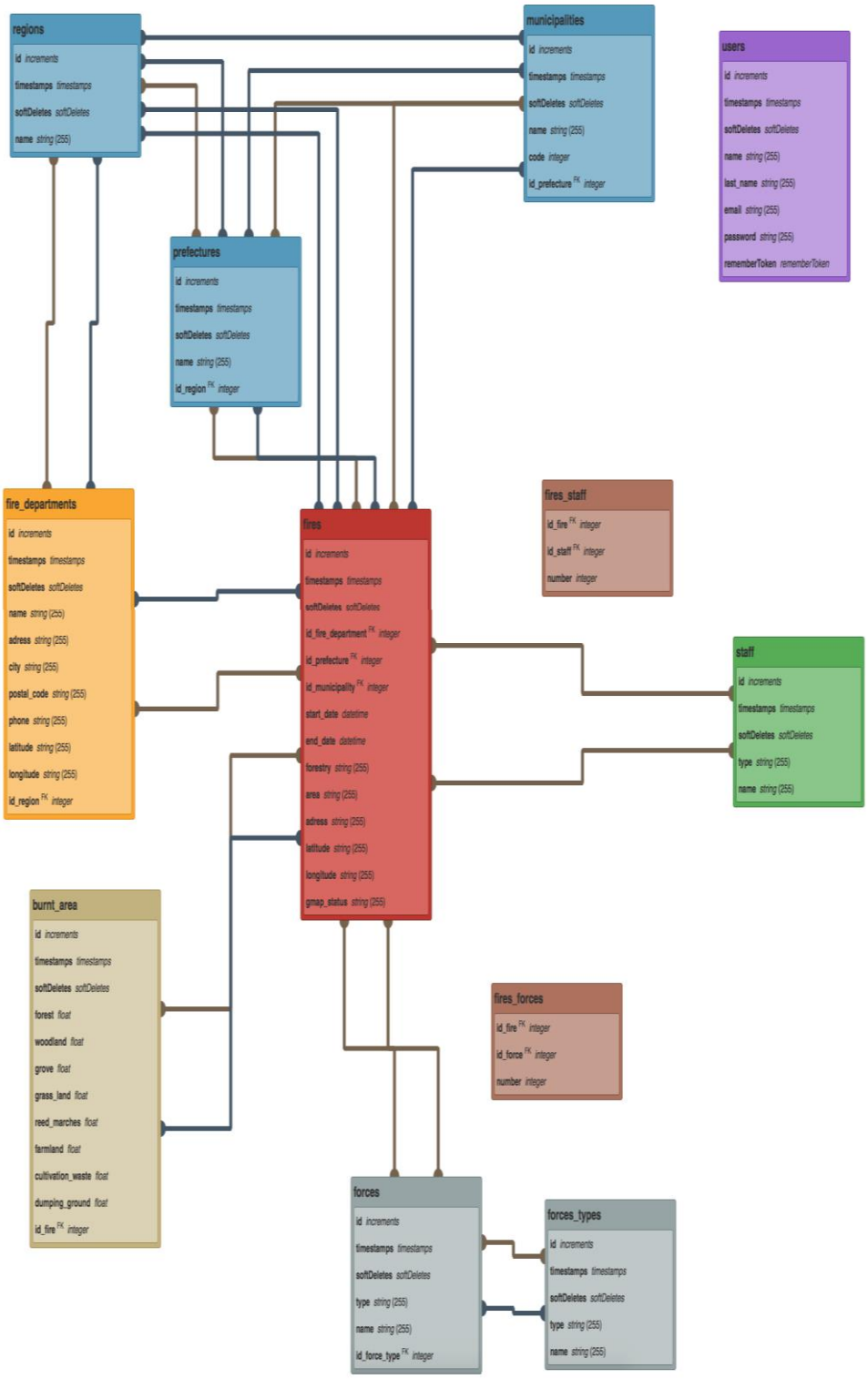




Εικόνα 22 – Σχέσεις μοντέλου Οντοτήτων-Συσχετίσεων

Το σχεσιακό μοντέλο κατασκευάστηκε με την βοήθεια σχετικής εφαρμογής από τη laravelsd.com, η οποία χρησιμοποιείται για την κατασκευή βάσεων δεδομένων που υποστηρίζονται από το εργαλείο Laravel. Η στήλη timestamps δημιουργεί τις στήλες created_at και updated_at, ενώ η στήλη softDeletes δημιουργεί την στήλη deleted_at.

Στο σχεσιακό μοντέλο απεικονίζονται οι πίνακες με τις στήλες τους, στις οποίες περιλαμβάνονται τα πεδία του πίνακα με το id και τα ξένα κλειδιά εάν υπάρχουν, καθώς και οι σχέσεις που έχει ο κάθε πίνακας-μοντέλο (σχέσεις belongsTo, hasMany, hasOne κ.τ.λ.) με άλλους πίνακες-μοντέλα. Οι σχέσεις αυτές απεικονίζονται με μπλέ και καφέ γραμμές, ανάλογα με την σχέση με την οποία συνδέονται οι πίνακες-μοντέλα (βλ. Εικόνα 23).



Εικόνα 23 – Σχεσιακό μοντέλο βάσης

ΚΕΦΑΛΑΙΟ 4: Η ΕΦΑΡΜΟΓΗ greekfires.gr

4.1 Εισαγωγή

Στο κεφάλαιο αυτό πραγματοποιείται η ανάλυση του τρόπου χρησιμοποίησης του συστήματος σε επίπεδο Back-end και Front-end. Επιπλέον, αναλύεται και ο σκοπός δημιουργίας της εφαρμογής greekfires.gr και παρουσιάζονται τα οφέλη που προκύπτουν από την δημιουργία της συγκεκριμένης εφαρμογής.

4.2 Σκοπός – Προσδοκώμενα αποτελέσματα

Ο στόχος της εφαρμογής αυτής είναι η δημιουργία ενός Front-end με το οποίο θα μπορεί ο κάθε πολίτης που θα το χρησιμοποιεί να ενημερωθεί σχετικά με τις καταγεγραμμένες πυρκαγιές της Ελλάδος, καθώς και η δημιουργία ενός Back-end, το οποίο αποτελεί ένα μοντέλο εισαγωγής και επεξεργασίας των πυρκαγιών, ικανό για να χρησιμοποιηθεί από την Πυροσβεστική.

Το Front-end της εφαρμογής, προσφέρει την δυνατότητα να απεικονιστούν οι πυρκαγιές επάνω στον χάρτη σύμφωνα με την χρονική τους περίοδο. Επιπλέον, μέσω στατιστικών, πραγματοποιείται πληροφόρηση των πολιτών σχετικά με το σύνολο των πυρκαγιών οι οποίες γίνονται ανά μήνα, με τις καταστροφικότερες πυρκαγιές με βάση τα καμένα στρέμματα, τις χρονικά μεγαλύτερες πυρκαγιές και τις περιοχές (περιφέρειες-νομοί) με τις περισσότερες πυρκαγιές.

Μέσω τώρα της Back-end εφαρμογής, γίνεται δυνατή η δημιουργία μοντέλου με το οποίο μπορεί να εισαχθεί το σύνολο των πυρκαγιών σε μία βάση δεδομένων, μέσω του αρχείου excel που δίνει η πυροσβεστική, καθώς και η δυνατότητα εισαγωγής και επεξεργασίας μίας νέας πυρκαγιάς και των πληροφοριών αυτής (προσωπικό, καμένη έκταση, πυροσβεστικά μέσα, γεωγραφικά στοιχεία). Επιπλέον, παρέχεται η δυνατότητα στον χρήστη αυτής, να εξάγει το σύνολο των πυρκαγιών σε ένα αρχείο CSV, παρόμοιας μορφής με το αρχείο excel της Πυροσβεστικής.

Για την δημιουργία του μοντέλου εισαγωγής των γεωγραφικών δεδομένων μίας πυρκαγιάς, δημιουργήθηκε ένα σύστημα επιλογής του νομού και του δήμου με βάση την περιφέρεια. Το μοντέλο αυτό προέκυψε από την λίστα των νομών και των δήμων με βάση το νομοσχέδιο του Καλλικράτη. Έτσι, δίνεται η δυνατότητα στον διαχειριστή της εφαρμογής να εισάγει με καθαρό και σωστό τρόπο τα ονόματα του νομού και του δήμου της πυρκαγιάς. Αυτό είναι σημαντικό, καθώς από την επεξεργασία του αρχείου excel της Πυροσβεστικής, διαπιστώθηκε η εσφαλμένη ονοματολογία νομών και δήμων, η οποία δεν ανταποκρίνεται στην αντίστοιχη που υπάρχει στο αρχείο με την λίστα των νομών-δήμων του Καλλικράτη.

Επιπλεόν, δημιουργήθηκε και ένα μοντέλο εισαγωγής της πυροσβεστικής σύμφωνα με την περιφέρεια στην οποία ανήκει. Η συλλογή αυτών των δεδομένων, έγινε με βάση τις αντίστοιχες λίστες περιφερειών-πυροσβεστικών υπηρεσιών που υπάρχουν στην ιστοσελίδα της Πυροσβεστικής Υπηρεσίας.

Επίσης, για την εισαγωγή της χρονικής περιόδου της πυρκαγιάς, χρησιμοποείται ένα plugin το οποίο επιτρέπει την πιο εύκολη εισαγωγή ημερομηνίας και ώρας και με το οποίο αποφεύγεται το να εισαχθεί ημερομηνία έναρξης της πυρκαγιάς μεταγενέστερης της ημερομηνίας κατάσβεσης. Αυτό είναι σημαντικό, καθώς το αρχείο excel της Πυροσβεστικής περιλαμβάνει τέτοιες περιπτώσεις λάθους. Εδώ φαίνεται και η συνεισφορά της εργασίας αυτής σε σχέση με τον καθαρισμό (data cleansing) των ανοικτών δημοσίων δεδομένων της Πυροσβεστικής Υπηρεσίας. Το Back-end της εφαρμογής, μπορεί να χρησιμοποιηθεί ανεξάρτητα από το Front-end και αποτελεί μία πολύ καλή λύση για μελλοντική χρήση του από την Πυροσβεστική παράλληλα με τη δημοσιοποίηση των σχετικών δεδομένων.

4.3 Περιγραφή Λειτουργίας

4.3.1 Back-end

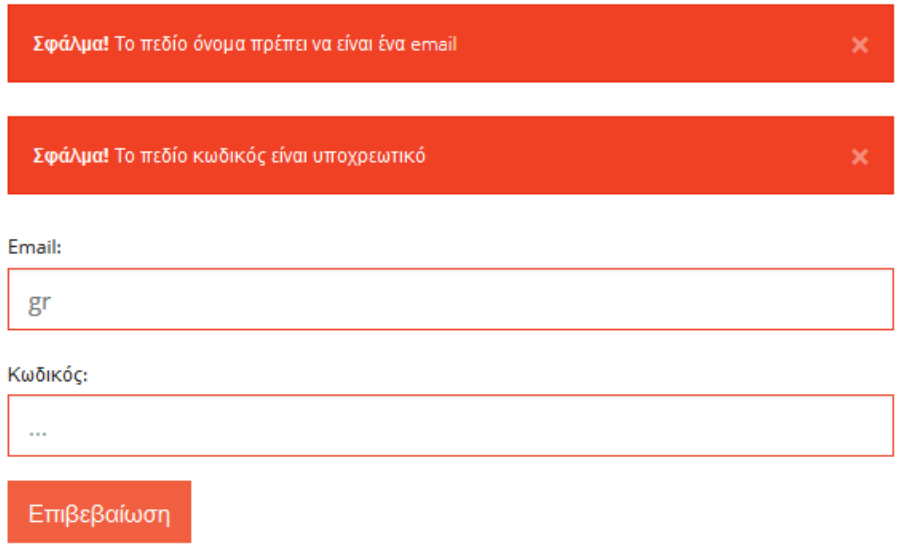
Το Back-end αποτελεί το μέρος της εφαρμογής την οποία διαχειρίζεται ο διαχειριστής της εφαρμογής, προκειμένου να εισάγει νέες πυρκαγιές, να επεξεργαστεί ήδη υπάρχουσες, να προσθέσει και να επεξεργαστεί νέα μέσα καθώς και νέο προσωπικό.

Σύνδεση στο Back-end

Αρχικά για να συνδεθεί ο χρήστης (διαχειριστής) στο σύστημα, πρέπει να εισάγει στην φόρμα το Συνθηματικό (Email) και τον Κωδικό του στην View login (βλ. Εικόνα 24).

Εικόνα 24 – Σύνδεση Διαχειριστή

Σε περίπτωση που κατά την επιβεβαίωση της φόρμας ο χρήστης δεν έχει εισάγει email και password ή εισάγει λανθασμένες τιμές τότε εμφανίζονται αντίστοιχα μηνύματα λάθους (βλ. Εικόνα 25).



Εικόνα 25 – Μηνύματα λάθους κατά την σύνδεση

Ο παραπάνω έλεγχος καθώς και η είσοδος στο σύστημα γίνονται στον AdminController από την συνάρτηση loginStore στο παρακάτω σημείο του κώδικα:

```
if (Auth::attempt($userdata)) {
    return Redirect::to('/dashboard');
} else {
```

```
        return Redirect::to('/admin')->withErrors($validation)->withInput()->with($data)->with(array('error'=>'login'));
    }
```

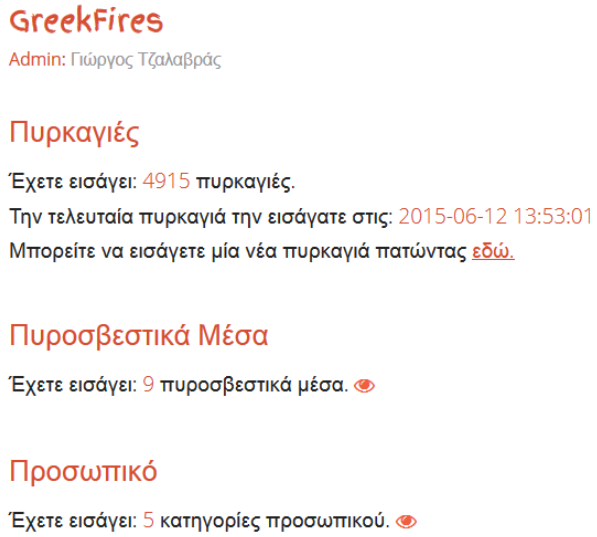
Εάν ο χρήστης εισήγαγε τα σωστά στοιχεία, όπως ελέγχει η συνθήκη Auth::attempt($userdata), τότε εμφανίζεται η View dashboard μέσω της route:

```
        Route::get('/dashboard', array('uses' => 'AdminController@dashboardCreate'));
```

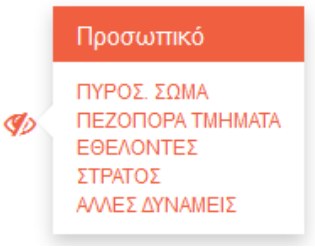
Αλλιώς, επιστρέφει στην σελίδα login με μηνύματα λάθους.

Σελίδα Dashboard

Έπειτα από την επιτυχή σύνδεση του χρήστη εμφανίζεται η σελίδα dashboard (βλ. Εικόνα 26), η οποία περιέχει πληροφορίες για τον χρήστη σχετικά με τις πυρκαγιές, τα πυροσβεστικά μέσα και το προσωπικό. Τα ονόματα των πυροσβεστικών μέσων καθώς και του προσωπικού εμφανίζονται μέσω των Bootstrap Popovers (βλ. Εικόνα 27).

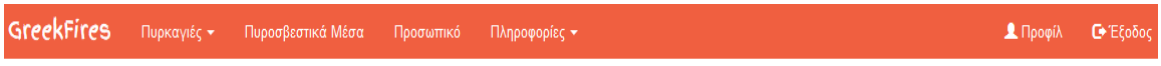


Εικόνα 26 – Σελίδα Dashboard



Εικόνα 27 – Popover

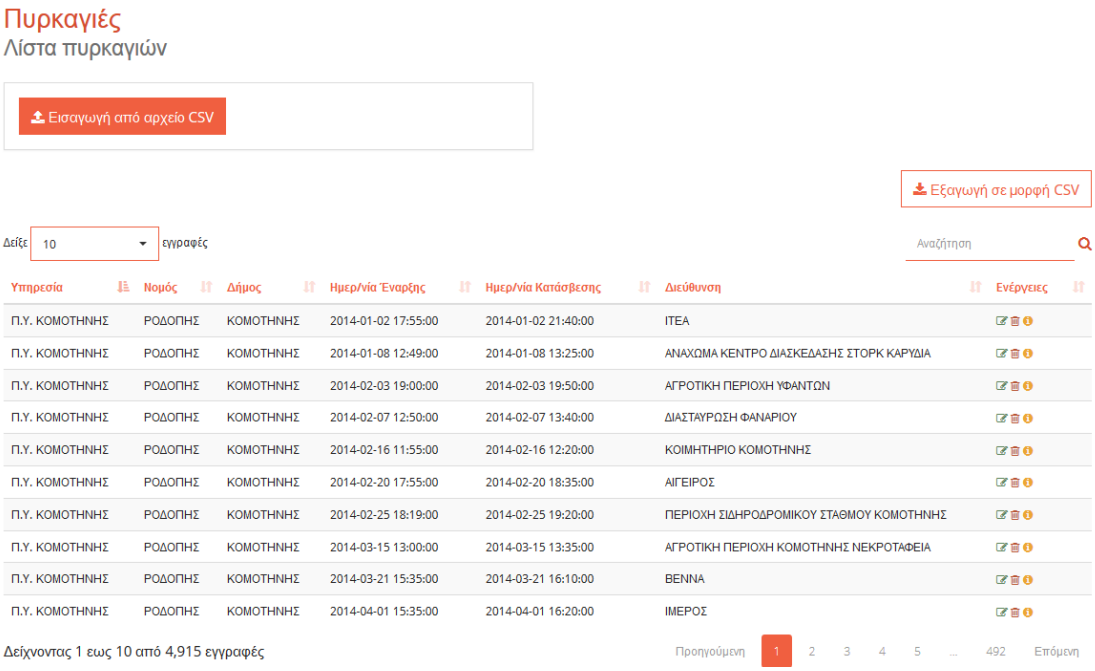
Επιπλέον, στο πάνω μέρος της σελίδας εμφανίζεται η λίστα με τις ενέργειες τις οποίες μπορεί να κάνει ο χρήστης στο σύστημα (βλ Εικόνα 28).



Εικόνα 28 – Μενού ενεργειών του διαχειριστή

Σελίδα Λίστας Πυρκαγιών

Από το μενού επιλέγοντας την υπολίστα Πυρκαγιές, εμφανίζονται οι δύο επιλογές ‘Λίστα Πυρκαγιών’ και ‘Εισαγωγή νέας πυρκαγιάς’. Με την επιλογή ‘Λίστα Πυρκαγιών’, εμφανίζονται όλες οι πυρκαγιές (βλ. Εικόνα 29).



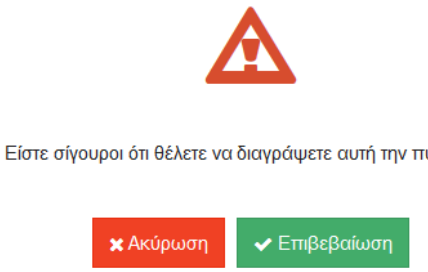
Εικόνα 29 – Λίστα Πυρκαγιών

Για την λίστα των πυρκαγιών, χρησιμοποιείται ένα jQuery plugin για την εμφάνιση των πινάκων, το Datatables plugin [36]. Λόγω του μεγάλου αριθμού των πυρκαγιών, η εισαγωγή αυτών γίνεται με την χρήση του AJAX, τα δεδομένα τα οποία επιστρέφονται από τον DataTablesController.

Σε καθεμία πυρκαγιά υπάρχουν κάποιες ενέργειες, όπως είναι η Επεξεργασία της πυρκαγιάς, η Διαγραφή της και οι Πληροφορίες τις, οι οποίες περιλαμβάνουν όλες τις πληροφορίες της πυρκαγιάς οι οποίες δεν εμφανίζονται στον πίνακα.

Με την επιλογή ‘Επεξεργασία’, γίνεται η μετάβαση στην σελίδα κατά την οποία μπορεί να γίνει η επεξεργασία της πυρκαγιάς.

Με την επιλογή ‘Διαγραφή, εμφανίζεται ένα παράθυρο επιβεβαίωσης (βλ. Εικόνα 30), σχετικά με το εάν επιθυμείται η διαγραφή της συγκεκριμένης πυρκαγιάς.



Εικόνα 30 – Παράθυρο επιβεβαίωσης Διαγραφής Πυρκαγιάς

Η διαδικασία αυτή έχει ως εξής:

1) Στην ιδιότητα data-id, η οποία υπάρχει στον σύνδεσμο με κλάση delete, υπάρχει το id της πυρκαγιάς.
```
<a href="#" class="delete" data-id="'.$rResult->fire_id.'" data-toggle="tooltip" data-placement="top" title="Διαγραφή"><i class="fa fa-trash-o"></i></a>
```

2) Με ένα κλικ τον σύνδεσμο .delete, εισάγεται σε μία μεταβλητή id_fire το id που περιέχει η ιδιότητα data-id. Στο παράθυρο (modal) με id #delete_fire_modal, υπάρχει μία φόρμα για την διαγραφή της πυρκαγιάς, το id της οποίας εισάγεται σε ένα Input τύπου hidden.

```
$('body').on('click', '.delete', function(){
        id_fire = $(this).attr('data-id');
        $('#delete_fire_modal #id_fire').val(id_fire);
        $("#delete_fire_modal").modal();
        return false;
});
```

3) Με την επιλογή ‘Επιβεβαίωση’ στο παράθυρο διαγραφής της πυρκαγιάς, υποβάλλεται η φόρμα το URL της οποίας είναι το fire/delete. Ο route ‘fire/delete’ χρησιμοποίει την μέθοδο fireDeleteStore του FireController, στην οποία διαγράφεται η πυρκαγιά.

Με την επιλογή ‘Πληροφορίες’, εμφανίζεται ένα παράθυρο με όλες τις πληροφορίες τις πυρκαγιάς η οποία επιλέχθηκε, με ανάλογο τρόπο όπως την διαγραφή της πυρκαγιάς. Η διαφορά εδώ είναι ότι τα αποτελέσματα επιστρέφονται με την βοήθεια του AJAX.

```
$('body').on('click', '.info', function(){
    id_fire = $(this).attr('data-id');
    $.ajax({
            type: "POST",
            url: "{{URL::to('get/fire/informations')}}",
            data: "&id_fire="+id_fire,
            contentType: "application/x-www-form-urlencoded;charset=UTF-8",
            success: function(response){
              if (response.success === true)
              {
                    $("#informations").html(response.informations);
                     $("#informations_modal").modal();
              }
                    }
          });
      return false;
});
```

Επίσης, στο πάνω μέρος της σελίδας ‘Λίστα Πυρκαγιών’, υπάρχουν δύο κουμπιά. Το ‘Εξαγωγή σε μορφή CSV’ και το ‘Εισαγωγή από αρχείο CSV’.

Εισαγωγή αρχείου CSV

Επιλέγοντας το κουμπί ‘Εισαγωγή από αρχείο CSV’, εμφανίζεται ένα παράθυρο με το οποίο επιλέγεται το αρχείο CSV, με βάση το οποίο γίνεται η εισαγωγή στην βάση δεδομένων των πυρκαγιών καθώς και των πληροφοριών των πυρκαγιών αυτών. Το αρχείο αυτό, πρέπει να έχει την ίδια μορφή με το αρχείο της πυροσβεστικής, δηλαδή τις ίδιες στήλες με την ίδια σειρά, χωρίς όμως την πρώτη γραμμή (γραμμή επικεφαλίδων). Έπειτα από την επιτυχή εισαγωγή του αρχείου, για να εκτελεστεί η διαδικασία χρειάζεται να πατηθεί το κουμπί ‘Αποθήκευση αρχείου’.

Με το πάτημα του κουμπιού ξεκινάει μία επαναληπτική διαδικασία, όπως αυτή ορίζεται στην συνάρτηση fileImport του FireController. Σε αυτή γίνονται τα εξής βήματα:

Α) Αποθήκευση της κάθε στήλης σε μεταβλητή:

```
$fire_department  = trim($data[0]);
```

Β) Κωδικοποίηση σε utf-8 των ελληνικών χαρακτήρων της μεταβλητής:

```
$fire_department = iconv('ISO-8859-7','UTF-8', $fire_department);
```

Γ) Έλεγχος για το εάν το όνομα της Πυροσβεστικής Υπηρεσίας, του Νομού και του Δήμου όπως αυτοί εισάγονται από το αρχείο, υπάρχουν στους αντίστοιχους πίνακες της βάσης δεδομένων (fire_departments, prefectures, municipalities):

```
if (count($fire_department_exists) > 0
&& count($prefecture_exists) > 0 &&
count($municipality_exists) > 0) {
        …
}
```

Δ) Έλεγχος για το εάν η πυρκαγιά ήδη υπάρχει στον πίνακα fires:

```
$fire_exists = Fire::where('start_date', '=', $fire->start_date)
                ->where('id_municipality', '=', $municipality_exists->id)
                ->where('id_prefecture', '=', $prefecture_exists->id)
                ->first();
```

Ε) Εάν υπάρχει τότε δεν εισάγεται στην βάση. Διαφορετικά εισάγεται στον πίνακα fires, ενώ εισάγονται και οι πληροφορίες της πυρκαγιάς στους αντίστοιχους πίνακες (burtn_area, fires_forces, fires_staff).

Κατά την παραπάνω διαδικασία, χρησιμοποιούνται count μεταβλητές, με σκοπό την ενημέρωση του χρήστη μέσω μηνύματος, σχετικά με το πόσες πυρκαγιές εισήχθησαν συνολικά επιτυχώς καθώς και πόσες απέτυχαν.

Επίσης, για τον εντοπισμό των latitude και longitude τιμών της πυρκαγιάς, χρησιμοποιείται η συνάρτηση getLatitudeAndLongitude. Η συγκεκριμένη συνάρτηση δέχεται ως εισόδους την χώρα (GREECE), τον νομό και τον δήμο, όπως δίνονται από το αρχείο CSV, καθώς και την μεταβλητή detail με αρχική τιμή 4:

```
getLatitudeAndLongitude($country, $prefecture, $municipality, $adress, 4)
```

Η συγκεκριμένη μεταβλητή δίνεται προκειμένου να εκτελεστεί μία επαναληπτική διαδικασία με βάση την οποία προκύπτει η διεύθυνση ($adress_output μεταβλητή), η οποία δίνεται στο URL του google maps

```
https://maps.google.com/maps/api/geocode/json?address=$adress_output&key=AIzaSyCksCbtBgHhhaGr5-YhbkWGcDEetL8VJXc&sensor=false
```

,όπως αυτό ορίζεται στην συνάρτηση.

Το URL του googlemaps επισρέφει ένα αρχείο JSON, από το οποίο προκύπτουν οι συντεταγμένες της δοθείσης διεύθυνσης:

```
$lat    = $json->{'results'}[0]->{'geometry'}->{'location'}->{'lat'};
$long   = $json->{'results'}[0]->{'geometry'}->{'location'}->{'lng'};
```

Εάν η διεύθυνση δεν βρεθεί, τότε εισάγεται ως σημείο της πυρκαγιάς το κέντρο συντεταγμένων της Ελλάδας με τιμές:

```
$lat = '39.074208', $long = '21.824312'.
```

Επίσης, στον πίνακα fires στην στήλη gmap_status εισάγεται η κατάσταση η οποία προκύπτει κατά την εύρεση της διεύθυνσης. Η κατάσταση αυτή μπορεί να έχει την τιμή OVER_QUERY_LIMIT. Αυτό σημαίνει ότι έχουν εκτελεστεί κατά την εισαγωγή του αρχείου πολλά ερωτήματα προς τον server της GoogleMaps και για προστασία η Google δεν επιστρέφει αποτελέσματα με συντεταγμένες. Έτσι, η εισαγωγή του αρχείου αποτυγχάνει να εισάγει τις ακριβείς συντεταγμένες. Επίσης, πρέπει να περάσουν 24 ώρες προκειμένου να ξαναγίνει η εισαγωγή του αρχείου, έτσι ώστε να υπάρξουν ακριβή αποτελέσματα. Αυτός είναι ο λόγος για τον οποίο χρειάζεται η εισαγωγή του αρχείου να γίνεται τμηματικά και με μικρότερο αριθμό εγγραφών.

Εξαγωγή αρχείου CSV

Επιλέγοντας το κουμπί ‘Εξαγωγή σε μορφή CSV’, δημιουργείται ένα αρχείο CSV, το οποίο παρουσιάζει τις πληροφορίες των πυρκαγιών όπως τις παρουσιάζει το αρχείο Excel της

πυροσβεστικής. Η εξαγωγή αυτή γίνεται από την συνάρτηση firesExport του FireController. Σε αυτή τρέχει μία επαναληπτική διαδικασία η οποία στην μεταβλητή $output εισάγει τις πληροφορίες της πυρκαγιάς.

Με την μεταβλητή:

```
$headers = array(
        'Content-Type' => 'application/vnd.ms-excel; charset=UTF-8',
        'Content-Disposition' => 'attachment; filename="fires.csv"',
);
```

δημιουργείται το αρχείο csv το οποίο επιστρέφεται στον χρήστη με την εντολή:

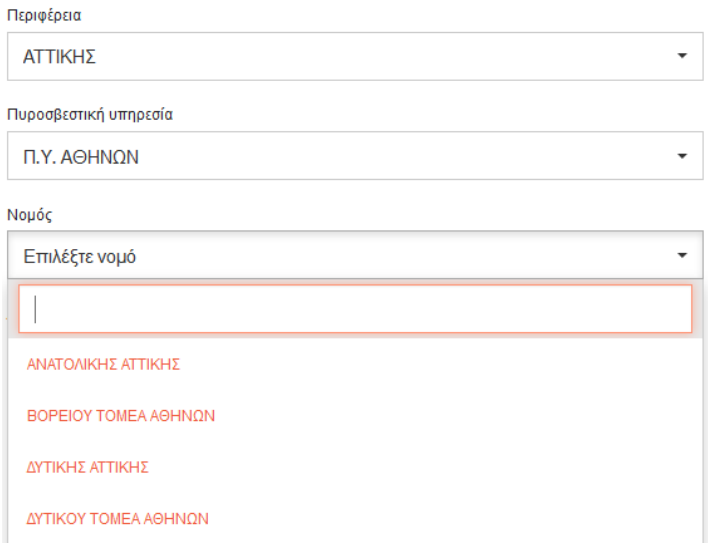
```
Response::make(rtrim($output, "\n"), 200, $headers).
```

Σελίδα Εισαγωγής - Επεξεργασίας Πυρκαγιών

Η εισαγωγή μίας νέας πυρκαγιάς γίνεται επιλέγοντας από το μενού την ‘Εισαγωγή νέας πυρκαγιάς’. Σε αυτήν την σελίδα ο χρήστης μπορεί να εισάγει τις πληροφορίες της φωτιάς, όπως είναι η πυροσβεστική υπηρεσία, ο νομός, ο δήμος, η ημερομηνία έναρξης, η ημερομηνία κατάσβεσης, το δασαρχείο, την περιοχή, την διεύθυνση καθώς και το σημείο πάνω στον χάρτη. Έπειτα από την εισαγωγή αυτών, η πυρκαγιά έχει εισαχθεί επιτυχώς και γίνεται η μετάβαση στην σελίδα της Επεξεργασίας της πυρκαγιάς, η οποία είναι η ίδια όπως η σελίδα της Εισαγωγής, με τη διαφορά ότι περιλαμβάνει τις τιμές που έχουν εισαχθεί, καθώς και τρία κουμπιά (Καμένη έκταση, Προσωπικό, Μέσα πυρόσβεσης) μέσω των οποίων εισάγονται οι πληροφορίες αυτών.

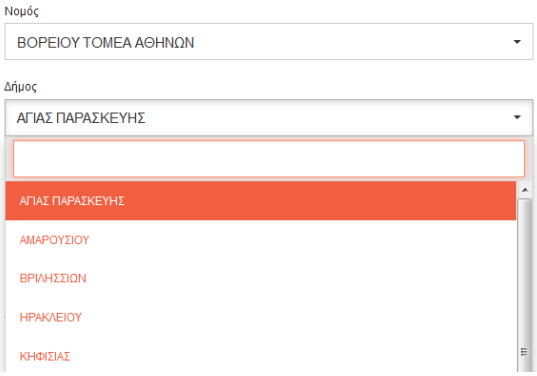
Για την εισαγωγή της πυροσβεστικής υπηρεσίας, του νομού και του δήμου χρησιμοποιούνται λίστες οι οποίες περιλαμβάνουν η κάθε μία την λίστα των αντίστοιχων πληροφοριών. Για την απλούστευση της επιλογής αυτών, κάθε μία λίστα ανανεώνει την επόμενη με βάση την επιλογή η οποία έχει γίνει (βλ Εικόνα 31). Για παράδειγμα, για την ενεργοποίηση των λιστών της πυροσβεστικής υπηρεσίας και του νομού, είναι υποχρεωτική η επιλογή πρώτα της περιφέρειας στην οποία αυτές ανήκουν. Έτσι, οι λίστες ανανεώνονται αυτόματα με βάση αυτή την επιλογή. Επίσης, η λίστα των δήμων ανανεώνεται με βάση τον νομό ο οποίος έχει επιλεχθεί και στον οποίο ανήκουν.

Για παράδειγμα, επιλέγοντας την περιφέρεια ΑΤΤΙΚΗΣ, ανανεώνονται οι λίστες της πυροσβεστικής υπηρεσίας και του νομού. Έτσι, περιλαμβάνονται στην λίστα μόνο αυτοί που αντιστοιχούν στην περιφέρεια ΑΤΤΙΚΗΣ.



Εικόνα 31 – Επιλογή νομού πυρκαγιάς

Επιπλέον, επιλέγοντας τον νομό ενεργοποιείται η λίστα των δήμων, η οποία περιλαμβάνει τους δήμους που ανήκουν στον νομό που επιλέχθηκε (βλ. Εικόνα 32).



Εικόνα 32 – Επιλογή δήμου πυρκαγιάς

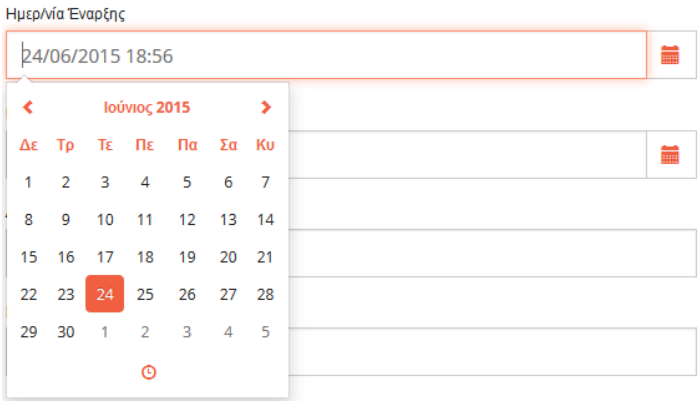
Η παραπάνω διαδικασία γίνεται με την χρήση AJAX. Για παράδειγμα, ανανεώνοντας την λίστα των περιφερειών, ο ακόλουθος κώδικας με την εντολή each, δίνει τις λίστες των πυροσβεστικών μέσων και των νομών:

```
$.each(response.fire_departments, function(){
        $('#id_fire_department').append('<option value="' + this.id + '">' + this.name + '</option>');
         $('#id_fire_department').selectpicker('refresh');
});

$('#id_prefecture').append('<option value="0" class="hide"></option>');
$.each(response.prefectures, function(){
        $('#id_prefecture').append('<option value="' + this.id + '">' + this.name + '</option>');
         $('#id_prefecture').selectpicker('refresh');
});
```

Με την εντολή append εισάγεται η κάθε γραμμή της λίστας.

Για την εισαγωγή ημερομηνιών έναρξης και κατάσβεσης, έχει χρησιμοποιηθεί ένα jQuery plugin το bootstrap-datetimepicker [37], το οποίο ανοίγει ένα παράθυρο ημερολογίου (βλ. Εικόνα 33), στο οποίο ο χρήστης μπορεί να επιλέξει την ημερομηνία που θέλει.



Εικόνα 33 – Εισαγωγή ημερομηνίας

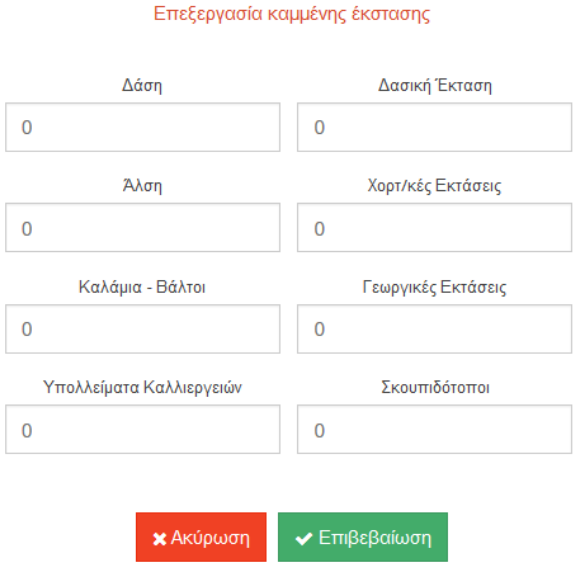
Τα πεδία δασαρχείο, περιοχή και διεύθυνση είναι τύπου Input και ο χρήστης έχει την ελεύθερη επιλογή να εισάγει ελεύθερα τις τιμές.

Τέλος, περιλαμβάνεται και ένας χάρτης GoogleMaps, η χρήση του οποίου γίνεται μέσω του plugin locationpicker [38]. Με αυτό το plugin, γίνεται δυνατή η επιλογή των συντεταγμένων Latitude και Longitude, με την μετακίνηση της πινέζας πάνω στον χάρτη ή της εισαγωγής διεύθυνσης στο autocomplete πεδίο ‘Διεύθυνση στον χάρτη’.

Οι παραπάνω διαδικασίες εφαρμόζονται και στην σελίδα ‘Επεξεργασία πυρκαγιάς’. Πρόκειται για την ίδια σελίδα η οποία χρησιμοποίει την ίδια view, όπως η εισαγωγή πυρκαγιάς. Αυτό έγινε για να αποφευχθεί η γραφή παραπάνω κώδικα, ο οποίος στην βάση του είναι ο ίδιος.

Για τα κομμάτια που προσθέτονται και αλλάζουν στην σελίδα αυτή, χρησιμοποιείται η μεταβλητή $section η οποία έχει την τιμή edit και η οποία δηλώνεται στην συνάρτηση fireEditCreate, με την οποία επιτυγχάνεται η προβολή της σελίδας Επεξεργασίας.

Στην σελίδα ‘Επεξεργασία πυρκαγιάς’ υπάρχουν τα τρία κουμπιά (Καμένη έκταση, Προσωπικό, Μέσα πυρόσβεσης) μέσω των οποίων εισάγονται οι πληροφορίες αυτών, όπως δείχνει η παρακάτω εικόνα για την Επεξεργασία καμένης έκτασης (Εικόνα 34).



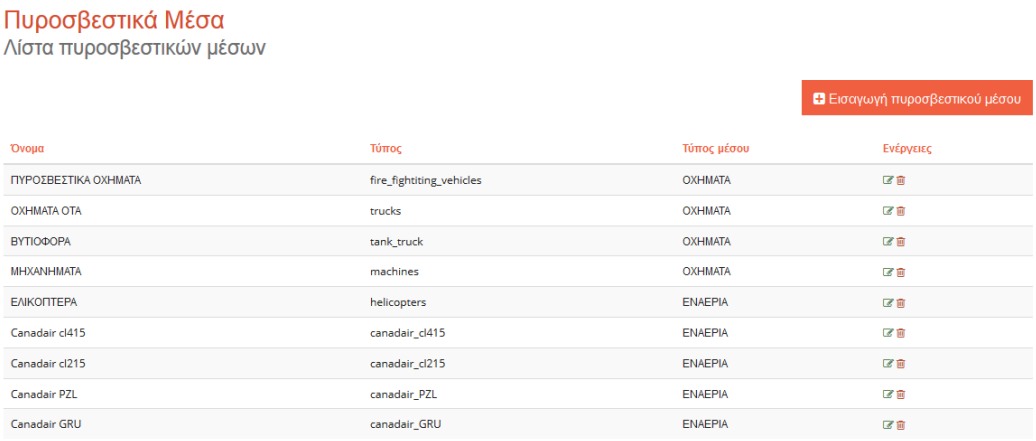
Εικόνα 34 – Εισαγωγή τιμών καμένης περιοχής

Και στις τρείς περιπτώσεις, δεν είναι δυνατή η εισαγωγή γραμμάτων, παρά μόνο αριθμών. Αυτό επιτυγχάνεται από την παρακάτω συνάρτηση:

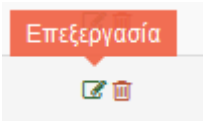
```
jQuery('.numbersOnly').keyup(function () {
    this.value = this.value.replace(/[^0-9\.]/g,'');
});
```

Σελίδα Λίστας Πυροσβεστικών Μέσων

Στην σελίδα αυτή παρουσιάζονται τα πυροσβεστικά μέσα τα οποία έχουν εισαχθεί από τον χρήστη (βλ. Εικόνα 35). Η view list.blade.php βρίσκεται στο μονοπάτι admin/forces και ο Controller ο οποίος χρησιμοποιείται είναι ο ForcesController. Οι στήλες είναι οι εξής: Όνομα, Τύπος, Τύπος μέσου (ΟΧΗΜΑΤΑ ή ΕΝΑΕΡΙΑ), Ενέργειες. Ο τύπος εισάγεται με κάτω παύλα ανάμεσα στις λέξεις π.χ. fire_fighting_vehicles και χρησιμοποιείται για το προγραμματιστικό μέρος. Σε κάθε μία εγγραφή υπάρχουν οι ενέργειες, με τις οποίες μπορεί να γίνει η Επεξεργασία (βλ. Εικόνα 36) ή η Διαγραφή ενός μέσου.

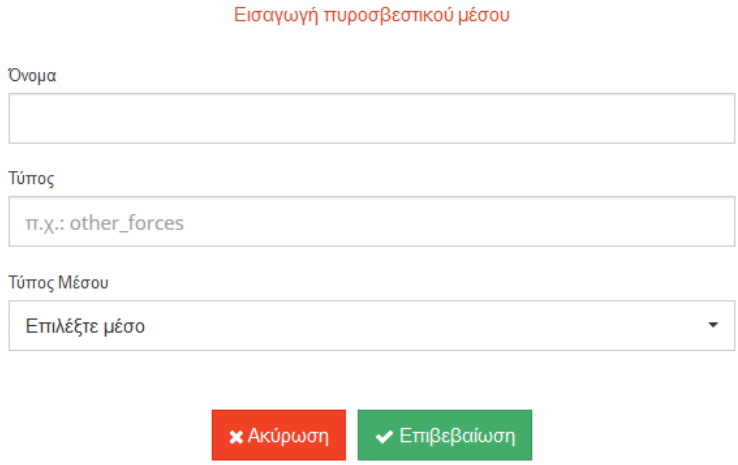


Εικόνα 35 – Σελίδα Λίστας Πυροσβεστικών Μέσων



Εικόνα 36 – Tooltip για την Επεξεργασία

Για την Εισαγωγή ενός νέου πυροσβεστικού μέσου, χρησιμοποιείται το κουμπί 'Εισαγωγή πυροσβεστικού μέσου'. Όταν αυτό πατηθεί, τότε εμφανίζεται ένα παράθυρο (modal) στον χρήστη (βλ. Εικόνα 37), στο οποίο εισάγει τις πληροφορίες του μέσου.

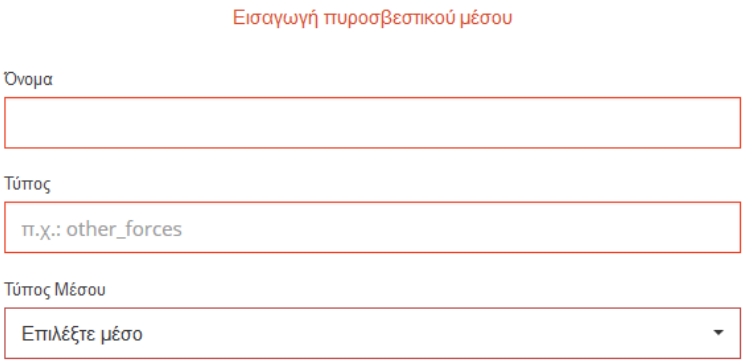


Εικόνα 37 – Εισαγωγή Πυροσβεστικού Μέσου

Για τον τύπο πυροσβεστικού μέσου, χρησιμοποιείται μία λίστα με δύο εγγραφές, τις εγγραφές ΟΧΗΜΑΤΑ – ΕΝΑΕΡΙΑ (βλ. Εικόνα 38).

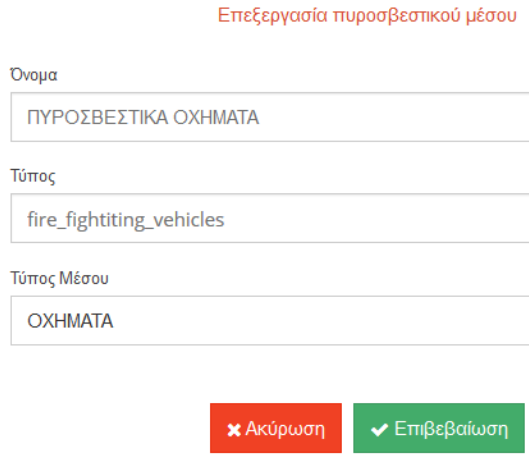
Εικόνα 38 – Εισαγωγή τύπου πυροσβεστικού μέσου (ΟΧΗΜΑΤΑ - ΕΝΑΕΡΙΑ)

Εάν κατά την εισαγωγή ενός μέσου δεν συμπληρωθούν όλες οι τιμές ή κάποια από αυτές, τότε η εισαγωγή αποτυγχάνει και τα Input αποκτούν κόκκινο χρώμα, το οποίο υποδηλώνει το λάθος κατά την εισαγωγή (βλ. Εικόνα 39).



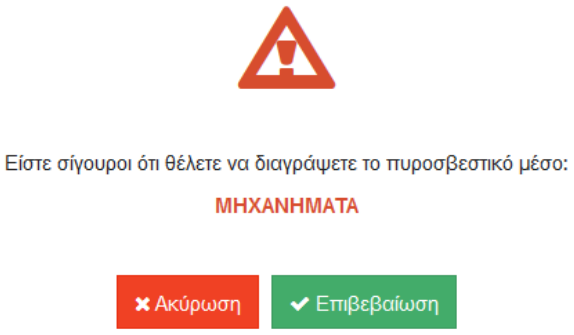
Εικόνα 39 – Έλλειψη πεδίων κατά την εισαγωγή Πυροσβεστικού Μέσου

Για την 'Επεξεργασία πυροσβεστικού μέσου' χρησιμοποιείται η ίδια διαδικασία, μόνο που το παράθυρο (modal) το οποίο ανοίγει περιλαμβάνει τις ήδη υπάρχουσες τιμές (βλ. Εικόνα 40). Σε περίπτωση μη εισαγωγής τιμών, τα Input όπως και στο παράθυρο 'Εισαγωγή πυροσβεστικού μέσου', παίρνουν το κόκκινο χρώμα και η επεξεργασία-ανανέωση του μέσου αποτυγχάνει.



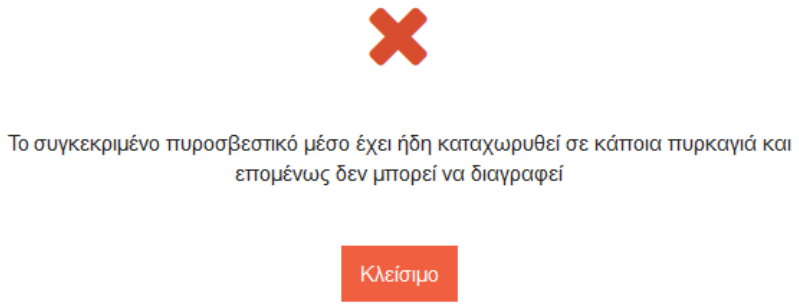
Εικόνα 40 – Επεξεργασία Πυροσβεστικού Μέσου

Για την Διαγραφή ενός Πυροσβεστικού μέσου, πατώντας το εικονίδιο της διαγραφής, εμφανίζεται ένα παράθυρο (βλ. Εικόνα 41), στο οποίο υπάρχει το όνομα του Πυροσβεστικού μέσου που έχει επιλεγεί για Διαγραφή.



Εικόνα 41 – Διαγραφή Μέσου

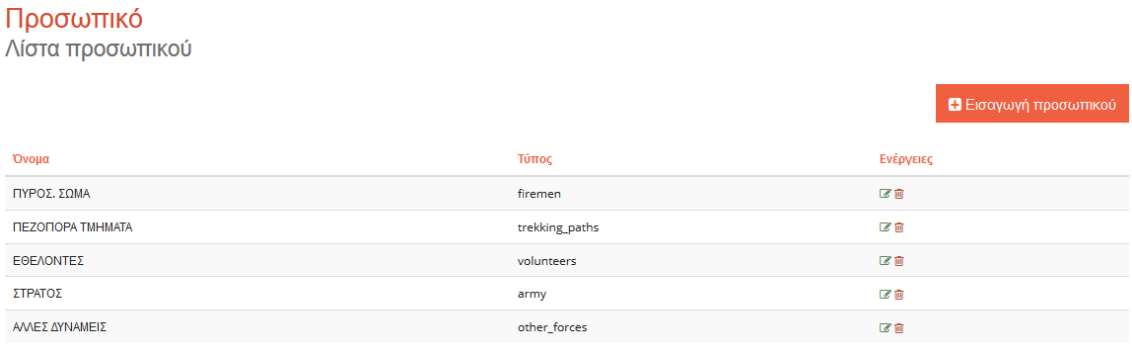
Με το κουμπί 'Επιβεβαίωση' γίνεται η διαγραφή του μέσου το οποίο διαγράφεται αυτόματα από την λίστα. Εάν το μέσο αυτό έχει ήδη συμμετάσχει στην πυρόσβεση μίας πυρκαγιάς, δηλαδή υπάρχει στον πίνακα fires_forces, τότε δεν είναι δυνατή η διαγραφή του και εμφανίζεται αντίστοιχο μήνυμα (βλ. Εικόνα 42).



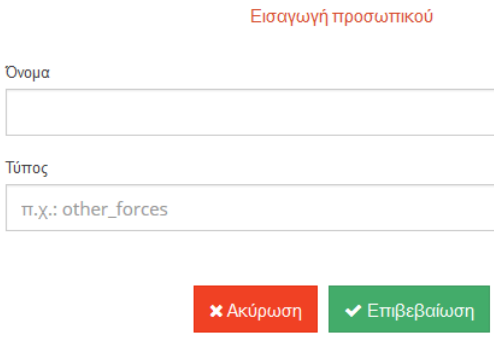
Εικόνα 42 – Μήνυμα μη διαγραφής πυροσβεστικού μέσου

Σελίδα Λίστας Προσωπικού

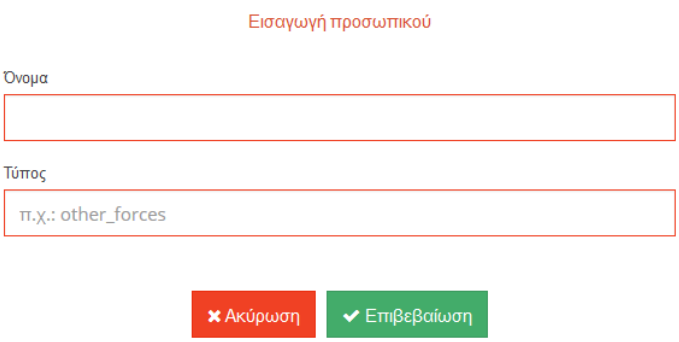
Η σελίδα Λίστας Προσωπικού (βλ. Εικόνα 43) παρουσιάζει όλους τους τύπους προσωπικού που παίρνουν μέρος σε μία πυρκαγιά. Η view list.blade.php βρίσκεται στο μονοπάτι admin/staff και ο Controller ο οποίος χρησιμοποιείται είναι ο StaffController. Οι διαδικασίες Εισαγωγής, Επεξεργασίας και Διαγραφής, γίνονται με τον ίδιο τρόπο όπως στην σελίδα της Λίστας Πυροσβεστικών Μέσων. Η διαφορά είναι ότι στο παράθυρο το οποίο εμφανίζεται για την Εισαγωγή και την Επεξεργασία του προσωπικού, δεν υπάρχει η λίστα του τύπου μέσου, καθώς αυτός υπάρχει μόνο για τα Πυροσβεστικά Μέσα. Παρακάτω δίνονται οι αντίστοιχες εικόνες των παραθύρων των ενεργειών του Προσωπικού (βλ. Εικόνα 44, Εικόνα 45, Εικόνα 46, Εικόνα 47, Εικόνα 48).



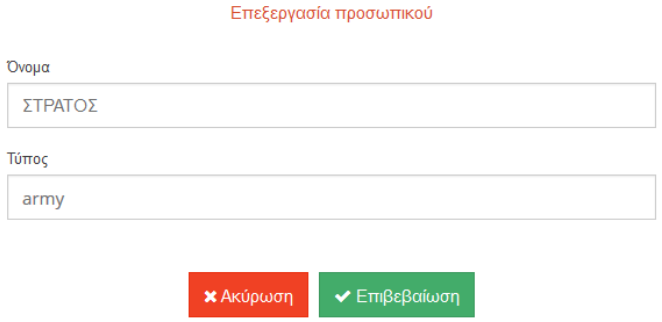
Εικόνα 43 - Σελίδα Λίστας Προσωπικού



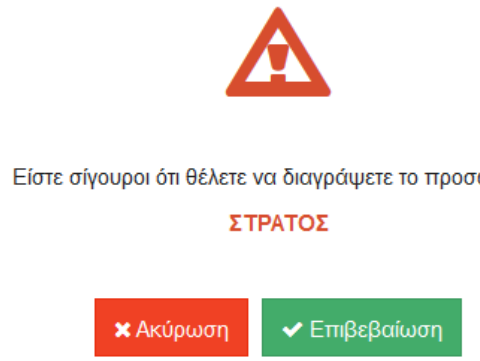
Εικόνα 44 – Εισαγωγή Προσωπικού



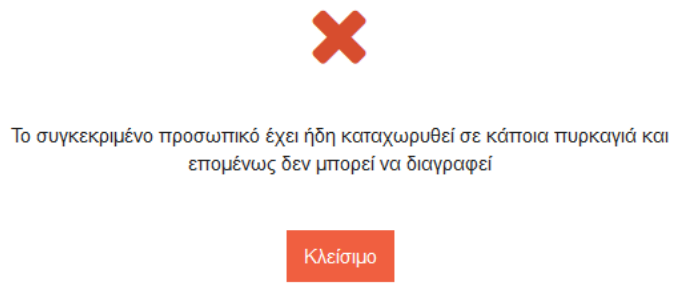
Εικόνα 45 - Έλλειψη πεδίων κατά την εισαγωγή Προσωπικού



Εικόνα 46 – Επεξεργασία προσωπικού



Εικόνα 47 – Διαγραφή Προσωπικού



Εικόνα 48 - Μήνυμα μη διαγραφής Προσωπικού

Σελίδες Περιφερειών, Νομών, Δήμων, Πυροσβεστικών Υπηρεσιών

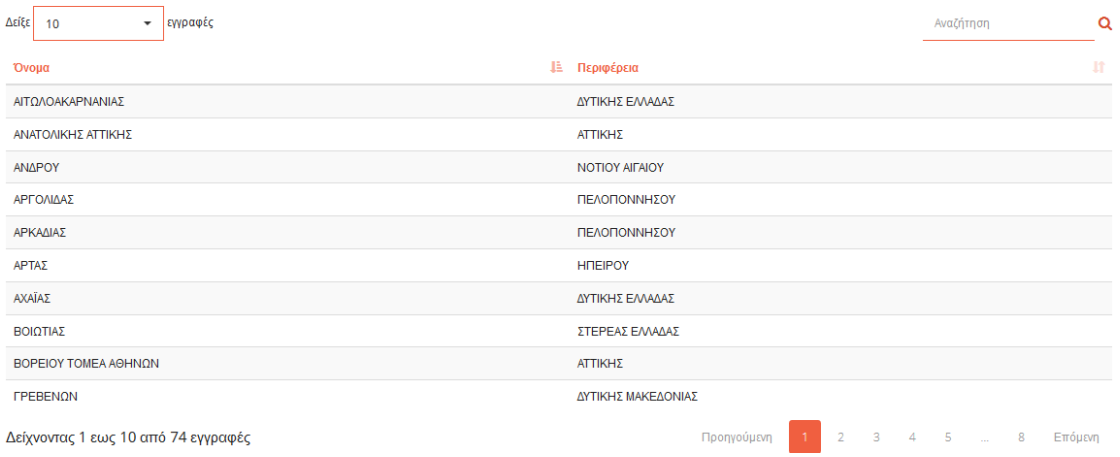
Στο μενού του χρήστη, υπάρχει η υπολίστα Πληροφορίες. Σε αυτήν, παρουσιάζονται, χρησιμοποιώντας το Datatable plugin, οι λίστες των Περιφερειών, Νομών, Δήμων και Πυροσβεστικών Υπηρεσιών. Σε αυτές δεν μπορεί να γίνει κάποια εισαγωγή, επεξεργασία ή διαγραφή, καθώς αποτελούν μέρος της πληροφόρησης απλά του χρήστη. Η μόνη διαφορά προκύπτει στην σελίδα των Δήμων, στην οποία μπορεί να γίνει η Εξαγωγή αυτών σε μορφή CSV (municipalities.csv), με σκοπό την ενημέρωση του χρήστη στο πως έχουν εισαχθεί οι δήμοι, ο νομός και η περιφέρεια αυτών στους πίνακες της βάσης δεδομένων (municipalities, prefectures, regions).

Παρακάτω δίνονται οι αντίστοιχες εικόνες από τις σελίδες των Περιφερειών, Νομών, Δήμων και Πυροσβεστικών Υπηρεσιών.

Η λίστα των Περιφερειών (βλ. Εικόνα 49) είναι η view list στο μονοπάτι admin/regions και χρησιμοποιείται ο RegionController.

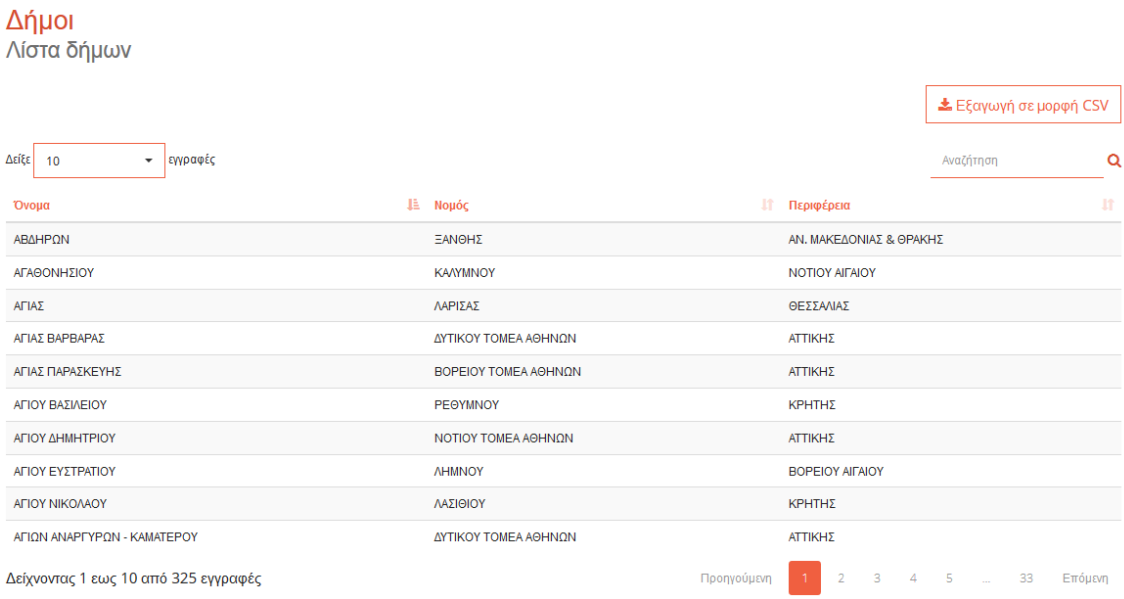
Εικόνα 49 – Λίστα Περιφερειών

Η λίστα των Νομών (βλ. Εικόνα 50) είναι η view list στο μονοπάτι admin/prefectures και χρησιμοποιείται ο PrefectureController.



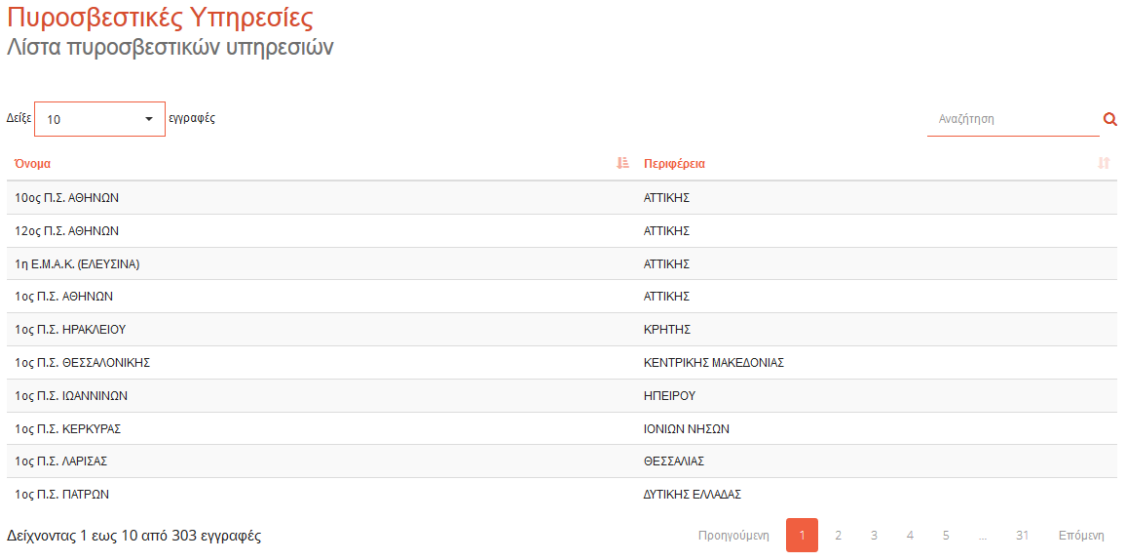
Εικόνα 50 – Λίστα Νομών

Η λίστα των Δήμων (βλ. Εικόνα 51) είναι η view list στο μονοπάτι admin/municipalities και χρησιμοποιείται ο MunicipalityController.



Εικόνα 51 – Λίστα Δήμων

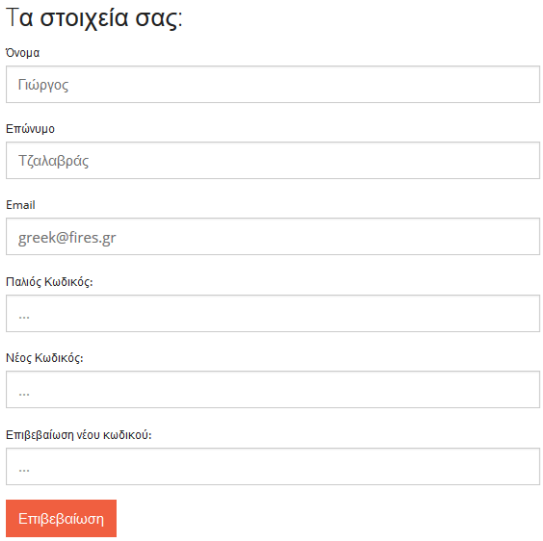
Η λίστα των Πυροσβεστικών Υπηρεσιών (βλ. Εικόνα 52) είναι η view list στο μονοπάτι admin/departments και χρησιμοποιείται ο FireDepartmentController.



Εικόνα 52 – Λίστα Πυροσβεστικών Υπηρεσιών

Σελίδα Προφίλ

Η σελίδα ‘Προφίλ’ (βλ. Εικόνα 53) είναι η view profile η οποία χρησιμοποίει τον AdminController και σε αυτή μπορεί να γίνει η επεξεργασία των στοιχείων του χρήστη (Admin), όπως και η αλλαγή κωδικού και email για την εισαγωγή στο σύστημα.

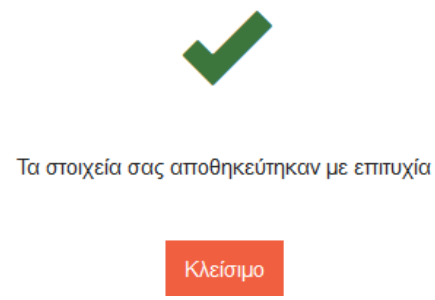


Εικόνα 53 – Σελίδα Προφίλ Χρήστη

Σε περίπτωση λανθασμένης εισαγωγής Παλιού Κωδικού, η αλλαγή των στοιχειών δεν γίνεται και εμφανίζεται κατάλληλο μήνυμα λάθους (βλ. Εικόνα 54). Το ίδιο συμβαίνει εάν δεν εισαχθούν τα πεδία 'Όνομα', 'Επώνυμο' και 'Email΄. Επίσης, το ίδιο συμβαίνει εάν δεν εισαχθεί ο ίδιος κωδικός στα πεδία 'Νέος Κωδικός' και 'Επιβεβαίωση νέου κωδικού'.

Εικόνα 54 – Μήνυμα λάθους εισαγωγής παλαιού κωδικού

Εάν ο χρήστης εισάγει επιτυχώς τις τιμές στα πεδία τότε αποθηκεύονται επιτυχώς τα δεδομένα και εμφανίζεται ένα παράθυρο που επαληθεύει την σωστή αλλαγή των δεδομένων (βλ. Εικόνα 55).



Εικόνα 55 – Επιτυχής αλλαγή στοιχειών του Προφίλ

Έξοδος από το σύστημα Back-end

Για να αποσυνδεθεί ο χρήστης από το σύστημα, χρειάζεται να πατήσει το κουμπί έξοδος. Αυτό διαγράφει τις τιμές Session του email και του κωδικού του χρήστη και έτσι δεν μπορεί ο χρήστης, εάν δεν συνδεθεί μέσω της σελίδας login, να μεταβεί σε οποιαδήποτε σελίδα του συστήματος Back-end.

Η συνάρτηση που εκτελεί την συγκεκριμένη διαδικασία έχει ως εξής:

```
public function logoutCreate()
{
    Auth::logout();
    Session::flush();
    return Redirect::to('/admin');
}
```

Με την εντολή Auth::logout() ο χρήστης αποσυνδέεται από το σύστημα και με την εντολή Session::flush() διαγράφονται οι τιμές Session του email και του κωδικού που εισήγαγε ο χρήστης, προκειμένου να μην μπορέσει να μεταβεί σε καμία σελίδα του Back-end στην οποία απαιτείται η σύνδεσή του.

4.3.2 Front-end

Το Front-end της εργασίας αποτελεί την κύρια σελίδα του site. Πρόκειται για την σελίδα την οποία μπορούν να επισκεφτούν όλοι οι χρήστες, τα στοιχεία της οποίας έχουν εισαχθεί από τον διαχειριστή του Back-end. Η σελίδα αυτή βασίζεται στην αρχιτεκτονική της One Page σελίδας, δηλαδή της σελίδας η οποία περιέχει όλη την πληροφορία σε μία σελίδα και η οποία χωρίζεται σε κάποια μέρη (sections).

Η σελίδα αυτή χωρίζεται στα εξής 4 μέρη:

Α) Είσοδος (Αρχική εικόνα της σελίδας)

Β) Χάρτης (Οι πυρκαγιές πάνω σε χρονικό χάρτη)

Γ) Στατιστικά (Διάγραμμα και Πίτες Στατιστικών)

Δ) Επικοινωνία (Φόρμα Επικοινωνίας μέσω αποστολής email)

Τα jQuery plugins τα οποία χρησιμοποιούνται είναι τα εξής:

1) bootrap-select.min.js [39]
   Πρόκειται για ένα plugin του Bootstrap, το οποίο βοηθάει στη δημιουργία πιο όμορφου στυλ μίας λίστας select. Η λίστα των νομών, με βάση την οποία φιλτράρονται οι πυρκαγιές πάνω στον χάρτη, χρησιμοποίει το plugin αυτό.

2) jquery.easing.js και scrolling-nav.js [40]
   Πρόκειται για δύο plugins, με τα οποία επιτυγχάνεται το scroll με κίνηση smooth της σελίδας.

3) chart.min.js [41]
   Είναι το plugin με το οποίο κατασκευάστηκαν το διάγραμμα και οι πίτες στη λειτουργία ‘Στατιστικά’.

4) bootstrap.newsbox.min.js [42]
   Στη λειτουργία ‘Στατιστικά’ χρησιμοποιείται το plugin αυτό για την προβολή 3 παραθύρων (των 10 μεγαλύτερων καμένων περιοχών, των 10 μεγαλύτερων πυρκαγιών καθώς και των 10 τελευταίων πυρκαγιών). Με το plugin επιτυγχάνεται η απεικόνιση αυτών σε μορφή νέων (tweets).

5) inview.min.js
   Με το συγκεκριμένο plugin επιτυγχάνεται το animation που χρησιμοποιείται στον κώδικα CSS, με το οποίο εμφανίζονται τα 3 παραπάνω παράθυρα με χρονική διαφορά.

6) Timemap (js) [43]
Στον φάκελο timemap του φακέλου js, χρησιμοποιούνται τα plugins που χρειάζονται για την δημιουργία του χρονικού Google Maps χάρτη καθώς και την απεικόνιση των πυρκαγιών επάνω σε αυτόν.

Είσοδος

Στη λειτουργία είσοδος, υπάρχει η κυρίως εικόνα του site με το logo και την λίστα των επιμέρους μερών της σελίδας (Χάρτης, Στατιστικά, Επικοινωνία), όπως απεικονίζει η Εικόνα 56.

Με την επιλογή ενός από αυτών γίνεται η μετάβαση στο συγκεκριμένο σημείο της σελίδας, ενώ με κλικ στο logo η σελίδα επιστρέφει στην αρχή. Με το πάτημα του κουμπιού 'Περισσότερα' (βλ. Εικόνα 57) η σελίδα μεταβαίνει στο επόμενο μέρος (Χάρτης).

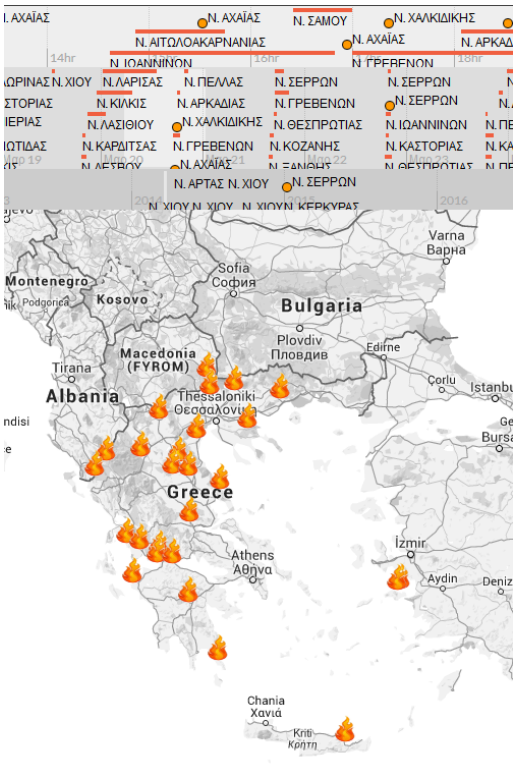
Εικόνα 56 – Μενού σελίδας Front-end

Εικόνα 57 – Κουμπί Περισσότερα

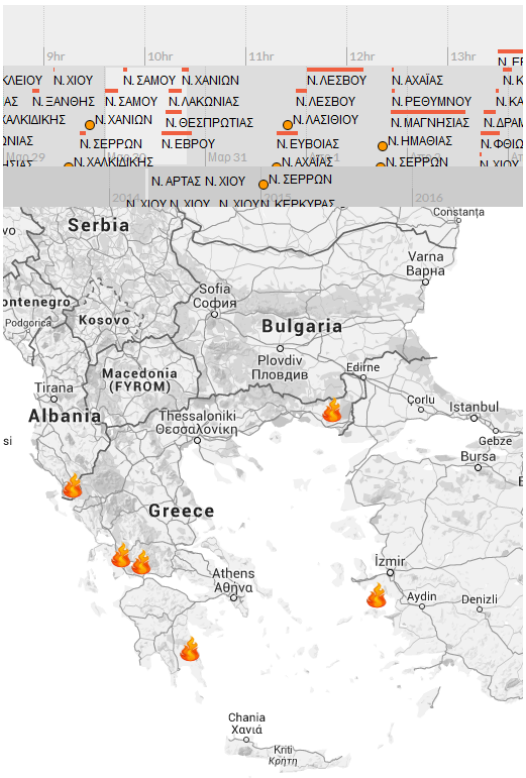
Χάρτης

Στο συγκεκριμένο μέρος γίνεται η απεικόνιση των πυρκαγιών πάνω στον χάρτη με βάση την ημερομηνία έναρξής τους. Για τον χάρτη έχει χρησιμοποιηθεί το plugin timemap το οποίο υπάρχει στον σύνδεσμο: code.google.com/p/timemap/. Μετακινώντας τις χρονικές μπάρες αριστερά και δεξιά, εμφανίζονται οι πυρκαγιές με βάση τις ημερομηνίες οι οποίες είναι ορατές στην οθόνη.

Για την κατανόηση της χρονολογικής απεικόνισης των πυρκαγιών επάνω στον χάρτη, δίνονται οι επόμενες δύο εικόνες (βλ Εικόνα 58, Εικόνα 59). Η εικόνα στα αριστερά απεικονίζει τις πυρκαγιές οι οποίες βρίσκονται σε εξέλιξη κατά την 20η Μαρτίου του 2014 και ώρες μεταξύ 7α.μ και 23π.μ. και η εικόνα στα δεξιά απεικονίζει τις πυρκαγιές κατά την 30η Μαρτίου του 2014 και ώρες μεταξύ 2α.μ. και 20π.μ..

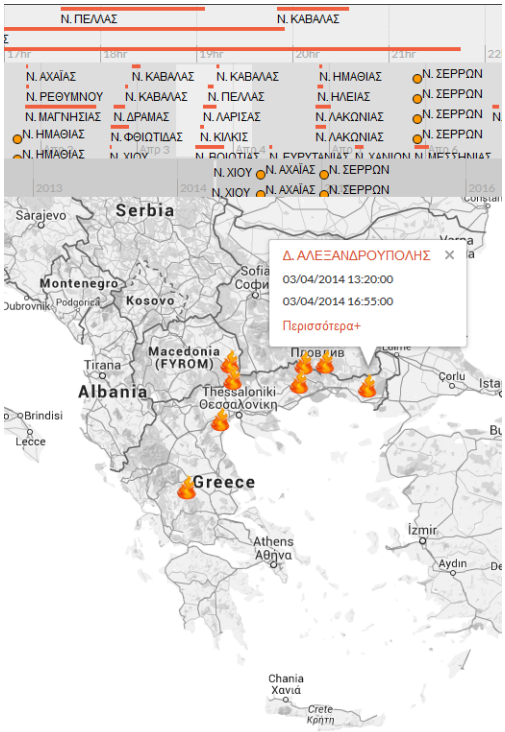


Εικόνα 58 – Πυρκαγιές 20-03-2014

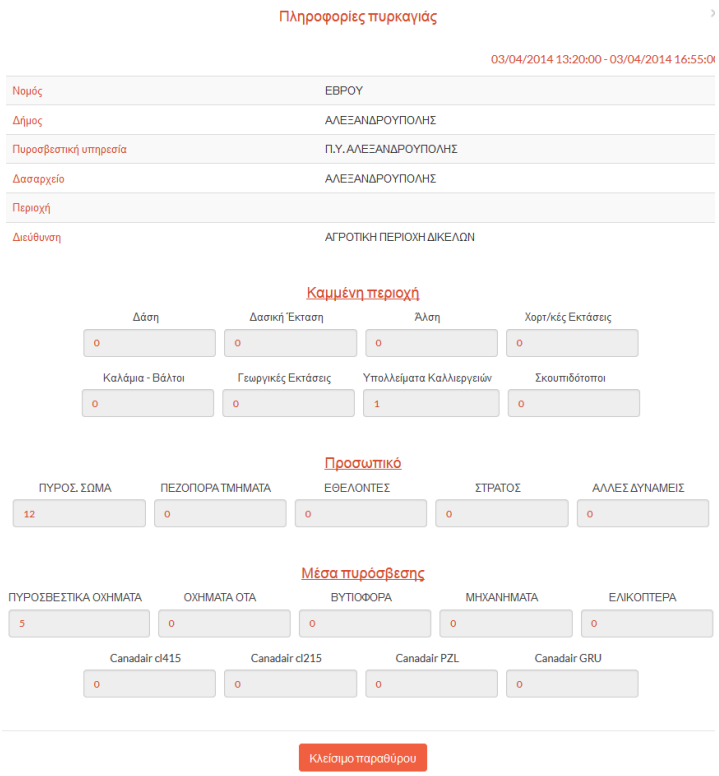


Εικόνα 59 – Πυρκαγιές 30-03-2014

Πατώντας πάνω σε μία πυρκαγιά στον χάρτη ή στο χρονολόγιο, εμφανίζεται το παράθυρο πάνω από το εικονίδιο της συγκεκριμένης πυρκαγιάς με κάποιες πληροφορίες (βλ. Εικόνα 60). Ο σύνδεσμος 'Περισσότερα +' ανοίγει ένα παράθυρο (modal), στο οποίο υπάρχουν όλες οι πληροφορίες της πυρκαγιάς (βλ. Εικόνα 61).



Εικόνα 60 – Επιλογή Πυρκαγιάς



Εικόνα 61 – Παράθυρο πληροφοριών πυρκαγιάς

Στατιστικά

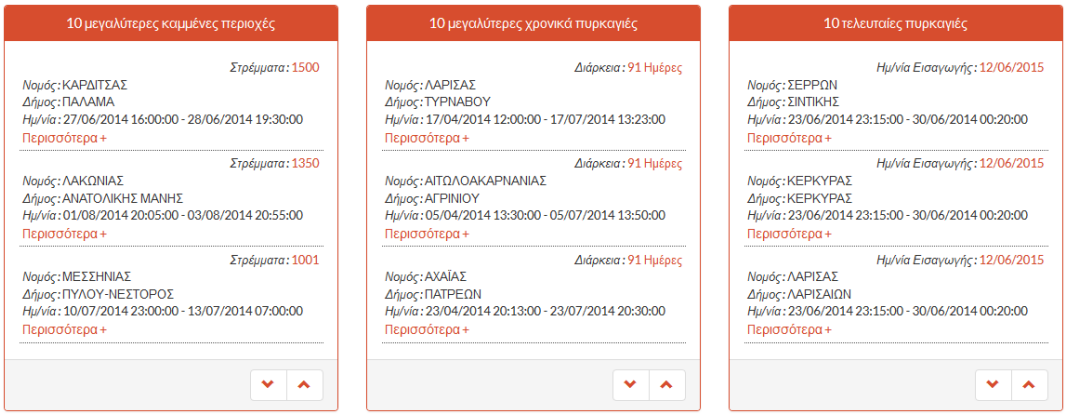
Η λειτουργία Στατιστικά χωρίζεται σε τρία τμήματα.

Στο πρώτο τμήμα υπάρχουν τρία παράθυρα με βάση τα οποία εμφανίζονται οι 10 μεγαλύτερες καμένες περιοχές, οι 10 μεγαλύτερες πυρκαγιές καθώς και οι 10 τελευταίες πυρκαγιές.

Στο δεύτερο τμήμα υπάρχει η κατανομή των μηνών με βάση των αριθμό των πυρκαγιών ανά μήνα.

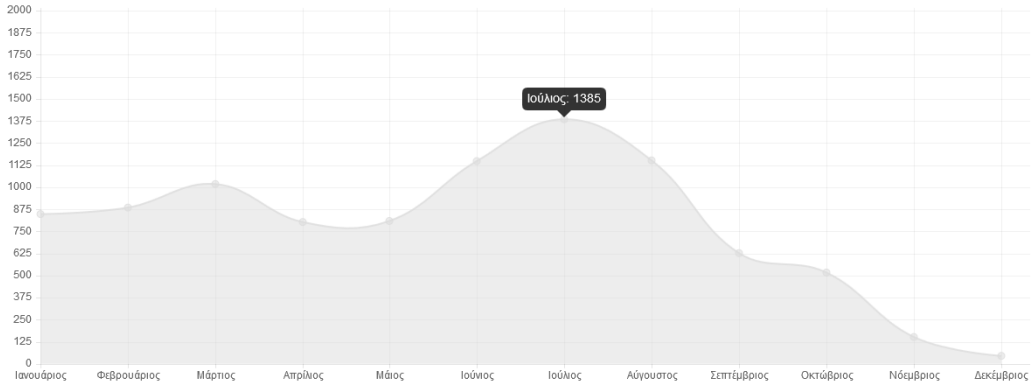
Στο τρίτο τμήμα υπάρχουν δύο πίτες, μία που εμφανίζει το σύνολο των πυρκαγιών ανά νομό (σε σύνολο 12 νομών) και μία δεύτερη που εμφανίζει το σύνολο των πυρκαγιών ανά περιφέρεια (13 περιφέρειες).

Στο πρώτο τμήμα, με τα βελάκια πάνω και κάτω εμφανίζεται η επόμενη ή η προηγούμενη πυρκαγιά ενώ με το κουμπί περισσότερα εμφανίζονται όλες οι πληροφορίες της πυρκαγιάς με την προβολή παραθύρου, όπως δείχνει η Εικόνα 62.



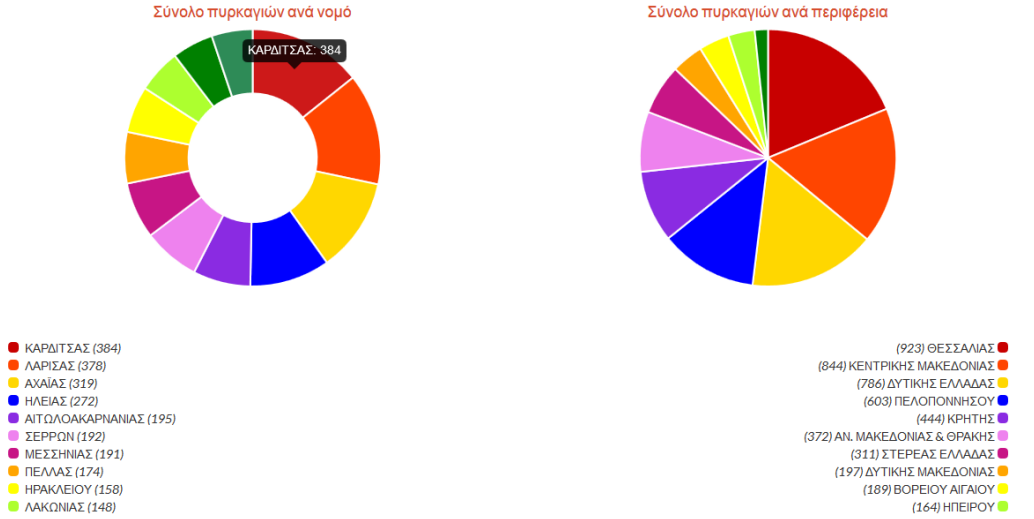
Εικόνα 62 – Πρώτο μέρος Στατιστικών

Στο δεύτερο μέρος απεικονίζεται το σύνολο των πυρκαγιών ανά μήνα, ενώ περνώντας τον κέρσορα πάνω από έναν μήνα, εμφανίζεται ο ακριβής αριθμός των πυρκαγιών του μήνα, όπως απεικονίζει η παρακάτω εικόνα (Εικόνα 63).



Εικόνα 63 – Δεύτερο μέρος Στατιστικών

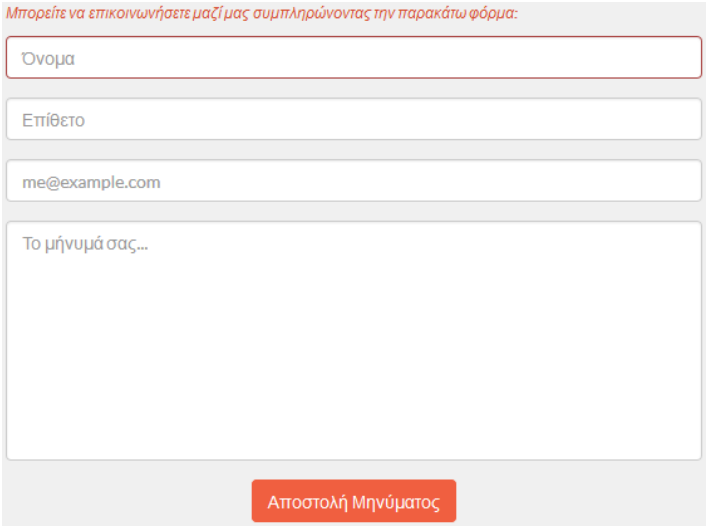
Το τρίτο μέρος περιέχει τις δύο πίτες (βλ. Εικόνα 64). Περνώντας τον κέρσορα πάνω από ένα κομμάτι της πίτας ή με ένα κλικ σε αυτό σε περίπτωση που η οθόνη είναι μικρή, εμφανίζεται ο επακριβής αριθμός των πυρκαγιών. Επίσης, κάτω από κάθε πίτα εμφανίζεται η λίστα των νομών-περιφερειών και το αντίστοιχο χρώμα που έχουν αυτοί επάνω στην πίτα.



Εικόνα 64 – Τρίτο μέρος Στατιστικών

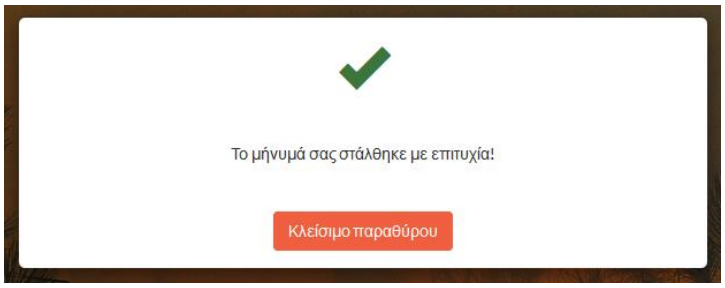
Επικοινωνία

Στο τελευταίο μέρος της σελίδας (βλ. Εικόνα 65) υπάρχει η φόρμα επικοινωνίας, μέσω της οποίας οποιοσδήποτε χρήστης της σελίδας μπορεί να στείλει ένα μήνυμα στο mail της σελίδας greekfires.gr. Σε περίπτωση μη σωστής εισαγωγής τιμής σε πεδίο της φόρμας, το πεδίο αυτό αποκτά κόκκινο χρώμα.



Εικόνα 65 – Φόρμα Επικοινωνίας

Σε περίπτωση επιτυχούς αποστολής του μηνύματος, ο χρήστης ενημερώνεται σχετικά με την προβολή αναλόγου μηνύματος επιτυχίας (βλ. Εικόνα 66).



Εικόνα 66 – Επιτυχής αποστολή μηνύματος

ΚΕΦΑΛΑΙΟ 5 : ΣΥΜΠΕΡΑΣΜΑΤΑ – ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ

Συνοψίζοντας την εργασία αυτή, αρχικά στο δεύτερο κεφάλαιο, δίνεται ο ορισμός των δημόσιων ανοικτών δεδομένων και παρουσιάζονται τα πρότυπα βάσει των οποίων παρουσιάζονται τα δεδομένα αυτά. Επιπλέον, γίνεται η ανάλυση του μοντέλου 5-star Open Data, το οποίο κατηγοριοποιεί σε πέντε βαθμίδες την «ανοικτότητα» των δεδομένων. Παράλληλα, αναλύεται επίσης και το μοντέλο Linked Open Data (LOD), το οποίο συνίσταται από διασυνδεδεμένα δεδομένα της πέμπτης βαθμίδας του μοντέλου 5-star Open Data. Επίσης, παρέχονται παραδείγματα εφαρμογών και πρωτοβουλιών, οι οποίες παρουσιάζουν διάφορα δημόσια δεδομένα. Τέλος, πραγματοποιείται και η ανάλυση των δημόσιων δεδομένων της Πυροσβεστικής Υπηρεσίας με τα οποία ασχολείται η συγκεκριμένη εργασία.

Στο τρίτο κεφάλαιο αναλύεται το προτεινόμενο σύστημα σε επίπεδο αρχιτεκτονικής και τεχνολογιών που χρησιμοποιήθηκαν, καθώς και η βάση δεδομένων που το στηρίζει. Αρχικά πραγματοποιείται η ανάλυση των τεχνολογιών σε επίπεδο client-side (front-end) και έπειτα σε επίπεδο server-side (back-end). Τέλος, παρουσιάζεται η δομή και οι συσχετίσεις όλων των πεδίων και των ιδιοτήτων αυτών (σχήμα βάσης δεδομένων) μέσω της παρουσίασης του σχεσιακού μοντέλου και του μοντέλου οντοτήτων-συσχετίσεων.

Έχοντας ήδη αναλυθεί οι τεχνολογίες που χρησιμοποιούνται στην εργασία αυτή και η βάση δεδομένων, στο τέταρτο κεφάλαιο παρουσιάζεται ο τρόπος λειτουργίας της εφαρμογής greekfires.gr τόσο σε front-end όσο και σε back-end επίπεδο, ενώ αναλύονται ο σκοπός της εφαρμογής με τα οφέλη τα οποία προκύπτουν από την δημιουργία της.

Από τα δημόσια και ανοιχτά δεδομένα που προσφέρονται από την Πυροσβεστική Υπηρεσία, δημιουργήθηκε λοιπόν ένας δικτυακός τόπος για την πληροφόρηση των πολιτών σχετικά με τις δασικές πυρκαγιές της Ελλάδος. Τα δεδομένα αυτά προέρχονται από τα δημόσια ανοιχτά δεδομένα που προσφέρονται από την Πυροσβεστική, τα οποία δίνονται από το αρχείο Excel (Δασικές Πυρκαγιές 2014) στην σελίδα Δημόσια Δεδομένα της ιστοσελίδας της Πυροσβεστικής (www.fireservice.gr). Παράλληλα, υλοποιήθηκε μία back-end εφαρμογή για την δομημένη εισαγωγή και επεξεργασία περιστατικών πυρκαγιών ανά την ευρύτερη περιφέρεια. Πέραν της ενίσχυσης της πληροφόρησης των πολιτών, η υπηρεσία αυτή μπορεί να χρησιμοποιηθεί από οποιονδήποτε ενδιαφερόμενο φορέα (ιδιαίτερα τον πάροχο των ανοικτών δεδομένων) με σκοπό την ομογενοποιημένη έκδοση των ανοικτών δεδομένων.

Στόχος της συγκεκριμένης εργασίας ήταν η λήψη και απεικόνιση των δημοσίων ανοιχτών δεδομένων της Πυροσβεστικής σχετικά με τις δασικές πυρκαγιές της Ελλάδος με άμεση εφαρμογή για το έτος 2014. Έτσι, δημιουργήθηκε ένας δικτυακός τόπος με λειτουργίες back end και front end που αλληλεπιδρούν προκειμένου να πραγματοποιηθεί λήψη, επεξεργασία, αποθήκευση, ανάκτηση και απεικόνιση πληροφοριών σχετικά με περιστατικά πυρκαγιών στην Ελληνική επικράτεια. Τα δύο αυτά μέρη έχουν ως εξής:

Α) το back-end, το οποίο διαλειτουργεί την διαχείριση και εισαγωγή πυρκαγιών μέσω ενός ρόλου διαχειριστή και

Β) το front-end, το οποίο απεικονίζει με χρονολογική σειρά τις πυρκαγιές πάνω σε έναν χάρτη, καθώς και επιμέρους στατιστικά τα οποία προκύπτουν από το σύνολο αυτών.

Η back-end εφαρμογή δημιουργήθηκε με τέτοιο τρόπο, ώστε να μπορεί να αποτελέσει ένα αυτόνομο module, βάσει του οποίου θα μπορεί να δημιουργηθεί ένα ανεξάρτητο front-end ανάλογα με τις ανάγκες της εκάστοτε εφαρμογής ή μία βάση δεδομένων η οποία θα περιέχει όλες τις πληροφορίες από περιστατικά πυρκαγιών.

Στο front-end δημιουργήθηκε ένας δικτυακός τόπος (greeekfires.gr) στον οποίο κάθε χρήστης-πολίτης μπορεί να ενημερώνεται σχετικά με περιστατικά πυρκαγιών. Πιο αναλυτικά, ο τελικός χρήστης μπορεί να βλέπει το σύνολο των περιστατικών πυρκαγιών πάνω σε διαδραστικό χάρτη με βάση τη διάρκεια τους. Επίσης, μπορεί να ενημερωθεί σχετικά με επαυξημένες και αναλυτικές πληροφορίες όπως αυτές παρέχονται από τα Ανοικτά Δημόσια Δεδομένα της Πυροσβεστικής Υπηρεσίας (όπως λ.χ. εμπλεκόμενο πυροσβεστικό τμήμα, έμψυχο προσωπικό, μέσα πυρόσβεσης κ.α.), καθώς και να ενημερωθεί με την χρήση στατιστικών πάνω στο σύνολο των δεδομένων αυτών (π.χ. μήνας με τις περισσότερες πυρκαγιές, νομοί-περιφέρειες με τις περισσότερες πυρκαγιές, δέκα μεγαλύτερες χρονικά πυρκαγιές, δέκα πιο καταστροφικές πυρκαγιές κ.α.).

Με την συγκεκριμένη μεταπτυχιακή εργασία όμως, ενδέχεται να προκύψουν και οφέλη όπως:

1) Χρήση του back-end από την ίδια την Πυροσβεστική Υπηρεσία. Έχει γίνει η κατάλληλη μελέτη για υποστήριξη όλων των πεδίων που εισάγονται χειροκίνητα στα Ανοικτά Δεδομένα της Πυροσβεστικής (σε μορφή excel).

2) Επέκταση του back-end στο πεδίο 'Πυροσβεστικές Υπηρεσίες', προκειμένου να εισαχθούν περισσότερες πληροφορίες σχετικά με αυτές (όπως για παράδειγμα διεύθυνση, τηλέφωνο, ταχυδρομικός κώδικας, γεωγραφικό μήκος και πλάτος). Έχοντας και γεω-συσχετισμένη πληροφορία σε ανοικτά πρότυπα η πληροφορία προς τον τελικό χρήστη θα ήταν πιο ακριβής. Μάλιστα, για τη συγκεκριμένη επέκταση έχουν ήδη δημιουργηθεί τα αντίστοιχα πεδία στον πίνακα 'fire_departments' στην βάση δεδομένων.

3) Τροποποίηση των στατιστικών στο front-end, προκειμένου να προβάλλονται οι πυρκαγιές με βάση το κάθε έτος.

4) Τροποποίηση εισαγωγής των γεωγραφικών δεδομένων της πυρκαγιάς από απλό σημείο (latitude-longitude) σε ένα σύνολο σημείων (πολύγωνο) αυτής. Αυτό βέβαια απαιτεί την κατάλληλη τροποποίηση του back και του front-end, καθώς και της βάσης δεδομένων. Στο back-end, επιβάλλεται η χρήση ενός άλλου plugin, με το οποίο θα μπορούσε να αποθηκευτεί η συγκεκριμένη πληροφορία. Σε επίπεδο front-end, το plugin Timemap προσφέρει την δυνατότητα εισαγωγής πολυγώνων με βάση το σύνολο των latitude και longitude σημείων των οποίων θα παρέχονται. Όμως ενδέχεται, η εισαγωγή πολλών δεδομένων να το καταστήσει ιδιαίτερα αργό (απαίτηση μνήμης). Τέλος, σε επίπεδο βάσης δεδομένων, θα χρειαζόταν να δημιουργηθεί ένας νέος πίνακας (και ένα νέο μοντέλο), ο οποίος θα περιλάμβανε όλα τα σημεία (latitude-longitude) του πολύγωνου. Ο πίνακας αυτός μπορεί να είναι ο 'locations' με πεδία τα id, latitude, longitude, id_fire.

5) Στο back-end της εφαρμογής, έχει προβλεφθεί η εισαγωγή πληροφορίας νέων πυροσβεστικών μέσων και προσωπικού. Για την αποτύπωση αυτών κατά την εισαγωγή και την εξαγωγή ενός αρχείου με το σύνολο των πυρκαγιών, θα απαιτείται προγραμματιστική παρέμβαση.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] *Σελίδα Πυροσβεστικής Υπηρεσίας www.fireservice.gr , τελευταία εύρεση 5/8/2015*
[2] *Κωδικολόγιο Καλλικράτειων Δήμων και Κοινοτήτων, http://www.ypes.gr/el/regions/programma/ , τελευταία εύρεση 5/8/2015*
[3] *Τι είναι τα ανοικτά δεδομένα, http://opendatahandbook.org/guide/el/what-is-open-data/, τελευταία εύρεση 5/8/2015*
[4] *Open data format, https://en.wikipedia.org/wiki/Open_format , τελευταία εύρεση 5/8/2015*
[5] *Plain text, https://en.wikipedia.org/wiki/Plain_text, τελευταία εύρεση 5/8/2015*
[6] *Comma-separated values, https://en.wikipedia.org/wiki/Comma-separated_values, τελευταία εύρεση 5/8/2015*
[7] *Extensible Markup Language (XML) , http://www.w3.org/XML/ , τελευταία εύρεση 5/8/2015*
[8] *Introducing JSON, http://json.org/ , τελευταία εύρεση 5/8/2015*
[9] *Resource Description Framework (RDF), http://www.w3.org/RDF/, https://en.wikipedia.org/wiki/Resource_Description_Framework , τελευταία εύρεση 5/8/2015*
[10] *5 Star Open Data, http://5stardata.info/ , τελευταία εύρεση 5/8/2015*
[11] *Linked Open Data, https://en.wikipedia.org/wiki/Linked_open_data , τελευταία εύρεση 5/8/2015*
[12] *Open Data , http://ec.europa.eu/digital-agenda/en/open-data-0 , τελευταία εύρεση 5/8/2015*
[13] *Europe's Public Data, http://publicdata.eu/dataset , τελευταία εύρεση 5/8/2015*
[14] *Open Data Paris, http://opendata.paris.fr , τελευταία εύρεση 5/8/2015*
[15] *Plateforme ouverte des données publiques françaises , www.data.gouv.fr, τελευταία εύρεση 5/8/2015*
[16] *Dati Open Data Services, www.dati.piemonte.it, τελευταία εύρεση 5/8/2015*
[17] *Dataportaal van de Nederlandse overheid , www.data.overheid.nl, τελευταία εύρεση 5/8/2015*
[18] *Opening up Government, www.data.gov.uk , τελευταία εύρεση 5/8/2015*
[19] *Δημόσια, Ανοικτά Δεδομένα, http://geodata.gov.gr/geodata/ , τελευταία εύρεση 5/8/2015*
[20] *Ανοικτή Διακυβέρνηση, http://www.opengov.gr/home/, τελευταία εύρεση 5/8/2015*
[21] *Lorax.gr - Πύλη για τη διευκόλυνση της πρόσβασης στα ανοικτά δημόσια δεδομένα, http://www.lorax.gr, τελευταία εύρεση 5/8/2015*
[22] *Ελληνική Αστυνομία, astynomia.gr, τελευταία εύρεση 5/8/2015*
[23] *Πυροσβεστικό Σώμα / Δημόσια Δεδομένα, http://www.fireservice.gr/pyr/site/home/LC+Secondary+Menu/opendata.csp , τελευταία εύρεση 5/8/2015*
[24] *Bootstrap, https://fr.wikipedia.org/wiki/Twitter_Bootstrap , τελευταία εύρεση 5/8/2015*
[25] *Bootstrap Grid, http://getbootstrap.com/css/#grid , τελευταία εύρεση 5/8/2015*
[26] *Bootstrap CSS, http://getbootstrap.com/css/ , τελευταία εύρεση 5/8/2015*
[27] *Bootstrap Media Queries, http://getbootstrap.com/css/#grid , τελευταία εύρεση 5/8/2015*
[28] *Blade Template, http://laravel.com/docs/4.2/templates , τελευταία εύρεση 5/8/2015*
[29] *Blade και HTML, http://laravel-recipes.com/ , τελευταία εύρεση 5/8/2015*
[30] *Laravel PHP Framework, https://en.wikipedia.org/wiki/Laravel , τελευταία εύρεση 5/8/2015*
[31] *Laravel Installation, http://laravel.com/docs/4.2 , τελευταία εύρεση 5/8/2015*
[32] *Model-View-controller, https://el.wikipedia.org/wiki/Model-view-controller , τελευταία εύρεση 5/8/2015*
[33] *Eloquent, http://laravel.com/docs/4.2/eloquent , τελευταία εύρεση 5/8/2015*
[34] *Eloquent Relationships, http://laravel.com/docs/4.2/eloquent#relationships , τελευταία εύρεση 5/8/2015*
[35] *Laravel Schema Builder, http://laravelsd.com/ , τελευταία εύρεση 5/8/2015*
[36] *Datatables, https://www.datatables.net/ , τελευταία εύρεση 5/8/2015*
[37] *Bootstrap Datetimepicker, https://eonasdan.github.io/bootstrap-datetimepicker/" , τελευταία εύρεση 5/8/2015*

[38] *Jquery Locationpicker, http://logicify.github.io/jquery-locationpicker-plugin/ , τελευταία εύρεση 5/8/2015*
[39] *Bootstrap Select, http://silviomoreto.github.io/bootstrap-select/ , τελευταία εύρεση 5/8/2015*
[40] *Scrolling Nav, http://startbootstrap.com/template-overviews/scrolling-nav/, τελευταία εύρεση 5/8/2015*
[41] *Chart JS, http://www.chartjs.org/docs/ , τελευταία εύρεση 5/8/2015*
[42] *Bootstrap News Ticker Plugin, http://www.jqueryscript.net/slider/Responsive-jQuery-News-Ticker-Plugin-with-Bootstrap-3-Bootstrap-News-Box.html , τελευταία εύρεση 5/8/2015*
[43] *Timemap JS, https://code.google.com/p/timemap/ , τελευταία εύρεση 5/8/2015*

ΠΑΡΑΡΤΗΜΑ: ΚΩΔΙΚΑΣ

ΑΡΧΕΙΟ routes.php

```
Route::get('/', array('uses' => 'HomeController@showHomePageCreate'));
Route::post('contact', array('uses' => 'HomeController@getContactStore'));

Route::get('fires/on/map/callback/{callback}', array('uses' => 'HomeController@showFiresOnMapCreate'));

Route::post('get/smallest/fire/date/by/prefecture/id', array('uses' => 'HomeController@getSmallestFireDateByPrefectureId'));
Route::post('get/fire/informations/by/id', array('uses' => 'HomeController@getFireInformationsById'));

Route::get('/admin', array('uses' => 'AdminController@loginCreate'));
Route::post('/admin/login', array('uses' => 'AdminController@loginStore'));
Route::get('/logout', array('as' => 'logout', 'uses' => 'AdminController@logoutCreate'));

if (Auth::check()) {

    /** DASHBOARD **/
    Route::get('/dashboard', array('uses' => 'AdminController@dashboardCreate'));

    /** FIRES **/
    Route::get('fires/list', array('as' => 'fires_list', 'uses' => 'FireController@firesListCreate'));
    Route::post('fires/list/datatables', array('uses' => 'DataTablesController@datatable'));
    Route::get('fire/insert', array('as' => 'fire_insert', 'uses' => 'FireController@fireInsertCreate'));
    Route::post('fire/insert', array('as' => 'fire_insert', 'uses' => 'FireController@fireInsertStore'));
    Route::get('fire/edit/{id}', array('as' => 'fire_edit', 'uses' => 'FireController@fireEditCreate'))->where('id', '[0-9]+');
    Route::post('fire/edit/{id}', array('as' => 'fire_edit', 'uses' => 'FireController@fireEditStore'))->where('id', '[0-9]+');
    Route::post('fire/delete', array('uses' => 'FireController@fireDeleteStore'));
    Route::post('get/fire/informations', array('uses' => 'FireController@getFireInformations'));

    Route::post('fires/export', array('uses' => 'FireController@firesExport'));
    Route::post('file/upload', array('uses' => 'FireController@fileUpload'));
    Route::post('file/import', array('uses' => 'FireController@fileImport'));
    Route::post('file/delete', array('uses' => 'FireController@fileDelete'));

    Route::post('fire/edit/{id}/burnt/area', array('uses' => 'FireController@fireEditBurntAreaStore'))->where('id', '[0-9]+');
    Route::post('fire/edit/{id}/staff', array('uses' => 'FireController@fireEditStaffStore'))->where('id', '[0-9]+');
    Route::post('fire/edit/{id}/fire/forces', array('uses' => 'FireController@fireEditFireForcesStore'))->where('id', '[0-9]+');
```

```
        Route::post('get/informations/by/region/id', array('uses' => 'FireController@getInformationsByRegionId'));
        Route::post('get/municipality/by/prefecture/id', array('uses' => 'FireController@getMunicipalityByPrefectureId'));

        /** FORCES **/
        Route::get('forces/list', array('as' => 'forces_list', 'uses' => 'ForceController@forcesListCreate'));
        Route::post('force/insert', array('as' => 'forces_list', 'uses' => 'ForceController@forceInsertStore'));
        Route::post('force/modify', array('as' => 'forces_list', 'uses' => 'ForceController@forceModifyStore'));
        Route::post('force/get/by/id', array('uses' => 'ForceController@forceGetById'));
        Route::post('force/delete', array('uses' => 'ForceController@forceDeleteStore'));

        /** STAFF **/
        Route::get('staff/list', array('as' => 'staff_list', 'uses' => 'StaffController@staffListCreate'));
        Route::post('staff/insert', array('as' => 'staff_list','uses' => 'StaffController@staffInsertStore'));
        Route::post('staff/modify', array('as' => 'staff_list','uses' => 'StaffController@staffModifyStore'));
        Route::post('staff/get/by/id', array('uses' => 'StaffController@staffGetById'));
        Route::post('staff/delete', array('uses' => 'StaffController@staffDeleteStore'));

        /** REGIONS **/
        Route::get('regions/list', array('as' => 'regions_list', 'uses' => 'RegionController@regionsListCreate'));

        /** PREFECTURES **/
        Route::get('prefectures/list', array('as' => 'prefectures_list', 'uses' => 'PrefectureController@prefecturesListCreate'));

        /** MUNICIPALITIES **/
        Route::get('municipalities/list', array('as' => 'municipalities_list', 'uses' => 'MunicipalityController@municipalitiesListCreate'));
        Route::post('municipalities/export', array('uses' => 'MunicipalityController@municipalitiesExport'));

        /** FIRE DEPARTMENTS **/
        Route::get('fire/departments/list', array('as' => 'fire_departments_list', 'uses' => 'FireDepartmentController@fireDepartmentsListCreate'));

        /** PROFILE **/
        Route::get('/profile', array('as' => 'profile', 'uses' => 'AdminController@profileCreate'));
        Route::post('/profile/edit', array('as' => 'profile', 'uses' => 'AdminController@profileStore'));
    }
```

BACK END

Αρχείο master.blade.php (main template των Back End views):

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>@yield('title')</title>

    {{HTML::style('css/admin/bootstrap.min.css')}}
    {{HTML::style('css/font-awesome.min.css')}}
    {{HTML::style('css/admin/style.css')}}
        @yield('extraStyle')

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>

    @include('admin.shared.header')

    <div class="container">
       @yield('content')
    </div>

        @yield('extraModal')

    {{HTML::script('js/jquery-1.11.2.min.js')}}
    {{HTML::script('js/bootstrap.min.js')}}
    @yield('extraScript')
  </body>
</html>
```

Αρχείο header.blade.php (header του Back End):

```
<div class="navbar navbar-default navbar-fixed-top">
  <div class="container">
    <div class="navbar-header">
```

```
        <a href="@if (Auth::check()) {{URL::to('/dashboard')}} @endif" class="navbar-brand">GreekFires</a>
        <button class="navbar-toggle" type="button" data-toggle="collapse" data-target="#navbar-main">
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
      </div>
      @if (Auth::check())
      <div class="navbar-collapse collapse" id="navbar-main">
        <ul class="nav navbar-nav">
          <li class="dropdown @if (Route::currentRouteName() == 'fires_list' || Route::currentRouteName() == 'fire_insert' || Route::currentRouteName() == 'fire_edit') {{'active'}} @endif">
            <a class="dropdown-toggle" data-toggle="dropdown" href="#" id="themes">Πυρκαγιές <span class="caret"></span></a>
            <ul class="dropdown-menu" aria-labelledby="themes">
              <li class="{{ Route::currentRouteName() == 'fires_list' ? 'active' : ''; }}"><a href="{{URL::to('fires/list')}}">Λίστα πυρκαγιών</a></li>
              <li class="{{ Route::currentRouteName() == 'fire_insert' ? 'active' : ''; }}"><a href="{{URL::to('fire/insert')}}">Εισαγωγή νέας πυρκαγιάς</a></li>
            </ul>
          </li>

          <li class="{{ Route::currentRouteName() == 'forces_list' ? 'active' : ''; }}">
            <a href="{{URL::to('forces/list')}}">Πυροσβεστικά Μέσα</a>
          </li>

          <li class="{{ Route::currentRouteName() == 'staff_list' ? 'active' : ''; }}">
            <a href="{{URL::to('staff/list')}}">Προσωπικό</a>
          </li>

          <li class="dropdown @if ( Route::currentRouteName() == 'regions_list' || Route::currentRouteName() == 'prefectures_list' || Route::currentRouteName() == 'municipalities_list' || Route::currentRouteName() == 'fire_departments_list' )
                                {{'active'}}
                              @endif">
            <a class="dropdown-toggle" data-toggle="dropdown" href="#" id="themes">Πληροφορίες <span class="caret"></span></a>
            <ul class="dropdown-menu" aria-labelledby="themes">
              <li class="{{ Route::currentRouteName() == 'regions_list' ? 'active' : ''; }}">
                <a href="{{URL::to('regions/list')}}">Περιφέρειες</a>
              </li>
              <li class="{{ Route::currentRouteName() == 'prefectures_list' ? 'active' : ''; }}">
                <a href="{{URL::to('prefectures/list')}}">Νομοί</a>
              </li>
              <li class="{{ Route::currentRouteName() == 'municipalities_list' ? 'active' : ''; }}">
```

```
            <a href="{{URL::to('municipalities/list')}}">Δήμοι</a>
          </li>
          <li class="divider"></li>
          <li class="{{ Route::currentRouteName() == 'fire_departments_list' ? 'active' : ''; }}">
            <a href="{{URL::to('fire/departments/list')}}">Πυροσβεστικές Υπηρεσίες</a>
          </li>
        </ul>
      </li>


    </ul>

    <ul class="nav navbar-nav navbar-right">
      <li class="{{ Route::currentRouteName() == 'profile' ? 'active' : ''; }}"><a href="{{URL::to('profile')}}"><span class="glyphicon glyphicon-user"></span> Προφίλ</a></li>
      <li class="{{ Route::currentRouteName() == 'logout' ? 'active' : ''; }}"><a href="{{URL::to('logout')}}"><span class="glyphicon glyphicon-log-out"></span> Έξοδος</a></li>
    </ul>

  </div>
  @endif
 </div>
</div>
```

View login.blade.php (Σελίδα σύνδεσης στην εφαρμογή):

```
@section('title')
  {{$title}}
@stop

@section('content')
    <div class="row">
        <div class="col-xs-6">

            @if ($errors->has('email'))
                <div class="alert alert-danger alert-dismissible" role="alert">
                  <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>
                  <strong>Σφάλμα!</strong> Το πεδίο όνομα πρέπει να είναι ένα email
                </div>
            @endif

            @if ($errors->has('password'))
                <div class="alert alert-danger alert-dismissible" role="alert">
                  <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>
                  <strong>Σφάλμα!</strong> Το πεδίο κωδικός είναι υποχρεωτικό
```

```
                    </div>
                @endif

                @if (Session::has('error') && (Session::get('error') == "login"))
                    <div class="alert alert-danger alert-dismissible" role="alert">
                     <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>
                     <strong>Σφάλμα!</strong> <p>Έχετε εισάγει λανθασμένα στοιχεία.</p>
                    </div>
                @endif

            {{ Form::open(array('url' => 'admin/login',  'method' => 'post')) }}
                <div class="form-group  @if ($errors->has('email')) has-error @endif">
                  <label for="email">Email:</label>
                  {{Form::text('email', Input::old('email'), array('id' => 'email', 'class' => 'form-control','placeholder' => 'Email'))}}
                </div>
                <div class="form-group @if ($errors->has('password')) has-error @endif">
                  <label for="password">Κωδικός:</label>
                  {{Form::password('password', array('id' => 'password', 'class' => 'form-control','placeholder' => '...'))}}
                </div>
                <button type="submit" class="btn btn-primary">Επιβεβαίωση</button>
                {{ Form::close() }}
            </div>
            <div class="col-xs-6"></div>
        </div>
    @stop
```

Συνάρτηση loginCreate του AdminController για την προβολή της login.blade.php:

```
protected $layout = 'admin.shared.master';

    public function loginCreate()
    {
        $data = array (   'title'=> 'Σύνδεση' );
        $this->layout->content = View::make('admin.login')->with($data);
    }
```

Route admin για την προβολή της login.blade.php:

```
Route::get('/admin', array('uses' => 'AdminController@loginCreate'));
```

Συνάρτηση loginStore του AdminController για την υποβολή της φόρμας σύνδεσης:

```
    public function loginStore()
    {
```

```
                $data = array (   'title'=> 'Σύνδεση' );

                $rules = array(
            'email'        => array('required', 'email'),
            'password' => array('required')
        );

                $validation = Validator::make(Input::all(), $rules);

                if ($validation->fails())
        {
          return Redirect::to('/admin')->withErrors($validation)->withInput()->with($data);
        }
        else
        {

                $userdata = array(
                    'email'     => Input::get('email'),
                    'password'  => Input::get('password')
                );

                // attempt to do the login
                if (Auth::attempt($userdata)) {
                        return Redirect::to('/dashboard');
                } else {
                    return Redirect::to('/admin')->withErrors($validation)->withInput()->with($data)-
>with(array('error'=>'login'));
                }

                }
        }
```

Route admin/login για την εκτέλεση της loginStore:

```
Route::post('/admin/login', array('uses' => 'AdminController@loginStore'));
```

View dashboard.blade.php (Πρώτη σελίδα προβολής μετά την σύνδεση):

```
@section('title')
   {{$title}}
@stop

@section('content')

        <div class="row">
                <div class="col-xs-12">
                    <h1>       <span       id="logo">GreekFires</span><br><span><small><span
class="orange">Admin:</span> {{$user->name}} {{$user->last_name}}</small></h1>
```

```
            </div>
        </div>

        <div class="row topmargin">
            <div class="col-xs-12">
                <h3 class="orange">Πυρκαγιές</h3>
                <h4 class="topmargin">Έχετε εισάγει: <span class="orange">{{$count_fires}}</span> @if ($count_fires == 1) πυρκαγιά. @else πυρκαγιές. @endif</h4>
                @if (isset($last_fire))<h4>Την τελευταία πυρκαγιά την εισάγατε στις: <span class="orange">{{$last_fire->created_at}}</span></h4>@endif
                <h4>Μπορείτε να εισάγετε μία νέα πυρκαγιά πατώντας <a href="{{URL::to('fire/insert')}}" class="text-underline orange">  εδώ.</a></h4>
            </div>
        </div>

        <div class="row topmargin">
            <div class="col-xs-12">
                <h3 class="orange">Πυροσβεστικά Μέσα</h3>
                <h4 class="topmargin">Έχετε εισάγει: <span class="orange">{{count($forces)}}</span> @if (count($forces) == 1) πυροσβεστικό μέσο. @else πυροσβεστικά μέσα. @endif
                    <button type="button" id="forces_popover" class="btn btn-lg btn-danger btn-popover" data-toggle="popover" title="Πυροσβεστικά Μέσα" data-content="@foreach ($forces as $force) {{$force->name}} <br> @endforeach"><i class="fa fa-eye"></i></button>
                </h4>

            </div>
        </div>

        <div class="row topmargin">
            <div class="col-xs-12">
                <h3 class="orange">Προσωπικό</h3>
                <h4 class="topmargin">Έχετε εισάγει: <span class="orange">{{count($staff)}}</span> @if (count($staff) == 1) κατηγορία προσωπικού. @else κατηγορίες προσωπικού. @endif
                    <button type="button" id="staff_popover" class="btn btn-lg btn-danger btn-popover" data-toggle="popover" title="Προσωπικό" data-content="@foreach ($staff as $staff_one) {{$staff_one->name}} <br> @endforeach"><i class="fa fa-eye"></i></button>
                </h4>
            </div>
        </div>

    @stop

    @section('extraScript')
        <script type="text/javascript">
            $(function() {
```

```
                $('[data-toggle="popover"]').popover({html: true});
            });

            $('#forces_popover').on('shown.bs.popover', function () {
              $(this).find('i').removeClass('fa-eye').addClass('fa-eye-slash');
            });

            $('#forces_popover').on('hidden.bs.popover', function () {
              $(this).find('i').addClass('fa-eye').removeClass('fa-eye-slash');
            });

            $('#staff_popover').on('shown.bs.popover', function () {
              $(this).find('i').removeClass('fa-eye').addClass('fa-eye-slash');
            });

            $('#staff_popover').on('hidden.bs.popover', function () {
              $(this).find('i').addClass('fa-eye').removeClass('fa-eye-slash');
            });
        </script>
    @stop
```

Συνάρτηση dashboardCreate του AdminController για την προβολή της dashboard.blade.php:

```
    public function dashboardCreate()
        {
            $user       = User::find(Auth::user()->id);
            $last_fire  = Fire::orderBy('created_at')->first();
            $count_fires = Fire::count();
            $forces     = Force::all();
            $staff      = Staff::all();

            $data = array (  'title'=> 'Dashboard', 'user' => $user , 'last_fire' => $last_fire, 'count_fires' => $count_fires, 'forces' => $forces, 'staff' => $staff);
            $this->layout->content = View::make('admin.dashboard')->with($data);
        }
```

Route dashboard για την εκτέλεση της dashboardCreate:

```
    Route::get('/dashboard', array('uses' => 'AdminController@dashboardCreate'));
```

View profile.blade.php (Σελίδα επεξεργασίας Προφίλ του διαχειριστή):

```
    @section('title')
       {{$title}}
    @stop

    @section('content')
        <div class="row">
```

```
                <div class="col-xs-6">
                        <h3>Τα στοιχεία σας:</h3>
                </div>
                <div class="col-xs-6"></div>
        </div>
        <div class="row">
                <div class="col-xs-6">

                        @if ($errors->has('name'))
                                <div class="alert alert-danger alert-dismissible" role="alert">
                                 <button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
                                 <strong>Σφάλμα!</strong> Το πεδίο όνομα είναι υποχρεωτικό
                                </div>
                        @endif

                        @if ($errors->has('last_name'))
                                <div class="alert alert-danger alert-dismissible" role="alert">
                                 <button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
                                 <strong>Σφάλμα!</strong> Το πεδίο επώνυμο είναι υποχρεωτικό
                                </div>
                        @endif

                        @if ($errors->has('email'))
                                <div class="alert alert-danger alert-dismissible" role="alert">
                                 <button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
                                 <strong>Σφάλμα!</strong> Το πεδίο email είναι υποχρεωτικό
                                </div>
                        @endif

                    @if ($errors->has('current_password'))
                                <div class="alert alert-danger alert-dismissible" role="alert">
                                 <button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
                                 <strong>Σφάλμα!</strong> Έχετε εισάγει λανθασμένο κωδικό
                                </div>
                        @endif

                        @if ($errors->has('new_password'))
                                <div class="alert alert-danger alert-dismissible" role="alert">
                                 <button type="button" class="close" data-dismiss="alert" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
                                 <strong>Σφάλμα!</strong> Το πεδίο νέο κωδικός είναι
υποχρεωτικό
                                </div>
```

```
                    @endif

                    @if ($errors->has('confirm_password'))
                        <div class="alert alert-danger alert-dismissible" role="alert">
                          <button type="button" class="close" data-dismiss="alert" aria-label="Close"><span aria-hidden="true">&times;</span></button>
                          <strong>Σφάλμα!</strong>  Έχετε εισάγει λανθασμένο νέο κωδικό
                        </div>
                    @endif

                {{ Form::open(array('url' => 'profile/edit',  'method' => 'post')) }}
                  <div class="form-group  @if ($errors->has('name')) has-error @endif">
                    <label for="name">Όνομα</label>
                    {{Form::text('name',  Input::old('name',  $user->name),  array('id'  => 'name', 'class' => 'form-control'))}}
                  </div>
                  <div  class="form-group  @if  ($errors->has('last_name'))  has-error @endif">
                    <label for="last_name"> Επώνυμο</label>
                    {{Form::text('last_name',  Input::old('last_name',  $user->last_name), array('id' => 'last_name', 'class' => 'form-control'))}}
                  </div>
                  <div class="form-group @if ($errors->has('email')) has-error @endif">
                    <label for="email"> Email</label>
                    {{Form::text('email', Input::old('email', $user->email), array('id' => 'email', 'class' => 'form-control'))}}
                  </div>
                  <div class="form-group @if ($errors->has('current_password')) has-error @endif">
                    <label for="current_password">Παλιός Κωδικός:</label>
                    {{Form::password('current_password', array('id' => 'current_password', 'class' => 'form-control','placeholder' => '...'))}}
                  </div>
                  <div class="form-group @if ($errors->has('new_password')) has-error @endif">
                    <label for="new_password">Νέος Κωδικός:</label>
                    {{Form::password('new_password', array('id' => 'new_password', 'class' => 'form-control','placeholder' => '...'))}}
                  </div>
                  <div class="form-group @if ($errors->has('confirm_password')) has-error @endif">
                    <label for="confirm_password">Επιβεβαίωση νέου κωδικού:</label>
                    {{Form::password('confirm_password', array('id' => 'confirm_password', 'class' => 'form-control','placeholder' => '...'))}}
                  </div>

                  <button type="submit" class="btn btn-primary">Επιβεβαίωση</button>
```

```
                    {{ Form::close() }}

                </div>
                <div class="col-xs-6"></div>
        </div>
    @stop


    @section('extraModal')
    <div class="modal fade" id="profile_changed">
      <div class="modal-dialog">
        <div class="modal-content">
         <div class="modal-header noborder">
           <h4 class="modal-title text-center text-success"><big><span class="glyphicon glyphicon-
ok"></span></big></h4>
         </div>
         <div class="modal-body text-center">
           <p>Τα στοιχεία σας αποθηκεύτηκαν με επιτυχία</p>
         </div>
         <div class="modal-footer noborder text-center">
           <button type="button" class="btn btn-primary" data-dismiss="modal">Κλείσιμο</button>
         </div>
        </div>
      </div>
    </div>
    @stop

    @section('extraScript')
        <script>
                @if (Session::has('profile') && (Session::get('profile') == "changed"))
                        $("#profile_changed").modal();
                @endif
        </script>
    @stop
```

Συνάρτηση profileCreate του AdminController για την προβολή της profile.blade.php:

```
    public function profileCreate()
        {
                $user = User::find(Auth::user()->id);
                $data = array (   'title'=> 'Προφίλ', 'user' => $user );
                $this->layout->content = View::make('admin.profile')->with($data);
        }
```

Route profile για την εκτέλεση της profileCreate:

```
    Route::get('/profile', array('as' => 'profile', 'uses' => 'AdminController@profileCreate'));
```

Συνάρτηση profileStore του AdminController για την υποβολή της φόρμας επεξεργασίας του προφίλ:

```
public function profileStore()
	{
		$rules = array(
			'name'                                  => array('required'),
			'last_name'                             => array('required'),
			'email'                                 => array('required', 'email'),
		'current_password'              => array('passcheck'),
			'confirm_password'                      => array('same:new_password')
		);

		Validator::extend('passcheck', function($attribute, $value, $parameters) {
		   return Hash::check($value, Auth::user()->password);
		});

		$messages = array(
		   'passcheck' => trans('validation.in'),
		);

		$validation = Validator::make(Input::all(), $rules);

		if ($validation->fails())
	{
		return Redirect::to('profile')->withErrors($validation)->withInput();
	}
	else
	{
		$user = User::findOrFail(Auth::user()->id);
		$user->email		= Input::get('email');
		$user->name		= Input::get('name');
			$user->last_name	= Input::get('last_name');

			if (Input::get('new_password') != '')
				$user->password		=
Hash::make(Input::get('new_password'));
			$user->save();

			return Redirect::to('profile')->with(array('profile' => 'changed'));
	}

	}
```

Route profile/edit για την εκτέλεση της profileStore:

```
Route::post('/profile/edit', array('as' => 'profile', 'uses' => 'AdminController@profileStore'));
```

View fires/list.blade.php για την λίστα των Πυρκαγιών:

```
@section('title')
   {{$title}}
@stop

@section('extraStyle')
   {{HTML::style('css/datatables/dataTables.bootstrap.css')}}
   {{HTML::style('css/bootstrap-datetimepicker.min.css')}}
   {{HTML::style('css/bootstrap-select.min.css')}}
@stop

@section('content')

      <div class="row">
            <div class="col-xs-12">
                  <h1>Πυρκαγιές</h1>
                  <p class="lead">Λίστα πυρκαγιών</p>
            </div>
      </div>


      <div class="hide">
   {{ Form::open(array('url' => 'file/upload/', 'files' => true, 'id' => 'upload_csv')) }}
      {{ Form::hidden('name', 'csv') }}
      {{ Form::file('document', array('id' => 'document_csv')) }}
   {{ Form::close() }}
      </div>

      <div class="row" id="upload">
            <div class="col-xs-12 col-md-6">
                  <div class="panel panel-default">
                        <div class="panel-body">
                        <p>
                              {{Form::hidden('csv', '', array('id' => 'csv'))}}
                              <button class="btn btn-info" id="save_file"><i
class="fa fa-download"></i> Αποθήκευση αρχείου </button><span id="response"></span>
                        </p>

                        <div id="show_csv">

                  </div>
                  </div>
            </div>
      </div>
```

```
<div class="row" id="file-container">
        <div class="col-xs-12 col-md-6">
          <div class="panel panel-default">
                <div class="panel-body">
                  <div class="row row-margin-top progress_csv" style="display:none;">
                    <div class="col-xs-12">
                      <div id="progress" class="progress">
                        <div class="progress-bar progress-bar-success" style="width: 0%;"></div>
                      </div>
                    </div>
                  </div>

                  <div class="row row-margin-top">
                    <div class="col-xs-12 col-md-6">
                      <button id="choose_file_csv" class="btn btn-primary "><i class="fa fa-upload"></i> Εισαγωγή από αρχείο CSV </button>
                    </div>
                  </div>
              </div>
                </div>
        </div>
</div>

<div class="row">
        <div class="col-xs-12">
                <button class="btn btn-info pull-right" id="export_csv"><span><i class="fa fa-download"></i></span> Εξαγωγή σε μορφή CSV</button>
        </div>
</div>
{{Form::close()}}

<div class="row topmargin">
        <div class="col-xs-12">

                <table class="table table-striped table-hover" id="fires">
                        <thead>
                                <tr>
                                        <th>Υπηρεσία</th>
                                        <th>Νομός</th>
                                        <th>Δήμος</th>
                                        <th>Ημερ/νία Έναρξης</th>
                                        <th>Ημερ/νία Κατάσβεσης</th>
                                        <th>Διεύθυνση</th>
                                        <th>Ενέργειες</th>
```

```
                                    </tr>
                                </thead>
                                <tbody>

                                </tbody>
                            </table>

                        </div>
                    </div>


    @stop

    @section('extraModal')


        <div class="modal fade" id="delete_fire_modal">
          <div class="modal-dialog">
            <div class="modal-content">
             <div class="modal-header noborder">
               <h4 class="modal-title text-center text-warning"><big><span class="glyphicon glyphicon-warning-sign"></span></big></h4>
             </div>
             {{ Form::open(array('url' => 'fire/delete',  'method' => 'post')) }}
             <div class="modal-body text-center">
                  <p>Είστε σίγουροι ότι θέλετε να διαγράψετε αυτή την πυρκαγιά;</p>
                  {{Form::hidden('id_fire', '', array('id' => 'id_fire'))}}
             </div>
             <div class="modal-footer noborder text-center">
               <button type="button" class="btn btn-danger" data-dismiss="modal"><i class="fa fa-times"></i> Ακύρωση</button>
                            <button type="submit" class="btn btn-success"><i class="fa fa-check"></i> Επιβεβαίωση</button>
             </div>
             {{Form::close()}}
            </div>
          </div>
        </div>

        <div class="modal fade" id="informations_modal">
        <div class="modal-dialog modal-lg">
          <div class="modal-content">
           <div class="modal-header noborder">
             <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden="true">&times;</span></button>
             <h4 class="modal-title text-center orange">Πληροφορίες πυρκαγιάς</h4>
```

```
          </div>
          <div class="modal-body text-center">
            <div class="row">
              <div class="col-xs-12">
                <div id="informations">

                </div>
              </div>
            </div>
          </div>
          <div class="modal-footer">
            <button type="button" class="btn btn-primary" data-dismiss="modal"> Κλείσιμο παραθύρου</button>
          </div>
        </div>
      </div>
    </div>

        <div class="modal fade" id="succes_modal">
          <div class="modal-dialog">
            <div class="modal-content">
              <div class="modal-header noborder">
                <h4 class="modal-title text-center text-success"><big><span class="glyphicon glyphicon-ok"></span></big></h4>
              </div>
              <div class="modal-body text-center">
                <p>Οι αλλαγές σας αποθηκεύτηκαν με επιτυχία</p>
              </div>
              <div class="modal-footer noborder text-center">
                <button type="button" class="btn btn-primary" data-dismiss="modal">Κλείσιμο</button>
              </div>
            </div>
          </div>
        </div>

        <div class="modal fade" id="succes_import_file_modal">
          <div class="modal-dialog">
            <div class="modal-content">
              <div class="modal-header noborder">
                <h4 class="modal-title text-center text-success"><big><span class="glyphicon glyphicon-ok"></span></big></h4>
              </div>
              <div class="modal-body text-center">
                <p>Οι αλλαγές σας αποθηκεύτηκαν με επιτυχία</p>
                <p id="count"></p>
```

```
            <p id="count_exists"></p>
           </div>
           <div class="modal-footer noborder text-center">
            <button type="button" class="btn btn-primary" data-dismiss="modal" id="close">Ανανέωση σελίδας</button>
           </div>
          </div>
         </div>
        </div>
    @stop

    @section('extraScript')
        {{HTML::script('js/datatables/jquery.dataTables.min.js')}}
        {{HTML::script('js/datatables/dataTables.bootstrap.min.js')}}
        {{HTML::script('js/moment.min.js')}}
        {{HTML::script('js/moment-gr.js')}}
        {{HTML::script('js/bootstrap-datetimepicker.min.js')}}
        {{HTML::script('js/bootstrap-select.min.js')}}
        {{HTML::script('js/jquery.form.min.js')}}
        <script type="text/javascript">
                $(function() {



                    $('#choose_file_csv').click(function(){
                $('#document_csv').click();
                return false;
                  });

                    $('body').delegate('#document_csv','change', function(){
                      $('#upload_csv').ajaxForm({ beforeSubmit: showRequest_csv, success: showResponse_csv, uploadProgress: showProgress_csv, dataType: 'json', resetForm: true}).submit();

                    });

                    function showRequest_csv(formData, jqForm, options) {
                      $('.progress_csv #progress .progress-bar').css('width', '0');
                      $('.progress_csv').fadeIn();
                      $(".progress_csv #validation-errors").hide().empty();
                      return true;
                    }

                    function showProgress_csv(event, position, total, percentComplete) {
                      var percentVal = percentComplete + '%';
                      $('.progress_csv .progress .progress-bar').css('width', percentVal);
```

```
                    }

                    function showResponse_csv(response, statusText, xhr, $form)  {
                     if(response.success == false) {
                        var arr = response.errors;
                        $.each(arr, function(index, value) {
                           if (value.length != 0) {
                              $(".progress_csv #validation-errors").append('<div class="alert
alert-error"><strong>'+ value +'</strong><div>');
                           }
                        });
                        $(".progress_csv #validation-errors").show();
                     } else {
                        $('.progress_csv #progress .progress-bar').css('width', '100%');
                        setTimeout(function(){ $('.progress_csv').fadeOut(); },200);

                        $('#csv').val(response.file);
                        $('#file-container').hide();

                        $("#upload").show();
                              $('#show_csv').show();
                        $('#show_csv').html("<a
href="'+response.destinationPath+response.file+'" class='nomargin' id='file_csv_name'> <i class='fa
fa-eye'></i></a> <a href='#'  id='delete_csv'><i class='fa fa-trash-o'></i>  </a>" +response.file+"");
                     }
                    }


                    $('body').on('click', '#save_file', function(){

                    csv = $(this).parent().find('#csv').val();

                          $.ajax({
                    type: "POST",
                    url: "{{URL::to('file/import')}}",
                    data: "&csv="+csv,
                    contentType: "application/x-www-form-urlencoded;charset=UTF-8",
                    beforeSend: function() {
                                          $('#response').html('{{      HTML::image("images/ajax-
loader.gif", "", array("class" => "")) }}');
                                          },
                    success: function(response){
                     if (response.success === true)
                     {
                          $("#succes_import_file_modal  #count").html("Οι  "+response.count+"  /
"+response.row+" εγγραφές μεταφέρθηκαν με επιτυχία");
```

```
                $("#succes_import_file_modal
#count_exists").html(response.count_fire_exists+" εγγραφές υπάρχουν ήδη");
                  $("#succes_import_file_modal").modal({keyboard:false, backdrop:'static'});
                  $('#response').html('');
                }
                }
            });
                return false;
              });



                $('body').on('click', '#delete_csv', function(){

                        csv = $('#csv').val();

                $.ajax({
            type: "POST",
            url: "{{URL::to('file/delete')}}",
            data: "&csv="+csv,
            contentType: "application/x-www-form-urlencoded;charset=UTF-8",
            success: function(response){
              if (response.success === true)
              {
                $("#upload").hide();
                                        $("#file-container").show();
              }
                }
            });
                return false;
                });


                $('body').on('click', '#close', function(){
                        window.location.href = "{{URL::to('fires/list')}}";
                });


                $('[data-toggle="tooltip"]').tooltip();

                var table = $('#fires').DataTable({
                        "bProcessing": true,
                        "bServerSide": true,
                        "sAjaxSource": "{{ URL::to('fires/list/datatables') }}",
                        "fnServerData": function ( sSource, aoData, fnCallback ) {
                                $.ajax( {
```

```
                                        "dataType": 'json',
                                        "type": "POST",
                                        "url": sSource,
                                        "data": aoData,
                                        "success": fnCallback
                                        } )
                                },

                                "language": {
                        "url": "{{URL::to('lang/greek.json')}}"
                        },
                        "lengthMenu": [ [10, 25, 50, -1], [10, 25, 50, "Όλες"] ],
                        "fnInitComplete": function(oSettings, json) {
                                        var datatable = this;
                                        var search_input = datatable.closest('.dataTables_wrapper').find('div[id$=_filter] input');
                                        search_input.attr('placeholder', (oSettings && oSettings.oLanguage && oSettings.oLanguage.sSearchPlaceholder) ? oSettings.oLanguage.sSearchPlaceholder : 'Αναζήτηση');
                                        var length_sel = datatable.closest('.dataTables_wrapper').find('div[id$=_length] select');
                                        length_sel.selectpicker().selectpicker('setStyle', 'btn-sm', 'add');
                                },
                        });



                $('body').on('click', '.delete', function(){

                        id_fire = $(this).attr('data-id');

                        $('#delete_fire_modal #id_fire').val(id_fire);
                        $("#delete_fire_modal").modal();
                        return false;
                });


                 $('body').on('click', '.info', function(){

                        id_fire = $(this).attr('data-id');

                        $.ajax({
                type: "POST",
                url: "{{URL::to('get/fire/informations')}}",
                data: "&id_fire="+id_fire,
                contentType: "application/x-www-form-urlencoded;charset=UTF-8",
```

```
                    success: function(response){
                     if (response.success === true)
                     {
                            $("#informations").html(response.informations);
                       $("#informations_modal").modal();
                     }
                            }
                });
                            return false;
                    });

                    @if (Session::has('success') && Session::get('success'))
                            $("#succes_modal").modal();
                    @endif
                });
        </script>
@stop
```

Συνάρτηση firesListCreate του FireController για την προβολή της λίστας των Πυρκαγιών:

```
public function firesListCreate() {
    $fires = Fire::all();
    $data  = array('title' => 'Πυρκαγιές', 'fires' => $fires);
    $this->layout->content = View::make('admin.fires.list')->with($data);
  }
```

Route fires/list για την εκτέλεση της firesListCreate:

```
Route::get('fires/list', array('as' => 'fires_list', 'uses' => 'FireController@firesListCreate'));
```

Συνάρτηση datatable του DatatableController για την χρήση AJAX στον πίνακα της λίστας των Πυρκαγιών:

```
function datatable ()
                {
                            /*
                             * Script:    DataTables server-side script for PHP and MySQL
                             * Copyright: 2010 - Allan Jardine
                             * License:   GPL v2 or BSD (3-point)
                             */

                            /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                             * Easy set variables
                             */

                            /* Array of database columns which should be read and sent back to
DataTables. Use a space where
```

```
                        * you want to insert a non-database field (for example a counter or static
image)
                        */
                        $aColumns = array( 'id_fire_department', 'id_prefecture', 'id_municipality',
'start_date', 'end_date', 'adress' );

                        /* Indexed column (used for fast and accurate table cardinality) */
                        $sIndexColumn = "id";

                        /* DB table to use */

                        $sTable = "fires";

                        /* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                        * If you just want to use the basic configuration for DataTables with PHP
server-side, there is
                        * no need to edit below this line
                        */



                        /*
                        * Paging
                        */
                        $sLimit = "";
                        if ( isset( $_POST['iDisplayStart'] ) && $_POST['iDisplayLength'] != '-1' )
                        {
                                $sLimit = "LIMIT ".( $_POST['iDisplayStart'] ).", ".
                                        ( $_POST['iDisplayLength'] );
                        }


                        /*
                        * Ordering
                        */
                        if ( isset( $_POST['iSortCol_0'] ) )
                        {
                                $sOrder = "ORDER BY  ";
                                for ( $i=0 ; $i<intval( $_POST['iSortingCols'] ) ; $i++ )
                                {
                                        if ( $_POST[ 'bSortable_'.intval($_POST['iSortCol_'.$i]) ]
== "true" )
                                        {
                                                $sOrder .= $aColumns[ intval(
$_POST['iSortCol_'.$i] ) ]."
                                                        ".( $_POST['sSortDir_'.$i] ) .", ";
                                        }
                                }
```

```
                $sOrder = substr_replace( $sOrder, "", -2 );
                if ( $sOrder == "ORDER BY" )
                {
                        $sOrder = "";
                }
        }


        /*
         * Filtering
         * NOTE this does not match the built-in DataTables filtering which does it
         * word by word on any field. It's possible to do here, but concerned about
efficiency
         * on very large tables, and MySQL's regex functionality is very limited
         */
        $sWhere = "";
        if ( $_POST['sSearch'] != "" )
        {
                $sWhere = "WHERE (";
                for ( $i=0 ; $i<count($aColumns) ; $i++ )
                {
                        $sWhere .= 'fires.'.$aColumns[$i]." LIKE '%".(
$_POST['sSearch'] )."%' OR ";
                }
                $sWhere = substr_replace( $sWhere, "", -3 );
                $sWhere .= ')';
        }

        /* Individual column filtering */
        for ( $i=0 ; $i<count($aColumns) ; $i++ )
        {
                if ( $_POST['bSearchable_'.$i] == "true" && $_POST['sSearch_'.$i]
!= '' )
                {
                        if ( $sWhere == "" )
                        {
                                $sWhere = "WHERE ";
                        }
                        else
                        {
                                $sWhere .= " AND ";
                        }
                        $sWhere .= 'fires.'.$aColumns[$i]." LIKE
'%".($_POST['sSearch_'.$i])."%' ";
                }
```

```
                }


                /*
                 * SQL queries
                 * Get data to display
                 */

                $sQuery = "
                        SELECT SQL_CALC_FOUND_ROWS fires.id_fire_department,
fires.id_prefecture, fires.id_municipality, fires.start_date, fires.end_date, fires.adress, fires.id as
fire_id, fire_departments.name as fire_department_name, prefectures.name as prefecture_name,
municipalities.name as municipality_name
                        FROM fires
                        LEFT JOIN fire_departments ON fire_departments.id =
fires.id_fire_department
                        LEFT JOIN prefectures ON prefectures.id = fires.id_prefecture
                        LEFT JOIN municipalities ON municipalities.id =
fires.id_municipality
                        $sWhere
                        $sOrder
                        $sLimit
                ";


        $rResults = DB::select($sQuery, array());


                /* Data set length after filtering */
                $sQuery = "
                        SELECT FOUND_ROWS() as total
                ";

                $aResultFilterTotal = DB::select($sQuery, array());
                $iFilteredTotal = $aResultFilterTotal[0]->total;

                /* Total data set length */
                $sQuery = "
                        SELECT COUNT(".$sIndexColumn.") as total
                        FROM   $sTable
                ";

                $aResultTotal = DB::select($sQuery, array());
                $iTotal = $aResultTotal[0]->total;
```

```
                        /*
                         * Output
                         */
                        $output = array(
                                "sEcho" => intval($_POST['sEcho']),
                                "iTotalRecords" => $iTotal,
                                "iTotalDisplayRecords" => $iFilteredTotal,
                                "aaData" => array(),
                        );

                        foreach ( $rResults as $rResult )
                        {
                                $row = array();
                                for ( $i=0 ; $i<count($aColumns) ; $i++ )
                                {
                                        if ( $aColumns[$i] == "id_fire_department" )
                                        {
                                                $row[] = $rResult->fire_department_name;
                                        }
                                        else if ( $aColumns[$i] == "id_prefecture" )
                                        {
                                                $row[] = $rResult->prefecture_name;
                                        }
                                        else if ( $aColumns[$i] == "id_municipality" )
                                        {
                                                $row[] = $rResult->municipality_name;
                                        }
                                        else if ( $aColumns[$i] != ' ' )
                                        {
                                                /* General output */
                                                $row[] = $rResult->{$aColumns[$i]};
                                        }
                                }
                                $row[] = '
                                        <a              href="'.URL::to('fire/edit/'.$rResult->fire_id).'"
class="modify" data-toggle="tooltip" data-placement="top" title="Επεξεργασία"><i class="fa fa-
pencil-square-o"></i></a>
                                        <a href="#" class="delete" data-id="'.$rResult->fire_id.'"
data-toggle="tooltip" data-placement="top" title="Διαγραφή"><i class="fa fa-trash-o"></i></a>
                                        <a href="#" class="info" data-id="'.$rResult->fire_id.'" data-
toggle="tooltip" data-placement="top" title="Πληροφορίες"><i class="fa fa-info-circle"></i></a>
                                        ';
                                $output['aaData'][] = $row;
                        }
```

```
                echo json_encode( $output );
            }
```

Route fires/list/datatables για την εκτέλεση της συνάρτησης datatable:

```
Route::post('fires/list/datatables', array('uses' => 'DataTablesController@datatable'));
```

Συνάρτηση fireDeleteStore του FireController για την διαγραφή Πυρκαγιάς:

```
public function fireDeleteStore()
 {
   $rules = array(
      'id_fire' => array('required')
   );

   $validation = Validator::make(Input::all(), $rules);

   if ($validation->fails())
   {
      return Redirect::to('fires/lst/')->withErrors($validation)->withInput();
   }
   else
   {

     $fire = Fire::find(Input::get('id_fire'));

     $burtn_area = $fire->burnt_area()->first();
     $burtn_area->forceDelete();

     $fire_forces = $fire->forces()->get();
     $fire_staff  = $fire->staff()->get();

     foreach ($fire_forces as $force) {
       $fire->forces()->detach($force->id);
     }

     foreach ($fire_staff as $staff) {
       $fire->staff()->detach($staff->id);
     }

     $fire->forceDelete();

      return Redirect::to('fires/list/')->with(array('success'=> true));
   }
 }
```

Route fires/delete για την εκτέλεση της συνάρτησης fireDeleteStore:

```
Route::post('fire/delete', array('uses' => 'FireController@fireDeleteStore'));
```

Συνάρτηση getFireInformations του FireController για την προβολή των πληροφοριών της Πυρκαγιάς:

```
public function getFireInformations()
  {
   $rules = array(
      'id_fire' => array('required')
   );

   $validation = Validator::make(Input::all(), $rules);

   if ($validation->fails())
   {
      return Response::json(array('success' => false));
   }
   else
   {

    $fire = Fire::find(Input::get('id_fire'));

    $fire_department = $fire->fire_department()->first();
    $prefecture      = $fire->prefecture()->first();
    $municipality    = $fire->municipality()->first();
    $burtn_area      = $fire->burnt_area()->first();
    $fire_forces     = $fire->forces()->get();
    $fire_staff      = $fire->staff()->get();


     $informations = "";

       $informations .= "<div class='row'>";
        $informations .= "<div class='col-xs-12'>";

          $start_date = $fire->transformDateFromDataBase($fire->start_date);
          if ($fire->end_date != "0000-00-00 00:00:00") {
            $end_date = $fire->transformDateFromDataBase($fire->end_date);
            $informations .= "<p class='text-right orange'>".$start_date." - ".$end_date."</p>";
          }
          else
            $informations .= "<p class='text-right orange'>".$start_date."</p>";

        $informations .= "</div";
       $informations .= "</div";

       $informations .= "<div class='row'>";
        $informations .= "<div class='col-xs-12 nopadding'>";
```

```
$informations .= "<table class='table table-striped text-left'>";

  $informations .= "<tr>";
    $informations .="<td  class='orange'>Νομός";
    $informations .="</td>";
    $informations .="<td>".$prefecture->name."";
    $informations .="</td>";
  $informations .= "</tr>";

  $informations .= "<tr>";
    $informations .="<td class='orange'>Δήμος";
    $informations .="</td>";
    $informations .="<td>".$municipality->name."";
    $informations .="</td>";
  $informations .= "</tr>";

  $informations .= "<tr>";
    $informations .="<td class='orange'>Πυροσβεστική υπηρεσία";
    $informations .="</td>";
    $informations .="<td>".$fire_department->name."";
    $informations .="</td>";
  $informations .= "</tr>";

  $informations .= "<tr>";
    $informations .="<td class='orange'>Δασαρχείο";
    $informations .="</td>";
    $informations .="<td>".$fire->forestry."";
    $informations .="</td>";
  $informations .= "</tr>";

  $informations .= "<tr>";
    $informations .="<td class='orange'>Περιοχή";
    $informations .="</td>";
    $informations .="<td>".$fire->area."";
    $informations .="</td>";
  $informations .= "</tr>";

  $informations .= "<tr>";
    $informations .="<td class='orange'>Διεύθυνση";
    $informations .="</td>";
    $informations .="<td>".$fire->adress."";
    $informations .="</td>";
  $informations .= "</tr>";

 $informations .= "</table>";
$informations .= "</div";
```

```
$informations .= "</div";


$informations .= "<div class='row'>";
  $informations .= "<div class='col-xs-12'>";
    $informations .= "<h3>Καμμένη περιοχή</h3>";
    $informations .= '<ul class="list-unstyled list-inline text-center">';
      $informations .= '<li>';
        $informations .= '<div class="form-group">';
        $informations .= '<label for="forest">Δάση</label>';
         $informations .= '<input type="text" disabled class="form-control" value="'.$burtn_area->forest.'"';
        $informations .= '</div>';
      $informations .= '</li>';
      $informations .= '<li>';
        $informations .= '<div class="form-group">';
        $informations .= '<label for="woodland">Δασική Έκταση</label>';
         $informations .= '<input type="text" disabled class="form-control" value="'.$burtn_area->woodland.'"';
        $informations .= '</div>';
      $informations .= '</li>';
      $informations .= '<li>';
        $informations .= '<div class="form-group">';
        $informations .= '<label for="grove">Άλση</label>';
         $informations .= '<input type="text" disabled class="form-control" value="'.$burtn_area->grove.'"';
        $informations .= '</div>';
      $informations .= '</li>';
      $informations .= '<li>';
        $informations .= '<div class="form-group">';
        $informations .= '<label for="grass_land">Χορτ/κές Εκτάσεις</label>';
         $informations .= '<input type="text" disabled class="form-control" value="'.$burtn_area->grass_land.'"';
        $informations .= '</div>';
      $informations .= '</li>';
      $informations .= '<li>';
        $informations .= '<div class="form-group">';
        $informations .= '<label for="reed_marches">Καλάμια - Βάλτοι</label>';
         $informations .= '<input type="text" disabled class="form-control" value="'.$burtn_area->reed_marches.'"';
        $informations .= '</div>';
      $informations .= '</li>';
      $informations .= '<li>';
        $informations .= '<div class="form-group">';
        $informations .= '<label for="farmland">Γεωργικές Εκτάσεις</label>';
         $informations .= '<input type="text" disabled class="form-control" value="'.$burtn_area->farmland.'"';
```

```
                $informations .= '</div>';
              $informations .= '</li>';
              $informations .= '<li>';
                $informations .= '<div class="form-group">';
                $informations .= '<label for="cultivation_waste">Υπολλείματα Καλλιεργειών</label>';
                 $informations .= '<input type="text" disabled class="form-control" value="'.$burtn_area->cultivation_waste."";
                $informations .= '</div>';
              $informations .= '</li>';
              $informations .= '<li>';
                $informations .= '<div class="form-group">';
                $informations .= '<label for="dumping_ground">Σκουπιδότοποι</label>';
                 $informations .= '<input type="text" disabled class="form-control" value="'.$burtn_area->dumping_ground."";
                $informations .= '</div>';
              $informations .= '</li>';
            $informations .= '</ul>';
          $informations .= '</div>';
        $informations .= '</div>';


        $informations .= "<div class='row'>";
          $informations .= "<div class='col-xs-12'>";
            $informations .= "<h3>Προσωπικό</h3>";
            $informations .= '<ul class="list-unstyled list-inline text-center">';

              foreach ($fire_staff as $staff) {
                  $informations .= '<li>';
                    $informations .= '<div class="form-group">';
                    $informations .= '<label for="">'.$staff->name.'</label>';
                     $informations .= '<input type="text" disabled class="form-control" value="'.$staff->pivot->number."";
                    $informations .= '</div>';
                 $informations .= '</li>';
              }

            $informations .= '</ul>';
          $informations .= '</div>';
        $informations .= '</div>';


        $informations .= "<div class='row'>";
          $informations .= "<div class='col-xs-12'>";
            $informations .= "<h3>Μέσα πυρόσβεσης</h3>";
            $informations .= '<ul class="list-unstyled list-inline text-center">';
```

```
            foreach ($fire_forces as $force) {
                $informations .= '<li>';
                   $informations .= '<div class="form-group">';
                   $informations .= '<label for="">'.$force->name.'</label>';
                    $informations .= '<input type="text" disabled class="form-control" value="'.$force->pivot->number.'"';
                   $informations .= '</div>';
                 $informations .= '</li>';
            }

           $informations .= '</ul>';
         $informations .= '</div>';
       $informations .= '</div>';


    return Response::json(array(  'success' => true, 'informations' => $informations ));

  }
 }
```

Route get/fire/informations για την εκτέλεση της συνάρτησης getFireInformations:

```
Route::post('get/fire/informations', array('uses' => 'FireController@getFireInformations'));
```

Συνάρτηση firesExport του FireController για την εξαγωγή σε CSV των Πυρκαγιών:

```
public function firesExport()
 {

   $fires_id = Fire::all();

    $output = iconv("utf-8", "iso-8859-7", "Υπηρεσία;Νομός;Ημερ/νία Έναρξης;Ώρα Έναρξης;Ημερ/νία Κατάσβεσης;Ώρα Κατάσβεσης;Δασαρχείο;Δήμος;Περιοχή;Διεύθυνση;Δάση;Δασική Έκταση;Αλση;Χορτ/κές Εκτάσεις;Καλάμια - Βάλτοι;Γεωργικές Εκτάσεις;Υπολλείματα Καλλιεργειών;Σκουπιδότοποι;ΠΥΡΟΣ. ΣΩΜΑ;ΠΕΖΟΠΟΡΑ ΤΜΗΜΑΤΑ;ΕΘΕΛΟΝΤΕΣ;ΣΤΡΑΤΟΣ;ΑΛΛΕΣ ΔΥΝΑΜΕΙΣ;ΠΥΡΟΣ. ΟΧΗΜ.;ΟΧΗΜ. ΟΤΑ;ΒΥΤΙΟΦΟΡΑ;ΜΗΧΑΝΗΜΑΤΑ;ΕΛΙΚΟΠΤΕΡΑ;Α/Φ CL415;Α/Φ CL215;Α/Φ PZL;Α/Φ GRU.")."\n\n";

    if (!empty($fires_id)) {

      foreach ($fires_id as $fire)
      {
        ini_set('max_execution_time', 180);

        $output .= iconv("utf-8", "iso-8859-7", $fire->fire_department()->first()->name .';');
        $output .= iconv("utf-8", "iso-8859-7", $fire->prefecture()->first()->name .';');
```

```
$start_date = explode(" ",$fire->transformDateFromDataBase($fire->start_date));
$output .= utf8_decode($start_date[0]  .';');
$output .= utf8_decode($start_date[1]  .';');

$end_date = $fire->end_date;

if ($end_date != "0000-00-00 00:00:00") {
  $end_date = explode(" ",$fire->transformDateFromDataBase($fire->end_date));
  $output .= utf8_decode($end_date[0]  .';');
  $output .= utf8_decode($end_date[1]  .';');
}
else {
  $output .= utf8_decode(" ".';');
}

$output .= iconv("utf-8", "iso-8859-7", $fire->forestry .';');
$output .= iconv("utf-8", "iso-8859-7", $fire->municipality()->first()->name .';');
$output .= iconv("utf-8", "iso-8859-7", $fire->area .';');
$output .= iconv("utf-8", "iso-8859-7", $fire->adress .';');



$burtn_area  = $fire->burnt_area()->first();

$output .= utf8_decode($burtn_area->forest  .';');
$output .= utf8_decode($burtn_area->woodland  .';');
$output .= utf8_decode($burtn_area->grove  .';');
$output .= utf8_decode($burtn_area->grass_land  .';');
$output .= utf8_decode($burtn_area->reed_marches  .';');
$output .= utf8_decode($burtn_area->farmland  .';');
$output .= utf8_decode($burtn_area->cultivation_waste  .';');
$output .= utf8_decode($burtn_area->dumping_ground  .';');

$fire_staff  = $fire->staff()->get();
foreach ($fire_staff as $staff) {
   $output .= utf8_decode( $staff->pivot->number .';');
}

$fire_forces = $fire->forces()->get();
foreach ($fire_forces as $force) {
   $output .= utf8_decode( $force->pivot->number .';');
}


$output .= ";";
$output .= "\n";
```

```
        }
      }

      $headers = array(
        'Content-Type' => 'application/vnd.ms-excel; charset=UTF-8',
        'Content-Disposition' => 'attachment; filename="fires.csv"',
      );

      return Response::make(rtrim($output, "\n"), 200, $headers);
    }
```

Route fires/export για την εκτέλεση της συνάρτησης firesExport:

```
Route::post('fires/export', array('uses' => 'FireController@firesExport'));
```

Συνάρτηση fileUpload του FireController για την εισαγωγή αρχείου CSV:

```
public function fileUpload() {
      $file  = Input::file('document');
      $input = array('document' => $file);

      $destinationPath = public_path().'/uploads/';
      $filename = Str::slug($file->getClientOriginalName()).".".$file->getClientOriginalExtension();

      Input::file('document')->move($destinationPath, $filename);

      $destination = '../uploads/';

      return Response::json(['success' => true, 'file' => $filename, 'time' => time(), 'destinationPath' => $destination]);
    }
```

Route file/upload για την εκτέλεση της συνάρτησης fileUpload:

```
Route::post('file/upload', array('uses' => 'FireController@fileUpload'));
```

Συνάρτηση fileDelete του FireController για την διαγραφή αρχείου CSV:

```
public function fileDelete() {

    $filename = public_path().'/uploads/'.Input::get('csv');

    if (File::exists($filename)) {
      File::delete($filename);
    }

    return Response::json(['success' => true]);
  }
```

Route file/delete για την εκτέλεση της συνάρτησης fileDelete:

```
Route::post('file/delete', array('uses' => 'FireController@fileDelete'));
```

Συνάρτηση fileImport του FireController για την αποθήκευση των τιμών του αρχείου CSV:

```
public function fileImport() {

   $rules = array(
      'csv' => array('required')
   );

   $validation = Validator::make(Input::all(), $rules);

   if ($validation->fails())
   {
      return Response::json(array('success' => false));
   }
   else
   {

     $old_fires = Fire::all();

     $row   = 0;
     $count = 0;
     $count_fire_exists = 0;
     $fire_exists = false;
     if (($handle = fopen(public_path().'/uploads/'.Input::get('csv'), "r")) !== FALSE) {

        while (($datas = fgets($handle, 4096)) !== FALSE) {

          $data = explode(';', $datas);


          $fire_exists = false;

          $row++;

          $fire_department          = trim($data[0]);
          $prefecture               = trim($data[1]);
          $start_date               = trim($data[2]);
          $start_time               = trim($data[3]);
          $end_date                 = trim($data[4]);
          $end_time                 = trim($data[5]);
          $forestry                 = trim($data[6]);
          $municipality             = trim($data[7]);
          $area                     = trim($data[8]);
          $adress                   = trim($data[9]);
```

```
            $forest                 = trim($data[10]);
            $woodland               = trim($data[11]);
            $grove                  = trim($data[12]);
            $grass_land             = trim($data[13]);
            $reed_marches           = trim($data[14]);
            $farmland               = trim($data[15]);
            $cultivation_waste      = trim($data[16]);
            $dumping_ground         = trim($data[17]);

            $firemen                = trim($data[18]);
            $trekking_paths         = trim($data[19]);
            $volunteers             = trim($data[20]);
            $army                   = trim($data[21]);
            $other_forces           = trim($data[22]);

            $fire_fightiting_vehicles = trim($data[23]);
            $trucks                 = trim($data[24]);
            $tank_truck             = trim($data[25]);
            $machines               = trim($data[26]);
            $helicopters            = trim($data[27]);
            $canadair_cl415         = trim($data[28]);
            $canadair_cl215         = trim($data[29]);
            $canadair_PZL           = trim($data[30]);
            $canadair_GRU           = trim($data[31]);


            $fire_department = iconv('ISO-8859-7','UTF-8', $fire_department);
            $prefecture      = iconv('ISO-8859-7','UTF-8', $prefecture);
            $forestry        = iconv('ISO-8859-7','UTF-8', $forestry);
            $municipality    = iconv('ISO-8859-7','UTF-8', $municipality);
            $area            = iconv('ISO-8859-7','UTF-8', $area);
            $adress          = iconv('ISO-8859-7','UTF-8', $adress);


            $fire_department_exists = FireDepartment::where('name', 'like', $fire_department)-
>first();
            $prefecture_exists      = Prefecture::where('name', 'like', $prefecture)->first();
            $municipality_exists    = Municipality::where('name', 'like', $municipality)->first();

            if (count($fire_department_exists) > 0 && count($prefecture_exists) > 0 &&
count($municipality_exists) > 0) {

                $fire = new Fire();

                $date_start       = $start_date." ".$start_time;
```

```
                $fire->start_date = $fire->transformDateForDataBase($date_start);


                $fire_exists = Fire::where('start_date', '=', $fire->start_date)->where('id_municipality',
'=', $municipality_exists->id)->where('id_prefecture', '=', $prefecture_exists->id)->first();
                if (count($fire_exists) > 0)
                  $fire_exists = true;

                if ($fire_exists == false) {

                   if ($end_date != "") {
                     $date_end       = $end_date." ".$end_time;
                     $fire->end_date = $fire->transformDateForDataBase($date_end);
                   } else {
                     $date_end       = "";
                     $fire->end_date = $date_end;
                   }

                   $country = "GREECE";

                   $location = $this->getLatitudeAndLongitude($country, $prefecture, $municipality,
$adress, 4);


                   $fire->latitude     = $location[0];
                   $fire->longitude    = $location[1];
                   $fire->gmap_status  = $location[2];


                   $fire->forestry = $forestry;
                   $fire->area     = $area;
                   $fire->adress   = $adress;

                   $fire->id_fire_department = $fire_department_exists->id;
                   $fire->id_prefecture      = $prefecture_exists->id;
                   $fire->id_municipality    = $municipality_exists->id;


                   $fire->created_at = date("Y-m-d H:i:s");
                   $fire->updated_at = date("Y-m-d H:i:s");
                   $fire->save();

                   $burtn_area                = new BurntArea();
                   $burtn_area->id_fire       = $fire->id;
                   $burtn_area->forest        = $forest;
                   $burtn_area->woodland      = $woodland;
```

```
                $burtn_area->grove            = $grove;
                $burtn_area->grass_land       = $grass_land;
                $burtn_area->reed_marches     = $reed_marches;
                $burtn_area->farmland         = $farmland;
                $burtn_area->cultivation_waste = $cultivation_waste;
                $burtn_area->dumping_ground   = $dumping_ground;
                $burtn_area->save();


                $staff_array = array($firemen, $trekking_paths, $volunteers, $army, $other_forces);

                $i = 0;
                $staff_all = Staff::all();
                foreach ($staff_all as $staff) {
                  $fire_staff          = new FireStaff();
                  $fire_staff->id_fire  = $fire->id;
                  $fire_staff->id_staff = $staff->id;
                  $fire_staff->number   = $staff_array[$i];
                  $fire_staff->save();
                  $i++;
                }

                $forces_array = array($fire_fightiting_vehicles, $trucks, $tank_truck, $machines, $helicopters, $canadair_cl415, $canadair_cl215 , $canadair_PZL, $canadair_GRU );

                $i = 0;
                $forces = Force::all();
                foreach ($forces as $force) {
                  $fire_force          = new FireForce();
                  $fire_force->id_fire  = $fire->id;
                  $fire_force->id_force = $force->id;
                  $fire_force->number   = $forces_array[$i];
                  $fire_force->save();
                  $i++;
                }

                $count++;
              } else {
                $count_fire_exists++;
              }
            }
          }
          fclose($handle);
        }
```

```
        return Response::json(array(  'success' => true, 'count' => $count, 'row' => $row, 'count_fire_exists' => $count_fire_exists ));
      }
    }
```

Route file/import για την εκτέλεση της συνάρτησης fileImport:

```
Route::post('file/import', array('uses' => 'FireController@fileImport'));
```

Συνάρτηση getLatitudeAndLongitude του FireController για την εύρεση του γεωγραφικού σημείου της Πυρκαγιάς:

```
public function getLatitudeAndLongitude($country, $prefecture, $municipality, $adress, $detail) {
    $adress_output = "";

    if ($detail >=1 )
      $adress_output .= $country."+";
    if ($detail >=2)
      $adress_output .= $prefecture."+";
    if ($detail >=3)
      $adress_output .= $municipality."+";
    if ($detail >=4)
      $adress_output .= $adress."+";

      $adress_output = urlencode($adress_output);

    $json = file_get_contents("https://maps.google.com/maps/api/geocode/json?address=$adress_output&key=AIzaSyCksCbtBgHhhaGr5-YhbkWGcDEetL8VJXc&sensor=false");
    $json = json_decode($json);

    if ($json->{'status'} == "ZERO_RESULTS")
    {
      return $this->getLatitudeAndLongitude($country, $prefecture, $municipality, $adress, ($detail-1));
    } else if ($json->{'status'} == "OK") {

      $lat = $json->{'results'}[0]->{'geometry'}->{'location'}->{'lat'};
      $long = $json->{'results'}[0]->{'geometry'}->{'location'}->{'lng'};

      return array($lat, $long, $json->{'status'});
    }
    else
    {

      $lat = '39.074208';
      $long = '21.824312';
      return array($lat, $long, $json->{'status'});
```

```
    }
  }
```

View fires/insert.blade.php για την εισαγωγή και επεξεργασία μίας Πυρκαγιάς:

```
@section('title')
    {{$title}}
@stop

@section('extraStyle')
    {{HTML::style('css/bootstrap-select.min.css')}}
    {{HTML::style('css/bootstrap-datetimepicker.min.css')}}
    <style>
        body {
                padding-bottom: 400px;
        }
    </style>
@stop

@section('content')
        <div class="row">
                <div class="col-xs-12">
                        <h1>Πυρκαγιές</h1>
                        @if (isset($section) && $section == 'edit')
                                <p class="lead">Επεξεργασία πυρκαγιάς</p>
                        @else
                                <p class="lead">Εισαγωγή πυρκαγιάς</p>
                        @endif
                </div>
        </div>

        <div class="row topmargin">

                <div class="col-xs-12 col-md-6">
                @if (isset($section) && $section == 'edit')
                        <p class="lead">Τοποθεσία πυρκαγιάς</p>
                @endif

                @if (isset($section) && $section == 'edit')
                        {{ Form::open(array('url' => 'fire/edit/'.$id,   'method' => 'post', 'id' =>
'submit_form')) }}
                @else
                        {{ Form::open(array('url' => 'fire/insert',    'method' => 'post', 'id' =>
'submit_form')) }}
                @endif
                        <div class="form-group @if ( $errors->has('id_region')) error @endif">
                                <label for="id_region"> Περιφέρεια</label>
```

```
                    <select class="form-control selectpicker" title='Επιλέξτε περιφέρεια' id="id_region" name="id_region">
                        <option class="hide" value="0"></option>
                        @foreach ($regions as $region)
                            <option value="{{$region->id}}" @if (isset($section) && $section == 'edit') @if ( ($region->id) == $region_id) {{'selected'}} @endif @endif>{{$region->name}}</option>
                        @endforeach
                    </select>
                </div>

                <div class="form-group @if ( $errors->has('id_fire_department')) error @endif">
                    <label for="id_fire_department"> Πυροσβεστική υπηρεσία</label>
                    <select class="form-control selectpicker" title='Επιλέξτε υπηρεσία' id="id_fire_department" name="id_fire_department" data-live-search="true" disabled>
                        @if (isset($section) && $section == 'edit')
                            <option class="hide" value="0"></option>
                            @foreach ($fire_departments as $fire_department)
                                <option value="{{$fire_department->id}}" @if ( ($fire->fire_department()->first()->id) == $fire_department->id) {{'selected'}} @endif >{{$fire_department->name}}</option>
                            @endforeach
                        @endif
                    </select>
                </div>



                <div class="form-group @if ( $errors->has('id_prefecture')) error @endif">
                    <label for="id_prefecture"> Νομός</label>
                    <select class="form-control selectpicker" title='Επιλέξτε νομό' id="id_prefecture" name="id_prefecture" data-live-search="true" disabled>
                        @if (isset($section) && $section == 'edit')
                            <option class="hide" value="0"></option>
                            @foreach ($prefectures as $prefecture)
                                <option value="{{$prefecture->id}}" @if ( ($fire->prefecture()->first()->id) == $prefecture->id) {{'selected'}} @endif >{{$prefecture->name}}</option>
                            @endforeach
                        @endif
                    </select>
                </div>

                <div class="form-group @if ( $errors->has('id_municipality')) error @endif">
                    <label for="id_municipality"> Δήμος</label>
                    <select class="form-control selectpicker" title='Επιλέξτε δήμο' id="id_municipality" name="id_municipality" data-live-search="true" disabled>
```

```
                                    @if (isset($section) && $section == 'edit')
                                        <option class="hide" value="0"></option>
                                        @foreach ($municipalities as $municipality)
                                            <option value="{{$municipality->id}}" @if (
($fire->municipality()->first()->id) == $municipality->id) {{'selected'}} @endif >{{$municipality->
name}}</option>
                                        @endforeach
                                    @endif
                                </select>
                            </div>

                            <div class="form-group @if ( $errors->has('started_at')) has-error @endif">
                                <label for="started_at"> Ημερ/νία Έναρξης</label>
                                 <div class='input-group date' id='started_at'>
                                    @if (isset($section) && $section == 'edit')
                                        {{Form::text('started_at',   Input::old('started_at',
$start_date), array('id' => 'started_at', 'class' => 'form-control'))}}
                                    @else
                                        {{Form::text('started_at',  Input::old('started_at'),
array('id' => 'started_at', 'class' => 'form-control'))}}
                                    @endif
                                <span class="input-group-addon">
                                    <span class="glyphicon glyphicon-calendar"></span>
                                </span>
                            </div>
                            </div>

                            <div class="form-group @if ( $errors->has('ended_at')) has-error @endif">
                                <label for="ended_at"> Ημερ/νία Κατάσβεσης</label>
                                 <div class='input-group date' id='ended_at'>
                                    @if (isset($section) && $section == 'edit')
                                        {{Form::text('ended_at',     Input::old('ended_at',
$end_date), array('id' => 'ended_at', 'class' => 'form-control'))}}
                                    @else
                                        {{Form::text('ended_at',    Input::old('ended_at'),
array('id' => 'ended_at', 'class' => 'form-control'))}}
                                    @endif
                                <span class="input-group-addon">
                                    <span class="glyphicon glyphicon-calendar"></span>
                                </span>
                            </div>
                            </div>


                    <div class="form-group">
                                <label for="forestry"> Δασαρχείο</label>
                                @if (isset($section) && $section == 'edit')
```

```
                                        {{Form::text('forestry', Input::old('forestry', $fire->forestry), array('id' => 'forestry', 'class' => 'form-control'))}}
                                @else
                                        {{Form::text('forestry', Input::old('forestry'), array('id' => 'forestry', 'class' => 'form-control'))}}
                                @endif
                        </div>

                        <div class="form-group">
                                <label for="area"> Περιοχή</label>
                                @if (isset($section) && $section == 'edit')
                                        {{Form::text('area', Input::old('area', $fire->area), array('id' => 'area', 'class' => 'form-control'))}}
                                @else
                                        {{Form::text('area', Input::old('area'), array('id' => 'area', 'class' => 'form-control'))}}
                                @endif
                        </div>

                        <div class="form-group">
                                <label for="adress"> Διεύθυνση</label>
                                @if (isset($section) && $section == 'edit')
                                        {{Form::text('adress', Input::old('adress', $fire->adress), array('id' => 'adress', 'class' => 'form-control'))}}
                                @else
                                        {{Form::text('adress', Input::old('adress'), array('id' => 'adress', 'class' => 'form-control'))}}
                                @endif
                        </div>

                        <div class="form-group">
                                <label for="map-address"> Διεύθυνση στον χάρτη</label>
                   <input type="text" class="form-control" id="map-address"/>
                </div>


                <div id="map" style="width: 100%; height: 400px;"></div>
                <div class="form-group col-md-6 col-xs-12  @if ( $errors->has('latitude')) has-error @endif">
                                <label for="latitude"> Latitude</label>
                                {{Form::text('latitude','', array('id' => 'latitude', 'class' => 'form-control'))}}
                        </div>

                        <div class="form-group col-xs-6 col-xs-12 @if ( $errors->has('longitude')) has-error @endif">
                                <label for="longitude"> Longitude</label>
```

```
                    {{Form::text('longitude', '', array('id' => 'longitude', 'class' => 'form-control'))}}
                </div>

                <div class="form-group">
                    <button type="submit" class="btn btn-primary topmargin">@if (isset($section) && $section == 'edit') <i class="fa fa-check"></i> Επιβεβαίωση @else <i class="fa fa-plus-square"></i> Εισαγωγή @endif</button>
                </div>

            {{Form::close()}}
            </div>


            <div class="col-md-6 col-xs-12">
                @if (isset($section) && $section == 'edit')
                    <p class="lead">Πληροφορίες πυρκαγιάς</p>
                    <div class="row">
                        <div class="col-xs-12">
                            <a href="#" class="btn btn-primary" id="burnt_area_button"><i class="fa fa-fire"></i> Καμμένη έκσταση</a>
                        </div>
                    </div>
                    <div class="row topmargin-10">
                        <div class="col-xs-12">
                            <a href="#" class="btn btn-primary" id="staff_button"><i class="fa fa-users"></i> Προσωπικό</a>
                        </div>
                    </div>
                    <div class="row topmargin-10">
                        <div class="col-xs-12">
                            <a href="#" class="btn btn-primary" id="fire_forces_button"><i class="fa fa-fighter-jet"></i> <i class="fa fa-car"></i> Μέσα πυρόσβεσης</a>
                        </div>
                    </div>
                @endif
            </div>

        </div>

    @stop

    @section('extraModal')

        @if (isset($section) && $section == 'edit')
```

```
                <div class="modal fade" id="modify_burnt_area_modal">
                  <div class="modal-dialog">
                    <div class="modal-content">
                     <div class="modal-header noborder">
                       <h5 class="modal-title text-center nomargin">Επεξεργασία καμμένης έκστασης</h5>
                     </div>
                          {{ Form::open(array('url' => 'fire/edit/'.$id.'/burnt/area', 'method' => 'post')) }}
                                  <div class="row topmargin">
                                          <div class="col-xs-12">
                                                  <ul class="list-unstyled list-inline text-center">
                                                          <li>
                                                                  <div class="form-group">
                                                                          <label for="forest">Δάση</label>
                                                                          {{Form::text('forest', Input::old('forest', $fire->burnt_area()->first()->forest), array('id' => 'forest', 'class' => 'form-control numbersOnly'))}}
                                                                  </div>
                                                          </li>
                                                          <li>
                                                                  <div class="form-group">
                                                                          <label for="woodland">Δασική Έκταση</label>
                                                                          {{Form::text('woodland', Input::old('woodland', $fire->burnt_area()->first()->woodland), array('id' => 'woodland', 'class' => 'form-control numbersOnly'))}}
                                                                  </div>
                                                          </li>
                                                          <li>
                                                                  <div class="form-group">
                                                                          <label for="grove">Άλση</label>
                                                                          {{Form::text('grove', Input::old('grove', $fire->burnt_area()->first()->grove), array('id' => 'grove', 'class' => 'form-control numbersOnly'))}}
                                                                  </div>
                                                          </li>
                                                          <li>
                                                                  <div class="form-group">
                                                                          <label for="grass_land">Χορτ/κές Εκτάσεις</label>
                                                                          {{Form::text('grass_land', Input::old('grass_land', $fire->burnt_area()->first()->grass_land), array('id' => 'grass_land', 'class' => 'form-control numbersOnly'))}}
                                                                  </div>
                                                          </li>
                                                          <li>
```

```
                                        <div class="form-group">
                                            <label
for="reed_marches">Καλάμια - Βάλτοι</label>

        {{Form::text('reed_marches',    Input::old('reed_marches',    $fire->burnt_area()->first()-
>reed_marches), array('id' => 'reed_marches', 'class' => 'form-control numbersOnly'))}}
                                        </div>
                                    </li>
                                    <li>
                                        <div class="form-group">
                                            <label
for="farmland">Γεωργικές Εκτάσεις</label>
                                            {{Form::text('farmland',
Input::old('farmland', $fire->burnt_area()->first()->farmland), array('id' => 'farmland', 'class' => 'form-
control numbersOnly'))}}
                                        </div>
                                    </li>
                                    <li>
                                        <div class="form-group">
                                            <label
for="cultivation_waste">Υπολλείματα Καλλιεργειών</label>

        {{Form::text('cultivation_waste', Input::old('cultivation_waste', $fire->burnt_area()->first()-
>cultivation_waste), array('id' => 'cultivation_waste', 'class' => 'form-control numbersOnly'))}}
                                        </div>
                                    </li>
                                    <li>
                                        <div class="form-group">
                                            <label
for="dumping_ground">Σκουπιδότοποι</label>

        {{Form::text('dumping_ground', Input::old('dumping_ground', $fire->burnt_area()->first()-
>dumping_ground), array('id' => 'dumping_ground', 'class' => 'form-control numbersOnly'))}}
                                        </div>
                                    </li>
                                </ul>
                            </div>
                        </div>
                        <div class="modal-footer noborder text-center">
                    <button type="button" class="btn btn-danger" data-dismiss="modal"><i
class="fa fa-times"></i> Ακύρωση</button>
                        <button type="submit" class="btn btn-success"><i class="fa fa-
check"></i> Επιβεβαίωση</button>
                    </div>
                    {{ Form::close() }}
                </div>
            </div>
        </div>
```

```
<div class="modal fade" id="modify_staff_modal">
  <div class="modal-dialog">
    <div class="modal-content">
     <div class="modal-header noborder">
       <h5 class="modal-title text-center nomargin">Επεξεργασία προσωπικού</h5>
     </div>
          {{ Form::open(array('url' => 'fire/edit/'.$id.'/staff',  'method' => 'post')) }}
               <div class="row topmargin">
                    <div class="col-xs-12">
                         <ul class="list-unstyled list-inline text-center">
                              @foreach ($fires_staff as $staff)
                                   <li>
                                        <div class="form-group">
                                             <label
for="staff_type_{{$staff->id}}">{{$staff->name}}</label>

        {{Form::hidden('staff_ids[]', $staff->id)}}

        {{Form::text('staff_type_'.$staff->id, Input::old('staff_type_'.$staff->id,$staff->pivot->number),
array('id' => 'staff_type_'.$staff->id, 'class' => 'form-control numbersOnly'))}}
                                        </div>
                                   </li>
                              @endforeach
                         </ul>
                    </div>
               </div>
               <div class="modal-footer noborder text-center">
                    <button type="button" class="btn btn-danger" data-dismiss="modal"><i
class="fa fa-times"></i> Ακύρωση</button>
                    <button type="submit" class="btn btn-success"><i class="fa fa-
check"></i> Επιβεβαίωση</button>
               </div>
               {{ Form::close() }}
    </div>
  </div>
</div>


<div class="modal fade" id="modify_fire_forces_modal">
  <div class="modal-dialog">
    <div class="modal-content">
     <div class="modal-header noborder">
       <h5 class="modal-title text-center nomargin">Επεξεργασία μέσων
πυρόσβεσης</h5>
     </div>
          {{ Form::open(array('url' => 'fire/edit/'.$id.'/fire/forces', 'method' => 'post')) }}
```

```
                                <div class="row topmargin">
                                    <div class="col-xs-12">
                                        <ul class="list-unstyled list-inline text-center">
                                            @foreach ($fires_forces as $fires_force)
                                                <li>
                                                    <div class="form-group">
                                                        <label
for="fires_forces_type_{{$fires_force->id}}">{{$fires_force->name}}</label>

        {{Form::hidden('fire_forces_ids[]', $fires_force->id)}}

        {{Form::text('fires_forces_type_'.$fires_force->id,
Input::old('fires_forces_type_'.$fires_force->id,  $fires_force->pivot->number),  array('id'  =>
'fires_forces_type_'.$fires_force->id, 'class' => 'form-control numbersOnly'))}}
                                                    </div>
                                                </li>
                                            @endforeach
                                        </ul>
                                    </div>
                                </div>
                                <div class="modal-footer noborder text-center">
                            <button type="button" class="btn btn-danger" data-dismiss="modal"><i
class="fa fa-times"></i> Ακύρωση</button>
                                <button type="submit" class="btn btn-success"><i class="fa fa-
check"></i> Επιβεβαίωση</button>
                            </div>
                            {{ Form::close() }}
                          </div>
                        </div>
                      </div>

        @endif

        <div class="modal fade" id="succes_modal">
          <div class="modal-dialog">
            <div class="modal-content">
             <div class="modal-header noborder">
                <h4 class="modal-title text-center text-success"><big><span class="glyphicon
glyphicon-ok"></span></big></h4>
              </div>
              <div class="modal-body text-center">
                <p>Οι αλλαγές σας αποθηκεύτηκαν με επιτυχία</p>
              </div>
              <div class="modal-footer noborder text-center">
                <button type="button" class="btn btn-primary" data-
dismiss="modal">Κλείσιμο</button>
              </div>
            </div>
```

```
        </div>
      </div>

@stop

@section('extraScript')
    {{HTML::script('js/bootstrap-select.min.js')}}
    {{HTML::script('js/moment.min.js')}}
    {{HTML::script('js/moment-gr.js')}}
    {{HTML::script('js/bootstrap-datetimepicker.min.js')}}
    {{HTML::script('http://maps.google.com/maps/api/js?sensor=false&libraries=places')}}
    {{HTML::script('js/locationpicker.jquery.min.js')}}

    <script type="text/javascript">
        $(function() {

            $('.selectpicker').selectpicker();
            $('[data-toggle="tooltip"]').tooltip();

            $('#submit_form').on('submit', function(){
                $('#id_fire_department').removeAttr('disabled');
                $('#id_prefecture').removeAttr('disabled');
                $('#id_municipality').removeAttr('disabled');
            });

         $('body').on('change', '#id_region', function(){
            id_region = $(this).val();

            $.ajax({
        type: "POST",
        url: "{{URL::to('get/informations/by/region/id')}}",
        data: "&id_region="+id_region,
        contentType: "application/x-www-form-urlencoded;charset=UTF-8",
        success: function(response){
          if (response.success === true)
          {
            $('#id_fire_department').removeAttr('disabled');
            $('#id_fire_department').html('');
            $.each(response.fire_departments, function(){
           $('#id_fire_department').append('<option value="' + this.id + '">' + this.name +
'</option>');
           $('#id_fire_department').selectpicker('refresh');
        });

            $('#id_prefecture').removeAttr('disabled');
        $('#id_prefecture').html('');
```

```
                $('#id_prefecture').append('<option value="0" class="hide"></option>');
                        $.each(response.prefectures, function(){
                    $('#id_prefecture').append('<option value="' + this.id + '">' + this.name +
'</option>');
                    $('#id_prefecture').selectpicker('refresh');
                  });
                  $('#id_municipality').html('');
                  $('#id_municipality').selectpicker('refresh');
                   }
                        }
               });
                        return false;
                 });

                        $('body').on('change', '#id_prefecture', function(){
                        id_prefecture = $(this).val();

                        $.ajax({
                  type: "POST",
                  url: "{{URL::to('get/municipality/by/prefecture/id')}}",
                  data: "&id_prefecture="+id_prefecture,
                  contentType: "application/x-www-form-urlencoded;charset=UTF-8",
                  success: function(response){
                   if (response.success === true)
                   {
                        $('#id_municipality').removeAttr('disabled');
                        $('#id_municipality').html('');
                        $.each(response.municipalities, function(){
                    $('#id_municipality').append('<option value="' + this.id + '">' + this.name +
'</option>');
                    $('#id_municipality').selectpicker('refresh');
                  });
                   }
                        }
               });
                        return false;
                 });

                        $('#started_at').datetimepicker({
                                format : 'DD/MM/YYYY HH:mm'
                        });
            $('#ended_at').datetimepicker({
                format: 'DD/MM/YYYY HH:mm'
            });
            $("#started_at").on("dp.change", function (e) {
                $('#ended_at').data("DateTimePicker").minDate(e.date);
```

```
    });
    $("#ended_at").on("dp.change", function (e) {
        $('#started_at').data("DateTimePicker").maxDate(e.date);
    });


    $('#map').locationpicker({
        @if (isset($section) && $section == 'edit')
        location: {latitude: {{$fire->latitude}}, longitude: {{$fire->longitude}}},
    @else
        location: {latitude: 41.133352, longitude: 24.88905},
    @endif
    radius: 0,
    inputBinding: {
        latitudeInput: $('#latitude'),
        longitudeInput: $('#longitude'),
       locationNameInput: $('#map-address')
    },
    enableAutocomplete: true,

    onchanged: function (currentLocation, radius, isMarkerDropped) {
        $("#latitude").val(currentLocation.latitude);
        $("#longitude").val(currentLocation.longitude);
    },
    oninitialized: function(component) {}
});


            @if (isset($section) && $section == 'edit')

                $('body').on('click', '#burnt_area_button', function(){
                $("#modify_burnt_area_modal").modal();
                return false;
                });

                $('body').on('click', '#staff_button', function(){
                $("#modify_staff_modal").modal();
                return false;
                });

                $('body').on('click', '#fire_forces_button', function(){
                $("#modify_fire_forces_modal").modal();
                return false;
                });

                jQuery('.numbersOnly').keyup(function () {
```

```
                    this.value = this.value.replace(/[^0-9\.]/g,'');
                });

                @endif

                @if (Session::has('success') && Session::get('success'))
                $("#succes_modal").modal();
                return false;
            @endif

        });
    </script>
@stop
```

Συνάρτηση fireInsertCreate του FireController για την εισαγωγή νέας Πυρκαγιάς:

```
public function fireInsertCreate() {
  $regions         = Region::all();
  $fire_departments = FireDepartment::all();
  $data     = array(  'title' => 'Εισαγωγή Πυρκαγιάς',
                'regions' => $regions,
                'fire_departments' => $fire_departments);
  $this->layout->content = View::make('admin.fires.insert')->with($data);
 }
```

Route fire/insert για την εκτέλεση της fireInsertCreate:

```
Route::get('fire/insert', array('as' => 'fire_insert', 'uses' => 'FireController@fireInsertCreate'));
```

Συνάρτηση fireInsertStore του FireController για την υποβολή της φόρμας της νέας Πυρκαγιάς:

```
public function fireInsertStore()
 {
  $rules = array(
   'id_region'         => array('required'),
   'id_fire_department' => array('required'),
   'id_prefecture'      => array('required'),
   'id_municipality'    => array('required'),
   'started_at'        => array('required'),
   'latitude'          => array('required'),
   'longitude'         => array('required')
  );

  $validation = Validator::make(Input::all(), $rules);

  if ($validation->fails())
  {
   return Redirect::to('fire/insert')->withErrors($validation)->withInput();
```

```
    }
    else
    {

        $fire                   = new Fire();
        $fire->id_fire_department = Input::get('id_fire_department');
        $fire->id_prefecture      = Input::get('id_prefecture');
        $fire->id_municipality    = Input::get('id_municipality');


        $start_date = $fire->transformDateForDataBase(Input::get('started_at'));

        if (Input::has('ended_at') && Input::get('ended_at') != "")
          $end_date = $fire->transformDateForDataBase(Input::get('ended_at'));
        else
          $end_date = Input::get('ended_at');


        $fire->start_date = $start_date;
        $fire->end_date   = $end_date;
        $fire->forestry   = Input::get('forestry');
        $fire->area       = Input::get('area');
        $fire->adress     = Input::get('adress');
        $fire->latitude   = Input::get('latitude');
        $fire->longitude  = Input::get('longitude');

        $fire->created_at = date("Y-m-d H:i:s");
        $fire->updated_at = date("Y-m-d H:i:s");
        $fire->save();

        $burtn_area = new BurntArea();
        $burtn_area->id_fire = $fire->id;
        $burtn_area->save();


        $staff_all = Staff::all();
        foreach ($staff_all as $staff) {
          $fire_staff          = new FireStaff();
          $fire_staff->id_fire  = $fire->id;
          $fire_staff->id_staff = $staff->id;
          $fire_staff->save();
        }


        $forces = Force::all();
        foreach ($forces as $force) {
```

```
        $fire_force          = new FireForce();
        $fire_force->id_fire  = $fire->id;
        $fire_force->id_force = $force->id;
        $fire_force->save();
      }

      return Redirect::to('/fire/edit/'.$fire->id);
   }

 }
```

Route fire/insert για την εκτέλεση της fireInsertStore:

```
Route::post('fire/insert', array('as' => 'fire_insert', 'uses' => 'FireController@fireInsertStore'));
```

Συνάρτηση fireEditCreate του FireController για την προβολή της επεξεργασίας Πυρκαγιάς:

```
public function fireEditCreate($id)
 {
    $fire            = Fire::find($id);
    $regions         = Region::all();
    $fire_departments = FireDepartment::all();
    $prefectures     = Prefecture::all();
    $municipalities   = Municipality::all();

    // $fires_staff = FireStaff::where('id_fire', '=', $fire->id)->get();
    $fires_staff  = $fire->staff()->get();
    $fires_forces = $fire->forces()->get();

    $region_id = $fire->prefecture()->first()->id_region;

    //$burtn_area = BurntArea::where('id_fire', '=', $fire->id)->first();
    $burtn_area = $fire->burnt_area()->first()->id;


    $start_date = $fire->transformDateFromDataBase ($fire->start_date);

    if ($fire->end_date != "")
      $end_date = $fire->transformDateFromDataBase ($fire->end_date);
    else
      $end_date = $fire->end_date;

    $data = array(
```

```
                'title'           => 'Πυρκαγιές',
                'fire'            => $fire,
                'burtn_area'      => $burtn_area,
                'section'         => 'edit',
                'regions'         => $regions,
                'fire_departments' => $fire_departments,
                'prefectures'     => $prefectures,
                'municipalities'  => $municipalities,
                'start_date'      => $start_date,
                'end_date'        => $end_date,
                'fires_staff'     => $fires_staff,
                'fires_forces'    => $fires_forces,
                'region_id'       => $region_id,
                'id'              => $id
              );

    $this->layout->content = View::make('admin.fires.insert')->with($data);
  }
```

Route fire/edit/{id} για την εκτέλεση της fireEditCreate:

```
Route::get('fire/edit/{id}', array('as' => 'fire_edit', 'uses' => 'FireController@fireEditCreate'))->where('id', '[0-9]+');
```

Συνάρτηση fireEditStore του FireController για την υποβολή της φόρμας επεξεργασίας της Πυρκαγιάς:

```
public function fireEditStore($id)
 {
  $rules = array(
   'id_region'         => array('required'),
   'id_fire_department' => array('required'),
   'id_prefecture'     => array('required'),
   'id_municipality'   => array('required'),
   'started_at'        => array('required'),
   'latitude'          => array('required'),
   'longitude'         => array('required')
  );

  $validation = Validator::make(Input::all(), $rules);

  if ($validation->fails())
  {
    return Redirect::to('fire/edit/'.$id)->withErrors($validation)->withInput();
  }
  else
  {
    if (Input::has('id_region') && Input::get('id_region') == 0)
      return Redirect::to('fire/edit/'.$id)->withErrors($validation)->withInput();
```

```
        else if (Input::has('id_fire_department') && Input::get('id_fire_department') == 0)
          return Redirect::to('fire/edit/'.$id)->withErrors($validation)->withInput();
        else if (Input::has('id_prefecture') && Input::get('id_prefecture') == 0)
          return Redirect::to('fire/edit/'.$id)->withErrors($validation)->withInput();
        else if (Input::has('id_municipality') && Input::get('id_municipality') == 0)
          return Redirect::to('fire/edit/'.$id)->withErrors($validation)->withInput();

        $fire = Fire::find($id);
        $fire->id_fire_department = Input::get('id_fire_department');
        $fire->id_prefecture      = Input::get('id_prefecture');
        $fire->id_municipality    = Input::get('id_municipality');


        $start_date = $fire->transformDateForDataBase(Input::get('started_at'));

        if (Input::has('ended_at') && Input::get('ended_at') != "")
          $end_date = $fire->transformDateForDataBase(Input::get('ended_at'));
        else
          $end_date = Input::get('ended_at');


        $fire->start_date = $start_date;
        $fire->end_date   = $end_date;
        $fire->forestry   = Input::get('forestry');
        $fire->area       = Input::get('area');
        $fire->adress     = Input::get('adress');
        $fire->latitude   = Input::get('latitude');
        $fire->longitude  = Input::get('longitude');
        $fire->updated_at = date("Y-m-d H:i:s");
        $fire->save();

        return Redirect::to('/fires/list')->with(array('success'=> true));
      }

    }
```

Route fire/edit/{id} για την εκτέλεση της fireEditStore:

```
Route::post('fire/edit/{id}', array('as' => 'fire_edit', 'uses' => 'FireController@fireEditStore'))->where('id', '[0-9]+');
```

Συνάρτηση fireEditBurntAreaStore του FireController για την υποβολή της φόρμας καμένης έκτασης της Πυρκαγιάς:

```
public function fireEditBurntAreaStore($id)
 {
    $fire = Fire::find($id);
    $burtn_area = $fire->burnt_area()->first();
```

```
        $burtn_area->forest           = Input::get('forest');
        $burtn_area->woodland         = Input::get('woodland');
        $burtn_area->grove            = Input::get('grove');
        $burtn_area->grass_land       = Input::get('grass_land');
        $burtn_area->reed_marches     = Input::get('reed_marches');
        $burtn_area->farmland         = Input::get('farmland');
        $burtn_area->cultivation_waste = Input::get('cultivation_waste');
        $burtn_area->dumping_ground   = Input::get('dumping_ground');
        $burtn_area->save();

        return Redirect::to('fire/edit/'.$fire->id)->with(array('success'=> true));
    }
```

Route fire/edit/{id}/burnt/area για την εκτέλεση της fireEditBurntAreaStore:

```
Route::post('fire/edit/{id}/burnt/area', array('uses' => 'FireController@fireEditBurntAreaStore'))->where('id', '[0-9]+');
```

Συνάρτηση fireEditStaffStore του FireController για την υποβολή της φόρμας προσωπικού της Πυρκαγιάς:

```
    public function fireEditStaffStore($id)
    {

        $rules = array(
          'staff_ids'  => array('required'),
        );

        $validation = Validator::make(Input::all(), $rules);

        if ($validation->fails())
        {
            return Redirect::to('fire/edit/'.$fire->id)->withErrors($validation)->withInput();
        }
        else
        {
          $staff_ids = Input::get('staff_ids');
          $fire = Fire::find($id);

          foreach ($staff_ids as $staff_id) {
            $fire->staff()->updateExistingPivot($staff_id, array('number' => Input::get('staff_type_'.$staff_id)), false);
          }

          return Redirect::to('fire/edit/'.$fire->id)->with(array('success'=> true));
        }
```

```
}
```

Route fire/edit/{id}/staff για την εκτέλεση της fireEditStaffStore:

```
Route::post('fire/edit/{id}/staff', array('uses' => 'FireController@fireEditStaffStore'))->where('id', '[0-9]+');
```

Συνάρτηση fireEditFireForcesStore του FireController για την υποβολή της φόρμας πυροσβεστικών μέσων της Πυρκαγιάς:

```
public function fireEditFireForcesStore($id)
{

  $rules = array(
    'fire_forces_ids'  => array('required'),
  );

  $validation = Validator::make(Input::all(), $rules);

  if ($validation->fails())
  {
     return Redirect::to('fire/edit/'.$fire->id)->withErrors($validation)->withInput();
  }
  else
  {
    $fire_forces_ids = Input::get('fire_forces_ids');
    $fire = Fire::find($id);

    foreach ($fire_forces_ids as $fire_force_id) {
      $fire->forces()->updateExistingPivot($fire_force_id, array('number' => Input::get('fires_forces_type_'.$fire_force_id)), false);
    }

    return Redirect::to('fire/edit/'.$fire->id)->with(array('success'=> true));
  }
}
```

Route fire/edit/{id}/fire/forces για την εκτέλεση της fireEditFireForcesStore:

```
Route::post('fire/edit/{id}/fire/forces', array('uses' => 'FireController@fireEditFireForcesStore'))->where('id', '[0-9]+');
```

Συναρτήση για την προβολή της λίστας Πυροσβεστικών Υπηρεσιών ανάλογα με την περιφέρεια:

```
public function getInformationsByRegionId() {
   $rules = array(
     'id_region' => array('required')
   );

   $validation = Validator::make(Input::all(), $rules);
```

```
        if ($validation->fails())
        {
          return Response::json(array('success' => false));
        }
        else
        {

          $region          = Region::find(Input::get('id_region'));
          $fire_departments = $region->fire_departments()->get();
          $prefectures      = Prefecture::where('id_region', '=', $region->id)->get();

          return Response::json(array(  'success' => true, 'fire_departments' => $fire_departments,
'prefectures' => $prefectures));
        }
      }
```

Route get/informations/by/region/id για την εκτέλεση της getInformationsByRegionId:

```
Route::post('get/informations/by/region/id', array('uses' => 'FireController@getInformationsByRegionId'));
```

Συναρτήση για την προβολή της λίστας Δήμων ανάλογα με τον νομό:

```
  public function getMunicipalityByPrefectureId() {
     $rules = array(
        'id_prefecture' => array('required')
     );

     $validation = Validator::make(Input::all(), $rules);

     if ($validation->fails())
     {
        return Response::json(array('success' => false));
     }
     else
     {

       $prefecture     = Prefecture::find(Input::get('id_prefecture'));
       $municipalities = Municipality::where('id_prefecture', '=', $prefecture->id)->get();

       return Response::json(array(  'success' => true, 'municipalities' => $municipalities));
     }
   }
```

Route get/municipality/by/prefecture/id για την εκτέλεση της getMunicipalityByPrefectureId:

```
Route::post('get/municipality/by/prefecture/id', array('uses' => 'FireController@getMunicipalityByPrefectureId'));
```

View forces/list.blade.php για την προβολή, εισαγωγή και επεξεργασία των πυροσβεστικών μέσων

```
@section('title')
  {{$title}}
@stop


@section('extraStyle')
  {{HTML::style('css/bootstrap-select.min.css')}}
@stop


@section('content')
    <div class="row">
        <div class="col-xs-12">
            <h1>Πυροσβεστικά Μέσα</h1>
            <p class="lead">Λίστα πυροσβεστικών μέσων</p>
        </div>
    </div>
    <div class="row">
        <div class="col-xs-12">
            <a href="#" class="btn btn-primary pull-right" id="insert"><i class="fa fa-plus-square"></i> Εισαγωγή πυροσβεστικού μέσου</a>
        </div>
    </div>
    <div class="row topmargin">
        <div class="col-xs-12">
            <table class="table table-striped table-hover ">
                <thead>
                    <tr>
                        <th>Όνομα</th>
                        <th>Τύπος</th>
                        <th>Τύπος μέσου</th>
                        <th>Ενέργειες</th>
                    </tr>
                </thead>
                <tbody>
                    @foreach ($forces as $force)
                    <tr id="force_{{$force->id}}">
                        <td>{{$force->name}}</td>
                        <td>{{$force->type}}</td>
                        <td>{{$force->forces_types()->first()->name}}</td>
                        <td>
```

```
                                                    <a href="#" class="modify" data-id="{{$force->id}}" data-toggle="tooltip" data-placement="top" title="Επεξεργασία"><i class="fa fa-pencil-square-o"></i></a>
                                                    <a href="#" class="delete" data-id="{{$force->id}}" data-toggle="tooltip" data-placement="top" title="Διαγραφή"><i class="fa fa-trash-o"></i></a>
                                                </td>
                                            </tr>
                                            @endforeach
                                        </tbody>
                                    </table>
                                </div>
                            </div>
                        @stop

                        @section('extraModal')

                            <div class="modal fade" id="insert_force_modal">
                              <div class="modal-dialog">
                                <div class="modal-content">
                                 <div class="modal-header noborder">
                                    <h5 class="modal-title text-center nomargin">Εισαγωγή πυροσβεστικού μέσου</h5>
                                 </div>
                                 {{ Form::open(array('url' => 'force/insert',  'method' => 'post')) }}
                                 <div class="modal-body">
                                        <div class="form-group  @if ($errors->has('name')) has-error @endif">
                                            <label for="name">Όνομα</label>
                                            {{Form::text('name', Input::old('name'), array('id' => 'name', 'class' => 'form-control'))}}
                                        </div>
                                        <div class="form-group @if ($errors->has('type')) has-error @endif">
                                            <label for="type"> Τύπος</label>
                                            {{Form::text('type', Input::old('type'), array('id' => 'type', 'class' => 'form-control', 'placeholder' => 'π.χ.: other_forces'))}}
                                        </div>
                                        <div class="form-group @if ( $errors->has('id_force_type') || (Session::has('force_type_empty') && Session::get('force_type_empty'))) error @endif">
                                            <label for="id_force_type"> Τύπος Μέσου</label>
                                            <select class="form-control selectpicker" title='Επιλέξτε μέσο' id="id_force_type" name="id_force_type">
                                                <option class="hide" value="0"></option>
                                                @foreach ($forces_types as $forces_type)
                                                <option value="{{$forces_type->id}}">{{$forces_type->name}}</option>
                                                @endforeach
                                            </select>
                                        </div>
                                </div>
```

```
            <div class="modal-footer noborder text-center">
              <button type="button" class="btn btn-danger" data-dismiss="modal"><i class="fa fa-
times"></i> Ακύρωση</button>
              <button type="submit" class="btn btn-success"><i class="fa fa-check"></i>
Επιβεβαίωση</button>
            </div>
            {{ Form::close() }}
           </div><!-- /.modal-content -->
         </div><!-- /.modal-dialog -->
        </div><!-- /.modal -->

        <div class="modal fade" id="modify_force_modal">
          <div class="modal-dialog">
            <div class="modal-content">
             <div class="modal-header noborder">
               <h5 class="modal-title text-center nomargin">Επεξεργασία πυροσβεστικού
μέσου</h5>
            </div>
            {{ Form::open(array('url' => 'force/modify',  'method' => 'post')) }}
            <div class="modal-body">
                        <div class="form-group  @if ($errors->has('name')) has-error @endif">
                                <label for="name">Όνομα</label>
                                {{Form::text('name', Input::old('name'), array('id' => 'name', 'class'
=> 'form-control'))}}
                        </div>
                        <div class="form-group @if ($errors->has('type')) has-error @endif">
                                <label for="type"> Τύπος</label>
                                {{Form::text('type', Input::old('type'), array('id' => 'type', 'class' =>
'form-control'))}}
                        </div>
                        <div class="form-group @if ( $errors->has('id_force_type') ||
(Session::has('force_type_empty') && Session::get('force_type_empty'))) has-error @endif">
                                <label for="id_force_type"> Τύπος Μέσου</label>
                                <select class="form-control selectpicker" title='Επιλέξτε μέσο'
id="id_force_type" name="id_force_type">
                                        <option class="hide" value="0"></option>
                                        @foreach ($forces_types as $forces_type)
                                        <option value="{{$forces_type->id}}" @if
(Input::old('id_force_type') == $forces_type->id) {{ 'selected' }} @endif>{{$forces_type-
>name}}</option>
                                        @endforeach
                                </select>
                        </div>
                        {{Form::hidden('id_force', '', array('id' => 'id_force'))}}
            </div>
            <div class="modal-footer noborder text-center">
              <button type="button" class="btn btn-danger" data-dismiss="modal"><i class="fa fa-
times"></i> Ακύρωση</button>
```

```
                <button type="submit" class="btn btn-success"><i class="fa fa-check"></i> Επιβεβαίωση</button>
              </div>
              {{ Form::close() }}
            </div>
          </div>
        </div>


        <div class="modal fade" id="delete_force_modal">
          <div class="modal-dialog">
            <div class="modal-content">
              <div class="modal-header noborder">
                <h4 class="modal-title text-center text-warning"><big><span class="glyphicon glyphicon-warning-sign"></span></big></h4>
              </div>
              <div class="modal-body text-center">
                  <p>Είστε σίγουροι ότι θέλετε να διαγράψετε το πυροσβεστικό μέσο:</p>
                  <p><strong><span id="force_name" class="orange"></span></strong></p>
                  {{Form::hidden('id_force', '', array('id' => 'id_force'))}}
              </div>
              <div class="modal-footer noborder text-center">
                <button type="button" class="btn btn-danger" data-dismiss="modal"><i class="fa fa-times"></i> Ακύρωση</button>
                            <button type="button" class="btn btn-success" id="delete_force_button"><i class="fa fa-check"></i> Επιβεβαίωση</button>
              </div>
            </div>
          </div>
        </div>

        <div class="modal fade" id="succes_modal">
          <div class="modal-dialog">
            <div class="modal-content">
              <div class="modal-header noborder">
                <h4 class="modal-title text-center text-success"><big><span class="glyphicon glyphicon-ok"></span></big></h4>
              </div>
              <div class="modal-body text-center">
                <p>Οι αλλαγές σας αποθηκεύτηκαν με επιτυχία</p>
              </div>
              <div class="modal-footer noborder text-center">
                <button type="button" class="btn btn-primary" data-dismiss="modal">Κλείσιμο</button>
              </div>
            </div>
          </div>
```

```
        </div>

        <div class="modal fade" id="delete_error_modal">
          <div class="modal-dialog">
            <div class="modal-content">
             <div class="modal-header noborder">
               <h4 class="modal-title text-center text-danger"><big><i class="fa fa-
times"></i></big></h4>
             </div>
             <div class="modal-body text-center">
               <p id="message"></p>
             </div>
             <div class="modal-footer noborder text-center">
               <button type="button" class="btn btn-primary" data-
dismiss="modal">Κλείσιμο</button>
             </div>
            </div>
          </div>
        </div>


    @stop

    @section('extraScript')
        {{HTML::script('js/bootstrap-select.min.js')}}
        <script type="text/javascript">
                $(function() {

                        $('.selectpicker').selectpicker();

                        $('[data-toggle="tooltip"]').tooltip();

                        $('body').on('click', '#insert', function(){
                                $('#insert_force_modal #name').parent().removeClass('has-error');
                        $('#insert_force_modal #type').parent().removeClass('has-error');
                        $('#insert_force_modal #id_force_type').parent().removeClass('error');

                        $("#insert_force_modal").modal();
                    });

                    @if (Session::has('error') && Session::get('error') == 'insert')
                        $("#insert_force_modal").modal();
                    @endif
```

```
    $('body').on('click', '.delete', function(){

            id_force = $(this).attr('data-id');

            $.ajax({
  type: "POST",
  url: "{{URL::to('force/get/by/id')}}",
  data: "&id_force="+id_force,
  contentType: "application/x-www-form-urlencoded;charset=UTF-8",
  success: function(response){
   if (response.success === true)
   {
    $('#delete_force_modal #force_name').html(response.force_name);
    $('#delete_force_modal #id_force').val(id_force);
    $("#delete_force_modal").modal();
   }
            }
});
            return false;
   });

    $('body').on('click', '#delete_force_button', function(){

            id_force = $("#delete_force_modal").find("#id_force").val();

            $.ajax({
  type: "POST",
  url: "{{URL::to('force/delete')}}",
  data: "&id_force="+id_force,
  contentType: "application/x-www-form-urlencoded;charset=UTF-8",
  success: function(response){
   if (response.success === true)
   {
            $('tr#force_'+id_force).remove();
    $("#delete_force_modal").modal('hide');
    $("#succes_modal").modal();
   } else {
            $("#delete_force_modal").modal('hide');
            $("#delete_error_modal #message").html(response.message);
            $("#delete_error_modal").modal();
   }
            }
});
$('#delete_force_modal #id_force').val('');
            return false;
   });
```

```
                $('body').on('click', '.modify', function(){

                        id_force = $(this).attr('data-id');

                        $('#modify_force_modal #name').parent().removeClass('has-error');
                        $('#modify_force_modal #type').parent().removeClass('has-error');
                        $('#modify_force_modal #id_force_type').parent().removeClass('error');

                        $.ajax({
                    type: "POST",
                    url: "{{URL::to('force/get/by/id')}}",
                    data: "&id_force="+id_force,
                    contentType: "application/x-www-form-urlencoded;charset=UTF-8",
                    success: function(response){
                     if (response.success === true)
                     {
                       $('#modify_force_modal #name').val(response.force_name);
                       $('#modify_force_modal #type').val(response.force_type);

                       $('#modify_force_modal #id_force_type').val(response.id_force_type);
                       $('#modify_force_modal #id_force_type').selectpicker('refresh');

                       $('#modify_force_modal #id_force').val(id_force);
                       $("#modify_force_modal").modal();
                     }
                        }
                });
                        return false;
                    });

                    @if (Session::has('error') && Session::get('error') == 'modify')
                        $("#modify_force_modal").modal();
                    @endif

                    @if (Session::has('success') && Session::get('success'))
                        $("#succes_modal").modal();
                    @endif

                });
        </script>
    @stop
```

Συνάρτηση forcesListCreate του ForceController για την προβολή των πυροσβεστικών μέσων:

```
public function forcesListCreate()
 {
  $forces       = Force::all();
  $forces_types = ForceType::all();
  $data = array ( 'title'=> 'Πυροσβεστικά Μέσα' , 'forces' => $forces, 'forces_types' => $forces_types );
  $this->layout->content = View::make('admin.forces.list')->with($data);
 }
```

Route forces/list για την εκτέλεση της forcesListCreate:

```
Route::get('forces/list', array('as' => 'forces_list', 'uses' => 'ForceController@forcesListCreate'));
```

Συνάρτηση forceInsertStore του ForceController για την υποβολή της φόρμας εισαγωγής ενός πυροσβεστικού μέσου:

```
public function forceInsertStore()
 {
  $rules = array(
   'name'         => array('required'),
   'type'         => array('required'),
   'id_force_type' => array('required')
  );

  $validation = Validator::make(Input::all(), $rules);

  if ($validation->fails())
  {
   if (Input::has('id_force_type') && Input::get('id_force_type') == 0)
     return                    Redirect::to('forces/list')->withErrors($validation)->withInput()->with(array('error'=>'insert', 'force_type_empty' => true));
    return                     Redirect::to('forces/list')->withErrors($validation)->withInput()->with(array('error'=>'insert'));
  }
  else
   {
    if (Input::has('id_force_type') && Input::get('id_force_type') == 0)
     return                    Redirect::to('forces/list')->withErrors($validation)->withInput()->with(array('error'=>'insert', 'force_type_empty' => true));

    $force              = new Force();
    $force->name        = Input::get('name');
    $force->type        = Input::get('type');
    $force->id_force_type = Input::get('id_force_type');
    $force->created_at  = date("Y-m-d H:i:s");
    $force->updated_at  = date("Y-m-d H:i:s");
    $force->save();

    return Redirect::to('/forces/list')->with(array('success'=> true));
```

```
    }
  }
```

Route force/insert για την εκτέλεση της forceInsertStore:

```
Route::post('force/insert', array('as' => 'forces_list', 'uses' => 'ForceController@forceInsertStore'));
```

Συνάρτηση forceModifyStore του ForceController για την υποβολή της φόρμας επεξεργασίας ενός πυροσβεστικού μέσου:

```
public function forceModifyStore()
  {
    $rules = array(
      'name'         => array('required'),
      'type'         => array('required'),
      'id_force_type' => array('required'),
      'id_force'     => array('required')
    );

    $validation = Validator::make(Input::all(), $rules);

    if ($validation->fails())
    {
      if (Input::has('id_force_type') && Input::get('id_force_type') == 0)
        return Redirect::to('forces/list')->withErrors($validation)->withInput()->with(array('error'=>'modify', 'force_type_empty' => true));
      return Redirect::to('forces/list')->withErrors($validation)->withInput()->with(array('error'=>'modify'));
    }
    else
    {
      if (Input::has('id_force_type') && Input::get('id_force_type') == 0)
        return Redirect::to('forces/list')->withErrors($validation)->withInput()->with(array('error'=>'modify', 'force_type_empty' => true));

      $force              = Force::find(Input::get('id_force'));
      $force->name        = Input::get('name');
      $force->type        = Input::get('type');
      $force->id_force_type = Input::get('id_force_type');
      $force->created_at  = date("Y-m-d H:i:s");
      $force->updated_at  = date("Y-m-d H:i:s");
      $force->save();

      return Redirect::to('forces/list')->with(array('success'=> true));
    }
  }
```

Route force/modify για την εκτέλεση της forceModifyStore:

```
Route::post('force/modify', array('as' => 'forces_list', 'uses' => 'ForceController@forceModifyStore'));
```

Συνάρτηση forceGetById του ForceController για την προβολή του πυροσβεστικού μέσου προς επεξεργασία:

```
public function forceGetById()
 {
   $rules = array(
      'id_force' => array('required')
   );

   $validation = Validator::make(Input::all(), $rules);

   if ($validation->fails())
   {
      return Response::json(array('success' => false));
   }
   else
   {
     $force = Force::find(Input::get('id_force'));
     return Response::json(array(  'success' => true, 'force_name' => $force->name, 'force_type' => $force->type, 'id_force_type' => $force->id_force_type));
   }

 }
```

Route force/get/by/id για την εκτέλεση της forceInsertStore:

```
Route::post('force/get/by/id', array('uses' => 'ForceController@forceGetById'));
```

Συνάρτηση forceDeleteStore του ForceController για την διαγραφή ενός πυροσβεστικού μέσου:

```
public function forceDeleteStore()
 {
   $rules = array(
      'id_force' => array('required')
   );

   $validation = Validator::make(Input::all(), $rules);

   if ($validation->fails())
   {
     return Response::json(array('success' => false));
   }
   else
   {
     $force = Force::find(Input::get('id_force'));
```

```
        $fires_count = $force->fires()->count();

        if ($fires_count > 0)
          return Response::json(array(  'success' => false, 'message' => 'Το συγκεκριμένο πυροσβεστικό μέσο έχει ήδη καταχωρυθεί σε κάποια πυρκαγιά και επομένως δεν μπορεί να διαγραφεί'));
        else {
          $force->delete();
          return Response::json(array(  'success' => true));
        }
       }

     }
```

Route force/delete για την εκτέλεση της forceDeleteStore:

```
Route::post('force/delete', array('uses' => 'ForceController@forceDeleteStore'));
```

View staff/list.blade.php για την προβολή, εισαγωγή και επεξεργασία του προσωπικού:

```
@section('title')
   {{$title}}
@stop

@section('content')
        <div class="row">
                <div class="col-xs-12">
                        <h1>Προσωπικό</h1>
                        <p class="lead">Λίστα προσωπικού</p>
                </div>
        </div>
        <div class="row">
                <div class="col-xs-12">
                        <a href="#" class="btn btn-primary pull-right" id="insert"><i class="fa fa-plus-square"></i> Εισαγωγή προσωπικού</a>
                </div>
        </div>
        <div class="row topmargin">
                <div class="col-xs-12">
                        <table class="table table-striped table-hover ">
                                <thead>
                                        <tr>
                                                <th>Όνομα</th>
                                                <th>Τύπος</th>
                                                <th>Ενέργειες</th>
                                        </tr>
                                </thead>
```

```
                        <tbody>
                            @foreach ($all_staff as $staff)
                            <tr id="staff_{{$staff->id}}">
                                <td>{{$staff->name}}</td>
                                <td>{{$staff->type}}</td>
                                <td>
                                    <a href="#" class="modify" data-id="{{$staff->id}}" data-toggle="tooltip" data-placement="top" title="Επεξεργασία"><i class="fa fa-pencil-square-o"></i></a>
                                    <a href="#" class="delete" data-id="{{$staff->id}}" data-toggle="tooltip" data-placement="top" title="Διαγραφή"><i class="fa fa-trash-o"></i></a>
                                </td>
                            </tr>
                            @endforeach
                        </tbody>
                    </table>
                </div>
            </div>
        @stop

        @section('extraModal')

            <div class="modal fade" id="insert_staff_modal">
              <div class="modal-dialog">
                <div class="modal-content">
                 <div class="modal-header noborder">
                   <h5 class="modal-title text-center nomargin">Εισαγωγή προσωπικού</h5>
                 </div>
                 {{ Form::open(array('url' => 'staff/insert',  'method' => 'post')) }}
                 <div class="modal-body">
                        <div class="form-group  @if ($errors->has('name')) has-error @endif">
                            <label for="name">Όνομα</label>
                            {{Form::text('name', Input::old('name'), array('id' => 'name', 'class' => 'form-control'))}}
                        </div>
                        <div class="form-group @if ($errors->has('type')) has-error @endif">
                            <label for="type"> Τύπος</label>
                            {{Form::text('type', Input::old('type'), array('id' => 'type', 'class' => 'form-control', 'placeholder' => 'π.χ.: other_forces'))}}
                        </div>
                 </div>
                 <div class="modal-footer noborder text-center">
                   <button type="button" class="btn btn-danger" data-dismiss="modal"><i class="fa fa-times"></i> Ακύρωση</button>
                    <button type="submit" class="btn btn-success"><i class="fa fa-check"></i> Επιβεβαίωση</button>
```

```
            </div>
            {{ Form::close() }}
           </div>
          </div>
         </div>


         <div class="modal fade" id="modify_staff_modal">
          <div class="modal-dialog">
           <div class="modal-content">
            <div class="modal-header noborder">
              <h5 class="modal-title text-center nomargin">Επεξεργασία προσωπικού</h5>
            </div>
            {{ Form::open(array('url' => 'staff/modify',  'method' => 'post')) }}
            <div class="modal-body">
                        <div class="form-group  @if ($errors->has('name')) has-error @endif">
                                <label for="name">Όνομα</label>
                                {{Form::text('name', Input::old('name'), array('id' => 'name', 'class'
=> 'form-control'))}}
                        </div>
                        <div class="form-group @if ($errors->has('type')) has-error @endif">
                                <label for="type"> Τύπος</label>
                                {{Form::text('type', Input::old('type'), array('id' => 'type', 'class' =>
'form-control'))}}
                        </div>
                        {{Form::hidden('staff_id', '', array('id' => 'staff_id'))}}
            </div>
            <div class="modal-footer noborder text-center">
              <button type="button" class="btn btn-danger" data-dismiss="modal"><i class="fa fa-
times"></i> Ακύρωση</button>
               <button type="submit" class="btn btn-success"><i class="fa fa-check"></i>
Επιβεβαίωση</button>
            </div>
            {{ Form::close() }}
           </div>
          </div>
         </div>


         <div class="modal fade" id="delete_staff_modal">
          <div class="modal-dialog">
           <div class="modal-content">
            <div class="modal-header noborder">
              <h4 class="modal-title text-center text-warning"><big><span class="glyphicon
glyphicon-warning-sign"></span></big></h4>
            </div>
            <div class="modal-body text-center">
```

```
                <p>Είστε σίγουροι ότι θέλετε να διαγράψετε το προσωπικό:</p>
                <p><strong><span id="staff_name" class="orange"></span></strong></p>
                {{Form::hidden('staff_id', '', array('id' => 'staff_id'))}}
            </div>
            <div class="modal-footer noborder text-center">
              <button type="button" class="btn btn-danger" data-dismiss="modal"><i class="fa fa-times"></i> Ακύρωση</button>
                            <button type="button" class="btn btn-success" id="delete_staff_button"><i class="fa fa-check"></i> Επιβεβαίωση</button>
            </div>
          </div>
        </div>
      </div>

      <div class="modal fade" id="succes_modal">
        <div class="modal-dialog">
          <div class="modal-content">
           <div class="modal-header noborder">
             <h4 class="modal-title text-center text-success"><big><span class="glyphicon glyphicon-ok"></span></big></h4>
            </div>
            <div class="modal-body text-center">
              <p>Οι αλλαγές σας αποθηκεύτηκαν με επιτυχία</p>
            </div>
            <div class="modal-footer noborder text-center">
              <button type="button" class="btn btn-primary" data-dismiss="modal">Κλείσιμο</button>
            </div>
          </div>
        </div>
      </div>

      <div class="modal fade" id="delete_error_modal">
        <div class="modal-dialog">
          <div class="modal-content">
           <div class="modal-header noborder">
             <h4 class="modal-title text-center text-danger"><big><i class="fa fa-times"></i></big></h4>
            </div>
            <div class="modal-body text-center">
              <p id="message"></p>
            </div>
            <div class="modal-footer noborder text-center">
              <button type="button" class="btn btn-primary" data-dismiss="modal">Κλείσιμο</button>
            </div>
          </div>
```

```
        </div>
      </div>

@stop

@section('extraScript')
      <script type="text/javascript">
            $(function() {

                  $('[data-toggle="tooltip"]').tooltip();

               $('body').on('click', '#insert', function(){
                     $("#insert_staff_modal").modal();
                     $('#insert_staff_modal #name').parent().removeClass('has-error');
                     $('#insert_staff_modal #type').parent().removeClass('has-error');
               });

               @if (Session::has('error') && Session::get('error') == 'insert')
                     $("#insert_staff_modal").modal();
               @endif


               $('body').on('click', '.modify', function(){

                     staff_id = $(this).attr('data-id');

                     $('#modify_staff_modal #name').parent().removeClass('has-error');
                     $('#modify_staff_modal #type').parent().removeClass('has-error');

                     $.ajax({
                  type: "POST",
                  url: "{{URL::to('staff/get/by/id')}}",
                  data: "&staff_id="+staff_id,
                  contentType: "application/x-www-form-urlencoded;charset=UTF-8",
                  success: function(response){
                    if (response.success === true)
                    {
                      $('#modify_staff_modal #name').val(response.staff_name);
                      $('#modify_staff_modal #type').val(response.staff_type);
                      $('#modify_staff_modal #staff_id').val(staff_id);
                      $("#modify_staff_modal").modal();
                    }
                     }
               });
                     return false;
               });
```

```
@if (Session::has('error') && Session::get('error') == 'modify')
        $("#modify_staff_modal").modal();
@endif

$('body').on('click', '.delete', function(){

        staff_id = $(this).attr('data-id');

        $.ajax({
  type: "POST",
  url: "{{URL::to('staff/get/by/id')}}",
  data: "&staff_id="+staff_id,
  contentType: "application/x-www-form-urlencoded;charset=UTF-8",
  success: function(response){
   if (response.success === true)
   {
    $('#delete_staff_modal #staff_name').html(response.staff_name);
    $('#delete_staff_modal #staff_id').val(staff_id);
    $("#delete_staff_modal").modal();
   }
        }
});
        return false;
  });


$('body').on('click', '#delete_staff_button', function(){

        detete_staff_id = $("#delete_staff_modal").find("#staff_id").val();

        $.ajax({
  type: "POST",
  url: "{{URL::to('staff/delete')}}",
  data: "&detete_staff_id="+detete_staff_id,
  contentType: "application/x-www-form-urlencoded;charset=UTF-8",
  success: function(response){
   if (response.success === true)
   {
        $('tr#staff_'+detete_staff_id).remove();
    $("#delete_staff_modal").modal('hide');
    $("#succes_modal").modal();
   } else {
        $("#delete_staff_modal").modal('hide');
        $("#delete_error_modal #message").html(response.message);
        $("#delete_error_modal").modal();
```

```
                    }
                        }
                });
                $('#delete_staff_modal #staff_id').val('');
                        return false;
                    });

                    @if (Session::has('success') && Session::get('success'))
                        $("#succes_modal").modal();
                    @endif
                });
        </script>
    @stop
```

Συνάρτηση staffListCreate του StaffController για την προβολή της λίστας του προσωπικού:

```
public function staffListCreate()
 {
   $all_staff = Staff::all();
   $data      = array ( 'title'=> 'Προσωπικό' , 'all_staff' => $all_staff );
   $this->layout->content = View::make('admin.staff.list')->with($data);
 }
```

Route staff/list για την εκτέλεση της staffListCreate:

```
Route::get('staff/list', array('as' => 'staff_list', 'uses' => 'StaffController@staffListCreate'));
```

Συνάρτηση staffInsertStore του StaffController για την εισαγωγή νέου προσωπικού:

```
public function staffInsertStore()
 {
   $rules = array(
    'name'       => array('required'),
    'type'       => array('required')
   );

   $validation = Validator::make(Input::all(), $rules);

   if ($validation->fails())
   {
     return                    Redirect::to('staff/list')->withErrors($validation)->withInput()-
>with(array('error'=>'insert'));
   }
   else
    {
     $staff            = new Staff();
```

```
        $staff->name       = Input::get('name');
        $staff->type       = Input::get('type');
        $staff->created_at = date("Y-m-d H:i:s");
        $staff->updated_at = date("Y-m-d H:i:s");
        $staff->save();

        return Redirect::to('/staff/list')->with(array('success'=> true));
    }
  }
```

Route staff/insert για την εκτέλεση της staffInsertStore:

```
Route::post('staff/insert', array('as' => 'staff_list','uses' => 'StaffController@staffInsertStore'));
```

Συνάρτηση staffModifyStore του StaffController για την επεξεργασία του προσωπικού:

```
public function staffModifyStore()
  {
    $rules = array(
      'name'     => array('required'),
      'type'     => array('required'),
      'staff_id' => array('required')
    );

    $validation = Validator::make(Input::all(), $rules);

    if ($validation->fails())
    {
      return                    Redirect::to('staff/list')->withErrors($validation)->withInput()-
>with(array('error'=>'modify'));
    }
    else
     {
      $staff            = Staff::find(Input::get('staff_id'));
      $staff->name      = Input::get('name');
      $staff->type      = Input::get('type');
      $staff->created_at = date("Y-m-d H:i:s");
      $staff->updated_at = date("Y-m-d H:i:s");
      $staff->save();

      return Redirect::to('/staff/list')->with(array('success'=> true));
    }
  }
```

Route staff/modify για την εκτέλεση της staffModifyStore:

```
Route::post('staff/modify', array('as' => 'staff_list','uses' => 'StaffController@staffModifyStore'));
```

Συνάρτηση staffGetById του StaffController για την προβολή του προσωπικού προς επεξεργασία:

```
public function staffGetById()
 {
   $rules = array(
      'staff_id' => array('required')
   );

   $validation = Validator::make(Input::all(), $rules);

   if ($validation->fails())
   {
      return Response::json(array('success' => false));
   }
   else
   {
     $staff = Staff::find(Input::get('staff_id'));
     return Response::json(array(  'success' => true, 'staff_name' => $staff->name, 'staff_type' =>
$staff->type ));
   }

 }
```

Route staff/get/by/id για την εκτέλεση της staffGetById:

```
Route::post('staff/get/by/id', array('uses' => 'StaffController@staffGetById'));
```

Συνάρτηση staffDeleteStore του StaffController για την διαγραφή του προσωπικού:

```
public function staffDeleteStore()
 {
   $rules = array(
      'detete_staff_id' => array('required')
   );

   $validation = Validator::make(Input::all(), $rules);

   if ($validation->fails())
   {
      return Response::json(array('success' => false));
   }
   else
   {
     $staff = Staff::find(Input::get('detete_staff_id'));

     $fires_count = $staff->fires()->count();
```

```
    if ($fires_count > 0)
      return Response::json(array(  'success' => false, 'message' => 'Το συγκεκριμένο προσωπικό έχει ήδη καταχωρυθεί σε κάποια πυρκαγιά και επομένως δεν μπορεί να διαγραφεί'));
    else {
      $staff->delete();
      return Response::json(array(  'success' => true));
    }
   }
  }
```

Route staff/delete για την εκτέλεση της staffDeleteStore:

```
Route::post('staff/delete', array('uses' => 'StaffController@staffDeleteStore'));
```

View regions/list.blade.php για την προβολή της λίστας των περιφερειών:

```
@section('title')
   {{$title}}
@stop

@section('content')
    <div class="row">
        <div class="col-xs-12">
            <h1>Περιφέρειες</h1>
            <p class="lead">Λίστα περιφερειών</p>
        </div>
    </div>
    <div class="row topmargin">
        <div class="col-xs-12">
            <table class="table table-striped table-hover ">
                <thead>
                    <tr>
                        <th>Όνομα</th>
                    </tr>
                </thead>
                <tbody>
                    @foreach ($regions as $region)
                    <tr>
                        <td>{{$region->name}}</td>
                    </tr>
                    @endforeach
                </tbody>
            </table>
        </div>
    </div>
@stop
```

Συνάρτηση regionsListCreate του RegionController για την λίστα των περιφερειών:

```
public function regionsListCreate()
{
    $regions = Region::all();
    $data    = array ( 'title'=> 'Περιφέρειες' , 'regions' => $regions );
  $this->layout->content = View::make('admin.regions.list')->with($data);
}
```

Route regions/list για την εκτέλεση της regionsListCreate:

```
Route::get('regions/list', array('as' => 'regions_list', 'uses' => 'RegionController@regionsListCreate'));
```

View prefectures/list.blade.php για την προβολή της λίστας των νομών:

```
@section('title')
  {{$title}}
@stop

@section('extraStyle')
  {{HTML::style('css/datatables/dataTables.bootstrap.css')}}
  {{HTML::style('css/bootstrap-select.min.css')}}
@stop

@section('content')
    <div class="row">
        <div class="col-xs-12">
            <h1>Νομοί</h1>
            <p class="lead">Λίστα νομών</p>
        </div>
    </div>
    <div class="row topmargin">
        <div class="col-xs-12">
            <table class="table table-striped table-hover" id="prefectures">
                <thead>
                    <tr>
                        <th>Όνομα</th>

                        <th>Περιφέρεια</th>
                    </tr>
                </thead>
                <tbody>
                    @foreach ($prefectures as $prefecture)
                    <tr>
                        <td>{{$prefecture->name}}</td>
```

```
                                        <td>{{$prefecture->region()->first()->name}}</td>
                                    </tr>
                                    @endforeach
                                </tbody>
                            </table>
                        </div>
                    </div>
                @stop



                @section('extraScript')
                    {{HTML::script('js/datatables/jquery.dataTables.min.js')}}
                    {{HTML::script('js/datatables/dataTables.bootstrap.min.js')}}
                    {{HTML::script('js/bootstrap-select.min.js')}}
                    <script type="text/javascript">
                        $(function() {
                            $('#prefectures').DataTable({
                                "language": {
                        "url": "{{URL::to('lang/greek.json')}}"
                        },
                        "lengthMenu": [ [10, 25, 50, -1], [10, 25, 50, "Όλοι"] ],
                        "fnInitComplete": function(oSettings, json) {
                                    var datatable = this;
                                    var search_input = datatable.closest('.dataTables_wrapper').find('div[id$=_filter] input');
                                    search_input.attr('placeholder', (oSettings && oSettings.oLanguage && oSettings.oLanguage.sSearchPlaceholder) ? oSettings.oLanguage.sSearchPlaceholder : 'Αναζήτηση');
                                    var length_sel = datatable.closest('.dataTables_wrapper').find('div[id$=_length] select');
                                    length_sel.selectpicker().selectpicker('setStyle', 'btn-sm', 'add');
                                },
                            });
                        });
                    </script>
                @stop
```

Συνάρτηση prefecturesListCreate του PrefectureController για την λίστα των νομών:

```
public function prefecturesListCreate()
 {
    $prefectures = Prefecture::all();
    $data       = array ( 'title'=> 'Νομοί' , 'prefectures' => $prefectures );
  $this->layout->content = View::make('admin.prefectures.list')->with($data);
 }
```

Route prefectures/list για την εκτέλεση της prefecturesListCreate:

```
Route::get('prefectures/list', array('as' => 'prefectures_list', 'uses' => 'PrefectureController@prefecturesListCreate'));
```

View departments/list.blade.php για την προβολή της λίστας των πυροσβεστικών υπηρεσιών:

```
@section('title')
  {{$title}}
@stop

@section('extraStyle')
  {{HTML::style('css/datatables/dataTables.bootstrap.css')}}
  {{HTML::style('css/bootstrap-select.min.css')}}
@stop

@section('content')

    <div class="row">
        <div class="col-xs-12">
            <h1>Πυροσβεστικές Υπηρεσίες</h1>
            <p class="lead">Λίστα πυροσβεστικών υπηρεσιών</p>
        </div>
    </div>
    <div class="row topmargin">
        <div class="col-xs-12">
            <table class="table table-striped table-hover" id="fire_departments">
                <thead>
                    <tr>
                        <th>Όνομα</th>

                        <th>Περιφέρεια</th>
                    </tr>
                </thead>
                <tbody>
                    @foreach ($fire_departments as $fire_department)
                    <tr>
                        <td>{{$fire_department->name}}</td>
                        <td>{{$fire_department->region()->first()->name}}</td>
                    </tr>
                    @endforeach
                </tbody>
            </table>
        </div>
```

```
        </div>
    @stop



    @section('extraScript')
        {{HTML::script('js/datatables/jquery.dataTables.min.js')}}
        {{HTML::script('js/datatables/dataTables.bootstrap.min.js')}}
        {{HTML::script('js/bootstrap-select.min.js')}}
        <script type="text/javascript">
                $(function() {
                        $('#fire_departments').DataTable({
                                "language": {
                "url": "{{URL::to('lang/greek.json')}}"
                },
                "lengthMenu": [ [10, 25, 50, -1], [10, 25, 50, "Όλες"] ],
                "fnInitComplete": function(oSettings, json) {
                                        var datatable = this;
                                        var search_input = datatable.closest('.dataTables_wrapper').find('div[id$=_filter] input');
                                        search_input.attr('placeholder', (oSettings && oSettings.oLanguage && oSettings.oLanguage.sSearchPlaceholder) ? oSettings.oLanguage.sSearchPlaceholder : 'Αναζήτηση');
                                        var length_sel = datatable.closest('.dataTables_wrapper').find('div[id$=_length] select');
                                        length_sel.selectpicker().selectpicker('setStyle', 'btn-sm', 'add');
                                    },
                        });
                });
        </script>
    @stop
```

Συνάρτηση fireDepartmentsListCreate του FireDepartmentController για την λίστα των πυροσβεστικών υπηρεσιών:

```
    public function fireDepartmentsListCreate()
     {
        $fire_departments = FireDepartment::all();
        $data = array ( 'title'=> 'Πυροσβεστικές Υπηρεσίες' , 'fire_departments' => $fire_departments );
       $this->layout->content = View::make('admin.departments.list')->with($data);
     }
```

Route fire/departments/list για την εκτέλεση της fireDepartmentsListCreate:

```
    Route::get('fire/departments/list', array('as' => 'fire_departments_list', 'uses' => 'FireDepartmentController@fireDepartmentsListCreate'));
```

View municipalities/list.blade.php για την προβολή της λίστας των δήμων:

```
@section('title')
   {{$title}}
@stop

@section('extraStyle')
   {{HTML::style('css/datatables/dataTables.bootstrap.css')}}
   {{HTML::style('css/bootstrap-select.min.css')}}
@stop

@section('content')
        <div class="row">
                <div class="col-xs-12">
                        <h1>Δήμοι</h1>
                        <p class="lead">Λίστα δήμων</p>
                </div>
        </div>

        {{ Form::open(array('url' => 'municipalities/export',  'method' => 'post')) }}
        <div class="row">
                <div class="col-xs-12">
                        <button class="btn btn-info pull-right" type="submit"><span><i class="fa fa-
download"></i></span> Εξαγωγή σε μορφή CSV</button>
                </div>
        </div>
        {{Form::close()}}


        <div class="row topmargin">
                <div class="col-xs-12">
                        <table class="table table-striped table-hover" id="municipalities">
                                <thead>
                                        <tr>
                                                <th>Όνομα</th>
                                                <th>Νομός</th>
                                                <th>Περιφέρεια</th>
                                        </tr>
                                </thead>
                                <tbody>
                                        @foreach ($municipalities as $municipality)
                                        <tr>
                                                <td>{{$municipality->name}}</td>
                                                <td>{{$municipality->prefecture()->first()-
>name}}</td>
```

```
                                        <td>{{$municipality->prefecture()->first()->region()->first()->name}}</td>
                                    </tr>
                                    @endforeach
                                </tbody>
                            </table>
                        </div>
                    </div>
                @stop



                @section('extraScript')
                    {{HTML::script('js/datatables/jquery.dataTables.min.js')}}
                    {{HTML::script('js/datatables/dataTables.bootstrap.min.js')}}
                    {{HTML::script('js/bootstrap-select.min.js')}}
                    <script type="text/javascript">
                        $(function() {
                            $('#municipalities').DataTable({
                                "language": {
                        "url": "{{URL::to('lang/greek.json')}}"
                        },
                        "lengthMenu": [ [10, 25, 50, -1], [10, 25, 50, "Όλοι"] ],
                        "fnInitComplete": function(oSettings, json) {
                                    var datatable = this;
                                    var search_input = datatable.closest('.dataTables_wrapper').find('div[id$=_filter] input');
                                    search_input.attr('placeholder', (oSettings && oSettings.oLanguage && oSettings.oLanguage.sSearchPlaceholder) ? oSettings.oLanguage.sSearchPlaceholder : 'Αναζήτηση');
                                    var length_sel = datatable.closest('.dataTables_wrapper').find('div[id$=_length] select');
                                    length_sel.selectpicker().selectpicker('setStyle', 'btn-sm', 'add');
                                },
                            });
                        });
                    </script>
                @stop
```

Συνάρτηση municipalitiesListCreate του MunicipalityController για την λίστα των δήμων:

```
 public function municipalitiesListCreate()
 {
     $municipalities = Municipality::all();
     $data           = array ( 'title'=> 'Δήμοι' , 'municipalities' => $municipalities );
     $this->layout->content = View::make('admin.municipalities.list')->with($data);
 }
```

Route municipalities/list για την εκτέλεση της municipalitiesListCreate:

```
Route::get('municipalities/list', array('as' => 'municipalities_list', 'uses' => 'MunicipalityController@municipalitiesListCreate'));
```

Συνάρτηση municipalitiesExport του MunicipalityController για την εξαγωγή της λίστας των δήμων σε μορφή CSV:

```
public function municipalitiesExport()
 {

   $municipalities = Municipality::all();

   $output = iconv("utf-8", "iso-8859-7", "Δήμος;Νομός;Περιφέρεια")."\n\n";

    if (!empty($municipalities)) {

      foreach ($municipalities as $municipality)
      {
        $output .= iconv("utf-8", "iso-8859-7", $municipality->name .';');
        $output .= iconv("utf-8", "iso-8859-7", $municipality->prefecture->name .';');
        $output .= iconv("utf-8", "iso-8859-7", $municipality->prefecture()->first()->region()->first()->name .';');

        $output .= ";";
        $output .= "\n";
      }
    }

    $headers = array(
      'Content-Type' => 'application/vnd.ms-excel; charset=UTF-8',
      'Content-Disposition' => 'attachment; filename="municipalities.csv"',
    );

    return Response::make(rtrim($output, "\n"), 200, $headers);

 }
```

Route municipalities/export για την εκτέλεση της municipalitiesExport:

```
Route::post('municipalities/export', array('uses' => 'MunicipalityController@municipalitiesExport'));
```

Συνάρτηση logoutCreate του AdminController για την αποσύνδεση του διαχειριστή από το Back End κομμάτι:

```
public function logoutCreate()
{
      Auth::logout();
```

```
        Session::flush();
        return Redirect::to('/admin');
    }
```

Route logout του AdminController για την εκτέλεση της logoutCreate:

```
Route::get('/logout', array('as' => 'logout', 'uses' => 'AdminController@logoutCreate'));
```

FRONT END

View index.blade.php (Η main σελίδα του Front End):

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <title>Greek Fires</title>

    {{HTML::style('css/bootstrap.min.css')}}
    {{HTML::style('css/font-awesome.min.css')}}
    {{HTML::style('css/bootstrap-select.min.css')}}
    {{HTML::style('css/scrolling-nav.css')}}
    {{HTML::style('css/fires.css')}}

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
      <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->

  </head>

  <body data-spy="scroll" data-target="#navbar-fires">


    <nav class="navbar navbar-default navbar-fixed-top affix-top" id="navbar-fires">
      <div class="container">
        <div class="navbar-header">
          <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-
target="#navbar" aria-expanded="false" aria-controls="navbar">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
```

```
          <span class="icon-bar"></span>
         </button>
         <a class="navbar-brand page-scroll" href="#intro">GreekFires</a>
        </div>
        <div id="navbar" class="navbar-collapse collapse">

         <ul class="nav navbar-nav navbar-right">
          <li><a href="#mapsection" class="page-scroll">Χάρτης</a></li>
          <li><a href="#statistics" class="page-scroll">Στατιστικά</a></li>
          <li><a href="#contact"    class="page-scroll">Επικοινωνία</a></li>
         </ul>
        </div>
       </div>
      </nav>


    <section id="intro" class="intro-section">
      <div id="intro-text">
       <div class="container">
        <div class="row">
         <div class="col-xs-12">
          <h1 class="no-top-margin">GreekFires</h1>
          <p>Το site που παρουσιάζονται χρονικά πάνω σε χάρτη οι πυρκαγιές της Ελλάδος</p>
          <a href="#mapsection" class="btn btn-primary btn-intro page-scroll topmargin">Περισσότερα</a>
         </div>
        </div>
       </div>
      </div>
    </section>

    <section id="mapsection" class="mapsection-section">
     <div class="container-fluid nopadding">

      <div class="row-fluid">
       <div class="col-xs-12">
        <div class="form-group">
         <label> Επιλέξτε νομό:</label>
         <select id="tag_prefectures" class="selectpicker">
          <option value="">Όλοι οι νομοί</option>
          @foreach ($prefectures as $prefecture)
           <option value="{{$prefecture->id}}">{{$prefecture->name}}</option>
          @endforeach
         </select>
        </div>
```

```
        </div>
      </div>

      <div class="row-fluid">
        <div class="col-xs-12 nopadding">
          <div id="timemap" class="nopadding">
            <div id="timelinecontainer">
              <div id="timeline"></div>
            </div>

            <div id="mapcontainer">
              <div id="map"></div>

              <div id="show-no-results-message">
                <div class="alert alert-danger " role="alert">
                  Δεν υπάρχουνε πυρκαγιές σε αυτόν τον νομό!
                </div>
              </div>

              <div id="no-results"></div>

              <div id="map_loading">
                <div id="map_loading_container">
                  <img src="{{URL::to('images/ajax-loader-black.gif')}}" alt="" />
                  <h4>Φόρτωση πυρκαγιών...</h4>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </section>


  <section id="statistics" class="statistics-section">
    <div class="container">
      <div class="row topmargin">

        <div class="col-xs-12 col-md-4">
          <div class="panel panel-default" id="panel_1">
            <div class="panel-heading">
              <h5 class="panel-title text-center">10 μεγαλύτερες καμμένες περιοχές</h5>
            </div>
            <div class="panel-body">
              <div class="row">
```

```
                <div class="col-xs-12">
                  <ul class="news">
                    @foreach ($biggest_burnt_areas as $biggest_burnt_area)
                    <li class="news-item informations">
                      <p class="nomargin text-right"><em>Στρέμματα :</em> <span class="orange">{{$biggest_burnt_area->count_burnt_area}}</span></p>
                      <p class="nomargin"><em>Νομός :</em> {{$biggest_burnt_area->prefecture_name}}</p>
                      <p class="nomargin"><em>Δήμος :</em> {{$biggest_burnt_area->municipality_name}}</p>
                      <p class="nomargin"><em>Ημ/νία :</em> {{$biggest_burnt_area->transformDateFromDataBase($biggest_burnt_area->start_date)}} - {{$biggest_burnt_area->transformDateFromDataBase($biggest_burnt_area->end_date)}}</p>
                      <a href="#" data-id="{{$biggest_burnt_area->id}}">Περισσότερα +</a>
                    </li>
                    @endforeach
                  </ul>
                </div>
              </div>
            </div>
          </div>
        </div>


        <div class="col-xs-12 col-md-4">
          <div class="panel panel-default" id="panel_2">
            <div class="panel-heading">
              <h5 class="panel-title text-center">10 μεγαλύτερες χρονικά πυρκαγιές</h5>
            </div>
            <div class="panel-body">
              <div class="row">
                <div class="col-xs-12">
                  <ul class="news">
                    @foreach ($biggest_fires_by_date as $biggest_fire_by_date)
                    <?php
                      $date_start = date_create($biggest_fire_by_date->start_date);
                      $date_end   = date_create($biggest_fire_by_date->end_date);
                      $diff = date_diff($date_start,$date_end);
                    ?>
                    <li class="news-item informations">
                      <p class="nomargin text-right"><em>Διάρκεια :</em> <span class="orange">{{$diff->format("%a Ημέρες")}}</span></p>
                      <p class="nomargin"><em>Νομός :</em> {{$biggest_fire_by_date->prefecture_name}}</p>
                      <p class="nomargin"><em>Δήμος :</em> {{$biggest_fire_by_date->municipality_name}}</p>
```

```
                    <p class="nomargin"><em>Ημ/νία :</em> {{$biggest_fire_by_date->transformDateFromDataBase($biggest_fire_by_date->start_date)}} - {{$biggest_fire_by_date->transformDateFromDataBase($biggest_fire_by_date->end_date)}}</p>
                    <a href="#" data-id="{{$biggest_fire_by_date->id}}">Περισσότερα +</a>
                  </li>
                  @endforeach
                </ul>
              </div>
            </div>
          </div>
        </div>


        <div class="col-xs-12 col-md-4">
          <div class="panel panel-default" id="panel_3">
            <div class="panel-heading">
              <h5 class="panel-title text-center">10 τελευταίες πυρκαγιές</h5>
            </div>
            <div class="panel-body">
              <div class="row">
                <div class="col-xs-12">
                  <ul class="news">
                    @foreach ($latest_fires as $latest_fire)
                    <?php
                       $date_start = date_create($biggest_fire_by_date->start_date);
                       $date_end   = date_create($biggest_fire_by_date->end_date);
                       $diff = date_diff($date_start,$date_end);
                    ?>
                    <li class="news-item informations">
                      <p class="nomargin text-right"><em>Ημ/νία Εισαγωγής :</em> <span class="orange"><?php $date_added = explode(" ",$latest_fire->transformDateFromDataBase($latest_fire->created_at)) ?> {{$date_added[0]}}</span></p>
                      <p class="nomargin"><em>Νομός :</em> {{$latest_fire->prefecture->name}}</p>
                      <p class="nomargin"><em>Δήμος :</em> {{$latest_fire->municipality->name}}</p>
                      <p class="nomargin"><em>Ημ/νία :</em> {{$latest_fire->transformDateFromDataBase($biggest_fire_by_date->start_date)}} - {{$latest_fire->transformDateFromDataBase($biggest_fire_by_date->end_date)}}</p>
                      <a href="#" data-id="{{$latest_fire->id}}">Περισσότερα +</a>
                    </li>
                    @endforeach
                  </ul>
                </div>
              </div>
            </div>
```

```
          </div>
        </div>

    </div>
  </div>

  <div class="container">
    <div class="row topmargin-50">
      <div class="col-xs-12">
        <h4 class="orange text-center">Κατανομή συνόλου πυρκαγιών ανά μήνα</h4>
      </div>
    </div>
    <div class="row topmargin">
      <div class="col-xs-12">
        <div id="chart-months-bar"></div>
      </div>
    </div>
  </div>

  <div class="container">
    <div class="row topmargin">
      <div class="col-xs-12 nopadding">
        <canvas id="canvas-months-bar" height="150" width="400"></canvas>
      </div>
    </div>
  </div>

  <div class="container charts">
    <div class="row topmargin-50">
      <div class="col-xs-12 col-md-6 nopadding canvas-pie">
        <h4 class="orange text-center">Σύνολο πυρκαγιών ανά νομό</h4>
        <canvas id="chart-prefectures-doughnut"></canvas>
        <div id="info-prefectures-doughnut"></div>
      </div>
      <div class="col-xs-12 col-md-6 nopadding canvas-pie">
        <h4 class="orange text-center">Σύνολο πυρκαγιών ανά περιφέρεια</h4>
        <canvas id="chart-regions-pie"></canvas>
        <div id="info-regions-pie"></div>
      </div>
    </div>
  </div>
</section>

<section id="contact" class="contact-section">
  <div class="container">
    <div class="row">
```

```
            <div class="col-md-3 col-xs-12"></div>
            <div class="col-xs-12 col-md-6">
              <div class="row">
                <div class="col-xs-12">
                  <p class="orange"><small><em>Μπορείτε να επικοινωνήσετε μαζί μας συμπληρώνοντας την παρακάτω φόρμα: </em></small></p>
                </div>
              </div>
              {{ Form:: open(array('url' => 'contact', 'id' => 'contact_form')) }}
                <div class="form-group">
                  {{ Form:: text ('first_name', '', array('class' => 'form-control' , 'placeholder' => 'Όνομα', 'id' => 'first_name') )}}
                </div>
                <div class="form-group">
                  {{ Form:: text ('last_name', '', array('class' => 'form-control' , 'placeholder' => 'Επίθετο' , 'id' => 'last_name') )}}
                </div>
                <div class="form-group">
                  {{ Form:: email ('email', '', array('class' => 'form-control', 'placeholder' => 'me@example.com' , 'id' => 'email')) }}
                </div>
                <div class="form-group">
                  {{ Form:: textarea ('message', '', array('class' => 'form-control', 'placeholder' => 'Το μήνυμά σας...', 'size' => '30x8' , 'id' => 'message'))}}
                </div>
                <div class="form-group text-center">
                  {{ Form::submit('Αποστολή Μηνύματος', array('class' => 'btn btn-success', 'id' => 'contact-submit')) }}
                </div>
              {{ Form:: close() }}
            </div>
            <div class="col-md-3 col-xs-12"></div>
          </div>
        </div>

        <div class="container-fluid" id="footer">
          <div class="row">
            <div class="col-xs-12">
              <div class="container">
                <div class="row">
                  <div class="col-xs-6">
                    <p class="topmargin"><a href="mailto:gtzalavras@gmail.com">&copy; G.Tzalavras - 2015</a></p>
                  </div>
                  <div class="col-xs-6 text-right">
                    <p class="topmargin"><a href="#"><i class="fa fa-info-circle"></i> Πληροφορίες</a></p>
```

```
            </div>
          </div>
        </div>
      </div>
    </div>
  </section>


  <!-- MODALS -->
   <div class="modal fade" id="informations_modal">
     <div class="modal-dialog modal-lg">
       <div class="modal-content">
        <div class="modal-header noborder">
          <button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>
          <h4 class="modal-title text-center orange">Πληροφορίες πυρκαγιάς</h4>
        </div>
        <div class="modal-body text-center">
          <div class="row">
            <div class="col-xs-12">
              <div id="informations">

              </div>
            </div>
          </div>
        </div>
        <div class="modal-footer">
          <button type="button" class="btn btn-success" data-dismiss="modal"> Κλείσιμο
παραθύρου</button>
        </div>
      </div>
    </div>
  </div>


  <div class="modal fade" id="succes_modal">
  <div class="modal-dialog">
    <div class="modal-content">
     <div class="modal-header noborder">
       <h4 class="modal-title text-center text-success"><big><span class="glyphicon glyphicon-
ok"></span></big></h4>
      </div>
      <div class="modal-body text-center">
       <p>Το μήνυμά σας στάλθηκε με επιτυχία!</p>
      </div>
```

```
        <div class="modal-footer noborder text-center">
          <button type="button" class="btn btn-success" data-dismiss="modal">Κλείσιμο παραθύρου</button>
        </div>
      </div>
    </div>
  </div>


  {{HTML::script('js/jquery-1.11.2.min.js')}}
  {{HTML::script('http://maps.google.com/maps/api/js?sensor=false')}}
  {{HTML::script('js/bootstrap.min.js')}}
  {{HTML::script('js/jquery.easing.min.js')}}
  {{HTML::script('js/scrolling-nav.js')}}
  {{HTML::script('js/bootstrap-select.min.js')}}
  {{HTML::script('js/chart.min.js')}}
  {{HTML::script('js/jquery.bootstrap.newsbox.min.js')}}
  {{HTML::script('js/jquery.inview.min.js')}}
  {{HTML::script('js/jquery.blockUI.js')}}

  <map>
  {{HTML::script('js/timemap/lib/timeline-2.3.0.js')}}
  </map>
  {{HTML::script('js/timemap/lib/mxn/mxn.js?(googlev3)')}}
  {{HTML::script('js/timemap/src/timemap.js')}}
  {{HTML::script('js/timemap/src/manipulation.js')}}
  {{HTML::script('js/timemap/src/json.js')}}

  <script type="text/javascript">

    var tm = TimeMap.init({

      mapId: "map",               // Id of map div element (required)
      timelineId: "timeline",     // Id of timeline div element (required)
      options: {
        centerOnItems: false,
        mapZoom: 6,
        mapCenter: new mxn.LatLonPoint(39.074208, 21.824312),
      },
      dataDisplayedFunction: function () {
        $('#mapcontainer').unblock();
      },
      datasets: [
        {
          id : "map_fires",
          title: "JSONP Dataset",
```

```
            type: "jsonp",
            theme: "orange",
            options: {
               url: "{{ URL::to('progsvc.php') }}?callback=?",    // Must be a local URL
               iconSize :[25,30],        //mege8os gia ton pointer (marker)
               icon: "images/flame.png",
            }
         }
      ],
      bandInfo: [
         {
           width:          "30%",
           intervalUnit:   Timeline.DateTime.HOUR,
           intervalPixels: 100
         },
         {
           width:          "50%",
           intervalUnit:   Timeline.DateTime.DAY,
           intervalPixels: 100
        },
        {
          width:          "20%",
          intervalUnit:   Timeline.DateTime.YEAR,
          intervalPixels: 150,
        }
      ]
   });


   $('body').on('change', '#tag_prefectures', function(){

      window.selectedTag = $(this).val();
      // run filters
      tm.filter('map');
      tm.filter('timeline');

      var d;

      id_prefecture = $(this).val();

      $.ajax({
         type: "POST",
         url: "{{URL::to('get/smallest/fire/date/by/prefecture/id')}}",
         data: "&id_prefecture="+id_prefecture,
         contentType: "application/x-www-form-urlencoded;charset=UTF-8",
         success: function(response){
```

```
            if (response.success === true)
            {
              if (response.no_result === false) {
                 $("#no-results").fadeIn();
                 setTimeout(function(){$("#no-results").fadeOut(500);}, 3000);
                 $("#show-no-results-message .alert-danger").fadeIn('slow');
                 setTimeout(function(){$(".alert-danger").fadeOut(500);}, 3000);
                 tm.refreshTimeline();
              } else
                 d = response.start;
                 lazyLayout = true;
                 animated = true;
                 //tm.scrollToDate(d, lazyLayout, animated) ();
              }
            }

        });
    });

    $('body').on('click', '.informations a', function(e){
        id_fire = $(this).attr('data-id');
        $.ajax({
            type: "POST",
            url: "{{URL::to('get/fire/informations/by/id')}}",
            data: "&id_fire="+id_fire,
            contentType: "application/x-www-form-urlencoded;charset=UTF-8",
            success: function(response){
              if (response.success === true)
              {
                $("#informations").html(response.informations);
                $("#informations_modal").modal();
              }
            }
        });
        e.preventDefault();
    });



    var date_month = [{{ Fire::getNumberFireByMonth('01', '2014') }},{{
Fire::getNumberFireByMonth('02', '2014') }},{{ Fire::getNumberFireByMonth('03', '2014') }},{{
Fire::getNumberFireByMonth('04', '2014') }},{{ Fire::getNumberFireByMonth('05', '2014') }},{{
Fire::getNumberFireByMonth('06', '2014') }},{{ Fire::getNumberFireByMonth('07', '2014') }},{{
Fire::getNumberFireByMonth('08', '2014') }},{{ Fire::getNumberFireByMonth('09', '2014') }},{{
Fire::getNumberFireByMonth('10', '2014') }},{{ Fire::getNumberFireByMonth('11', '2014') }},{{
Fire::getNumberFireByMonth('12', '2014') }}];
```

```
        var colors = ["#C80000", "#FF4500", "#FFD700", "#0000FF", "#8A2BE2", "#EE82EE",
"#C71585" , "#FFA500", "#FFFF00", "#ADFF2F", "#008000", "#2E8B57"];
        var highlights = ["rgba(200, 0, 0, 0.9)", "rgba(255, 69, 0, 0.9)", "rgba(255, 215, 0, 0.9)",
                    "rgba(0, 0, 255, 0.9)", "rgba(138, 43, 226, 0.9)", "rgba(238, 130, 238, 0.9)",
                    "rgba(199, 21, 133, 0.9)" , "rgba(255, 165, 0, 0.9)", "rgba(255, 255, 0, 0.9)",
                    "rgba(173, 255, 47, 0.9)", "rgba(0, 128, 0, 0.9)", "rgba(46, 139, 87, 0.9)"
                ];

        var color_indicator     = 0;
        var highlight_indicator = 0;

        var color_indicator_regions     = 0;
        var highlight_indicator_regions = 0;

        var pieData = [
                    @foreach ($fires_by_prefecture as $fire_by_prefecture)
                    {
                        value: {{$fire_by_prefecture->count_prefectures}},
                        color: colors[color_indicator++],
                        highlight: highlights[highlight_indicator++],
                        label: "{{$fire_by_prefecture->name}}"
                    },
                    @endforeach
                ];

        var pieData_regions = [
                            @foreach ($fires_by_regions as $fires_by_region)
                             {
                                value: {{$fires_by_region->count_fires}},
                                color: colors[color_indicator_regions++],
                                highlight: highlights[highlight_indicator_regions++],
                                label: "{{$fires_by_region->name}}"
                             },
                            @endforeach
                        ];

        @if (Session::has('success') && Session::get('success'))
          $("#succes_modal").modal();
        @endif

      </script>

      {{HTML::script('js/fires.js')}}

    </body>
  </html>
```

Συνάρτηση showHomePageCreate στον HomeController για την προβολή των βασικών πληροφοριών της σελίδας index:

```
public function showHomePageCreate()
{
        $prefectures = Prefecture::all();

        $biggest_burnt_areas = Fire::where('end_date', '<>', '0000-00-00 00:00:00')
->select(DB::raw(' forest +  woodland +  grove +  grass_land +  reed_marches +  farmland + cultivation_waste + dumping_ground as count_burnt_area, fires.*, prefectures.name as prefecture_name, municipalities.name as municipality_name'))
->join('burnt_area','burnt_area.id_fire', '=', 'fires.id')
->join('prefectures','prefectures.id', '=', 'fires.id_prefecture')
->join('municipalities','municipalities.id', '=', 'fires.id_municipality')
->orderBy('count_burnt_area','desc')
->take(10)
->get();

    $biggest_fires_by_date = Fire::where('end_date', '<>', '0000-00-00 00:00:00')
->select(DB::raw('TIMESTAMPDIFF(SECOND, start_date, end_date) as difference, fires.*, prefectures.name as prefecture_name, municipalities.name as municipality_name'))
->join('prefectures','prefectures.id', '=', 'fires.id_prefecture')
->join('municipalities','municipalities.id', '=', 'fires.id_municipality')
->orderBy('difference','desc')
->take(10)
->get();

    $latest_fires = Fire::select('fires.*')
    ->orderBy('start_date','desc')
        ->take(10)
        ->get();

        $fires_by_prefecture = Fire::select(DB::raw('prefectures.name, count(id_prefecture) as count_prefectures'))
->join('prefectures', 'prefectures.id', '=', 'fires.id_prefecture')
->groupBy('fires.id_prefecture')
->orderBy('count_prefectures', 'desc')
->take(12)
->get();

        $fires_by_regions = Region::select(DB::raw('regions.name, count(fires.id) as count_fires'))
->join('prefectures', 'prefectures.id_region', '=', 'regions.id')
->join('fires', 'fires.id_prefecture', '=', 'prefectures.id')
->groupBy('regions.id')
->orderBy('count_fires', 'desc')
```

```
->get();


            $data  = array(  'prefectures'          => $prefectures,
                             'biggest_burnt_areas'   => $biggest_burnt_areas,
                             'biggest_fires_by_date' => $biggest_fires_by_date,
                             'latest_fires'          => $latest_fires,
                             'fires_by_prefecture'   => $fires_by_prefecture,
                             'fires_by_regions'                 => $fires_by_regions);
            return View::make('index')->with($data);
    }
```

Route για την προβολή της index view του HomeController:

```
Route::get('/', array('uses' => 'HomeController@showHomePageCreate'));
```

Συνάρτηση getSmallestFireDateByPrefectureId για την προβολή των πυρκαγιών ανά νομό:

```
public function getSmallestFireDateByPrefectureId() {
        $rules = array(
           'id_prefecture' => array('required')
        );

        $validation = Validator::make(Input::all(), $rules);

        if ($validation->fails())
        {
          return Response::json(array('success' => false));
        }
        else
        {
                $fire  = Fire::where('id_prefecture', '=', Input::get('id_prefecture'))->first();
                $no_result = false;

                if (isset($fire)) {
                        $no_result = true;
                        $start = $fire->start_date;
                } else {
                        $start = '';
                }

            return Response::json(array(   'success' => true, 'start' => $start, 'no_result' =>
$no_result));
        }
    }
```

Route get/smallest/fire/date/by/prefecture/id για την εκτέλεση της getSmallestFireDateByPrefectureId:

```
Route::post('get/smallest/fire/date/by/prefecture/id', array('uses' => 'HomeController@getSmallestFireDateByPrefectureId'));
```

Συνάρτηση getFireInformationsById για την προβολή των πληροφοριών της πυρκαγιάς:

```
public function getFireInformationsById()
   {
        $rules = array(
           'id_fire' => array('required')
        );

       $validation = Validator::make(Input::all(), $rules);

        if ($validation->fails())
        {
           return Response::json(array('success' => false));
        }
        else
        {

                    $fire = Fire::find(Input::get('id_fire'));

                    $fire_department = $fire->fire_department()->first();
                    $prefecture      = $fire->prefecture()->first();
                    $municipality    = $fire->municipality()->first();
                    $burtn_area      = $fire->burnt_area()->first();
                    $fire_forces     = $fire->forces()->get();
                    $fire_staff      = $fire->staff()->get();


                $informations = "";

                  $informations .= "<div class='row'>";
                    $informations .= "<div class='col-xs-12'>";

                    $start_date = $fire->transformDateFromDataBase($fire->start_date);
                    if ($fire->end_date != "0000-00-00 00:00:00") {
                            $end_date = $fire->transformDateFromDataBase($fire->end_date);
                            $informations .= "<p class='text-right orange'>".$start_date." - ".$end_date."</p>";
                    }
                    else
                            $informations .= "<p class='text-right orange'>".$start_date."</p>";
```

```
                $informations .= "</div";
            $informations .= "</div";

            $informations .= "<div class='row'>";
              $informations .= "<div class='col-xs-12 nopadding'>";
                 $informations .= "<table class='table table-striped text-left'>";

                        $informations .= "<tr>";
                                $informations .="<td  class='orange'>Νομός";
                                $informations .="</td>";
                                $informations .="<td>".$prefecture->name."";
                                $informations .="</td>";
                        $informations .= "</tr>";

                        $informations .= "<tr>";
                                $informations .="<td class='orange'>Δήμος";
                                $informations .="</td>";
                                $informations .="<td>".$municipality->name."";
                                $informations .="</td>";
                        $informations .= "</tr>";

                        $informations .= "<tr>";
                                $informations    .="<td    class='orange'>Πυροσβεστική
υπηρεσία";
                                $informations .="</td>";
                                $informations .="<td>".$fire_department->name."";
                                $informations .="</td>";
                        $informations .= "</tr>";

                        $informations .= "<tr>";
                                $informations .="<td class='orange'>Δασαρχείο";
                                $informations .="</td>";
                                $informations .="<td>".$fire->forestry."";
                                $informations .="</td>";
                        $informations .= "</tr>";

                        $informations .= "<tr>";
                                $informations .="<td class='orange'>Περιοχή";
                                $informations .="</td>";
                                $informations .="<td>".$fire->area."";
                                $informations .="</td>";
                        $informations .= "</tr>";

                        $informations .= "<tr>";
                                $informations .="<td class='orange'>Διεύθυνση";
```

```
                                $informations .="</td>";
                                $informations .="<td>".$fire->adress."";
                                $informations .="</td>";
                            $informations .= "</tr>";

                        $informations .= "</table>";
                        $informations .= "</div";
                    $informations .= "</div";


                    $informations .= "<div class='row'>";
                     $informations .= "<div class='col-xs-12'>";
                      $informations .= "<h3>Καμμένη περιοχή</h3>";
                      $informations .= '<ul class="list-unstyled list-inline text-center">';
                       $informations .= '<li>';
                         $informations .= '<div class="form-group">';
                         $informations .= '<label for="forest">Δάση</label>';
                          $informations .= '<input type="text" disabled class="form-control" value="'.$burtn_area->forest.'"';
                         $informations .= '</div>';
                       $informations .= '</li>';
                       $informations .= '<li>';
                         $informations .= '<div class="form-group">';
                         $informations .= '<label for="woodland">Δασική Έκταση</label>';
                          $informations .= '<input type="text" disabled class="form-control" value="'.$burtn_area->woodland.'"';
                         $informations .= '</div>';
                       $informations .= '</li>';
                       $informations .= '<li>';
                         $informations .= '<div class="form-group">';
                         $informations .= '<label for="grove">Άλση</label>';
                          $informations .= '<input type="text" disabled class="form-control" value="'.$burtn_area->grove.'"';
                         $informations .= '</div>';
                       $informations .= '</li>';
                       $informations .= '<li>';
                         $informations .= '<div class="form-group">';
                         $informations .= '<label for="grass_land">Χορτ/κές Εκτάσεις</label>';
                          $informations .= '<input type="text" disabled class="form-control" value="'.$burtn_area->grass_land.'"';
                         $informations .= '</div>';
                       $informations .= '</li>';
                       $informations .= '<li>';
                         $informations .= '<div class="form-group">';
                         $informations .= '<label for="reed_marches">Καλάμια - Βάλτοι</label>';
                          $informations .= '<input type="text" disabled class="form-control" value="'.$burtn_area->reed_marches.'"';
```

```
                    $informations .= '</div>';
                $informations .= '</li>';
                $informations .= '<li>';
                    $informations .= '<div class="form-group">';
                    $informations .= '<label for="farmland">Γεωργικές Εκτάσεις</label>';
                     $informations .= '<input type="text" disabled class="form-control" value="'.$burtn_area->farmland.'"';
                    $informations .= '</div>';
                $informations .= '</li>';
                $informations .= '<li>';
                    $informations .= '<div class="form-group">';
                    $informations .= '<label for="cultivation_waste">Υπολλείματα Καλλιεργειών</label>';
                     $informations .= '<input type="text" disabled class="form-control" value="'.$burtn_area->cultivation_waste.'"';
                    $informations .= '</div>';
                $informations .= '</li>';
                $informations .= '<li>';
                    $informations .= '<div class="form-group">';
                    $informations .= '<label for="dumping_ground">Σκουπιδότοποι</label>';
                     $informations .= '<input type="text" disabled class="form-control" value="'.$burtn_area->dumping_ground.'"';
                    $informations .= '</div>';
                $informations .= '</li>';
              $informations .= '</ul>';
            $informations .= '</div>';
          $informations .= '</div>';


          $informations .= "<div class='row'>";
            $informations .= "<div class='col-xs-12'>";
              $informations .= "<h3>Προσωπικό</h3>";
              $informations .= '<ul class="list-unstyled list-inline text-center">';

                foreach ($fire_staff as $staff) {
                    $informations .= '<li>';
                        $informations .= '<div class="form-group">';
                        $informations .= '<label for="">'.$staff->name.'</label>';
                         $informations .= '<input type="text" disabled class="form-control" value="'.$staff->pivot->number.'"';
                        $informations .= '</div>';
                     $informations .= '</li>';
                }

              $informations .= '</ul>';
            $informations .= '</div>';
          $informations .= '</div>';
```

```
                    $informations .= "<div class='row'>";
                      $informations .= "<div class='col-xs-12'>";
                        $informations .= "<h3>Μέσα πυρόσβεσης</h3>";
                        $informations .= '<ul class="list-unstyled list-inline text-center">';

                          foreach ($fire_forces as $force) {
                              $informations .= '<li>';
                                 $informations .= '<div class="form-group">';
                                 $informations .= '<label for="">'.$force->name.'</label>';
                                  $informations .= '<input type="text" disabled class="form-control"
value="'.$force->pivot->number.'"';
                                 $informations .= '</div>';
                               $informations .= '</li>';
                          }

                        $informations .= '</ul>';
                      $informations .= '</div>';
                    $informations .= '</div>';


        return Response::json(array(  'success' => true, 'informations' => $informations ));

      }

     }
```

Route get/fire/informations/by/id για την εκτέλεση της getFireInformationsById:

```
Route::post('get/fire/informations/by/id', array('uses' => 'HomeController@getFireInformationsById'));
```

Συνάρτηση showFiresOnMapCreate για την προβολή πυρκαγιών επάνω στον χάρτη:

```
public function showFiresOnMapCreate ($callback = '') {

        $cb = $callback;

        $fires = Fire::all();

        $map =  "$cb([";

        $total_fires = count($fires);
        $i = 0;
        foreach ($fires as $fire) {

            $start_date  = explode(" ",$fire->start_date);
```

```
                        if ($fire->end_date != "0000-00-00 00:00:00")
                                $end_date    = explode(" ",$fire->end_date);

                        if ($i != $total_fires && $i != 0) $map .= ',';

                        $i++;
                        $map .= '{';
                        $map .= '"title":"N. '.$fire->prefecture->name.'",';

                        if ($fire->end_date != "0000-00-00 00:00:00")
                                $end_date_info                =                "<h6>".$fire-
>transformDateFromDataBase($fire->end_date)."</h6>";
                        else
                                $end_date_info = "";

                        $map .= '"options" : { tags: ['."".$fire->prefecture->id."".'] , infoHtml: "<div
class='."informations".'><h5 class='."nomargin".'>Δ. '.$fire->municipality->name.'</h5><h6>'.$fire-
>transformDateFromDataBase($fire->start_date).'</h6>'.$end_date_info.'<a href='."#".' data-
id='.$fire->id.'>Περισσότερα+</a></div>" },';
                        $map .= '"start":"'.$start_date[0]."T".$start_date[1].'+00:00",';

                        if ($fire->end_date != "0000-00-00 00:00:00")
                                $map .= '"end":"'.$end_date[0]."T".$end_date[1].'+00:00",';

                        $map .= '"point":{"lat":'.$fire->latitude.',"lon":'.$fire->longitude.'}';
                        $map .= '}';
                }

                $map .= "])";

                echo $map;

        }
```

Route fires/on/map/callback/{callback} για την εκτέλεση της showFiresOnMapCreate':

```
Route::get('fires/on/map/callback/{callback}', array('uses' =>
'HomeController@showFiresOnMapCreate'));
```

Συνάρτηση getContactStore για την υποβολή της φόρμας επικοινωνίας:

```
public function getContactStore(){

                $data = Input::all();

                $rules = array (
                        'first_name' => 'required|alpha',
```

```
                    'last_name'  => 'required|alpha',
                    'email'      => 'required|email',
                    'message'    => 'required'
                );

                $validator  = Validator::make ($data, $rules);

                if ($validator -> passes()) {
                    Mail::send('emails.mail', $data, function($message) use ($data)
                    {
                        $message->from($data['email']  ,  $data['first_name'].'
'.$data['last_name']);
                        $message->to('gtzalavras@gmail.com',  'my  name')-
>cc('gtzalavras@gmail.com')->subject('Contact Greek Fires');
                    });
                    return Redirect::to('/')->with(array('success' => true));
                } else {
                    return  Redirect::to('/')->withErrors($validator)->withInput()-
>with(array('success' => false));
                }
        }
```

Route contact για την εκτέλεση της getContactStore:

```
Route::post('contact', array('uses' => 'HomeController@getContactStore'));
```

View emails/email.blade.php για την απεικόνιση του απεσταλμένου email στον διαχειριστή:

```
<?php
    $first_name    = Input::get('first_name');
    $last_name     = Input::get ('last_name');
    $phone_number  = Input::get('phone_number');
    $email         = Input::get ('email');
    $message       = Input::get ('message');
    $date_time     = date("F j, Y, g:i a");
?>

<h3 style="font-family: 'Finger Paint', cursive; color: #f05f40;">GreekFires</h3>
<p>
Όνομα: {{$first_name}} <br>
Επίθετο: {{$last_name}} <br>
Email: {{$email}} <br><br>
Μήνυμα: {{$message}}  <br><br>
Ημερομηνία: {{$date_time}} <br>
</p>
```

Αρχείο fires.js για την εφαρμογή των jQuery plugins:

```
$(function(){

    $('.selectpicker').selectpicker();

    $(".news").bootstrapNews({
        newsPerPage: 3,
        autoplay: false,
    });

    // Offset for Main Navigation
    $('#navbar-fires').affix({
        offset: {
            top: 100
        }
    })


    $('#mapcontainer').block({
      message: $('#map_loading'),
      css: { opacity: 1 },
      overlayCSS:  {
        opacity: 0.5,
        cursor: null,
        backgroundColor: '#ccc',
      },
    });

    var barChartData = {
        labels : ["Ιανουάριος","Φεβρουάριος","Μάρτιος","Απρίλιος","Μάιος","Ιούνιος","Ιούλιος","Αύγουστος","Σεπτέμβριος","Οκτώβριος","Νόεμβριος","Δεκέμβριος"],
        datasets : [
            {
                fillColor : "rgba(220,220,220,0.5)",
                strokeColor : "rgba(220,220,220,0.8)",
                highlightFill: "rgba(220,220,220,0.75)",
                highlightStroke: "rgba(220,220,220,1)",
                data : date_month
            }
        ]

    };
```

```
var styledMapType = new google.maps.StyledMapType([
    {
        featureType: "water",
        elementType: "all",
        stylers: [
          { saturation: 0 },
          { lightness: 100 }
        ]
    },
    {
        featureType: "all",
        elementType: "all",
        stylers: [
          { saturation: -100 }
        ]
    }
], {
    name: "white",
});


// set the map to our custom style
var gmap = tm.getNativeMap();
gmap.mapTypes.set("white", styledMapType);
gmap.setMapTypeId("white");

// filter function for tags
var hasSelectedTag = function(item) {
    // if no tag was selected, everything passes
    return !window.selectedTag || (
            // item has tags?
            item.opts.tags &&
            // tag found? (note - will work in IE; Timemap extends the Array prototype if necessary)
            item.opts.tags.indexOf(window.selectedTag) >= 0
        );
};

// add our new function to the map and timeline filters
tm.addFilter("map", hasSelectedTag); // hide map markers on fail
tm.addFilter("timeline", hasSelectedTag); // hide timeline events on fail


$('#statistics').one('inview', function (event, visible, topOrBottomOrBoth) {
  if (visible == true) {
    $("#panel_1").addClass('panel_1_animation');
```

```
        $("#panel_2").addClass('panel_2_animation');
        $("#panel_3").addClass('panel_3_animation');
      } else {
      }
    });

    $('#chart-months-bar').one('inview', function (event, visible, topOrBottomOrBoth) {
      if (visible == true) {
        var chart_months_bar = document.getElementById("canvas-months-bar").getContext("2d");
        window.bar_months = new Chart(chart_months_bar).Line(barChartData, {
            responsive : true
        });
      } else {
      }
    });

    $('#chart-prefectures-doughnut').one('inview', function (event, visible, topOrBottomOrBoth) {
      if (visible == true) {
        var chart_prefectures_doughnut = document.getElementById("chart-prefectures-
doughnut").getContext("2d");
        prefectures_doughnut = new Chart(chart_prefectures_doughnut).Doughnut(pieData, {
          responsive : true ,
          legendTemplate : '<ul class="chart-legend list-unstyled"><% for (var i=0;
i<segments.length; i++){%><li><span class="background-color-info" style="background-
color:<%=segments[i].fillColor%>;"></span>
<%if(segments[i].label){%><%=segments[i].label%><%}%>
<span><em>(<%=segments[i].value%>)</em></span>  </li><%}%></ul>'
        });

        var helpers = Chart.helpers;
        var info_prefectures_doughnut = document.getElementById('info-prefectures-doughnut');
        info_prefectures_doughnut.innerHTML = prefectures_doughnut.generateLegend();

        function sizing() {
          if ($(window).width() < 768 ) {

            helpers.each(info_prefectures_doughnut.firstChild.childNodes, function (legendNode,
index) {
              helpers.addEvent(legendNode, 'click', function () {
                var activeSegment = prefectures_doughnut.segments[index];
                activeSegment.save();
                prefectures_doughnut.showTooltip([activeSegment]);
                activeSegment.restore();
              });
            });

          } else  {
```

```
            helpers.each(info_prefectures_doughnut.firstChild.childNodes, function (legendNode,
index) {
              helpers.addEvent(legendNode, 'mouseover', function () {
                var activeSegment = prefectures_doughnut.segments[index];
                activeSegment.save();
                prefectures_doughnut.showTooltip([activeSegment]);
                activeSegment.restore();
              });
            });
            helpers.addEvent(info_prefectures_doughnut.firstChild, 'mouseout', function () {
               prefectures_doughnut.draw();
            });


          }
        }

        $(document).ready(sizing);
        $(window).resize(sizing);

      } else {
      }
    });

    $('#chart-regions-pie').one('inview', function (event, visible, topOrBottomOrBoth) {
      if (visible == true) {
        var chart_regions_pie = document.getElementById("chart-regions-pie").getContext("2d");
        pie_regions = new Chart(chart_regions_pie).Pie(pieData_regions, {
            responsive : true,
            legendTemplate : '<ul class="chart-legend list-unstyled text-right"><% for (var i=0;
i<segments.length; i++){%><li> <span><em>(<%=segments[i].value%>)</em> </span>
<%if(segments[i].label){%><%=segments[i].label%><%}%> <span class="background-color-info"
style="background-color:<%=segments[i].fillColor%>;"></span> </li><%}%></ul>'
        });

        var helpers = Chart.helpers;
        var info_regions_pie = document.getElementById('info-regions-pie');
        info_regions_pie.innerHTML = pie_regions.generateLegend();

        function sizing() {
          if ($(window).width() < 768 ) {

            helpers.each(info_regions_pie.firstChild.childNodes, function (legendNode, index) {
            helpers.addEvent(legendNode, 'click', function(){
              var activeSegment = pie_regions.segments[index];
```

```
                activeSegment.save();
                activeSegment.fillColor = activeSegment.highlightColor;
                pie_regions.showTooltip([activeSegment]);
                activeSegment.restore();
              });
            });


          } else  {

            helpers.each(info_regions_pie.firstChild.childNodes, function (legendNode, index) {
            helpers.addEvent(legendNode, 'mouseover', function(){
              var activeSegment = pie_regions.segments[index];
                activeSegment.save();
                activeSegment.fillColor = activeSegment.highlightColor;
                pie_regions.showTooltip([activeSegment]);
                activeSegment.restore();
              });
            });
            helpers.addEvent(info_regions_pie.firstChild, 'mouseout', function(){
              pie_regions.draw();
            });
          }
        }

        $(document).ready(sizing);
        $(window).resize(sizing);

      } else {
      }
    });

      /* form submit errors */
      $('body').on('click', '#contact-submit', function(e){

        email = '';

        var testEmail = /^[A-Z0-9._%+-]+@([A-Z0-9-]+\.)+[A-Z]{2,4}$/i;
        if (testEmail.test($("#email").val()))
          email = 'yes';
        else
           email = 'no';

        if ($("#first_name").val() !="" && $("#last_name").val() !="" &&  $("#email").val() !="" &&
email != 'no' && $("#message").val() !="") {
          $( "#contact_form" ).submit();
```

```
        } else {
          $("#first_name").parent().removeClass('has-error');
          $("#last_name").parent().removeClass('has-error');
          $("#email").parent().removeClass('has-error');
          $("#message").parent().removeClass('has-error');

          if ($("#first_name").val() == "") {
            $("#first_name").parent().addClass('has-error');
           return false;
          }
          else if ($("#last_name").val() == "") {
            $("#last_name").parent().addClass('has-error');
            return false;
          }
          else if ($("#email").val() == "" || email == 'no') {
            $("#email").parent().addClass('has-error');
            return false;
          }
          else if ($("#message").val() == "") {
            $("#message").parent().addClass('has-error');
            return false;
          }
          e.preventDefault();
        }

    });

});
```