




## Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	<b>Θεωρητική ανασκόπηση των πλαισίων εργασίας εφαρμογών ιστού και πρακτική εφαρμογή στα Spring και Struts2.</b> <b>General reference at Web Application Frameworks and creation of a Web Application at Spring and Struts2.</b>
Όνοματεπώνυμο φοιτητή	<b>Κατσογιάννης Νικόλαος του Σπυρίδωνος</b>
Αριθμός μητρώου	<b>ΜΠΣΠ 13046</b>
Κατεύθυνση	<b>Δικτυοκεντρικά Πληροφορικά Συστήματα</b>
Επιβλέπων Συνεπιβλέπων	<b>Δουληγέρης Χρήστος, Καθηγητής</b> <b>Δρ. Σαράντης Μητρόπουλος, Επιστημονικός Συνεργάτης</b>



Πανεπιστήμιο Πειραιώς-Τμήμα Πληροφορικής  
Πρόγραμμα Μεταπτυχιακών Σπουδών στα  
Προηγμένα Συστήματα Πληροφορικής

Ημερομηνία παράδοσης **Ιούνιος 2016**

---

**Τριμελής Εξεταστική Επιτροπή**

(υπογραφή)

(υπογραφή)

(υπογραφή)

Χρήστος Δουληγέρης

Δημήτριος Βέργαδος

Παναγιώτης  
Κοτζανικολάου

## ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ .....	6
ABSTRACT .....	6
Ευχαριστίες.....	7
ΚΕΦΑΛΑΙΟ 1 .....	8
ΤΑ ΠΛΑΙΣΙΑ ΕΡΓΑΣΙΑΣ ΛΟΓΙΣΜΙΚΟΥ .....	8
Πλαίσια εργασίας λογισμικού .....	8
ΚΕΦΑΛΑΙΟ 2.....	11
ΤΑ ΠΛΑΙΣΙΑ ΕΡΓΑΣΙΑΣ ΕΦΑΡΜΟΓΩΝ ΙΣΤΟΥ.....	11
Ιστορική εξέλιξη των πλαισίων εργασίας εφαρμογών ιστού.....	11
Η πολυπλοκότητα των εφαρμογών ιστού .....	11
Τα πλαίσια εργασίας εφαρμογών ιστού .....	12
Η λειτουργία των πλαισίων εργασίας εφαρμογών ιστού .....	14
ΚΕΦΑΛΑΙΟ 3 .....	15
Τα πλαίσια εργασίας εφαρμογών ιστού ανά γλώσσα προγραμματισμού.....	15
Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα C.....	15
Το πλαίσιο εργασίας εφαρμογών ιστού libonion.....	15
Το πλαίσιο εργασίας εφαρμογών ιστού Kore.....	15
Το πλαίσιο εργασίας εφαρμογών ιστού KClone.....	16
Το πλαίσιο εργασίας εφαρμογών ιστού ESP .....	16
Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα C++.....	17
Το πλαίσιο εργασίας εφαρμογών ιστού CppCMS .....	17
Το πλαίσιο εργασίας εφαρμογών ιστού Wt.....	17
Το πλαίσιο εργασίας εφαρμογών ιστού Tntnet.....	17
Το πλαίσιο εργασίας εφαρμογών ιστού TreeFrog .....	18
Το πλαίσιο εργασίας εφαρμογών ιστού Platinum .....	19
Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα ColdFusion Markup Language .....	19
Το πλαίσιο εργασίας εφαρμογών ιστού CFWheels .....	19
Το πλαίσιο εργασίας εφαρμογών ιστού Fusebox.....	20
Το πλαίσιο εργασίας εφαρμογών ιστού FarCry Core .....	21

Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα D .....	21
Το πλαίσιο εργασίας εφαρμογών ιστού vibe.d .....	21
Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα Haskell .....	22
Το πλαίσιο εργασίας εφαρμογών ιστού Happstack .....	22
Το πλαίσιο εργασίας εφαρμογών ιστού Snap.....	22
Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα Java .....	22
Το πλαίσιο εργασίας εφαρμογών ιστού Struts .....	22
Το πλαίσιο εργασίας εφαρμογών ιστού Spring.....	23
Το πλαίσιο εργασίας εφαρμογών ιστού Play.....	23
Το πλαίσιο εργασίας εφαρμογών ιστού Grails.....	24
Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα JavaScript.....	24
Το πλαίσιο εργασίας εφαρμογών ιστού Ember.js .....	24
Το πλαίσιο εργασίας εφαρμογών ιστού Angular.js .....	24
Το πλαίσιο εργασίας εφαρμογών ιστού Meteor.....	25
Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα Scala .....	26
Το πλαίσιο εργασίας εφαρμογών ιστού Play.....	26
Το πλαίσιο εργασίας εφαρμογών ιστού Scalatra .....	26
Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα Perl .....	26
Το πλαίσιο εργασίας εφαρμογών ιστού catalyst.....	26
Το πλαίσιο εργασίας εφαρμογών ιστού Mojolicious.....	27
Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα PHP .....	27
Το πλαίσιο εργασίας εφαρμογών ιστού Symfony .....	27
Το πλαίσιο εργασίας εφαρμογών ιστού Symfony .....	27
Το πλαίσιο εργασίας εφαρμογών ιστού CodeIgniter .....	28
Το πλαίσιο εργασίας εφαρμογών ιστού CakePHP .....	28
Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα Python.....	29
Το πλαίσιο εργασίας εφαρμογών ιστού Django.....	29
Το πλαίσιο εργασίας εφαρμογών ιστού TurboGears 2 .....	29
Το πλαίσιο εργασίας εφαρμογών ιστού WEB2PY .....	29
Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα Ruby .....	30
Το πλαίσιο εργασίας εφαρμογών ιστού Ruby on Rails.....	30
Το πλαίσιο εργασίας εφαρμογών ιστού Sinatra.....	30
Το πλαίσιο εργασίας εφαρμογών ιστού Cuba.....	30
Το πλαίσιο εργασίας εφαρμογών ιστού ASP.NET .....	30

ΚΕΦΑΛΑΙΟ 4 .....	31
ΤΟ ΠΛΑΙΣΙΟ ΕΡΓΑΣΙΑΣ ΕΦΑΡΜΟΓΩΝ ΙΣΤΟΥ STRUTS2.....	31
Παρουσίαση του πλαισίου εργασίας εφαρμογών ιστού Struts 2.....	31
Πλεονεκτήματα και μειονεκτήματα του Struts 2 .....	31
Λειτουργία του Struts 2.....	32
Δημιουργία εφαρμογής στο Struts2 .....	33
Δημιουργία εφαρμογής online quiz στο Struts2 .....	34
ΚΕΦΑΛΑΙΟ 5 .....	43
ΤΟ ΠΛΑΙΣΙΟ ΕΡΓΑΣΙΑΣ ΕΦΑΡΜΟΓΩΝ ΙΣΤΟΥ SPRING.....	43
Παρουσίαση του πλαισίου εργασίας εφαρμογών ιστού Spring .....	43
Χαρακτηριστικά του Spring .....	43
Αρχιτεκτονική του Spring .....	43
Δημιουργία Εφαρμογής quiz στο Spring .....	45
ΚΕΦΑΛΑΙΟ 6 .....	56
Συμπεράσματα και πιθανές επεκτάσεις .....	56
ΒΙΒΛΙΟΓΡΑΦΙΑ .....	57

## ΠΕΡΙΛΗΨΗ

Το αντικείμενο της παρούσας διατριβής αφορά τα πλαίσια εργασίας εφαρμογών ιστού (Web Application Frameworks). Αρχικά, γίνεται αναφορά στα χαρακτηριστικά και στους λόγους χρησιμοποίησης των πλαισίων εργασίας λογισμικού (software frameworks) και στη συνέχεια γίνεται αντίστοιχη αναφορά στα πλαίσια εργασίας εφαρμογών ιστού. Παρουσιάζονται τα κυριότερα πλαίσια εργασίας εφαρμογών ιστού για τις κυριότερες γλώσσες προγραμματισμού. Παρουσιάζονται επίσης τα χαρακτηριστικά και ο τρόπος λειτουργίας των Struts2 και Spring. Τέλος, δημιουργήθηκε μια εφαρμογή online quiz στα πλαίσια εργασίας εφαρμογών ιστού Struts2 και Spring τα οποία βασίζονται στη γλώσσα Java.

## ABSTRACT

The subject of this thesis is related to Web Application Frameworks. Firstly, the main features of software frameworks as well the reasons to use them, are mentioned. A similar approach is followed about Web Application Frameworks. Moreover, the main Web Application Frameworks for the most popular programming languages are presented. The main features of Struts2 and Spring are presented and their functionality is explained. Finally, an online quiz application is created, based on the Web Application Frameworks for the Java programming language, Struts2 and Spring.

## **Ευχαριστίες**

Θα ήθελα να ευχαριστήσω τον καθηγητή κ. Δουληγέρη Χρήστο για την καθοδήγησή του και την υποστήριξή του κατά τη διάρκεια εκπόνησης της παρούσας διατριβής. Επιπλέον θέλω να ευχαριστήσω τον Επιστημονικό συνεργάτη Δρ. Σαράντη Μητρόπουλο για τις πολύτιμες συμβουλές του. Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου που με υποστήριξε όλο αυτό το διάστημα.

## ΚΕΦΑΛΑΙΟ 1

### ΤΑ ΠΛΑΙΣΙΑ ΕΡΓΑΣΙΑΣ ΛΟΓΙΣΜΙΚΟΥ

#### Πλαίσια εργασίας λογισμικού

Ένα πλαίσιο εργασίας λογισμικού (software framework), είναι ένας επαναχρησιμοποιούμενος τρόπος σχεδίασης λογισμικού, ο οποίος απαιτεί συστατικά μέρη λογισμικού (software components) ώστε να εφαρμοστεί [1]. Τα πλαίσια εργασίας λογισμικού παρέχουν κοινή λειτουργικότητα σε μια κατηγορία εφαρμογών [2].

Τα κυριότερα χαρακτηριστικά τα οποία συνήθως έχει ένα πλαίσιο εργασίας λογισμικού είναι τα ακόλουθα [3]:

- **Διαχωρισμός του κώδικα (Code Modularity):** Τα πλαίσια εργασίας λογισμικού βοηθούν το λογικό διαχωρισμό του κώδικα, αποκρύπτοντας τις συνεχώς μεταβαλλόμενες λεπτομέρειες μιας υλοποίησης λογισμικού μέσω της χρησιμοποίησης σταθερών διεπαφών (stable interfaces). Με το διαχωρισμό του κώδικα η συντήρηση του λογισμικού πραγματοποιείται με μεγαλύτερη ευκολία και ο κώδικας γίνεται ευκολότερα κατανοητός.

- **Επαναχρησιμοποίηση του κώδικα (Code reusability):** Οι σταθερές διεπαφές που παρέχουν τα πλαίσια εργασίας λογισμικού, συμβάλλουν στην επαναχρησιμοποίηση του κώδικα. Αυτό επιτυγχάνεται με τον ορισμό συστατικών μερών λογισμικού τα οποία προορίζονται για γενική χρήση και μπορούν να χρησιμοποιηθούν σε διαφορετικές εφαρμογές. Η επαναχρησιμοποίηση του κώδικα αξιοποιεί την προηγούμενη γνώση και εμπειρία των προγραμματιστών σε ένα συγκεκριμένο τομέα, με συνέπεια την αύξηση της παραγωγικότητας των προγραμματιστών, της ποιότητας, της απόδοσης και της διαλειτουργικότητας του λογισμικού.

- **Επεκτασιμότητα (Extensibility):** Ένα πλαίσιο εργασίας λογισμικού είναι επεκτάσιμο, διότι παρέχει μεθόδους που επιτρέπουν στις εφαρμογές να επεκτείνουν τις σταθερές τους διεπαφές. Η επεκτασιμότητα είναι απαραίτητη για την παραμετροποίηση των χαρακτηριστικών (features) και των υπηρεσιών (services) μιας εφαρμογής.

- **Αντιστροφή του ελέγχου (Inversion of control):** Η αντιστροφή του ελέγχου επιτρέπει σε ένα πλαίσιο εργασίας λογισμικού να καθορίζει τις σχετικές με μια εφαρμογή μεθόδους, οι οποίες καλούνται ως απόκριση σε εξωτερικά γεγονότα (external events). Τα εξωτερικά γεγονότα μπορεί να είναι πακέτα μηνυμάτων τα οποία φτάνουν σε μια θύρα επικοινωνίας (communication port), ή μηνύματα μέσω παραθύρων (window messages) τα οποία προέρχονται από τους τελικούς χρήστες μια εφαρμογής.

Στη συνέχεια αναφέρονται τα κυριότερα πλεονεκτήματα της χρήσης των πλαισίων εργασίας λογισμικού [2]. Η χρήση κώδικα ο οποίος ήδη έχει υλοποιηθεί, δοκιμαστεί και χρησιμοποιηθεί από άλλους προγραμματιστές, έχει ως συνέπεια την αύξηση της αξιοπιστίας του λογισμικού και τη μείωση του προγραμματιστικού χρόνου. Σε μια επιχείρηση η επαναχρησιμοποίηση του κώδικα μειώνει το κόστος ανάπτυξης του λογισμικού. Οι ομάδες ανάπτυξης λογισμικού μπορούν να διασπαστούν σε μία ομάδα που προγραμματίζει το πλαίσιο εργασίας λογισμικού και σε μία άλλη ομάδα που προγραμματίζει την τελική εφαρμογή. Αυτός ο διαχωρισμός των εργασιών επιτρέπει σε κάθε ομάδα να έχει πιο σαφείς στόχους και να δουλέψει στο κομμάτι εκείνο που έχει μεγαλύτερη εμπειρία.

Τα πλαίσια εργασίας λογισμικού συνήθως παρέχουν χαρακτηριστικά ασφαλείας τα οποία συχνά απαιτούνται από μια κοινή κατηγορία εφαρμογών. Συνεπώς, κάθε εφαρμογή φτιαγμένη με ένα πλαίσιο εργασίας λογισμικού, επωφελείται από την



παρεχόμενη ασφάλεια χωρίς πρόσθετο χρόνο προγραμματισμού και συνεπώς χωρίς πρόσθετο κόστος. Τα πλαίσια εργασίας λογισμικού βοηθούν στην υιοθέτηση των βέλτιστων τεχνικών και κανόνων για μια συγκεκριμένη πλατφόρμα. Για παράδειγμα, ένα πλαίσιο εργασίας λογισμικού για σταθερούς υπολογιστές το οποίο υποστηρίζει γραφικά περιβάλλοντα διεπαφής (desktop GUI framework), συνήθως έχει τη δυνατότητα δημιουργίας γραμμών εργαλείων (toolbars) και κουμπιών (buttons), τα οποία είναι κοινά με αυτά του λειτουργικού συστήματος στο οποίο εφαρμόζεται το πλαίσιο εργασίας λογισμικού. Τα πλαίσια εργασίας λογισμικού διευκολύνουν τον προγραμματισμό με την υιοθέτηση προτύπων σχεδίασης λογισμικού και βέλτιστων πρακτικών. Για παράδειγμα, υπάρχουν πολλά πλαίσια εργασίας λογισμικού τα οποία είναι βασισμένα στην αρχιτεκτονική σχεδίασης λογισμικού Model-View-Controller (MVC).

Οι αναβαθμίσεις σε ένα πλαίσιο εργασίας λογισμικού συμβάλλουν στη βελτίωση της λειτουργικότητας μιας εφαρμογής χωρίς να απαιτείται πρόσθετος προγραμματισμός. Για παράδειγμα, στην περίπτωση όπου ένα πλαίσιο εργασίας λογισμικού σχετικό με το ηλεκτρονικό εμπόριο (e-commerce) υποστηρίζει μια νέα μέθοδο πληρωμής (μετά από αναβάθμιση), αυτή η επιλογή γίνεται αυτόματα διαθέσιμη στον τελικό χρήστη της εφαρμογής, χωρίς να απαιτείται πρόσθετη εργασία από την πλευρά προγραμματιστή.

Ωστόσο, μπορεί να υπάρξουν και μειονεκτήματα από τη χρησιμοποίηση ενός πλαισίου εργασίας λογισμικού [2]:

Λόγω του ότι ο κώδικας ενός πλαισίου εργασίας λογισμικού είναι γενικής χρήσης και δεν είναι βελτιστοποιημένος για μια συγκεκριμένη περίπτωση χρήσης, μπορεί να υπάρξει μείωση της απόδοσης της εφαρμογής. Ωστόσο, σε αρκετές περιπτώσεις η μείωση αυτή αντισταθμίζεται από την ποιότητα και τη γρήγορη ανάπτυξη της τελικής εφαρμογής. Οι προγραμματιστές χρειάζονται αρκετή εκπαίδευση ώστε να μάθουν ένα πλαίσιο εργασίας λογισμικού έτσι ώστε να το χρησιμοποιούν σωστά και αποτελεσματικά.

Τα σφάλματα λογισμικού και τα κενά ασφαλείας ενός πλαισίου εργασίας λογισμικού, είναι πιθανό να επηρεάσουν κάθε εφαρμογή η οποία το χρησιμοποιεί. Η επιλογή ενός πλαισίου εργασίας λογισμικού πρέπει να γίνεται με προσοχή, διότι δεν είναι όλα τα πλαίσια εργασίας λογισμικού κατάλληλα για ένα συγκεκριμένο έργο σχετικό με την ανάπτυξη λογισμικού. Συνεπώς, η χρησιμοποίηση ενός μη κατάλληλου πλαισίου εργασίας λογισμικού μπορεί να έχει σημαντικές επιπτώσεις για ένα έργο.

Παραδείγματα πλαισίων εργασίας λογισμικού αποτελούν τα ακόλουθα:

Το MacApp [4] το οποίο ήταν το πρώτο αντικειμενοστρεφές πλαίσιο εργασίας λογισμικού. Ήταν γραμμένο στη γλώσσα ObjectPascal (μια επέκταση της γλώσσας Pascal) και φτιαγμένο για το λειτουργικό σύστημα Macintosh της εταιρείας Apple. Χαρακτηριστικό του ήταν η διευκόλυνση που παρείχε στον προγραμματιστή για τη δημιουργία μιας πλήρους και ισχυρής εφαρμογής για το λειτουργικό σύστημα Macintosh.

Το πλαίσιο εργασίας λογισμικού Dabo [5], είναι σχεδιασμένο για τη δημιουργία εφαρμογών για σταθερούς υπολογιστές. Υποστηρίζει τα λειτουργικά συστήματα Windows, OS X και Linux. Είναι γραμμένο σε γλώσσα Python, σε συνδυασμό με την εργαλειοθήκη λογισμικού wxPython, η οποία δημιουργεί γραφικό περιβάλλον διεπαφής για εφαρμογές λογισμικού.

Το πλαίσιο εργασίας λογισμικού Jt [6] (Java Pattern Oriented Framework), είναι βασισμένο σε πρότυπο σχεδιασμού με αποστολή μηνυμάτων. Έχει ως σκοπό τη γρήγορη ανάπτυξη εφαρμογών σε γλώσσα Java. Το πρότυπο σχεδιασμού με αποστολή μηνυμάτων που χρησιμοποιεί, έχει τη δυνατότητα να υλοποιεί πολλά ευρέως χρησιμοποιούμενα

πρότυπα σχεδιασμού καθώς και δυνατότητες για υποστήριξη Υπηρεσιοστρεφών Αρχιτεκτονικών (Service Oriented Architectures – SOA) και Επιχειρησιακού Διαύλου Υπηρεσιών (Enterprise Service Bus – ESB).

Στο κεφάλαιο 2 αναφέρονται οι λόγοι που οδήγησαν στη χρήση των πλαισίων εργασίας εφαρμογών ιστού και παρουσιάζονται τα κυριότερα χαρακτηριστικά τους. Στο κεφάλαιο 3 παρουσιάζονται τα κυριότερα πλαίσια εργασίας εφαρμογών ιστού για τις κυριότερες γλώσσες προγραμματισμού.

Στο κεφάλαιο 4 παρουσιάζεται η λειτουργία του πλαισίου εργασίας εφαρμογών ιστού Struts2 για τη γλώσσα Java και δίνεται ένα παράδειγμα χρήσης του μέσω της δημιουργίας μιας εφαρμογής online quiz.

Στο κεφάλαιο 5 παρουσιάζεται η λειτουργία του πλαισίου εργασίας εφαρμογών ιστού Spring για τη γλώσσα Java και δίνεται ένα παράδειγμα χρήσης του μέσω της δημιουργίας μιας εφαρμογής online quiz.

Στο κεφάλαιο 6 αναφέρονται τα συμπεράσματα όσον αφορά τη χρήση των πλαισίων εργασίας εφαρμογών ιστού γενικά, αλλά και συγκεκριμένα για τα Spring και Struts2. Τέλος, στο ίδιο κεφάλαιο προτείνονται πιθανές επεκτάσεις της εφαρμογής online quiz.

## ΚΕΦΑΛΑΙΟ 2

### ΤΑ ΠΛΑΙΣΙΑ ΕΡΓΑΣΙΑΣ ΕΦΑΡΜΟΓΩΝ ΙΣΤΟΥ

#### Ιστορική εξέλιξη των πλαισίων εργασίας εφαρμογών ιστού

Οι πρώτες εφαρμογές ιστού χρησιμοποιούσαν το πρωτόκολλο Common Gateway Interface (CGI)[7]. Πριν τη χρησιμοποίηση του πρωτοκόλλου CGI, οι εξυπηρετητές ιστού επέστεφαν ως απόκριση μόνο στατικά αρχεία. Το CGI όριζε ένα μηχανισμό ο οποίος αντιστοιχούσε ορισμένους υπερσυνδέσμους σε εκτελέσιμα προγράμματα αντί για αρχεία. Για αυτούς τους υπερσυνδέσμους ο εξυπηρετητής ανακαλούσε τα εκτελέσιμα προγράμματα ως μια υπο-διεργασία (subprocess). Η υπο-διεργασία αυτή δέχονταν τον υπερσύνδεσμο ως είσοδο, παρήγαγε μια ιστοσελίδα HTML και εισήγαγε τα περιεχόμενα της σελίδας στην έξοδο. Η σελίδα αυτή, επιστρέφονταν στο φυλλομετρητή που είχε αιτηθεί για αυτή. Τα προγράμματα σχετικά με το CGI, ήταν γραμμένα σε διάφορες γλώσσες όπως η C++ και η Perl.

Ο μηχανισμός βάσει του οποίου λειτουργούσε το CGI ήταν άνευ καταστάσεων (stateless). Αν μια εφαρμογή βασισμένη στο CGI χρειαζόνταν να διατηρήσει μια κατάσταση για μια σειρά από αιτήσεις, έπρεπε να αποθηκεύσει τις πληροφορίες αυτές σε ένα αρχείο, ή σε μια βάση δεδομένων. Για κάθε νέα αίτηση έπρεπε να ξαναφορτώσει την κατάστασή της. Γενικότερα η ιδιότητα άνευ καταστάσεων καθιστά πολυπλοκότερη την ανάπτυξη των εφαρμογών ιστού. Ωστόσο, βοηθάει στην κλιμάκωση (scaling) μέσω της κατανομής των εισερχόμενων αιτήσεων σε μια συστοιχία (cluster) εξυπηρετητών χωρίς να υπάρχει το πρόβλημα ποιος εξυπηρετητής διατηρεί την κατάσταση για αυτές τις αιτήσεις. Επίσης, βοηθάει στη δημιουργία ισχυρών εφαρμογών ιστού διότι σε περίπτωση σφάλματος στον εξυπηρετητή (όσον αφορά τις αιτήσεις) δεν υπάρχει κατάσταση (state) ώστε αυτή να χαθεί.

Τα προγράμματα που έκαναν χρήση του CGI, αντικαταστάθηκαν γρήγορα από τα πλαίσια εργασίας εφαρμογών ιστού πρώτης γενιάς, όπως τα Java Servlets. Στα πλαίσια εργασίας εφαρμογών ιστού πρώτης γενιάς, το σύστημα χρόνου εκτέλεσης (runtime system) για μια συγκεκριμένη γλώσσα είναι άμεσα σχετιζόμενο με τον εξυπηρετητή ιστού. Τα πλαίσια εργασίας εφαρμογών ιστού πρώτης γενιάς εισήγαγαν τη χρήση των προτύπων σχεδίασης για την απλοποίηση της δημιουργίας σελίδων HTML. Το πρότυπο σχεδίασης είναι ένα έγγραφο HTML το οποίο έχει επισημανθεί με κώδικα στη γλώσσα του πλαισίου εργασίας εφαρμογών ιστού. Τα πρότυπα σχεδίασης χρησιμοποιούνται ευρέως από την πλειοψηφία των πλαισίων εργασίας εφαρμογών ιστού.

Τα πλαίσια εργασίας εφαρμογών ιστού δεύτερης γενιάς είναι αυτά τα οποία εξετάζονται στην παρούσα διατριβή.

#### Η πολυπλοκότητα των εφαρμογών ιστού

Η ανάπτυξη εφαρμογών ιστού είναι πολύπλοκη για πολλούς λόγους [7]. Η λειτουργικότητα μιας εφαρμογής ιστού είναι κατανεμημένη και ένα μέρος της μπορεί να εκτελείται σε έναν εξυπηρετητή (σχετικά με τα δεδομένα της εφαρμογής), ενώ ένα άλλο στους φυλλομετρητές (σχετικά με τους χρήστες της εφαρμογής). Οι εφαρμογές ιστού ενσωματώνουν μια πληθώρα διαφορετικών τεχνολογιών όπως τις γλώσσες HTML, JavaScript, CSS και XML, καθώς και τα πρωτόκολλα HTTP και SSL. Επίσης στην πλευρά του εξυπηρετητή (server-side) εκτελούνται μία ή και παραπάνω γλώσσες όπως η PHP, Python, Java και Ruby. Επιπλέον, οι εφαρμογές ιστού πρέπει να υποστηρίζουν πολλούς

διαφορετικούς φυλλομετρητές οι οποίοι όμως ενδέχεται να μην έχουν πλήρως συμβατά χαρακτηριστικά σε σχέση με την εφαρμογή ιστού.

Οι εφαρμογές ιστού υποστηρίζουν ταυτόχρονα πολλούς χρήστες μέσω ενός εξυπηρετητή. Έτσι, διευκολύνεται η συνεργασία αλλά οι προγραμματιστές πρέπει να επιλύσουν τα θέματα της ταυτόχρονης πρόσβασης των χρηστών καθώς και της κλιμάκωσης. Οι εφαρμογές ιστού πρέπει να είναι παραμετροποιήσιμες. Τα περιβάλλοντα γραφικής διεπαφής χρήστη πριν την εποχή του παγκόσμιου ιστού, προτεινόταν να είναι ομοιόμορφα. Ωστόσο οι εφαρμογές ιστού έχουν η κάθε μία τα δικά της χαρακτηριστικά και τη δική της εμφάνιση. Έτσι, γίνεται δυσκολότερη η χρήση επαναχρησιμοποιούμενων στοιχείων λογισμικού. Η δημόσια προσβασιμότητα των εφαρμογών ιστού έχει ως αποτέλεσμα τη δημιουργία κενών ασφάλειας και θεμάτων απώλειας της ιδιωτικότητας. Τέλος, η πολυπλοκότητα του περιβάλλοντος της ανάπτυξης λογισμικού στον παγκόσμιο ιστό καθιστά πιθανή τη «δημιουργία» κενών ασφάλειας από μη έμπειρους προγραμματιστές (π.χ. μια εφαρμογή μπορεί να είναι ευάλωτη σε επιθέσεις τύπου SQL Injection).

### **Τα πλαίσια εργασίας εφαρμογών ιστού**

Ένα πλαίσιο εργασίας εφαρμογών ιστού (web application framework), αποτελεί έναν τύπο πλαισίου εργασίας λογισμικού, σχεδιασμένο για τη δημιουργία εφαρμογών ιστού (web applications) [8]. Παραδείγματα της λειτουργικότητας που παρέχουν τα πλαίσια εργασίας εφαρμογών ιστού, είναι η διαχείριση συν(session management), η διατήρηση των δεδομένων της εφαρμογής (data persistence) και τα συστήματα προτύπων σχεδίασης (template systems). Με τη χρησιμοποίηση ενός κατάλληλου πλαισίου εργασίας εφαρμογών ιστού, ο προγραμματιστής εξοικονομεί πολύ χρόνο κατά τη δημιουργία μιας εφαρμογής ιστού, ή ενός ιστότοπου (web site).

Στη συνέχεια, αναφέρονται τα κυριότερα γενικά χαρακτηριστικά των πλαισίων εργασίας εφαρμογών ιστού. Ένα βασικό χαρακτηριστικό των εφαρμογών ιστού, αποτελεί η αποθήκευση των πληροφοριών και η δημιουργία ιστοσελίδων οι οποίες είναι βασισμένες σε αποθηκευμένες πληροφορίες. Αντίθετα με τις στατικές (static) ιστοσελίδες, η πλειοψηφία των σελίδων των εφαρμογών ιστού, δημιουργούνται δυναμικά από τα δεδομένα τα οποία διατηρεί η εφαρμογή. Παράδειγμα αποτελεί ένα σύστημα διαχείρισης περιεχομένου όπου ο χρήστης δημοσιεύει ένα άρθρο μέσω μιας σελίδας διαχείρισης. Η σελίδα αυτή θα έχει πολλαπλά πεδία για το άρθρο όπως τον τίτλο, το κυρίως περιεχόμενο το συγγραφέα και την ημερομηνία δημιουργίας. Στη συνέχεια, η σελίδα αυτή θα δημιουργηθεί δυναμικά στην περίπτωση όπου ένας χρήστης ζητήσει το άρθρο από την εφαρμογή ιστού. Έτσι, κάθε άρθρο έχει την ίδια γενική δομή και ο εκδότης του άρθρου θα μπορεί να επεξεργάζεται το περιεχόμενο ενός άρθρου χωρίς η επεξεργασία αυτή να γίνεται απευθείας στον εξυπηρετητή ή να χρειάζεται ο χρήστης να γνωρίζει τη γλώσσα HTML.

Επειδή οι εφαρμογές ιστού διαφέρουν μεταξύ τους, τα πλαίσια εργασίας εφαρμογών ιστού, διαθέτουν χαρακτηριστικά τα οποία βοηθούν στη διατήρηση των δεδομένων. Ένα παράδειγμα αποτελεί η χρήση μιας σταθερής διεπαφής προγραμματισμού εφαρμογών (API), η οποία επιτρέπει την πρόσβαση της εφαρμογής ιστού σε πολλά διαφορετικά συστήματα αποθήκευσης δεδομένων.

Επίσης, τα πλαίσια εργασίας εφαρμογών ιστού παρέχουν αυτόματη ή απλοποιημένη αποθήκευση αντικειμένων που αντιπροσωπεύουν δεδομένα μέσω της

τεχνικής Object-Relational Mapping (ORM) για παράδειγμα. Η τεχνική ORM καθιστά εφικτή τη διευθυνσιοδότηση, την πρόσβαση και το χειρισμό αντικειμένων χωρίς να χρειάζεται να γνωρίζουμε πώς τα αντικείμενα αυτά σχετίζονται με τις πηγές δεδομένων τους [9]. Επιπλέον, παρέχουν βελτιώσεις στην απόδοση όπως η χρησιμοποίηση κρυφής μνήμης πάνω από το επίπεδο της βάσης δεδομένων.

Τέλος, τα πλαίσια εργασίας εφαρμογών ιστού, παρέχουν τη δημιουργία βάσεων δεδομένων SQL και τον έλεγχο της ακεραιότητας των δεδομένων (data integrity check). Για παράδειγμα, ελέγχουν τις σχέσεις μεταξύ των δεδομένων και επαληθεύουν ότι τα απαιτούμενα πεδία σε μια βάση δεδομένων είναι συμπληρωμένα.

Οι στατικοί ιστότοποι διαχειρίζονται τους επισκέπτες τους ως ανώνυμους. Οι εφαρμογές ιστού όμως συχνά απαιτούν τη δημιουργία λογαριασμών χρηστών (user accounts) και τη διατήρηση των πληροφοριών μεταξύ των διάφορων σελίδων τους. Τα πλαίσια εργασίας εφαρμογών ιστού, παρέχουν τη δυνατότητα διαχείρισης λογαριασμού χρηστών, έτσι ώστε οι χρήστες να μπορούν να εγγραφούν (register), να συνδεθούν (login) σε μια εφαρμογή ιστού καθώς και να επαναπροσδιορίσουν τον κωδικό πρόσβασης τους (password reset).

Σε πολλές περιπτώσεις, οι σελίδες των εφαρμογών ιστού είναι προσβάσιμες μόνο από αυθεντικοποιημένους χρήστες. Τα πλαίσια εφαρμογών ιστού μπορούν αυτόματα να ελέγξουν αν ο χρήστης είναι αυθεντικοποιημένος και να απαιτήσουν την αυθεντικοποίηση (authentication) του χρήστη πριν δημιουργήσουν μια σελίδα για ένα συγκεκριμένο χρήστη.

Όταν ένας χρήστης αυθεντικοποιηθεί, το πλαίσιο εργασίας εφαρμογών ιστού μπορεί να ελέγξει τις άδειες πρόσβασης (permissions) για το χρήστη αυτόν. Για παράδειγμα, αυτό καθίσταται εφικτό με τη χρήση του ελέγχου πρόσβασης βασισμένο σε ρόλους (role based access control). Συνήθως τα χαρακτηριστικά ασφάλειας ελέγχονται από τον προγραμματιστή μέσω του κώδικα, ή μέσω του διαχειριστή της εφαρμογής ιστού π.χ. μέσω μιας σελίδας διαχείρισης.

Οι προγραμματιστές αναθέτουν μέρος του περιεχομένου μιας εφαρμογής ιστού στην κρυφή μνήμη με σκοπό την αύξηση της απόδοσής της. Το περιεχόμενο που βρίσκεται στην κρυφή μνήμη δε δημιουργείται εκ νέου σε κάθε αίτηση που αφορά μια σελίδα της εφαρμογής ιστού.

Τα πρότυπα σχεδίασης για ιστοσελίδες συμβάλλουν στο διαχωρισμό της επιχειρησιακής λογικής από την λογική της εμφάνισης, κάτι το οποίο θεωρείται αποδεκτή πρακτική. Ένα πλαίσιο εργασίας εφαρμογών ιστού συνήθως παρέχει ένα ή περισσότερα συστήματα προτύπων σχεδίασης με σταθερές διεπαφές. Η πλειοψηφία των πλαισίων εργασίας εφαρμογών ιστού διαθέτουν πεδία τα οποία προσδιορίζουν οι προγραμματιστές, ή έτοιμες δομές δεδομένων οι οποίες προϋπάρχουν στα πρότυπα σχεδίασης. Το σύστημα προτύπων σχεδίασης, μπορεί να αποτελεί κάτι απλό, όπως αντικατάσταση μιας λέξης κλειδί (keyword) ή κάτι πιο σύνθετο όπως η αυτόματη δημιουργία περιεχομένου π.χ. πεδία φόρμας (form fields) από παραμέτρους.

Οι δυναμικοί ιστότοποι συνήθως έχουν ένα μέρος τους δημιουργημένο για τους διαχειριστές τους. Έτσι, ο ιστότοπος μπορεί να ρυθμιστεί κατάλληλα και να τροποποιηθούν τα δεδομένα του εφόσον χρειάζεται. Οι διαχειριστές μπορούν να δημιουργήσουν λογαριασμούς χρηστών, να αλλάξουν το περιεχόμενο της σελίδας κ.ά. Τα πλαίσια εργασίας εφαρμογών ιστού βοηθούν στη δημιουργία διεπαφών διαχειριστή παράγοντας μία δομή πλοήγησης (navigation structure) και παρέχοντας συνηθισμένα στοιχεία διεπαφής σε πεδία φόρμών π.χ. ένα πεδίο για ημερομηνία με τη χρήση

ημερολογίου. Τέλος, μπορεί να υποστηρίζουν την αυτόματη παραγωγή σελίδων επεξεργασίας και εμφάνισης στοιχείων από διατηρούμενα δεδομένα.

### Η λειτουργία των πλαισίων εργασίας εφαρμογών ιστού

Όλες οι εφαρμογές ιστού δέχονται μεμονωμένες αιτήσεις HTTP και παράγουν τις αντίστοιχες αποκρίσεις. Υπάρχουν πολλοί τρόποι επίτευξης της παραπάνω διαδικασίας διότι εξαρτάται σε μεγάλο βαθμό από τον τύπο του εξυπηρετητή.

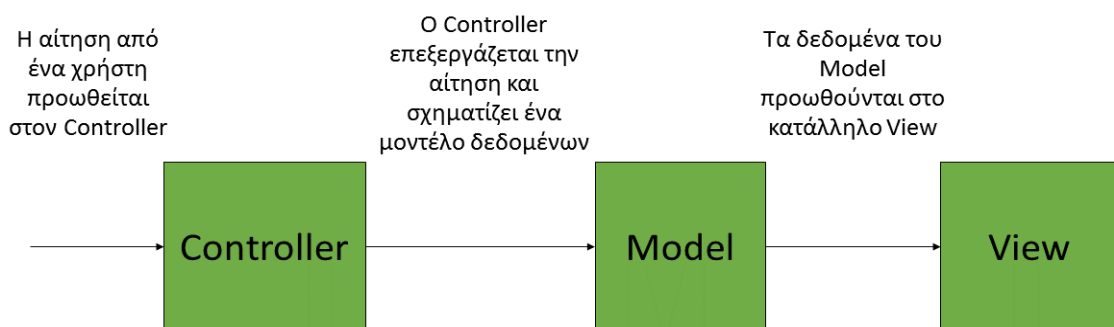
Η πιο διαδομένη αρχιτεκτονική σχεδιασμού των πλαισίων εργασίας εφαρμογών ιστού είναι η αρχιτεκτονική Model-View-Controller (MVC).

Το Model έχει σχέση με τη λογική που σχετίζεται με τα δεδομένα της εφαρμογής. Το Model μπορεί να αντιπροσωπεύει τα δεδομένα τα οποία μεταφέρονται μεταξύ του View και του Controller, καθώς και όσα δεδομένα σχετίζονται με την επιχειρησιακή λογική (business logic).

Το View έχει σχέση με τη λογική της διεπαφής χρήστη (user interface- UI) της εφαρμογής.

Ο Controller «συνδέει» το Model με το View, ώστε να επεξεργαστεί την επιχειρησιακή λογική και τις εισερχόμενες αιτήσεις της εφαρμογής. Επιπλέον, διαχειρίζεται τα δεδομένα μέσω του Model και αλληλεπιδρά με το View ώστε να εμφανίσει την απόκριση στο χρήστη.

Ο πηγαίος κώδικας του πλαισίου εργασίας εφαρμογών ιστού δέχεται ως είσοδο τον υπερσύνδεσμο και αναθέτει τη διεκπεραίωση της ενέργειας στον κατάλληλο Controller. Ο Controller ενεργεί μαζί με το Model των δεδομένων της εφαρμογής και επιτρέπει στο View να δημιουργήσει το περιεχόμενο της τελικής απόκρισης το οποίο θα εμφανίσει ο φυλλομετρητής. Τα παραπάνω παρουσιάζονται στην Εικόνα 1:



Εικόνα 1: Η αρχιτεκτονική MVC

## **ΚΕΦΑΛΑΙΟ 3**

### **ΤΑ ΠΛΑΙΣΙΑ ΕΡΓΑΣΙΑΣ ΕΦΑΡΜΟΓΩΝ ΙΣΤΟΥ ΑΝΑ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ**

Στο παρόν κεφάλαιο παρουσιάζονται τα κυριότερα πλαίσια εργασίας εφαρμογών ιστού ανά γλώσσα προγραμματισμού. Γίνεται αναφορά στα χαρακτηριστικά και στη λειτουργία τους.

#### **Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα C**

##### **Το πλαίσιο εργασίας εφαρμογών ιστού libonion**

Το πλαίσιο εργασίας εφαρμογών ιστού libonion, αποτελεί μια ελαφριά βιβλιοθήκη η οποία βοηθάει στη δημιουργία εξυπηρετητών ιστού στη γλώσσα C [10]. Οι εξυπηρετητές ιστού μπορεί να είναι εφαρμογές ιστού, ή ένα μέσο επέκτασης μιας εφαρμογής σε C ώστε αυτή να αποκτήσει λειτουργικότητα σχετική και συμβατή με τον παγκόσμιο ιστό.

Ο χρήστης μπορεί να δημιουργήσει χειριστές (handlers), οι οποίοι υποστηρίζουν την ανάγνωση δεδομένων από ένα αρχείο, ή την εγγραφή δεδομένων σε ένα αρχείο. Ωστόσο, υπάρχουν πολλοί προεπιλεγμένοι χειριστές οι οποίοι εξυπηρετούν τις πιο συχνά εμφανιζόμενες ανάγκες.

Το libonion μπορεί να χρησιμοποιηθεί για τη δημιουργία μικρών και αυτόνομων εφαρμογών ιστού, καθώς και για την προσθήκη λειτουργικότητας HTTP σε υπάρχοντα προγράμματα σε C. Μια ακόμα περίπτωση χρήσης του libonion, είναι η δημιουργία υπηρεσιών οι οποίες χρειάζεται να χρησιμοποιούν μικρό μέρος της μνήμης και του επεξεργαστή λόγω της φύσης του συστήματος, όπως τα ενσωματωμένα συστήματα (embedded systems).

Το libonion έχει πολλά παραδείγματα χρήσης το οποία μπορεί να είναι απλά αλλά και σύνθετα. Υπάρχει και πρακτική εφαρμογή του στην πλατφόρμα υλικού (hardware platform) Aisoy (<http://www.aisoy.com/>), η οποία χρησιμοποιείται για την ανάπτυξη εφαρμογών τεχνητής νοημοσύνης. Το libonion, χρησιμοποιείται για το συντονισμό της επικοινωνίας με το ρομπότ, για την ενημέρωση του λογισμικού συστήματος (system software) καθώς και για την δημιουργία εφαρμογών για το ρομπότ. Χάρη στη χρήση του libonion η πλατφόρμα Aisoy αποτελείται από 1000 γραμμές κώδικα σε C και άλλες 500 γραμμές από κώδικα σε HTML και σε JavaScript.

##### **Το πλαίσιο εργασίας εφαρμογών ιστού Kore**

Το πλαίσιο εργασίας εφαρμογών ιστού Kore[11], είναι εύκολο στη χρήση και μέσω αυτού δημιουργούνται κλιμακούμενες (scalable) διεπαφές προγραμματισμού εφαρμογών για τον παγκόσμιο ιστό (web APIs). Το Kore έχει ως στόχο την ασφάλεια, την κλιμάκωση και τη γρήγορη ανάπτυξη τέτοιων API.

Το Kore υποστηρίζει τα πρωτόκολλα SNI, SPDY/3.1, HTTP/1.1. και WebSocket. Επιπλέον, διαθέτει ενσωματωμένη επαλήθευση παραμέτρων (parameter validation) και επιτρέπει μόνο τη δημιουργία συνδέσεων τύπου HTTPS (ασφαλείς). Το Kore διαθέτει

επίσης, εργασίες που τρέχουν στο παρασκήνιο (background tasks) οι οποίες καταναλώνουν ένα μικρό μέρος των υπολογιστικών πόρων. Πρόσθετα, διαθέτει χειριστές σελίδων τύπου non-blocking, έναν εύκολο στη χρήση μηχανισμό αυθεντικοποίησης και τη δυνατότητα άμεσης και ταυτόχρονης φόρτωσης πολλών βιβλιοθηκών λογισμικού (software libraries). Παρέχει υποστήριξη για βάσεις δεδομένων τύπου PostgreSQL, προεπιλεγμένη κρυπτογράφηση μέσω του πρωτοκόλλου TLS και φόρτωση της εφαρμογής ιστού ως μια προ-μεταγλωττισμένη (pre-compiled) δυναμική βιβλιοθήκη λογισμικού. Τέλος διαθέτει αρχιτεκτονική οδηγούμενη από γεγονότα (event driven), με διεργασίες τύπου workers που τρέχουν ανά πυρήνα του επεξεργαστή και δομικές μονάδες λογισμικού (modules) οι οποίες μπορούν να επαναφορτώνονται άμεσα, ακόμα και όταν την ίδια στιγμή αποστέλλεται περιεχόμενο στον φυλλομετρητή.

Η λειτουργία του Kore έχει δοκιμαστεί στις ακόλουθες πλατφόρμες λογισμικού: Linux 3.2, OpenBSD 5.3, FreeBSD 10 και OSX 10.8.2 ή σε νεότερες εκδόσεις τους. Παράδειγμα χρήσης του Kore είναι η μετάδοση περιεχομένου video μέσω της γλώσσας HTML5.

### **Το πλαίσιο εργασίας εφαρμογών ιστού KClone**

Το πλαίσιο εργασίας εφαρμογών ιστού KClone [12] είναι ένα πλήρως λειτουργικό πλαίσιο εργασίας εφαρμογών ιστού, ειδικά σχεδιασμένο για ενσωματωμένα συστήματα και συσκευές.

Τα βασικά του χαρακτηριστικά είναι η δημιουργία εξυπηρετητών ιστού τύπου HTTP ή HTTPS οι οποίοι υποστηρίζουν πολλές διαφορετικές πλατφόρμες λογισμικού όπως τα Windows XP, GNU/Linux 2.x και Darwin 7, 8 / MacOSX (εκδόσεις 10.3, 10.4, 10.5). Το KClone χρησιμοποιεί ένα μικρό μέρος της μνήμης του συστήματος. Επιπλέον, ο εξυπηρετητής που φτιάχνεται μέσω του KClone μπορεί να ανακτήσει το περιεχόμενό του από το τοπικό σύστημα αρχείων, ή από το σύστημα αρχείων του ενσωματωμένου συστήματος, ή μέσω του πρωτοκόλλου CGI, ή και με συνδυασμό των παραπάνω μεθόδων.

Η εκκίνηση μιας νέας εφαρμογής σε KClone σε ένα νέο υπολογιστή είναι απλά η δημιουργία ενός αρχείου τύπου Makefile το οποίο αποτελείται από πέντε γραμμές. Δε χρειάζεται χειροκίνητη (manual) λήψη του αρχείου, διότι γίνεται αυτόματη ρύθμιση από το KClone. Το KClone είναι ανοικτού κώδικα και όλα τα συστατικά του μέρη είναι διαθέσιμα για δωρεάν λήψη από τον ιστότοπό του. Επίσης, το KClone διαθέτει τη δυνατότητα δημιουργίας δυναμικών ιστοσελίδων γραμμένες σε κώδικα σε C (ή και σε C++) οι οποίες ενσωματώνεται σε κώδικα HTML μέσω ετικετών (tags) που γράφονται ως εξής: <% code %>. Τέλος, το KClone παρέχει βιβλιοθήκες λογισμικού για πρόσβαση σε όλα τα συνηθισμένα στοιχεία που απαιτούνται για μια εφαρμογή ιστού όπως η διαχείριση συνόδου, το ανέβασμα αρχείων (file uploading) και τα cookies.

### **Το πλαίσιο εργασίας εφαρμογών ιστού ESP**

Το πλαίσιο εργασίας εφαρμογών ιστού ESP [13] καθιστά εύκολη τη δημιουργία γρήγορων και δυναμικών εφαρμογών ιστού και ιστοσελίδων. Οι εφαρμογές που είναι γραμμένες με το ESP χρησιμοποιούν τη γλώσσα C για προγραμματισμό από την πλευρά του εξυπηρετητή, κάτι το οποίο επιτρέπει εύκολη πρόσβαση στα δεδομένα του συστήματος, ή στο υλικό (hardware).

Το ESP βασίζεται στην αρχιτεκτονική MVC και αν οι σελίδες ιστού ή οι ελεγκτές (controllers) τροποποιηθούν, ο κώδικας επαναμεταγλωττίζεται και ξαναφορτώνεται. Αυτό συνεπάγεται αυξημένη απόδοση και ευελιξία για τις εφαρμογές ιστού. Το ESP διατίθεται



με εμπορική άδεια, καθώς και με άδεια ανοικτού κώδικα τύπου GNU GPL για μη εμπορική χρήση.

## **Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα C++**

### **Το πλαίσιο εργασίας εφαρμογών ιστού CppCMS**

Το πλαίσιο εργασίας εφαρμογών ιστού CppCMS [14] χαρακτηρίζεται από υψηλή απόδοση και διαφέρει από την πλειοψηφία των άλλων πλαισίων εργασίας εφαρμογών ιστού όπως το Django για Python και τα Java Servlets. Πρώτον, το CppCMS είναι σχεδιασμένο και ρυθμισμένο έτσι ώστε να διαχειρίζεται πολύ υψηλό φόρτο εργασίας. Δεύτερον, χρησιμοποιεί μοντέρνα C++ ως κύρια γλώσσα ανάπτυξης ώστε να πετύχει τον πρώτο στόχο. Τρίτον, είναι σχεδιασμένο για τη δημιουργία και υπηρεσιών ιστού αλλά και ιστότοπων.

Το CppCMS είναι ανοικτού κώδικα, αλλά διαθέτει και εμπορική άδεια σε περίπτωση που ο προγραμματιστής χρειάζεται να δημιουργήσει εμπορικό λογισμικό με το CppCMS. Τέλος το CppCMS, υποστηρίζει τα λειτουργικά συστήματα Linux, Windows, Solaris, FreeBSD και Mac OS X.

### **Το πλαίσιο εργασίας εφαρμογών ιστού Wt**

Το πλαίσιο εργασίας εφαρμογών ιστού Wt [15] είναι μια βιβλιοθήκη της C++ με στόχο την ανάπτυξη εφαρμογών ιστού. Η διεπαφή προγραμματισμού εφαρμογών την οποία διαθέτει έχει ως βάση τα widgets (widget-centric) τα οποία είναι γραφικά στοιχεία που χρησιμοποιούνται στον παγκόσμιο ιστό.

Το Wt βοηθάει τους προγραμματιστές «αφαιρώντας» πολλές λεπτομέρειες της υλοποίησης που είναι σχετικές με τον παγκόσμιο ιστό, συμπεριλαμβάνοντας τα πρωτόκολλα μεταξύ πελάτη-εξυπηρετητή όπως τα HTTP και WebSockets. Επίσης, το Wt απαλλάσσει τον προγραμματιστή από χειρισμό της HTML μέσω της JavaScript και αντιμετωπίζει τα θέματα συμβατότητας με διαφορετικούς φυλλομετρητές. Συνεπώς ο προγραμματιστής μπορεί να εστιάσει στην ουσιαστική λειτουργικότητα της εφαρμογής που αναπτύσσει, μαζί με ένα σύνολο πλήρως λειτουργικών widgets.

Οι εφαρμογές ιστού που μπορούν να γραφτούν με το Wt, χαρακτηρίζονται από υψηλή απόδοση και πολυπλοκότητα και είναι πλήρως εξατομικευμένες. Επίσης, χρησιμοποιούν την τεχνολογία Ajax και είναι προσβάσιμες από μηχανές αναζήτησης και βελτιστοποιημένες για αυτές (Search Engine Optimized). Άλλη περίπτωση χρήσης του Wt αποτελεί η δημιουργία γραφικών διεπαφών χρήστη στον παγκόσμιο ιστό (Web-based GUIs) για ενσωματωμένα συστήματα. Τέλος, μέσω του Wt μπορούν να δημιουργηθούν Web-based GUIs που απαιτούν ενσωμάτωση με υπάρχουσες βιβλιοθήκες της C++ (π.χ. για επιστημονικές εφαρμογές) ή με υπάρχουσες εφαρμογές της C++ για σταθερούς υπολογιστές (desktop applications).

### **Το πλαίσιο εργασίας εφαρμογών ιστού Tntnet**

Το πλαίσιο εργασίας εφαρμογών ιστού Tntnet [16] αποτελεί έναν εξυπηρετητή ιστού που επιτρέπει τη δημιουργία εφαρμογών ιστού σε γλώσσα C++. Το Tntnet διαθέτει μια πρότυπη

γλώσσα την `ecpp` όπου ο κώδικας σε C++ μπορεί να ενσωματωθεί σε γλώσσα HTML όπως γίνεται με τη γλώσσα PHP και τις Java Server Pages (JSP). Οι σελίδες που δημιουργούνται μέσω του Tntnet μεταγλωττίζονται και συνδέονται σε μια διαμοιραζόμενη βιβλιοθήκη. Ως αποτέλεσμα, δημιουργείται μια εφαρμογή ιστού που είναι αποδοτική και συμπαγής.

Το Tntnet παρέχει υποστήριξη για το πρωτόκολλο SSL, αντιστοίχιση των υπερσυνδέσμων (URL mapping) και ένα API το οποίο υποστηρίζει το πρωτόκολλο HTTP, το ανέβασμα αρχείων μέσω του HTTP, καθώς και την αυθεντικοποίηση. Τέλος, παρέχει ρυθμίσεις σχετικές με τις κεφαλίδες (headers) του HTTP, ρυθμίσεις σχετικές με τα cookies και την αυτόματη διαχείριση συνόδου (session management).

Το Tntnet διατίθεται δωρεάν με άδεια τύπου Lesser General Public License και είναι συμβατό με αρκετές διανομές του λειτουργικού συστήματος GNU/Linux, με το λειτουργικό σύστημα Solaris και με άλλα λειτουργικά συστήματα που είναι βασισμένα στο Unix. Ωστόσο, δεν είναι συμβατό με τα λειτουργικά συστήματα Windows και OS X.

### **Το πλαίσιο εργασίας εφαρμογών ιστού TreeFrog**

Το πλαίσιο εργασίας εφαρμογών ιστού Treefrog [17] είναι υψηλής απόδοσης και πλήρους στοίβας full-stack) και υποστηρίζει τα πρωτόκολλα HTTP και Websocket. Λόγω του ότι από την πλευρά του εξυπηρετητή το Treefrog είναι γραμμένο σε C++, οι εφαρμογές ιστού εκτελούνται γρηγορότερα από ότι σε μια γλώσσα σεναρίου ενεργειών (scripting language). Όσον αφορά την ανάπτυξη των εφαρμογών ιστού το TreeFrog παρέχει ένα αντικειμενοσχεσιακό σύστημα αντιστοίχισης (Object/Relational mapping system) και πρότυπα συστήματα βασισμένα στην αρχιτεκτονική MVC. Στόχος του είναι η επίτευξη υψηλής παραγωγικότητας μέσω της πολιτικής όπου δίνεται προτεραιότητα στις συμβάσεις σε σχέση με τις ρυθμίσεις της εφαρμογής ιστού (convention over configuration).

Το TreeFrog είναι συμβατό με τα λειτουργικά συστήματα Windows, Mac OS, Linux και με τα λειτουργικά συστήματα που είναι βασισμένα στο Unix. Είναι εφικτή η δημιουργία εφαρμογών ιστού με το TreeFrog οι οποίες τρέχουν σε διαφορετικά λειτουργικά συστήματα, μέσω της επαναμεταγλώττισης του πηγαίου κώδικα.

Στο επίπεδο του Controller (ελεγκτή), ο προγραμματιστής μέσω του TreeFrog αποκτά εύκολα τα δεδομένα τα οποία αναπαριστούν την αίτηση και την αντίστοιχη απόκριση HTTP, καθώς και την αντίστοιχη σύνοδο του HTTP. Επίσης, το TreeFrog παρέχει μερικά προεπιλεγμένα χαρακτηριστικά όπως η αυθεντικοποίηση εισόδου, ο έλεγχος πρόσβασης και η επικύρωση στοιχείων σε μια διαδικτυακή φόρμα. Επιπλέον, παρέχει ένα μηχανισμό δρομολόγησης για την κλήση μεθόδων από τον κατάλληλο ελεγκτή μέσω του URL το οποίο ζητείται από τον πελάτη, στις κατάλληλες δράσεις (Actions). Δεν υπάρχει λόγος να γράφονται οι κανόνες ένας προς ένας στο αρχείο ρυθμίσεων, ή για την κατανομή αιτήσεων που απαιτούν κάποια δράση.

Στο επίπεδο του View (όψης), ο κώδικας C++ μπορεί να ενσωματωθεί σε αρχεία HTML μέσω των ετικετών `<% ...%>`. Αυτό σημαίνει ότι ο κώδικας μπορεί να γραφτεί με τρόπο παρόμοιο με αυτόν μιας γλώσσας σεναρίου ενεργειών (scripting language). Πρόσθετα, προσφέρει ένα νέο σύστημα προτύπων σχεδίασης το οποίο διαχωρίζει τη λογική της παρουσίασης από τα πρότυπα. Το πρότυπο είναι γραμμένο σε γλώσσα HTML και έτσι περιγράφεται καθαρά η λογική του αρχείου που είναι γραμμένο σε C++. Έτσι διευκολύνεται η συνεργασία μεταξύ των προγραμματιστών και των σχεδιαστών.

Στο επίπεδο του μοντέλου που ονομάζεται SqlObject στο TreeFrog, παρέχεται ένας αντικειμενο-σχεσιακός αντιστοιχητής (object relational mapper). Οι προγραμματιστές εφαρμογών μέσω αυτού, μπορούν να εστιάσουν στην ανάπτυξη της επιχειρησιακής λογικής χωρίς να γράψουν πολλές γραμμές κώδικα. Ωστόσο, αν χρειάζεται να χρησιμοποιηθούν δεδομένα σε σύνθετες περιπτώσεις, η γλώσσα SQL μπορεί να χρησιμοποιηθεί άμεσα και λόγω της χρήσης των αντικαταστατών (placeholders), τα ερωτήματα (queries) προς τη βάση δεδομένων εκτελούνται εύκολα και με ασφάλεια. Τέλος, το TreeFrog είναι συμβατό με τις βάσεις δεδομένων MySQL, PostgreSQL, SQLite, DB2 και Oracle.

Το TreeFrog συνδέεται με το πλαίσιο εργασίας λογισμικού Qt το οποίο χρησιμοποιείται για τη δημιουργία γραφικών διεπαφών χρήστη. Το TreeFrog είναι ανοικτού κώδικα.

### **Το πλαίσιο εργασίας εφαρμογών ιστού Platinum**

Το πλαίσιο εργασίας εφαρμογών ιστού Platinum[18] βοηθάει τον προγραμματιστή να γράφει εφαρμογές υψηλής απόδοσης για διαφορετικές πλατφόρμες λογισμικού με τον ίδιο κύριο κώδικα (base code). Είναι εύκολο στη χρήση και ενσωματώνεται εύκολα σε υπάρχοντα πλαίσια εργασίας εφαρμογών ιστού και σε υπάρχουσες βιβλιοθήκες λογισμικού.

Το Platinum μπορεί να χρησιμοποιηθεί σε ενσωματωμένες συσκευές, σε σταθερούς υπολογιστές, ή σε μεγάλους εξυπηρετητές. Υποστηρίζει τα λειτουργικά συστήματα Windows, Windows CE, Mac OS-X, Linux, QNX και Unix. Μέσω του Platinum ο προγραμματιστής αποφεύγει τον προγραμματισμό χαμηλού επιπέδου (low-level programming) και εστιάζει στην εφαρμογή που έχει να δημιουργήσει.

Το Platinum υποστηρίζει κείμενο με κωδικοποίηση Unicode, πολλαπλά νήματα (multi-threading), σειριακή διάταξη (serialization), επεξεργασία μέσω XML, πρόσθετα (plugins) και άλλες δυνατότητες. Η υψηλή απόδοσή του επιτυγχάνεται με τη χρήση χαρακτηριστικών μοντέρνας C++. Με το Platinum μπορούν να γραφούν εμπορικές εφαρμογές καθώς και εφαρμογές ανοικτού κώδικα λόγω της άδειας (τύπου LGPL) που διαθέτει.

### **Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα ColdFusion Markup Language**

#### **Το πλαίσιο εργασίας εφαρμογών ιστού CFWheels**

Το πλαίσιο εργασίας εφαρμογών ιστού CFWheels [19] είναι ανοικτού κώδικα και είναι βασισμένο στο πλαίσιο εργασίας εφαρμογών ιστού Ruby On Rails. Το CFWheels παρέχει γρήγορη ανάπτυξη εφαρμογών μέσω ενός συστήματος οργάνωσης του κώδικα και διαθέτει εκτενή και ενημερωμένη (up-to date) τεκμηρίωση.

Το CFWheels έχει ως στόχο την απλοποίηση του κώδικα. Η έννοια του αντικειμενοστρεφούς προγραμματισμού (object-oriented programming) στο CFWheels είναι απλή και έχει ως κέντρο τα δεδομένα (data-centric). Επίσης, στο CFWheels από προεπιλογή θα επιστραφεί ένα σύνολο αποτελεσμάτων σχετικά με ένα ερώτημα σε μια βάση δεδομένων, όταν υπάρχουν πολλαπλές εγγραφές σε αυτήν. Το CFWheels ενθαρρύνει τη χρήση των βέλτιστων πρακτικών, αλλά δε θα παρουσιάσει σφάλμα σε περίπτωση που ακολουθηθεί διαφορετική πρακτική από τον προγραμματιστή. Επιπλέον, χρησιμοποιείται

κώδικας της ColdFusion Markup Language (CFML) εξ' ολοκλήρου και δε γίνεται χρήση άλλων γλωσσών όπως της XML. Τέλος αν ο προγραμματιστής παρατηρήσει ότι το CFWheels δεν καλύπτει τις ανάγκες του, ή το πρόγραμμα μπορεί να υλοποιηθεί με πιο απλό τρόπο, μπορεί να προγραμματίσει σε απλή γλώσσα CMFL χωρίς δυσκολία.

Το CFWheels χρησιμοποιεί την πολιτική όπου δίνεται προτεραιότητα στις συμβάσεις σε σχέση με τις ρυθμίσεις της εφαρμογής ιστού (convention over configuration). Συνεπώς, αντί να χρειάζεται η δήλωση πληθώρας μεταβλητών σχετικές με τις ρυθμίσεις του συστήματος, το CFWheels χρησιμοποιεί προεπιλεγμένες ρυθμίσεις έτσι ώστε να μη χρειάζεται η δήλωση των σχετικών μεταβλητών.

Ο κώδικας που χρησιμοποιεί το CFWheels είναι σαφής και δεν επαναλαμβάνεται. Επίσης, το CFWheels βασίζεται στην αρχιτεκτονική MVC, συνεπώς ο κώδικας που γράφεται από τον προγραμματιστή υπάγεται στην κατηγορία είτε του Model, είτε του View, είτε του Controller. Τέλος, το CFWheels βασίζεται στην τεχνική Object/Relational Mapping η οποία είναι υπεύθυνη για το χειρισμό της αντιστοίχισης των αντικειμένων στη μνήμη ως προς το πώς είναι αποθηκευμένα στη βάση δεδομένων. Έτσι, μπορεί να γίνει αντικατάσταση γραψίματος ερωτημάτων προς τη βάση δεδομένων με απλές μεθόδους (π.χ. record.save()).

### **Το πλαίσιο εργασίας εφαρμογών ιστού Fusebox**

Το πλαίσιο εργασίας εφαρμογών ιστού Fusebox [20] βοηθάει στην εύκολη δημιουργία και συντήρηση των εφαρμογών ιστού και διευκολύνει την ομαδική ανάπτυξη λογισμικού. Αυτό επιτυγχάνεται επειδή βοηθάει τον προγραμματιστή να δομήσει την εφαρμογή του και να διαχωρίσει τα συστατικά της στοιχεία σε διακριτά μέρη. Το Fusebox αποτελείται από ένα σύνολο βασικών αρχείων τα οποία διαχειρίζονται πολλές βασικές λειτουργίες κατά τη δημιουργία μιας διαδικτυακής εφαρμογής.

Μια εφαρμογή στο Fusebox αποτελείται από κυκλώματα (Circuits). Ένα Circuit, αντιστοιχεί σε μία διαδρομή (directory) μέσα στην εφαρμογή. Μέσα σε κάθε κύκλωμα υπάρχουν μια ή περισσότερες «FuseActions». Ένα FuseAction είναι ένας χειριστής αιτήσεων (request handler). Ένα FuseAction λειτουργεί εκτελώντας ένα ή περισσότερα «Fuses». Τα Fuses είναι ξεχωριστά πρότυπα σχεδίασης σε CFML, όπως αρχεία απεικόνισης.

Το Fusebox χρησιμοποιεί αρχεία XML για να ορίσει τις ρυθμίσεις της εφαρμογής και για να δηλώσει τι θα κάνει κάθε Circuit και πως γίνεται η εκτέλεση των FuseActions μέσα σε κάθε Circuit. Υπάρχει ένα βασικό σύνολο της γραμματικής της XML, το οποίο χρησιμοποιείται για τη δημιουργία αυτών των αρχείων.

Το Fusebox είναι πλήρως επεκτάσιμο μέσω προσθέτων που επιτρέπουν την επέκταση της βασικής λειτουργικότητάς του χωρίς τροποποίηση των βασικών αρχείων του. Υποστηρίζει επίσης λεξικά (lexicons) που επιτρέπουν την επέκταση της γραμματικής της XML με ετικέτες οριζόμενες από το χρήστη. Ως αποτέλεσμα, το Fusebox είναι πολύ ευέλικτο και μπορεί να εξυπηρετήσει ένα ευρύ πεδίο προγραμματιστικών αναγκών.

Το Fusebox δεν επιβάλλει τη χρήση του μοντέλου MVC, ή τον αντικειμενοστρεφή προγραμματισμό στον προγραμματιστή. Παρόλα αυτά είτε κάθε μια ξεχωριστά, είτε και οι δύο προγραμματιστικές προσεγγίσεις ταυτόχρονα μπορούν να χρησιμοποιηθούν με το Fusebox.

### **Το πλαίσιο εργασίας εφαρμογών ιστού FarCry Core**

Το πλαίσιο εργασίας εφαρμογών ιστού FarCry Core [21], είναι διαδομένο και προσανατολισμένο στον επιχειρηματικό χώρο και επιτρέπει την ανάπτυξη γρήγορων και επεκτάσιμων εφαρμογών. Έχει σχεδιαστεί για μηχανές της CFML, όπως οι Lucee, Railo και η Adobe ColdFusion και μπορεί να τρέξει σε οποιοδήποτε περιβάλλον μπορεί να εκτελεστεί η γλώσσα Java συμπεριλαμβανομένου του υπολογιστικού νέφους.

Το FarCry Core δεν περιορίζει τον προγραμματιστή με τύπους περιεχομένων που είναι έτοιμα για χρήση χωρίς περαιτέρω επεξεργασία (out-of-the-box). Ο προγραμματιστής μπορεί να κατασκευάσει και να αναπτύξει εξατομικευμένους τύπους περιεχομένου γρήγορα και εύκολα με το FarCry Builder, ή να τους γράψει στον προτιμώμενο για αυτόν επεξεργαστή (editor).

Η προτεραιότητα στις συμβάσεις σε σχέση με τις ρυθμίσεις της εφαρμογής ιστού και οι βέλτιστες πρακτικές είναι το δυνατό σημείο του προγραμματιστικού μοντέλου του FarCry Core. Ο προγραμματιστής «ξοδεύει» λιγότερο χρόνο στη σκέψη και την αναζήτηση και έχει περισσότερο χρόνο να αναπτύξει την εφαρμογή του.

Το «Webtop» που διαθέτει το FarCry Core είναι ένα πλήρως εξατομικευμένο και επεκτάσιμο πάνελ για τη διαχείριση της εφαρμογής (administration panel). Μια σαφής και φιλική για τον χρήστη διεπαφή καθιστά τη διαχείριση περιεχομένου απλή.

### **Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα D**

#### **Το πλαίσιο εργασίας εφαρμογών ιστού vibe.d**

Το πλαίσιο εργασίας εφαρμογών ιστού vibe.d [22] είναι υψηλού επιπέδου και βασισμένο στην αρχιτεκτονική REST. Το vibe.d υποστηρίζει υλοποιήσεις πελατών (clients), εξυπηρετητών (servers), καθώς και διαμεσολαβητών (proxies). Επιπλέον, έχει έτοιμους οδηγούς (drivers) για την υποστήριξη των βάσεων δεδομένων MongoDB και Redis. Πρόσθετα, διαθέτει εργαλειοθήκη συγχρονισμού διεργασιών (concurrency toolkit) και υποστηρίζει λειτουργίες εισόδου-εξόδου (I/O operations) χαμηλού επιπέδου.

Επιπλέον, το vibe.d διαθέτει ασύγχρονο μηχανισμό εισόδου-εξόδου (I/O) με στόχο τη γρήγορη εκτέλεση των διεργασιών και την ελάχιστη δυνατή χρησιμοποίηση της μνήμης. Το vibe.d διαθέτει πρότυπα σχεδίασης σε γλώσσα HTML τα οποία ονομάζονται Diet. Τα συγκεκριμένα πρότυπα σχεδίασης μειώνουν σημαντικά την επιβάρυνση της σύνταξης της HTML, η οποία οφείλεται στη σήμανση των στοιχείων της με ετικέτες (tags). Τέλος, τα πρότυπα σχεδίασης Diet επιτρέπουν τον ορισμό δυναμικών στοιχείων HTML στην ίδια γραμμή (inline). Τα δυναμικά αυτά στοιχεία είναι συνήθως γραμμένα σε γλώσσα JavaScript.

Ο κώδικας του vibe.d μεταγλωττίζεται σε κώδικα γλώσσας μηχανής και ως αποτέλεσμα, ο χρόνος εκτέλεσης του κώδικα ελαχιστοποιείται. Πρόσθετα, το vibe.d υποστηρίζει την τεχνολογία εκτέλεσης σε πολλαπλά νήματα (multi-threading) και σύντομα θα προστεθεί και η δυνατότητα ενσωματωμένης εξισορρόπησης φόρτου εργασίας (load balancing).

Το vibe.d διαθέτει μια συμπαγή διεπαφή προγραμματισμού εφαρμογών με τις πιο κοινές προεπιλεγμένες ρυθμίσεις. Τέλος, το vibe.d διαθέτει πλήρη υποστήριξη για διαχείριση σφαλμάτων μέσω εξαιρέσεων (exception-based) καθώς και το σύστημα πακέτων (packages) DUB μέσω του οποίου επιτυγχάνεται απλή πρόσβαση σε βιβλιοθήκες επέκτασης (extension libraries) φτιαγμένες από τρίτους.

## **Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα Haskell**

### **Το πλαίσιο εργασίας εφαρμογών ιστού Happstack**

Το πλαίσιο εργασίας εφαρμογών ιστού Happstack [23], είναι ανοικτού κώδικα και διαθέτει έτοιμες βιβλιοθήκες. Κάποιες από αυτές αφορούν τη δημιουργία προτύπων σχεδίασης (templating) και τη διατήρηση δεδομένων (data persistence). Το Haskell διαθέτει και βιβλιοθήκες σχετικές με το πρωτόκολλο HTTP και την επικύρωση δεδομένων (data validation).

Στόχος του Happstack είναι η δημιουργία ανεξάρτητων μερών τα οποία μπορούν να εκμεταλλεύονται πλήρως την ταχύτητα, την ασφάλεια και τη δύναμη της εκφραστικότητας της γλώσσας Haskell. Το Happstack εκτελείται στα λειτουργικά συστήματα GNU/Linux, OS X, FreeBSD και Windows [24].

Το κύριο μέρος του Haskell είναι ο εξυπηρετητής που ονομάζεται happstack (happstack-server). Ο εξυπηρετητής happstack διαθέτει έναν εξυπηρετητή HTTP με πολλές δυνατότητες, δυνατότητα διαχείρισης αρχείων και άλλα. Επιπλέον, διαθέτει ξεχωριστά τα ακόλουθα πακέτα: Το safecopy το οποίο διαθέτει σειριακή διάταξη τύπων δεδομένων (datatype serialization) και υποστήριξη για εγκατάσταση της εφαρμογής ιστού σε άλλο λειτουργικό σύστημα. Το acid-type το οποίο αποτελεί ένα ισχυρό σύστημα αποθήκευσης το οποίο είναι μη-σχεσιακό (No SQL) και το οποίο υποστηρίζει τους τύπους δεδομένων της γλώσσας Haskell. Τέλος, διαθέτει πληθώρα βιβλιοθηκών λογισμικού οι οποίες είναι κατασκευασμένες από τρίτους.

### **Το πλαίσιο εργασίας εφαρμογών ιστού Snap**

Το πλαίσιο εργασίας εφαρμογών ιστού Snap [25], είναι απλό και φτιαγμένο για λειτουργικά συστήματα τύπου Unix. Πρόσφατα προστέθηκε υποστήριξη για το λειτουργικό σύστημα Windows χωρίς όμως να έχουν γίνει πλήρεις δοκιμές. Το Snap τείνει να γίνει το de-facto πλαίσιο εργασίας εφαρμογών ιστού για τη Haskell για τους ακόλουθους λόγους: Παρουσιάζει υψηλή απόδοση και έχει αυστηρά κριτήρια σχεδίασης. Επίσης, είναι απλό στη χρήση, ισχυρό και διαθέτει πλήρη τεκμηρίωση με ευκολία στην εκτέλεση δοκιμών.

Το Snap διαθέτει μια γρήγορη βιβλιοθήκη εξυπηρετητή σε HTTP και μία λογική και σαφή μονάδα για προγραμματισμό σχετικό με τον παγκόσμιο ιστό. Επιπλέον το Snap διαθέτει ένα σύστημα προτύπων σχεδίασης βασισμένο στην HTML για τη δημιουργία ιστοσελίδων, οι οποίες επιτρέπουν το συνδυασμό ετικετών σε Haskell με την αντίστοιχη λειτουργικότητα χωρίς να χρειάζεται άλλη γλώσσα προγραμματισμού όπως η PHP. Επίσης, διαθέτει ένα σύστημα υψηλού επιπέδου που ονομάζεται Snaplets, το οποίο βοηθάει στη δημιουργία δομημένων εφαρμογών ιστού. Τέλος, διαθέτει ενσωματωμένα Snaplets για τη δημιουργία προτύπων σχεδίασης, για την αυθεντικοποίηση και για τη διαχείριση συνόδου.

## **Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα Java**

### **Το πλαίσιο εργασίας εφαρμογών ιστού Struts**

Το πλαίσιο εργασίας εφαρμογών ιστού Struts [26][27] είναι ανοικτού κώδικα και είναι σχεδιασμένο έτσι ώστε να βοηθάει τον προγραμματιστή να δημιουργεί εφαρμογές ιστού

που χρησιμοποιούν την αρχιτεκτονική MVC. Το Struts αποτελείται από τρία βασικά μέρη: Ένα χειριστή αιτήσεων (request handler), ο οποίος παρέχεται από τον προγραμματιστή της εφαρμογής ιστού και αντιστοιχίζεται σε ένα συγκεκριμένο URI (Uniform Resource Identifier). Ένα χειριστή αποκρίσεων (response handler), ο οποίος αναθέτει τον έλεγχο της εφαρμογής σε έναν άλλον πόρο (resource) ο οποίος διαχειρίζεται την απόκριση. Τέλος, διαθέτει και μία βιβλιοθήκη ετικετών (tag library) η οποία βοηθάει τους προγραμματιστές στη δημιουργία διαδραστικών εφαρμογών που είναι βασισμένες σε φόρμες (form-based) με σελίδες εξυπηρετητή (server pages). Η αρχιτεκτονική του Struts καθώς και οι ετικέτες είναι συμβατές με ευρέως χρησιμοποιούμενες ορολογίες σε σχέση με την τεχνολογία (buzzword compliant). Το Struts συνεργάζεται καλά με συμβατικές εφαρμογές τύπου REST και με τεχνολογίες όπως οι SOAP και AJAX.

Το Struts2 αποτελεί την τρέχουσα έκδοση του Struts. Το Struts2 είναι πλήρως λειτουργικό και μοντέρνο πλαίσιο εργασίας εφαρμογών ιστού.

### **Το πλαίσιο εργασίας εφαρμογών ιστού Spring**

Το πλαίσιο εργασίας εφαρμογών ιστού Spring [28][29], αναπτύχθηκε το 2003, είναι ελαφρύ και έχει τη δυνατότητα να συνεργάζεται με άλλα πλαίσια εργασίας εφαρμογών ιστού όπως τα Struts, Hibernate και JSF. Το Spring προορίζεται για εφαρμογές σχετικές με επιχειρήσεις (enterprise applications).

Ένα βασικό στοιχείο του Spring είναι η υποστήριξη της υποδομής σε επίπεδο εφαρμογών, δηλαδή το Spring στοχεύει στην επιχειρησιακή λογική σε επίπεδο εφαρμογών χωρίς να συνδέεται με συγκεκριμένα μόνο περιβάλλοντα ανάπτυξης (deployment environments). Κάποια από τα χαρακτηριστικά του Spring είναι: Dependency injection[30], δηλαδή μια μεθοδολογία ανάπτυξης λογισμικού η οποία εφαρμόζει την τεχνική Inversion of control και έχει ως σκοπό να δίνει σε ένα αντικείμενο (object) τις απαραίτητες μεταβλητές στιγμιότυπου (instance variables). Επιπλέον, το Spring υποστηρίζει το μοντέλο προγραμματισμού Aspect Oriented Programming[31] το οποίο έχει προταθεί ως μια τεχνική για τη βελτίωση του διαχωρισμού των εννοιών στο λογισμικό και βασίζεται στον αντικειμενοστρεφή προγραμματισμό (object-oriented programming) καθώς και στον διαδικαστικό προγραμματισμό (procedural programming). Πρόσθετα, το Spring διαθέτει υπο-πλαίσια σχετικά με εφαρμογές ιστού καθώς με υπηρεσίες ιστού τύπου Restful. Τέλος το Spring υποστηρίζει τεχνολογίες σχετικές με τη γλώσσα Java όπως τη Java Message Service (JMS) και τη Java Database Connectivity (JDBC).

### **Το πλαίσιο εργασίας εφαρμογών ιστού Play**

Το πλαίσιο εργασίας εφαρμογών ιστού Play [32] είναι ελαφρύ, άνευ καταστάσεων και παρέχει μια αρχιτεκτονική φιλική προς τον παγκόσμιο ιστό. Το Play έχει ως στόχο τη δημιουργία κλιμακούμενων εφαρμογών ιστού και υποστηρίζει επίσης και τη γλώσσα Scala.

Το Play είναι φιλικό προς τον προγραμματιστή με ασφαλή τυποποίηση (type safety) και ισχυρά εργαλεία δημιουργίας κώδικα (build tools). Το Play υποστηρίζει την αρχιτεκτονική τύπου Restful από προεπιλογή καθώς και πλαίσια εργασίας λογισμικού της γλώσσας CSS όπως το CoffeeScript, τη γλώσσα JSON καθώς και μη σχεσιακές βάσεις δεδομένων (no-sql databases). Επιπλέον, το Play υποστηρίζει ένα μεγάλο εύρος των υπαρχόντων για Java βιβλιοθηκών και διαθέτει μεγάλη και δραστήρια κοινότητα χρηστών.

### **Το πλαίσιο εργασίας εφαρμογών ιστού Grails**

Το πλαίσιο εργασίας εφαρμογών ιστού Grails [33], έχει ως στόχο την αύξηση της παραγωγικότητας του προγραμματιστή μέσω των προεπιλεγμένων ρυθμίσεων ώστε να ταιριάζουν με την πλειοψηφία των περιπτώσεων χρήσης καθώς και μέσω της τεχνικής που δίνεται προτεραιότητα στις συμβάσεις αντί για τις ρυθμίσεις. Επιπλέον, το Grails διαθέτει μεγάλο εύρος διεπαφών προγραμματισμού εφαρμογών.

Το Grails ενσωματώνεται με ευκολία στην εικονική μηχανή της Java (JVM) με στόχο την άμεση παραγωγικότητα και την παροχή ισχυρών χαρακτηριστικών. Μεταξύ αυτών των χαρακτηριστικών είναι: Υλοποίηση της τεχνικής ORM, υποστήριξη γλωσσών για συγκεκριμένους τομείς εφαρμογών (Domain Specific Languages), καθώς και η υποστήριξη του ασύγχρονου προγραμματισμού (Asynchronous programming).

### **Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα JavaScript**

#### **Το πλαίσιο εργασίας εφαρμογών ιστού Ember.js**

Το πλαίσιο εργασίας εφαρμογών ιστού Ember.js [34] ευνοεί τη χρήση σύντομου και περιεκτικού κώδικα μέσω των προτύπων σχεδίασης Handlebars τα οποία ενημερώνονται αυτόματα σε περίπτωση αλλαγής των δεδομένων της εφαρμογής. Το Ember.js βοηθάει στην αύξηση της παραγωγικότητας του προγραμματιστή μέσω των εύκολων στη χρήση API που διαθέτει.

Επιπλέον, το Ember.js διαθέτει τα «Components». Τα Components επιτρέπουν στον προγραμματιστή να γράφει δικές του ετικέτες HTML, με τη χρήση των Handlebars για την περιγραφή της σήμανσης. Η παραμετροποιήσιμη συμπεριφορά όσον αφορά τα Components, επιτυγχάνεται μέσω της JavaScript. Πρόσθετα, η παρουσίαση δεδομένων σε JSON στο Ember.js, διευκολύνεται με τη χρήση της γλώσσας jQuery.

Με το Ember.js είναι δυνατή η δημιουργία εφαρμογών ιστού με πολλές διαφορετικές ιστοσελίδες και τη δημιουργία εύχρηστων και εύκολων στην απομνημόνευση υπερσυνδέσμων.

#### **Το πλαίσιο εργασίας εφαρμογών ιστού Angular.js**

Το πλαίσιο εργασίας εφαρμογών ιστού Angular.js [35], βοηθάει στη δημιουργία δυναμικών εφαρμογών ιστού μέσω της δημιουργίας από τον προγραμματιστή παραμετροποιήσιμη γλώσσας HTML. Η προαναφερθείσα γλώσσα HTML είναι εκφραστική, εύκολη στην κατανόηση και γρήγορη στην ανάπτυξη.

Το Angular.js είναι επεκτάσιμο και συνεργάζεται με άλλες βιβλιοθήκες λογισμικού, καθώς και παραμετροποιήσιμο διότι κάθε χαρακτηριστικό του μπορεί να τροποποιηθεί από τον προγραμματιστή ή να αντικατασταθεί από άλλο.

Η υποστήριξη της δυναμικής διασύνδεσης δεδομένων (data binding) που διαθέτει το Angular.js ενημερώνει το View όταν αλλάζει το Model και αντίστροφα. Με αυτόν τον τρόπο η διαχείριση του Document Object Model (DOM) γίνεται αυτόματα. Στο Angular.js, το Model αποτελείται από απλά αντικείμενα της JavaScript (plain old JavaScript objects). Με αυτόν τον τρόπο, η επαναχρησιμοποίηση του κώδικα, η δοκιμή και η συντήρησή του γίνονται με ευκολία.



Το Angular.js επιτρέπει στον προγραμματιστή να ορίζει τους κανόνες της επικύρωσης των στοιχείων που υποβάλλονται σε μια διαδικτυακή φόρμα (form validation) χωρίς εκείνος να χρειάζεται να γράψει κώδικα σε JavaScript. Επίσης, το Angular.js διαθέτει φίλτρα και ειδικές οδηγίες (directives) ώστε να υποστηρίξει τη διάθεση της εφαρμογής σε πολλές διαφορετικές γλώσσες και τοπικές ρυθμίσεις (locales).

### **Το πλαίσιο εργασίας εφαρμογών ιστού Meteor**

Το πλαίσιο εργασίας εφαρμογών ιστού Meteor [36] δίνει τη δυνατότητα σχεδιασμού εφαρμογών διαδικτύου καθώς και κινητών εφαρμογών σε JavaScript χρησιμοποιώντας την ίδια βάση κώδικα. Με το Meteor, είναι δυνατή η δημιουργία ενιαίας JavaScript δηλαδή, ο ίδιος κώδικας εκτελείται στον πελάτη και στο υπολογιστικό νέφος, όπως επίσης και στους φυλλομετρητές και στις κινητές συσκευές. Αυτό, επιτυγχάνεται μέσω του ενιαίου συστήματος δημιουργίας κώδικα.

Με το Meteor είναι εφικτή η δημιουργία εφαρμογής η οποία εγκαθίσταται στα κινητά λειτουργικά συστήματα iOS και Android καθώς και η δημιουργία εφαρμογής ιστού η οποία φορτώνει κατ' απαίτηση (on demand).

Τα στοιχεία του γραφικού περιβάλλοντος διεπαφής παρέχουν την εμφάνιση και απόκριση μίας ολοκληρωμένης εφαρμογής, χρησιμοποιώντας το υπο-πλαίσιο Blaze του Meteor, ή μέσω ενσωμάτωσης με τα πλαίσια εργασίας εφαρμογών ιστού Angular.js και ReactJS.

Το Meteor παρέχει μία πλατφόρμα σχετική με το υπολογιστικό νέφος που ονομάζεται Galaxy η οποία απλοποιεί σε μεγάλο βαθμό την κλιμάκωση την ανάπτυξη και τον έλεγχο εφαρμογών που κατασκευάζονται με αυτό. Το Galaxy παρέχει μετρήσεις σχετικές με τη διαχείριση των εφαρμογών, έτσι ώστε να εξασφαλίζεται η ορθή και αποτελεσματική λειτουργία τους.

Με μία απλή εντολή, η εφαρμογή υλοποιείται σε επίπεδο παραγωγής και ενημερώνονται με την τελευταία έκδοση όλοι οι συνδεδεμένοι περιηγητές και οι συσκευές που τη χρησιμοποιούν.

Το Galaxy διασφαλίζει ότι μπορούν να συντονιστούν οι ενημερώσεις του κώδικα που εκτελείται στην πλευρά του πελάτη με ακρίβεια με σκοπό να εκτελεστεί μία αδιάλειπτη και αυτόματη ενημέρωση της εφαρμογής για όλους τους πελάτες, ακόμα κι όταν εκείνη τη στιγμή τη χρησιμοποιούν .

Όταν τα δεδομένα αλλάζουν ο κάθε χρήστης που χρησιμοποιεί μια εφαρμογή γραμμένη στο Meteor παίρνει τα πιο πρόσφατα δεδομένα με αποτελεσματικό τρόπο και με αξιοπιστία.

Μέσω των βιβλιοθηκών λογισμικού που διαθέτει το Meteor ο προγραμματιστής δε χρειάζεται να αφιερώσει ιδιαίτερο χρόνο ώστε να διαχειριστεί και να ενσωματώσει στοιχεία και κώδικα. Έτσι, έχει τη δυνατότητα να επικεντρωθεί στον τρόπο σχεδίασης της εφαρμογής του.

## **Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα Scala**

### **Το πλαίσιο εργασίας εφαρμογών ιστού Play**

Το πλαίσιο εργασίας εφαρμογών ιστού Play [32], παρέχει μια αρχιτεκτονική φιλική προς τον παγκόσμιο ιστό η οποία είναι ελαφριά και άνευ καταστάσεων (stateless). Το Play έχει ως στόχο τη δημιουργία κλιμακούμενων εφαρμογών ιστού και υποστηρίζει επίσης και τη γλώσσα Java.

Το Play είναι φιλικό προς τον προγραμματιστή και διαθέτει ασφαλή τυποποίηση (type safety) και ισχυρά εργαλεία δημιουργίας κώδικα (build tools). Το Play υποστηρίζει την αρχιτεκτονική τύπου Restful από προεπιλογή, καθώς και πλαίσια εργασίας λογισμικού της γλώσσας CSS όπως το CoffeeScript, τη γλώσσα JSON και μη σχεσιακές βάσεις δεδομένων (no-sql databases). Επιπλέον, το Play υποστηρίζει ένα μεγάλο εύρος των υπάρχοντων βιβλιοθηκών λογισμικού για Scala και διαθέτει μεγάλη και δραστήρια κοινότητα χρηστών.

### **Το πλαίσιο εργασίας εφαρμογών ιστού Scalatra**

Το πλαίσιο εργασίας εφαρμογών ιστού Scalatra [37], είναι απλό micro-framework και διατίθεται δωρεάν. Συνδυάζει τις δυνατότητες της γλώσσας Scala με τις δυνατότητες της εικονικής μηχανής της Java με στόχο τη γρήγορη δημιουργία ιστότοπων και API υψηλής απόδοσης. Το Scalatra βασίζεται στην αρχιτεκτονική MVC1[38] και εκτελείται σε Servlet Containers όπως ο Apache Tomcat.

## **Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα Perl**

### **Το πλαίσιο εργασίας εφαρμογών ιστού catalyst**

Το πλαίσιο εργασίας εφαρμογών ιστού catalyst [39] είναι ανοικτού κώδικα και βασίζεται στην αρχιτεκτονική MVC. Για τη βάση δεδομένων (Model) χρησιμοποιεί την τεχνική ORM. Για το View το catalyst υποστηρίζει πολλές γλώσσες προτύπων σχεδίασης όπως HTML και Mason 2.

Το catalyst διαθέτει δικό του εξυπηρετητή για την ανάπτυξη της εφαρμογής, ο οποίος καταναλώνει λίγους πόρους. Ο εξυπηρετητής του catalyst επανακκινεί αυτόματα όταν αλλάξουν κάποια στοιχεία της εφαρμογής, συνεπώς προκύπτουν άμεσα αποτελέσματα στην πλευρά του χρήστη. Το catalyst λειτουργεί στους πιο διαδεδομένους εξυπηρετητές ιστού χωρίς πολλές και χρονοβόρες ρυθμίσεις. Επιπλέον το catalyst διαθέτει πρόσθετα που είναι έτοιμα για χρήση και τα οποία αφορούν διαχείριση συνόδου, αυθεντικοποίηση χρηστών, χρησιμοποίηση κρυφής μνήμης και άλλα.

### **Το πλαίσιο εργασίας εφαρμογών ιστού Mojolicious**

Το πλαίσιο εργασίας εφαρμογών ιστού Mojolicious [40], είναι πραγματικού χρόνου (real-time web framework), ανοικτού κώδικα και επιτρέπει την τροποποίηση μεμονωμένων αρχείων πρωτοτύπων σε ισχυρές εφαρμογές ιστού. Το Mojolicious παρέχει αρκετές έτοιμες δυνατότητες όπως υποστήριξη προσθέτων, προτύπων σχεδίασης της Perl, υποστήριξη της αρχιτεκτονικής Restful και διαχείριση συνεδριών.

Επιπλέον το Mojolicious παρέχει μια εργαλειοθήκη για ανάπτυξη λογισμικού στον παγκόσμιο ιστό (η οποία είναι ανεξάρτητη από το ποιο πλαίσιο εργασίας εφαρμογών ιστού θα χρησιμοποιηθεί) με της εξής δυνατότητες: Υλοποίηση πλήρους στοίβας HTTP, υλοποίηση WebSocket για πελάτη/εξυπηρετητή μέσω του πρωτοκόλλου IPv6, υποστήριξη του Transport Socket Layer(TLS), cookies και άλλα. Επίσης, διαθέτει ενσωματωμένο εξυπηρετητή ιστού τύπου non-blocking εισόδου/εξόδου (I/O) με σκοπό τη δημιουργία κλιμακούμενων στην απόδοση υπηρεσιών ιστού. Πρόσθετα, διαθέτει συντακτικό αναλυτή (parser) για JSON και HTML/XML με υποστήριξη για selector της γλώσσας CSS (Cascading Style Sheets). Τέλος το Mojolicious διαθέτει ένα αντικειμενοστρεφές API εύκολο στην κατανόηση, το οποίο εκτελείται σε διαφορετικές πλατφόρμες λογισμικού.

### **Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα PHP**

#### **Το πλαίσιο εργασίας εφαρμογών ιστού Symfony**

Το πλαίσιο εργασίας εφαρμογών ιστού Laravel [41][42], είναι ανοικτού κώδικα και διαθέτει εύκολη στην κατανόηση και εκφραστική σύνταξη. Επίσης, το Laravel διευκολύνει συνηθισμένες εργασίες οι οποίες χρησιμοποιούνται στην πλειοψηφία των εργασιών σχετικών με τον παγκόσμιο ιστό όπως η αυθεντικοποίηση και η διαχείριση συνεδριών.

Επιπλέον, το Laravel διαθέτει επαρκή και ενημερωμένη τεκμηρίωση και οι περισσότερες λειτουργίες του είναι εύκολες στην κατανόηση καθώς και στην υλοποίηση [43]. Οι λειτουργίες του Laravel βασίζονται σε πρότυπα της βιομηχανίας και χρειάζονται λίγες ρυθμίσεις για τη σωστή και αποτελεσματική λειτουργία της εφαρμογής.

#### **Το πλαίσιο εργασίας εφαρμογών ιστού Symfony**

Το πλαίσιο εργασίας εφαρμογών ιστού Symfony [44] είναι φτιαγμένο με βάση τα Symfony Components τα οποία είναι ένα σύνολο ανεξάρτητων και επαναχρησιμοποιήσιμων στοιχείων σε γλώσσα PHP. Το Symfony είναι βασισμένο στην αρχιτεκτονική MVC [45].

Το Symfony υποστηρίζει έξυπνα URL (smart URLs), αυτόματη δημιουργία κώδικα (code generation), εύκολη δημιουργία προτύπων σχεδίασης, αυτόματη επικύρωση στοιχείων διαδικτυακών φορμών και άλλα. Η σχεδίαση μιας εφαρμογής ιστού στο Symfony είναι λίγο

διαφορετική σε σχέση με τα άλλα πλαίσια εργασίας εφαρμογών ιστού αλλά και σε σχέση με τη δημιουργία χωρίς της χρήση αυτών.

Το Symfony περιέχει μια διανομή η οποία περιέχει μια άδεια εφαρμογή και μια βάση δεδομένων SQLite με έτοιμες ρυθμίσεις. Αυτά διατίθενται απλά με τη λήψη της συγκεκριμένης διανομής χωρίς καμιά περαιτέρω ενέργεια. Το Symfony για την εκτέλεσή του χρειάζεται έναν εξυπηρετητή που υποστηρίζει την έκδοση 5 της PHP και να μπορεί ο προγραμματιστής να έχει πρόσβαση στο κέλυφος του εξυπηρετητή (shell).

### **Το πλαίσιο εργασίας εφαρμογών ιστού CodeIgniter**

Το πλαίσιο εργασίας εφαρμογών ιστού CodeIgniter [46], είναι υψηλής απόδοσης και έχει ως στόχο τη δημιουργία πλήρως λειτουργικών εφαρμογών ιστού με πληθώρα χαρακτηριστικών. Το CodeIgniter έχει πολύ μικρό μέγεθος (μόλις 2MB) και στη λήψη του συμπεριλαμβάνεται και ο οδηγός χρήσης του (user guide). Για τη λειτουργία του CodeIgniter χρειάζεται η έκδοση 5.2.4 της PHP και μπορεί να λειτουργήσει στις περισσότερες πλατφόρμες φιλοξενίας εφαρμογών ιστού. Επιπλέον, το CodeIgniter υποστηρίζει τις πιο κοινές βάσεις δεδομένων μεταξύ αυτών και τη MySQL.

Το CodeIgniter είναι ευέλικτο σε θέματα συγγραφής κώδικα και επιτρέπει στον προγραμματιστή να χρησιμοποιήσει δικές του συμβάσεις ονομασίας ή συγγραφής κώδικα. Ο μόνος περιορισμός που θέτει είναι κάποιοι κανόνες για να αντιμετωπιστούν «συγκρούσεις» (conflicts) κλάσεων με το ίδιο όνομα. Επιπλέον, το CodeIgniter συνιστά τη χρήση της αρχιτεκτονικής MVC από τον προγραμματιστή χωρίς όμως να την καθιστά υποχρεωτική.

Το CodeIgniter συνεργάζεται με πρόσθετα φτιαγμένα από τρίτους με σκοπό την επέκταση της λειτουργικότητάς του. Οι ρυθμίσεις που χρειάζονται για τη λειτουργία του CodeIgniter είναι πολύ λίγες διότι υποστηρίζει την τεχνική όπου δίνεται προτεραιότητα στις συμβάσεις σε σχέση με τις ρυθμίσεις. Το CodeIgniter διαθέτει μια ενσωματωμένη εργαλειοθήκη προτύπων σχεδίασης αλλά σε περίπτωση που ο προγραμματιστής χρησιμοποιεί το δικό του, είναι η εφικτή η υποστήριξη άλλων προτύπων σχεδίασης μέσω προσθέτων.

### **Το πλαίσιο εργασίας εφαρμογών ιστού CakePHP**

Το πλαίσιο εργασίας εφαρμογών ιστού CakePHP [47] έχει ως στόχο τη απλή και γρήγορη δημιουργία εφαρμογών ιστού οι οποίες έχουν λίγες γραμμές κώδικα. Το CakePHP είναι σχεδιασμένο για την έκδοση 5.4 της PHP ή νεότερες. Το CakePHP διαθέτει ενσωματωμένη και έτοιμη την πρόσβαση σε βάση δεδομένων, την επικύρωση την αυθεντικοποίηση τη μετάφραση σε διαφορετικές γλώσσες και άλλα.

Το CakePHP δε διαθέτει πολύπλοκα αρχεία ρυθμίσεων της XML ή της γλώσσας σειριοποίησης δεδομένων YAML. Συνεπώς ο προγραμματιστής ρυθμίζει τη βάση δεδομένων και μπορεί να ξεκινήσει να γράφει κώδικα. Βασίζεται στην αρχιτεκτονική MVC. Τέλος το CakePHP δίνει έμφαση στην ασφάλεια επειδή διαθέτει ενσωματωμένα εργαλεία για επικύρωση εισόδου και προστασία από επιθέσεις XSRF (Cross Site Forgery) και SQL Injection.

## **Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα Python**

### **Το πλαίσιο εργασίας εφαρμογών ιστού Django**

Το πλαίσιο εργασίας εφαρμογών ιστού Django [48] είναι υψηλού επιπέδου και βοηθάει στη γρήγορη ανάπτυξη λογισμικού. Το Django είναι ανοικτού κώδικα και διατίθεται δωρεάν. Το Django επίσης παρέχει πολλές έτοιμες λειτουργίες για πολλές συνηθισμένες εργασίες που αφορούν την ανάπτυξη λογισμικού στον παγκόσμιο ιστό. Σε αυτές περιλαμβάνονται η αυθεντικοποίηση του χρήστη, η διαχείριση περιεχομένου, χάρτες ιστότοπων (site maps) και άλλες.

Το Django δίνει έμφαση στην ασφάλεια και βοηθά τον προγραμματιστή να «αποφεύγει» λάθη όπως το SQL Injection, το Cross Site Scripting και το Cross-Site Request Forgery. Το σύστημα που παρέχει για την αυθεντικοποίηση των χρηστών παρέχει έναν ασφαλή τρόπο για τη διαχείριση των ονομάτων των χρηστών (usernames) και των κωδικών πρόσβασης (passwords). Επιπλέον, το Django παρέχει κλιμακούμενη απόδοση ανάλογα με τις απαιτήσεις της εφαρμογής.

### **Το πλαίσιο εργασίας εφαρμογών ιστού TurboGears 2**

Το πλαίσιο εργασίας εφαρμογών ιστού TurboGears2 [49] ξεκίνησε ως micro-framework και πλέον είναι full-stack framework. Η δεύτερη έκδοση του TurboGears βασίστηκε στα πλαίσια εργασίας εφαρμογών ιστού Django, Rails καθώς και στην πρώτη έκδοση του TurboGears.

Η διαδικασία συγγραφής κώδικα στο TurboGears 2 είναι απλή. Επιπλέον, το TurboGears 2 υποστηρίζει πολλές διαφορετικές βάσεις δεδομένων και υποστηρίζει την τεχνική ORM. Το TurboGears 2 διαθέτει ένα σύστημα γραφικών στοιχείων (widgets) ώστε να διευκολύνεται η δημιουργία απαιτητικών εφαρμογών οι οποίες χρησιμοποιούν την τεχνολογία AJAX. Πρόσθετα, το TurboGears 2 υποστηρίζει πολλά μορφώματα ανταλλαγής δεδομένων (data-exchange formats).

Το TurboGears 2 είναι επεκτάσιμο και διαθέτει ένα εύχρηστο σύστημα δημιουργίας προτύπων σχεδίασης. Τέλος το TurboGears 2 είναι συμβατό με τις εκδόσεις της Python: 2.6, 2.7, 3.2, 3.3 και 3.4.

### **Το πλαίσιο εργασίας εφαρμογών ιστού WEB2PY**

Το πλαίσιο εργασίας εφαρμογών ιστού web2py [50] διατίθεται δωρεάν, είναι ανοικτού κώδικα και έχει ως στόχο τη γρήγορη ανάπτυξη εφαρμογών ιστού οι οποίες θα είναι αποδοτικές, κλιμακούμενες, ασφαλείς και θα διαθέτουν βάση δεδομένων.

Το web2py διαθέτει έναν αποδοτικό εξυπηρετητή ο οποίος υποστηρίζει πολλαπλά νήματα (multi-threading). Επιπλέον, το web2py διαθέτει μια βάση δεδομένων SQL και μια διεπαφή βασισμένη στον παγκόσμιο ιστό. Το web2py δεν εξαρτάται από λογισμικό φτιαγμένο από τρίτους, αλλά το υποστηρίζει εφόσον χρειαστεί.

Το web2py διαθέτει ένα ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού (Integrated Development Environment - IDE) το οποίο είναι βασισμένο στον παγκόσμιο ιστό (web-based). Με το προαναφερθέν περιβάλλον ανάπτυξης λογισμικού, ο προγραμματιστής μπορεί να δημιουργήσει να τροποποιήσει και να διαχειριστεί την εφαρμογή του με τη

χρήση ενός φυλλομετρητή. Επιπλέον, κάθε στιγμιότυπο (instance) του web2py μπορεί να εκτελεί περισσότερες από μία ιστοσελίδες, χρησιμοποιώντας διαφορετικές βάσεις δεδομένων. Τέλος, το web2py έχει επαρκή τεκμηρίωση και δραστήρια κοινότητα χρηστών.

## **Πλαίσια εργασίας εφαρμογών ιστού για τη γλώσσα Ruby**

### **Το πλαίσιο εργασίας εφαρμογών ιστού Ruby on Rails**

Το πλαίσιο εργασίας εφαρμογών ιστού Ruby on Rails [51] είναι ανοικτού κώδικα και βασίζεται στην τεχνική όπου δίνεται προτεραιότητα στις συμβάσεις που χρησιμοποιούνται σε σχέση με τις ρυθμίσεις. Το Ruby on Rails βασίζεται στο βέλτιστο τρόπο που μπορεί να γίνει κάποια εργασία και προτείνεται ο προγραμματιστής να ακολουθεί το συγκεκριμένο τρόπο. Ο τρόπος που προτείνεται από το Ruby on Rails δεν είναι υποχρεωτικός, ωστόσο αν ο προγραμματιστής ακολουθήσει διαφορετική μεθοδολογία υπάρχει μεγάλη πιθανότητα να μη φτάσει στον επιθυμητό στόχο, τουλάχιστον όχι άμεσα.

### **Το πλαίσιο εργασίας εφαρμογών ιστού Sinatra**

Το πλαίσιο εργασίας εφαρμογών ιστού Sinatra [52], είναι μια γλώσσα για χρήση σε ένα συγκεκριμένο τομέα εφαρμογών (Domain Specific Language- DSL) με σκοπό τη δημιουργία εφαρμογών ιστού γρήγορα και αποτελεσματικά.

### **Το πλαίσιο εργασίας εφαρμογών ιστού Cuba**

Το πλαίσιο εργασίας εφαρμογών ιστού Cuba [53] είναι ένα micro-framework για ανάπτυξη εφαρμογών στον παγκόσμιο ιστό και διαθέτει πολλά πρότυπα σχεδίασης μέσω του Tilt[54] το οποίο είναι μια γενική διεπαφή η οποία συνδυάζει διαφορετικά πρότυπα σχεδίασης που είναι φτιαγμένα στη γλώσσα Ruby. Ο σκοπός του Tilt είναι να γενικεύσει αυτά τα πρότυπα σχεδίασης έτσι ώστε αυτά να μπορούν να χρησιμοποιηθούν σε περισσότερες από μία εφαρμογές του παγκόσμιου ιστού.

### **Το πλαίσιο εργασίας εφαρμογών ιστού ASP.NET**

Το πλαίσιο εργασίας εφαρμογών ιστού ASP.NET [55] διατίθεται δωρεάν από την εταιρεία Microsoft και έχει ως σκοπό τη δημιουργία εφαρμογών ιστού και ιστότοπων με τη χρήση HTML, CSS και JavaScript. Επιπλέον με το ASP.NET είναι δυνατή η δημιουργία API σχετικά με τον παγκόσμιο ιστό (WEB APIs), ιστότοπων σχεδιασμένων για κινητές συσκευές, η χρήση τεχνολογιών πραγματικού χρόνου όπως η Web Sockets και άλλα.

Για τη δημιουργία ιστότοπων χρησιμοποιούνται τρεις διαφορετικές προσεγγίσεις: η προσέγγιση ASP.NET Web Forms χρησιμοποιεί ελέγχους και ένα μοντέλο βασισμένο σε γεγονότα (event-model) με σκοπό την ανάπτυξη του λογισμικού βασισμένη σε στοιχεία (component-based development). Η προσέγγιση ASP.NET MVC βασίζεται στο διαχωρισμό των «ευθυνών» και έτσι βοηθάει στην ανάπτυξη λογισμικού οδηγούμενη από δοκιμές (test-driven development). Η προσέγγιση ASP.NET Web Pages δίνει έμφαση σε ένα μοντέλο στο οποίο υπάρχει μια μεμονωμένη σελίδα όπου συνδυάζεται ο κώδικας με τη σήμανση της HTML. Αναλόγως τις ανάγκες του, ο προγραμματιστής μπορεί να συνδυάσει αυτές τις τεχνικές εφόσον χρειάζεται.

Θεωρητική ανασκόπηση των πλαισίων εργασίας εφαρμογών ιστού και πρακτική εφαρμογή στα Spring και Struts2.

Το ASP.NET περιλαμβάνει το ASP.NET Web API για τη δημιουργία υπηρεσιών ιστού που βασίζονται στην αρχιτεκτονική Restful οι οποίες επιστρέφουν δεδομένα σε μορφές αναπαράστασης συμβατές με τον παγκόσμιο ιστό όπως JSON. Επιπλέον, το ASP.NET Web API μπορεί να παρέχει υπηρεσίες δεδομένων σε εφαρμογές για κινητές συσκευές όπως για Android, iOS και Windows Phone. Τα ASP.NET Web API μπορούν να χρησιμοποιηθούν σε οποιαδήποτε εφαρμογή για τον παγκόσμιο ιστό φτιαγμένη με το ASP.NET.

Η βιβλιοθήκη λογισμικού ASP.NET SignalR που παρέχει το ASP.NET καθιστά εφικτή τη δημιουργία αμφίδρομης επικοινωνίας μεταξύ πελάτη-εξυπηρετητή σε πραγματικό χρόνο (real-time). Η βιβλιοθήκη SignalR υποστηρίζει καινούργια πρότυπα όπως το WebSocket, διατηρώντας ωστόσο τη συμβατότητα με παλαιότερες εκδόσεις φυλλομετρητών. Τέλος, η SignalR υποστηρίζει clients που έχουν λειτουργικό Android ή iOS, clients σε γλώσσα C# όπως Windows Phone και Windows 8 και clients σε γλώσσα JavaScript.

Το ASP.NET παρέχει στις εφαρμογές για κινητές συσκευές πλαίσια εργασίας εφαρμογών για responsive design όπως το Twitter Bootstrap. Τέλος, το ASP.NET και το Visual Studio το οποίο είναι ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού, παρέχουν δυνατότητες όπως ενσωματωμένοι επεξεργαστές (editors) για τις HTML5 και CSS3 καθώς και ανέβασμα της εφαρμογής ιστού μέσα από το Visual Studio.

## ΚΕΦΑΛΑΙΟ 4

### ΤΟ ΠΛΑΙΣΙΟ ΕΡΓΑΣΙΑΣ ΕΦΑΡΜΟΓΩΝ ΙΣΤΟΥ STRUTS2

#### Παρουσίαση του πλαισίου εργασίας εφαρμογών ιστού Struts 2

Το πλαίσιο εργασίας εφαρμογών ιστού Struts 2, υποστηρίζει τη δημιουργία εφαρμογών ιστού επιχειρησιακού επιπέδου σε γλώσσα Java οι οποίες είναι βασισμένες στην αρχιτεκτονική MVC [56][57]. Το Struts 2 διατίθεται δωρεάν και είναι ανοικτού κώδικα [58]. Το πλαίσιο εργασίας εφαρμογών Struts δημιουργήθηκε το 2000 και η έκδοση 1 κυκλοφόρησε το 2001 [59]. Η έκδοση 2 του Struts κυκλοφόρησε το 2007 και είναι βασισμένη στην 1 έκδοση του Struts και στο πλαίσιο εργασίας εφαρμογών ιστού WebWork.

#### Πλεονεκτήματα και μειονεκτήματα του Struts 2

Το Struts 2 διαθέτει τα ακόλουθα χαρακτηριστικά:

- POJO Actions

Στο Struts 2 οι κλάσεις Java που υλοποιούν κάποια ενέργεια στην εφαρμογή (Action classes) είναι τύπου Plain Old Java Objects – POJOs. Αυτό σημαίνει ότι είναι απλές κλάσεις της Java χωρίς να είναι απαραίτητο να κληρονομούν άλλες κλάσεις ή να υλοποιούν μια διεπαφή (interface).

- Υποστήριξη της τεχνολογίας AJAX

Το Struts 2 υποστηρίζει την τεχνολογία AJAX με σκοπό την αύξηση της απόδοσης της εφαρμογής. Για το σκοπό αυτό παρέχει σχετικά tags για AJAX, παρόμοια με αυτά του Struts 2.

- Συνεργασία με άλλα πλαίσια εργασίας εφαρμογών ιστού

Το Struts 2 μπορεί να ενσωματώσει σε μια εφαρμογή που είναι κατασκευασμένη με αυτό άλλα πλαίσια εργασίας εφαρμογών ιστού όπως τα Spring και Hibernate.

- Υποστήριξη πολλών τύπων σχετικών με αποτελέσματα

Το Struts 2 ως αποτέλεσμα μια ενέργειας στην εφαρμογή, παρέχει μια σελίδα σχετική με τα αποτελέσματα (results) στο επίπεδο του View. Η σελίδα αυτή μπορεί να είναι τύπου JSP, Freemarker ή Velocity.

- Υποστήριξη δικών του tags (Struts 2 tags)

Το Struts 2 αποτελείται από ένα σύνολο δικών του tags με σκοπό τη διευκόλυνση της δημιουργίας μιας εφαρμογής ιστού. Τα tags αυτά μπορεί να είναι σχετικά με το User Interface, με δεδομένα (data tags) και άλλα.

- Υποστήριξη επικύρωσης δεδομένων

Το Struts 2 διαθέτει ενσωματωμένο μηχανισμό επικύρωσης δεδομένων που είναι επαναχρησιμοποιήσιμος.

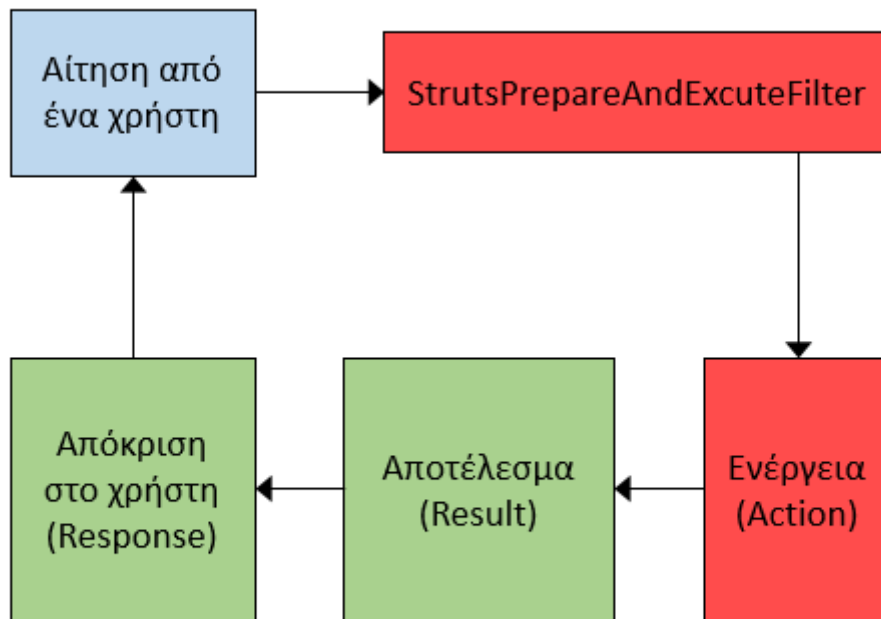
- Υποστήριξη προεπιλεγμένων ρυθμίσεων και συμβάσεων (conventions)

Το Struts 2 διαθέτει τις πιο κοινές προεπιλεγμένες ρυθμίσεις έτοιμες ώστε να μην απαιτούνται χρονοβόρες και πολλές πρόσθετες αλλαγές. Επίσης μπορούν να χρησιμοποιηθούν ρυθμίσεις βασισμένες στις συμβάσεις (convention based configuration), όπου είναι ακόμα πιο απλός τρόπος και δεν απαιτείται το αρχείο struts.xml.

## **Λειτουργία του Struts 2**

Η λειτουργία του Struts 2 παρουσιάζεται συνοπτικά στο παρακάτω σχήμα: (Εικόνα 2)





**Εικόνα 2: Η λειτουργία του Struts2**

Ένας χρήστης αιτείται κάποια ενέργεια σε μια εφαρμογή π.χ. σύνδεση (login). Η αίτηση αυτή επεξεργάζεται από το StrutsPrepareAndExcuteFilter που είναι ο Controller και αντιστοιχίζει την αίτηση σε μια ενέργεια (Action). Η ενέργεια αφορά την επιχειρησιακή λογική (business logic) και μπορεί να χειριστεί ένα αντικείμενο, ή να εκτελέσει ένα ερώτημα σε μια βάση δεδομένων. Το αποτέλεσμα (result) της ενέργειας μπορεί να είναι μια σελίδα JSP και το αποτέλεσμα επιστρέφεται ως απόκριση προς το χρήστη (response).

### **Δημιουργία εφαρμογής στο Struts2**

Στην ενότητα αυτή αναφέρονται τα στοιχεία λογισμικού που απαιτούνται για τη δημιουργία μιας εφαρμογής ιστού στο Struts2 καθώς και γενικά στοιχεία για τον τρόπο λειτουργίας της εφαρμογής ιστού.

Καταρχήν, χρειάζεται το Java Development Kit (JDK) το οποίο είναι διαθέσιμο στην ιστοσελίδα: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>. Επιπλέον, χρειάζεται ο Apache Tomcat στον οποίον θα εκτελούνται οι εφαρμογές. Ο Apache Tomcat είναι διαθέσιμος για λήψη από την ιστοσελίδα: <http://tomcat.apache.org/download-80.cgi>. Χρειάζεται και ένα Java IDE όπως το Eclipse το οποίο είναι διαθέσιμο για λήψη από την ιστοσελίδα: <http://www.eclipse.org/downloads/> Πρέπει να επιλεγεί το Eclipse IDE for Java EE Developers. Τέλος, χρειάζεται το Struts 2 το οποίο διατίθεται για λήψη στην ιστοσελίδα: <https://struts.apache.org/download.cgi>. Για την παρούσα εφαρμογή επιλέχθηκε το Struts 2.3.24.3 (η επιλογή Full Distribution struts-2.3.24.3-

all.zip ). Το αρχείο .zip με το Struts 2 πρέπει να αποσυμπεστεί σε ένα φάκελο της επιλογής του προγραμματιστή.

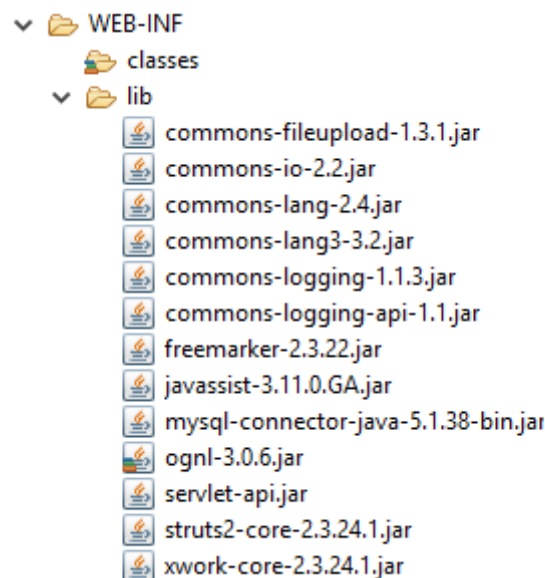
Για να αρχίσει η δημιουργία μιας εφαρμογής ιστού σε Struts2 πρέπει να ανοίξει ο χρήστης το Eclipse IDE και να πάει στο μενού File -> New -> Dynamic Web Project. Δίνεται ένα όνομα στο project (π.χ. HelloStruts2) και πατιέται Next (αφήνοντας τα υπόλοιπα ως έχουν). Πρέπει να επιλεγθεί στην επόμενη οθόνη η επιλογή Generate Web.xml deployment descriptor. Τέλος πατιέται το Finish.

### Δημιουργία εφαρμογής online quiz στο Struts2

Για να γίνει επίδειξη της λειτουργίας του Struts2, δημιουργήθηκε μια εφαρμογή online quiz η οποία θα είναι διαθέσιμη μόνο σε εγγεγραμμένους χρήστες που θα συνδέονται στην εφαρμογή και θα κάνουν το quiz. Θα υπάρχει φόρμα εγγραφής, καθώς και σύνδεσης στην εφαρμογή όπου θα ελέγχονται τα στοιχεία που δίνει ο χρήστης. Θα υπάρχει button logout. Η σελίδα με τις ερωτήσεις θα έχει έναν υπερσύνδεσμο που θα οδηγεί στην επόμενη σελίδα. Τέλος ,θα υπάρχει σελίδα εμφάνισης των αποτελεσμάτων του quiz που έκανε ο χρήστης.

Αρχικά, χρειάζονται όλα τα στοιχεία λογισμικού που αναφέρθηκαν στην προηγούμενη ενότητα καθώς και μια βάση δεδομένων. Στην παρούσα εφαρμογή χρησιμοποιήθηκε η βάση δεδομένων MySQL. Η MySQL είναι διαθέσιμη για λήψη στην ιστοσελίδα: <http://www.mysql.com/downloads/> . Για τις ανάγκες της εφαρμογής έγινε λήψη του MySQL Workbench για το λειτουργικό σύστημα Windows το οποίο είναι διαθέσιμο για λήψη από την ιστοσελίδα: <http://dev.mysql.com/downloads/windows/>.

Στη συνέχεια, στο Eclipse IDE δημιουργήθηκε ένα νέο Dynamic Web Project (όπως στην προηγούμενη ενότητα) με όνομα Struts2Quiz. Μετά στο φάκελο WEB-INF δημιουργούμε ένα φάκελο με όνομα lib και προσθέτουμε κάποια από τα .jar του struts2. Για τις ανάγκες της εφαρμογής χρειάστηκαν αυτά που φαίνονται στην Εικόνα 3:



Εικόνα 3: Οι βιβλιοθήκες λογισμικού για την εφαρμογή

Το αρχείο .jar για τη MySQL βρίσκεται στην ιστοσελίδα <https://dev.mysql.com/downloads/connector/j/> και το αρχείο commons-logging-api-1.1

βρίσκεται στην ιστοσελίδα: [https://commons.apache.org/proper/commons-logging/download\\_logging.cgi](https://commons.apache.org/proper/commons-logging/download_logging.cgi)

Στο αρχείο web.xml, προστέθηκε το φίλτρο (filter) του Struts 2 το οποίο λειτουργεί ως Controller. Το φίλτρο αυτό, τοποθετείται ανάμεσα σε tags τύπου <filter-class> και έχει όνομα: org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter. Το αρχείο web.xml πρέπει να δείχνει όπως στην Εικόνα 4:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://java.sun.
<welcome-file-list>
  <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
<filter>
  <filter-name>struts2</filter-name>
  <filter-class>
    org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter
  </filter-class>
</filter>
<filter-mapping>
  <filter-name>struts2</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
</web-app>
```

Εικόνα 4: Το αρχείο web.xml της εφαρμογής

Στη συνέχεια, δημιουργούμε το αρχείο struts.xml το οποίο θα βρίσκεται στο φάκελο src. Το αρχείο struts.xml δίνει πληροφορίες για την ενέργεια που θα εκτελεστεί (action) και με βάση ένα αποτέλεσμα (π.χ. success) καθορίζει ποια σελίδα θα εμφανιστεί στο επίπεδο του View. Το αρχείο struts.xml έχει τα ακόλουθα tags: Το πρώτο είναι το <struts> το οποίο είναι το βασικό στοιχείο (root file) του struts.xml. Το δεύτερο tag είναι το <package> το οποίο περιγράφει ένα δομικό στοιχείο της εφαρμογής. Συνήθως επεκτείνει (extends) το πακέτο struts-default όπου ορίζονται interceptors και τύποι αποτελεσμάτων (result types).

Το τρίτο tag είναι το <action> το οποίο είναι υποστοιχείο του <package>. Το <action> αντιπροσωπεύει το Action που θα εκτελεστεί για μια συγκεκριμένη αίτηση. Το <action> περιλαμβάνει τα στοιχεία name, class και method. Αν ο προγραμματιστής δεν προσδιορίσει όνομα για το Action, τότε η μέθοδος που θα εκτελεστεί για το συγκεκριμένο Action είναι η execute(). Το τέταρτο tag είναι το result το οποίο είναι υποστοιχείο του <action>. Το <result> καθορίζει ανάλογα το αποτέλεσμα ποια σελίδα θα εμφανιστεί στο επίπεδο του view για ένα συγκεκριμένο Action. Στην Εικόνα 5 δίνεται ένα μέρος από το αρχείο struts.xml της εφαρμογής:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE struts PUBLIC "-//Apache Software Foundation//DTD
struts Configuration 2.1//EN" "http://struts.apache.org/dtds/struts-2.1.dtd">
<struts>

  <package name="default" extends="struts-default">

    <action name="register" class="com.quiz.struts2.RegisterFullAction">
      <result name="success">login.jsp</result>
      <result name="error">error.jsp</result>
      <result name="input">register.jsp</result>
    </action>

    <action name="loginprocess" class="com.quiz.struts2.LoginFull">
      <result name="success">main.jsp</result>
      <result name="error">loginerror.jsp</result>
    </action>

    <action name="profilepage" class="com.quiz.struts2.ProfilePage">
      <result name="success">profilepage.jsp</result>
      <result name="login">profileerror.jsp</result>
    </action>

    <action name="logout" class="com.quiz.struts2.LoginFull"
      method="logout">
      <result name="logout-success">logoutsuccess.jsp</result>
    </action>
```

### Εικόνα 5: Το αρχείο struts.xml της εφαρμογής

Παρατηρείται ότι η δομή του αρχείου struts.xml είναι όπως περιγράφηκε παραπάνω. Το Action για την εγγραφή θα έχει όνομα register και η κλάση Java που θα υλοποιεί το Action αναφέρεται στην κλάση RegisterFullAction. Αν η εγγραφή είναι επιτυχής (name=success) ο χρήστης παραπέμπεται στη σελίδα σύνδεσης login.jsp. Αν υπάρχει σφάλμα (name=error) ή δεν επικυρωθούν τα στοιχεία του στη φόρμα (name=input), τότε θα παραπεμφθεί στις σελίδες error.jsp και register.jsp(με σχετικό μήνυμα) αντίστοιχα.

Επίσης, για τη λειτουργικότητα της εγγραφής χρήστη (user registration) δίνονται οι κλάσεις που υλοποιούν την αντίστοιχη λειτουργικότητα: Η κλάση RegisterFullAction, υλοποιεί τη σχετική λειτουργικότητα (Εικόνα 6):

```
package com.quiz.struts2;
import com.opensymphony.xwork2.ActionSupport;

public class RegisterFullAction extends ActionSupport
{
    // Private variables that represent the DB fields.
    private String name,password,email;
    private int age;

    public void validate()
    {
        if(name.length()<1)
            addActionError("Please enter your username. It can't be blank.");
        if(password.length()<3)
            addActionError("Weak password given. Must be 3 or more characters.");
    }
    // Getters and setters methods for each field.
    public String getName()
    {
        return name;
    }

    public void setName(String name)
    {
        this.name = name;
    }

    public String getPassword()
    {
        return password;
    }

    public void setPassword(String password)
    {
        this.password = password;
    }
}
```

**Εικόνα 6: Η κλάση RegisterFullAction**

Παρατηρούμε ότι η κλάση RegisterFullAction επεκτείνει την κλάση ActionSupport του Struts2. Η κλάση ActionSupport υλοποιεί πολλές διεπαφές όπως τις Action, Validateable και TextProvider. Γενικότερα στο Struts2 μια κλάση που αντιπροσωπεύει ένα Action, συνήθως υλοποιείται επεκτείνοντας την κλάση ActionSupport. Η κλάση RegisterFullAction έχει τα πεδία name, password και email τύπου String και age τύπου int. Η μέθοδος validate() αφορά την επικύρωση του name και του password τα οποία δίνονται από τον χρήστη στη φόρμα εγγραφής. Αν υπάρξει σφάλμα επικύρωσης επιστρέφεται input και βάσει του αρχείου struts.xml δίνεται το αντίστοιχο View όπου θα φαίνεται η συμβολοσειρά της μεθόδου addActionError.

Υπάρχουν στη συνέχεια οι μέθοδοι getters και setters για ανάγνωση και εισαγωγή τιμών στα πεδία της κλάσης. Τέλος, στη Εικόνα 7, δίνεται η μέθοδος execute():

```

public String execute()
{
    // If the action object returns 1
    // return success, else return
    // error for zero(0).
    int i=RegisterDaoFull.save(this);
    if(i>0)
    {
        return "success";
    }
    return "error";
}
}

```

Εικόνα 7: Η μέθοδος execute της κλάσης RegisterFullAction

Αν γίνει επιτυχής καταχώρηση του χρήστη στη βάση δεδομένων μέσω της μεθόδου save της κλάσης RegisterDaoFull τότε η μεταβλητή i παίρνει την τιμή 1 και επιστρέφεται success, αλλιώς παίρνει την τιμή 0 και επιστρέφεται error. Οι τιμές αυτές χρησιμοποιούνται από το αρχείο struts.xml όπως εξηγήθηκε παραπάνω.

Τέλος, στην Εικόνα 8 δίνεται η κλάση RegisterDaoFull όπου γίνεται η σύνδεση και ενημέρωση της βάσης δεδομένων:

```

package com.quiz.struts2;
// SQL necessary imports.
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

public class RegisterDaoFull
{

public static int save(RegisterFullAction r)
{
    int status = 0;
    Connection conn = null;
    try
    {
        String URL = "jdbc:mysql://localhost:3306/struts_register";
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(URL, "root", "admin1235!");
        // Connection with the DB using URL, username and password.
        PreparedStatement ps=conn.prepareStatement("insert into struts_users values(?,?,?,?)");
        // Set the values.
        ps.setString(1,r.getName());
        ps.setString(2,r.getPassword());
        ps.setString(3,r.getEmail());
        ps.setInt(4,r.getAge());
        // ExecuteUpdate method returns either (1) the row count
        // for SQL Data Manipulation Language
        // (DML) statements or (2) 0 for SQL statements that return nothing.
        status = ps.executeUpdate();
    }

catch(Exception e)
{
    e.printStackTrace();
}

    return status;
}
}

```

Εικόνα 8: Η κλάση RegisterDaoFull

Η κλάση RegisterDaoFull διαχειρίζεται την αλληλεπίδραση με τη βάση δεδομένων. Υπάρχουν οι μεταβλητές status τύπου int και conn τύπου Connection. Στο block του try δίνονται τα στοιχεία για σύνδεση με τη βάση δεδομένων και οι δηλώσεις SQL που απαιτούνται για την καταχώρηση του χρήστη. Αν δεν προκύψει σφάλμα, τότε η status γίνεται 1 αλλιώς εκτελείται το block catch και η status γίνεται 0.

Στη συνέχεια δίνεται ο κώδικας της .jsp σελίδας για εγγραφή χρήστη register.jsp (Εικόνα 9):

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@ taglib uri="/struts-tags" prefix="s" %>
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
    <title>MyQuiz-Register</title>
    <style>
        .errorDiv
        {
            color:red;
        }
    </style>
</head>
<body>
    <s:if test="hasActionErrors()">
    <div class="errorDiv">
        <s:actionerror/>
    </div>
    </s:if>
    <s:form action="register">
    <s:textfield name="name" label="Username"></s:textfield>
    <s:password name="password" label="Password"></s:password>
    <s:textfield name="email" label="Email"></s:textfield>
    <s:textfield name="age" label="Age"></s:textfield>
    <s:submit cssStyle="float:left;" value="Register"></s:submit>
    </s:form>
</body>
</html>
```

**Εικόνα 9: Η σελίδα register.jsp**

Για τη φόρμα χρησιμοποιήθηκαν τα tags του Struts2 μέσω της οδηγίας taglib για τα struts tags. Μέσα στο tag body, φαίνεται ότι ελέγχεται μέσω του tag if για το Struts2(<s:if>, αν υπάρχουν λάθη. Αυτό γίνεται με τον έλεγχο της συνθήκης hasActionErrors() και μόνο αν έχει λάθη η φόρμα εμφανίζεται το errorDiv. Αλλιώς εμφανίζεται η φόρμα μέσω του Struts2 tag <s:form> η οποία έχει ως Action το register και υπάρχουν τα αντίστοιχα tags του struts μέσα στο tag <s:form>.

Η αρχική σελίδα της εφαρμογής είναι η index.jsp (welcome-file στο web.xml) και εμφανίζεται όταν ξεκινά να εκτελείται η εφαρμογή στον Tomcat(Εικόνα 10):



**Welcome guest. Please [Register](#) or [Login](#).**

©MyQuiz 2016.

### Εικόνα 10: Η αρχική σελίδα της εφαρμογής

Ο μηχανισμός σύνδεσης(login) του χρήστη στην εφαρμογή, υλοποιείται αντίστοιχα σε δύο κλάσεις μια που ελέγχει τα στοιχεία από τη βάση δεδομένων και μία άλλη που υλοποιεί τη σύνδεση και διαχειρίζεται τη σύνοδο(session) μέσω της υλοποίησης της διεπαφής του Struts2, της SessionAware. Η κλάση login βασίζεται στο να υλοποιεί τη διεπαφή SessionAware (implements SessionAware). Επίσης διαθέτει ένα αντικείμενο τύπου Map της Java το SessionMap το οποίο δηλώνεται ως εξής: `SessionMap<String,String> sessionmap;`. Υπάρχουν getters και setters μέθοδοι αντίστοιχα για τις μεταβλητές τύπου String username και password.

Τέλος, υπάρχουν τρεις μέθοδοι η execute(), η setSession() και η logout(). Στη μέθοδο execute() ελέγχεται αν το όνομα χρήστη αντιστοιχεί με κάποια εγγραφή στη βάση δεδομένων και τοποθετείται στη σύνοδο η μεταβλητή username μέσω της εντολής `sessionmap.put("username", username);`. Μετά αυτές τις δύο εντολές επιστρέφεται το String success. Σε περίπτωση που το login δεν επιτευχθεί, επιστρέφεται το String error.

Η μέθοδος setSession() καταχωρεί στη σύνοδο τη μεταβλητή login ως true για να παραμείνει ο χρήστης συνδεδεμένος. Αυτό γίνεται μέσω της εντολής: `sessionmap.put("login","true");`. Τέλος η μέθοδος logout() είναι υπεύθυνη για την αποσύνδεση του χρήστη. Καταρχήν αλλάζει τιμή η μεταβλητή login της προηγούμενης εντολής σε false και το αντικείμενο με όνομα sessionmap ακυρώνει τη σύνοδο μέσω της μεθόδου invalidate() που διαθέτει (`sessionmap.invalidate();`). Τέλος, η μέθοδος logout επιστρέφει τη συμβολοσειρά logout-success.

Αν ο χρήστης συνδεθεί επιτυχώς, τότε εμφανίζεται η σελίδα main.jsp όπου εμφανίζει το όνομα του χρήστη και τρεις υπερσυνδέσμους τους Profile, Take the quiz και Logout. Στη σελίδα main.jsp έχουν συμπεριληφθεί τα tags του Struts2 και με βάση το session που έχει δημιουργηθεί με το επιτυχές login το όνομα χρήστη εμφανίζεται μέσω της εντολής: `<s:property value="#session.username" />`. Οι υπερσύνδεσμοι ορίζουν το αντίστοιχο Action μέσω του Struts2 tag π.χ. για το profile είναι: `<s:url action="profilepage" />`.

Αν ο χρήστης πατήσει logout τότε η σύνοδος λήγει και επιστρέφεται στο χρήστη η αρχική σελίδα. Αν ο χρήστης πατήσει take the quiz τότε αρχίζει το quiz με ένα μέρος των ερωτήσεων όπου επιλέγει τη σωστή απάντηση. Μέσω του button next μεταφέρεται στις υπόλοιπες ερωτήσεις και όταν πατήσει Submit φαίνονται η επιλογή του καθώς και αν απάντησε σωστά ή όχι σε κάθε ερώτηση. Σε κάθε σελίδα με ερωτήσεις, καθώς και σε αυτή με τα αποτελέσματα υπάρχει ένα link μέσω του οποίου ο χρήστης μπορεί να επιστρέψει στη σελίδα main.jsp.

Επίσης η σελίδα του προφίλ χρήστη δεν είναι ορατή σε μη εγγεγραμμένους ή μη συνδεδεμένους χρήστες. Αυτό επιτυγχάνεται μέσω του ελέγχου του αν υπάρχει ενεργή



σύννοδος για έναν χρήστη. Σε αυτό το σημείο πρέπει να γίνουν import οι βιβλιοθήκες `javax.servlet.http.HttpSession` και `javax.servlet.http.HttpServletRequest`. Μέσω σχετικών μεθόδων γίνεται λήψη του `HttpSession` και μέσω της εντολής `String s=(String)session.getAttribute("login");`, λαμβάνεται η τιμή της μεταβλητής `login`. Αν είναι `null` ή κενή τότε επιστρέφεται το `String login` και ο χρήστης δεν βλέπει τη σελίδα `profile` και εμφανίζεται η σελίδα `login.jsp`. Αν υπάρχει σύννοδος επιστρέφεται το `String success` και η σελίδα `profilepage.jsp` εμφανίζεται στο χρήστη.

Ο μηχανισμός με τον οποίο υλοποιείται το quiz περιλαμβάνει την κλάση `Quiz` όπου υλοποιεί τη διεπαφή `SessionAware` για να διατηρούνται στη σύννοδο οι απαντήσεις του χρήστη. Επίσης συνδέεται με τη βάση δεδομένων για ληφθούν οι τιμές από τους πίνακες που έχουν τις ερωτήσεις και βρίσκονται στη βάση δεδομένων `MySQL`. Επίσης ελέγχεται αν η τιμή που βρίσκεται στη σύννοδο που αντιστοιχεί στην απάντηση του χρήστη αντιστοιχεί στη σωστή απάντηση που υπάρχει στη βάση δεδομένων. Αν ο χρήστης δώσει σωστή απάντηση, τότε εμφανίζεται το κείμενο `Correct`, αλλιώς εμφανίζεται το κείμενο `Wrong` στη σελίδα των αποτελεσμάτων. Τέλος υπάρχουν οι κλάσεις `QuizAnswers` και `QuizOptions` που αντιστοιχούν στους πίνακες της βάσης δεδομένων που έχουν τις σωστές απαντήσεις και τις ερωτήσεις με τις πιθανές επιλογές τους. Οι κλάσεις αυτές περιλαμβάνουν τα αντίστοιχα πεδία καθώς και μεθόδους `getters` και `setters` για τα πεδία αυτά.

Στην Εικόνα 11 παρουσιάζεται η πρώτη σελίδα με τις ερωτήσεις και στην Εικόνα 12 παρουσιάζεται η σελίδα των αποτελεσμάτων:

[Back to main page.](#)

## Questions

Category: Geography

1: What is the capital of France?

Madrid  Paris  Marseilles  Monte Carlo

2: What is the capital of Qatar?

Cairo  City of Qatar  Doha  Rabat

3: In which South American country is the Atacama desert?

Chile  Brazil  Argentina  Bolivia

4: What is the largest country in the world?

USA  China  Canada  Russia

Submit

Εικόνα 11: Η πρώτη σελίδα των ερωτήσεων

Your selection on question 1: Madrid **Wrong**  
Your selection on question 2 : Cairo **Wrong**  
Your selection on question 3: Chile **Correct**  
Your selection on question 4 : USA **Wrong**  
Your selection on question 5: Nebraska **Wrong**  
Your selection on question 6 : Qatar **Wrong**

[Back to main page.](#)

Εικόνα 12: Σελίδα των αποτελεσμάτων

## ΚΕΦΑΛΑΙΟ 5

### ΤΟ ΠΛΑΙΣΙΟ ΕΡΓΑΣΙΑΣ ΕΦΑΡΜΟΓΩΝ ΙΣΤΟΥ SPRING

#### Παρουσίαση του πλαισίου εργασίας εφαρμογών ιστού Spring

Το πλαίσιο εργασίας εφαρμογών ιστού Spring είναι μια ανοικτού κώδικα πλατφόρμα σε γλώσσα Java. Το Spring παρέχει την υποδομή για τη δημιουργία ισχυρών εφαρμογών σε Java εύκολα και γρήγορα [59]. Το Spring κυκλοφόρησε το 2003 και η τρέχουσα σταθερή έκδοση είναι η 4.2.6 [60].

#### Χαρακτηριστικά του Spring

Το Spring είναι το πιο δημοφιλές πλαίσιο εργασίας εφαρμογών ιστού σε Java και βοηθάει στη δημιουργία κώδικα επαναχρησιμοποίησης, υψηλής απόδοσης και εύκολο να δοκιμαστεί η σωστή λειτουργία του.

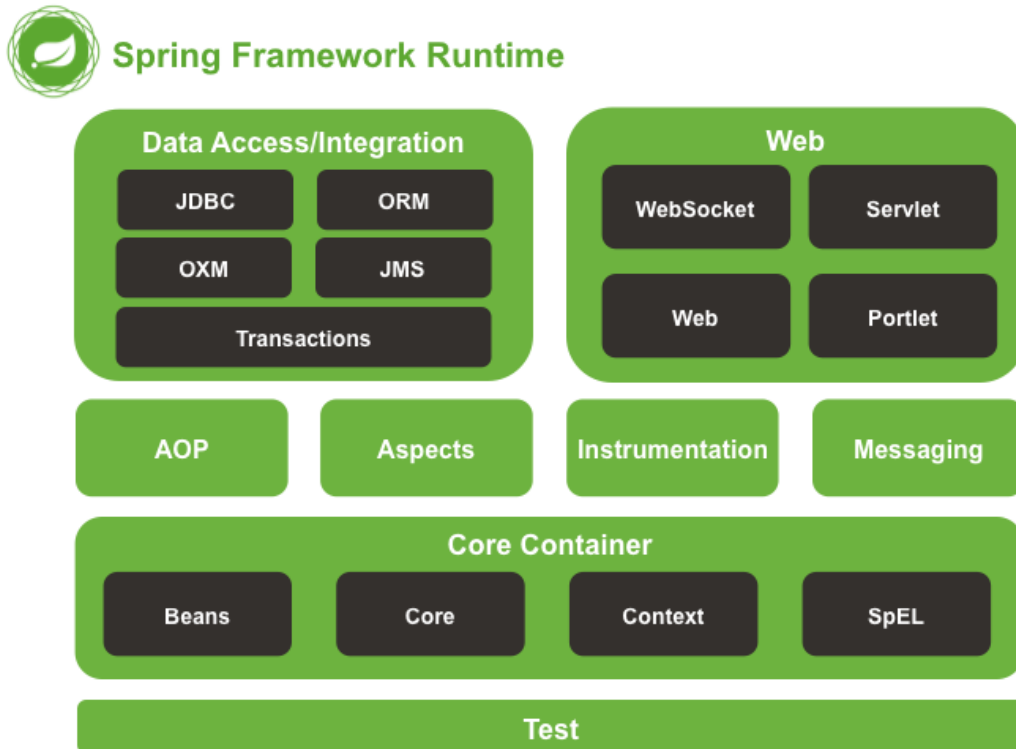
Το Spring βασίζεται σε Plain Old Java Objects – POJOs. Αυτό σημαίνει ότι δε χρειάζεται ένας Enterprise Java Bean- EJB container συνεπώς οι εφαρμογές σε Spring μπορούν να εκτελεστούν σε έναν container για servlets όπως ο Apache Tomcat. Επίσης τα POJOs είναι απλές κλάσεις της Java χωρίς να είναι απαραίτητο να κληρονομούν άλλες κλάσεις ή να υλοποιούν μια διεπαφή(interface). Η δομημένη οργάνωση του Spring βοηθάει στην λήψη μόνο των απαραίτητων δομικών μονάδων (modules) του Spring χωρίς να χρειάζεται να συμπεριληφθούν οι υπόλοιπες.

Το Spring χρησιμοποιεί υπάρχουσες τεχνολογίες όπως πλαίσια εργασίας λογισμικού για Object Relational Mapping και για καταγραφή (logging). Το στοιχείο του Spring που αφορά τον παγκόσμιο ιστό (Spring Web MVC) αποτελεί μια εναλλακτική λύση για το Struts ή άλλα πλαίσια εργασίας εφαρμογών ιστού λιγότερο δημοφιλή.

Το Spring βασίζεται στην τεχνική του Dependency Injection. Στόχος της τεχνικής αυτής είναι να μην υπάρχουν εξαρτήσεις μεταξύ των κλάσεων μιας εφαρμογής. Το Dependency Injection βοηθάει στη συνεργασία μεταξύ διαφορετικών κλάσεων αλλά παράλληλα αυτές είναι προγραμματιστικά ανεξάρτητες.

#### Αρχιτεκτονική του Spring

Η αρχιτεκτονική του Spring παρουσιάζεται στην Εικόνα 13 [61]:



**Εικόνα 13: Η αρχιτεκτονική του Spring**

Το επίπεδο του Test αφορά την υποστήριξη εκτέλεσης δοκιμών για τα στοιχεία του Spring με το Junit ή το TestNG.

Το επίπεδο του Core Container περιλαμβάνει το στοιχείο Core που αφορά τα βασικά μέρη του πλαισίου εργασία εφαρμογών ιστού όπως το Dependency Injection και το Inversion of Control. Το στοιχείο BeanFactory αποτελεί υλοποίηση της τεχνικής σχεδίασης λογισμικού Factory. Το στοιχείο Context είναι ένα μέσο πρόσβασης στα αντικείμενα που έχουν οριστεί. Τέλος το στοιχείο SpEL είναι μια γλώσσα έκφρασης (Expression Language) και αποτελεί επέκταση της γλώσσας έκφρασης που ορίζεται στην προδιαγραφή 2.1 των JSP.

Το επίπεδο του Aspect Oriented Programming αποτελεί υλοποίηση αυτής της τεχνικής προγραμματισμού που έχει ως σκοπό τη μείωση των εξαρτήσεων στον κώδικα.

Το επίπεδο του Data Access/Integration αποτελείται από τα στοιχεία JDBC, ORM, OXM, JMS και Transaction τα οποία αφορούν την αλληλεπίδραση με τη βάση δεδομένων.

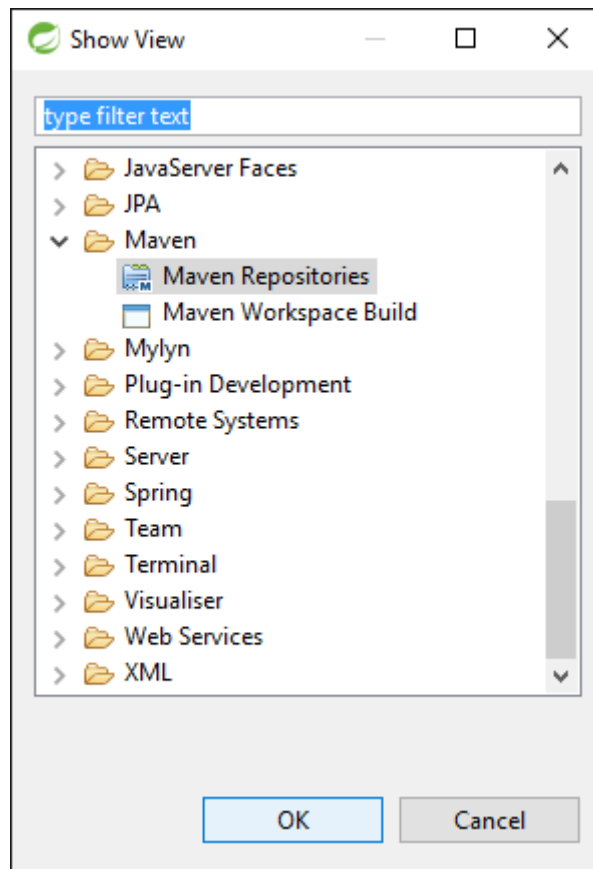
Το επίπεδο του Web αφορά τη σχετική με τις εφαρμογές ιστού λειτουργικότητα.

### **Δημιουργία Εφαρμογής quiz στο Spring**

Για να γίνει επίδειξη της λειτουργίας του Spring , δημιουργήθηκε μια εφαρμογή online quiz η οποία θα είναι διαθέσιμη μόνο σε εγγεγραμμένους χρήστες που θα συνδέονται στην εφαρμογή και θα κάνουν το quiz. Θα υπάρχει φόρμα εγγραφής, καθώς και σύνδεσης στην εφαρμογή όπου θα ελέγχονται τα στοιχεία που δίνει ο χρήστης, καθώς και button logout. Θα υπάρχει σελίδα με ερωτήσεις με button υποβολής τους. Μετά την υποβολή τους, θα εμφανίζονται στο χρήστη τα αποτελέσματά του σε κάθε ερώτηση. Τέλος, θα υπάρχει σελίδα στατιστικών επιτυχίας (για όλους τους χρήστες) ανά ερώτηση. Στην εφαρμογή θα υπάρχουν χαρακτηριστικά ασφάλειας.

Καταρχήν, αναφέρονται τα απαραίτητα στοιχεία λογισμικού που χρειάζονται για τη δημιουργία της εφαρμογής quiz. Χρειάζεται το Java Development Kit (JDK) το οποίο είναι διαθέσιμο στην ιστοσελίδα:<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>. Επιπλέον, χρειάζεται ο Apache Tomcat στον οποίο θα εκτελούνται οι εφαρμογές. Ο Apache Tomcat είναι διαθέσιμος για λήψη από την ιστοσελίδα: <http://tomcat.apache.org/download-80.cgi>. Ως IDE επιλέχθηκε το Spring Tool Suite (STS), το οποίο είναι βασισμένο στο Eclipse και είναι εξειδικευμένο για τη δημιουργία εφαρμογών σε Spring. Το Spring Tool Suite είναι διαθέσιμο για λήψη στην ιστοσελίδα: <https://spring.io/tools/sts>. Επιπλέον χρειάζεται το Apache Maven που διαχειρίζεται τις εξαρτήσεις του λογισμικού και είναι διαθέσιμο για λήψη στην ιστοσελίδα: <https://maven.apache.org/download.cgi>. Τέλος, ως βάση δεδομένων χρησιμοποιήθηκε η MySQL. Η MySQL είναι διαθέσιμη για λήψη στην ιστοσελίδα: <http://www.mysql.com/downloads/>. Για τις ανάγκες της εφαρμογής έγινε λήψη του MySQL Workbench για το λειτουργικό σύστημα Windows το οποίο είναι διαθέσιμο για λήψη από την ιστοσελίδα: <http://dev.mysql.com/downloads/windows>.

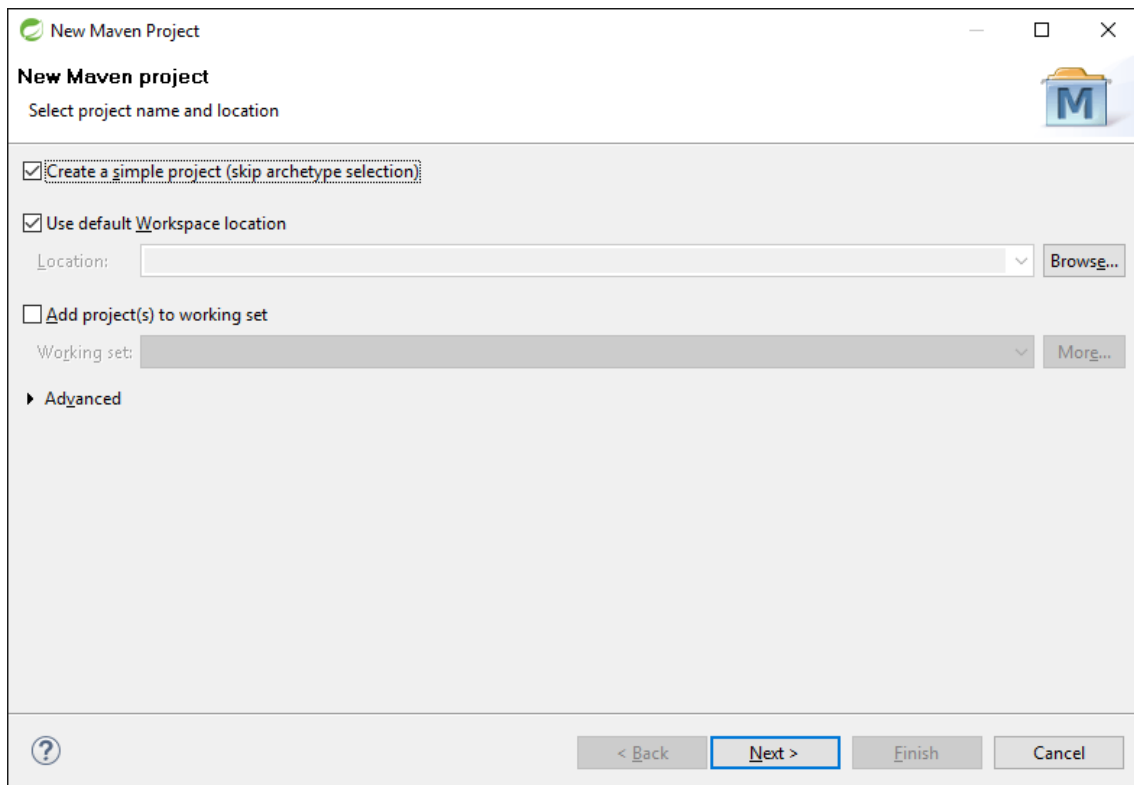
Αφού έχει γίνει λήψη και εγκατάσταση του απαραίτητου λογισμικού πρέπει να γίνει εκτέλεση του Spring Tool Suite. Στο menu του Spring Tool Suite στο Window->Show View Other-> επιλέγεται το Maven Repositories (Εικόνα 14):



**Εικόνα 14: Το View με όνομα Maven Repositories**

Στο View αυτό που φαίνεται στο κάτω μέρος του Spring Tool Suite, επιλέγεται το Global Repositories -> central και δεξί κλικ στο central επιλέγοντας Rebuild index. Στο παράθυρο που θα εμφανιστεί επιλέγεται OK. Αυτό γίνεται για να είναι διαθέσιμο το ευρετήριο του Maven ώστε να βρίσκονται εύκολα τα στοιχεία λογισμικού (dependencies) που χρειάζονται για την εφαρμογή.

Στη συνέχεια στο menu στο File -> New Maven Project επιλέγονται τα ακόλουθα(Εικόνα 15):



**Εικόνα 15: Δημιουργία νέου Maven Project.**

Πατώντας Next γίνονται κάποιες άλλες ρυθμίσεις και πατιέται Finish (Εικόνα 16):

New Maven Project

**New Maven project**

Configure project

Artifact

Group Id: com.myquiz

Artifact Id: myquiz2016

Version: 0.0.1-SNAPSHOT

Packaging: war

Name:

Description:

Parent Project

Group Id:

Artifact Id:

Version:

Browse... Clear

► Advanced

? < Back Next > Finish Cancel

**Εικόνα 16: Ρυθμίσεις του Maven Project**

Το αρχείο pom.xml του Maven διαχειρίζεται τα στοιχεία λογισμικού που χρειάζονται για την εφαρμογή. Στη Εικόνα 17 δίνεται ένα μέρος του pom.xml της τελικής εφαρμογής:



```

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.quiz</groupId>
  <artifactId>myquiz</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>

  <!-- Shared version number properties -->
  <properties>
    <org.springframework.version>4.2.5.RELEASE</org.springframework.version>
    <apache.tiles>3.0.3</apache.tiles>
    <spring.security.version>4.0.4.RELEASE</spring.security.version>
  </properties>

  <dependencies>

    <!-- Spring Security -->
    <dependency>
      <groupId>org.springframework.security</groupId>
      <artifactId>spring-security-web</artifactId>
      <version>${spring.security.version}</version>
    </dependency>

    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-validator</artifactId>
      <version>5.1.0.Final</version>
    </dependency>

    <dependency>
      <groupId>org.springframework.security</groupId>
      <artifactId>spring-security-config</artifactId>
      <version>${spring.security.version}</version>

```

Εικόνα 17: Ένα μέρος του pom.xml αρχείου

Στο αρχείο web.xml πρέπει να προστεθούν οι ακόλουθες γραμμές (Εικόνα 18):

```

<servlet>
<servlet-name>dispatcher</servlet-name>
<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
<load-on-startup>1</load-on-startup>
</servlet>

<servlet-mapping>
<servlet-name>dispatcher</servlet-name>
<url-pattern>/*</url-pattern>
</servlet-mapping>

```

Εικόνα 18: Προσθήκη του dispatcher servlet στο αρχείο web.xml

Το dispatcher servlet διαχειρίζεται τις αιτήσεις της εφαρμογής χρησιμοποιώντας την αντιστοίχιση που βρίσκεται στο url-pattern tag. Το /\* σημαίνει ότι το dispatcher servlet θα χειρίζεται όλες τις αιτήσεις. Όταν ο χρήστης αιτείται μια σελίδα, το dispatcher servlet θα χρειαστεί να υπάρχει ένα αρχείο .xml στο φάκελο WEB-INF και θα πρέπει να ονομάζεται σύμφωνα με το περιεχόμενο του tag servlet name ακολουθούμενο από το -servlet. Έτσι το αρχείο που χρειάζεται στην περίπτωση αυτή θα ονομάζεται dispatcher-servlet.xml. Σημειώνεται ότι το dispatcher-servlet.xml. πρέπει να δημιουργηθεί ως Spring Bean Configuration File μέσω του Spring Tool Suite.

Μέσω του tag , ορίζεται το πακέτο της java όπου περιέχει τον (ή τους) Controller:

Θεωρητική ανασκόπηση των πλαισίων εργασίας εφαρμογών ιστού και πρακτική εφαρμογή στα Spring και Struts2.

```
<context:component-scan base-package="com.quiz.controller">/context:component-scan
```

Δίνεται ο κώδικας του IndexController της εφαρμογής ο οποίος είναι υπεύθυνος για την αρχική σελίδα της εφαρμογής (Εικόνα 19):

```
package com.quiz.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.RequestMapping;

@Controller
public class IndexController {

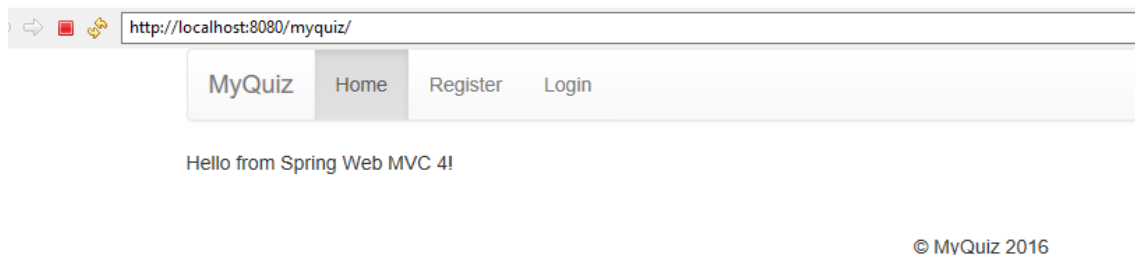
    @RequestMapping("")
    public String index0() {
        return "index";
    }

    @RequestMapping("/index")
    public String index() {
        return "index";
    }

}
```

Εικόνα 19: Ο IndexController

Για να οριστεί μια κλάση Java ως Controller πρέπει να έχει το Annotation @Controller και διαχειρίζεται τα URL μέσω του Annotation @RequestMapping. Παρατηρείται ότι υπάρχουν δύο μέθοδοι που επιστρέφουν το ίδιο String. Αυτό συμβαίνει σε περίπτωση που ο χρήστης ανοίξει την εφαρμογή χωρίς να δώσει στο URL το /index. Δηλαδή είτε ο χρήστης δώσει /index είτε δε δώσει /index στο URL της εφαρμογής θα δει τη σελίδα index.jsp (Εικόνα 20):



Εικόνα 20: Η αρχική σελίδα της εφαρμογής

Χρησιμοποιήθηκε το Apache Tiles για την εμφάνιση της ιστοσελίδας. Το Apache tiles είναι ένα πλαίσιο εργασίας λογισμικού το οποίο δημιουργεί και συνδυάζει μέρη μιας ιστοσελίδας με σκοπό την εμφάνισή τους στο χρήστη. Το Apache Tiles έχει σκοπό να μην επαναλαμβάνονται στις διάφορες σελίδες της εφαρμογής κάποια κοινά μέρη π.χ. footer.

Θεωρητική ανασκόπηση των πλαισίων εργασίας εφαρμογών ιστού και πρακτική εφαρμογή στα Spring και Struts2.

Το Spring υποστηρίζει το Apache Tiles. Για να ενσωματωθεί το Apache Tiles σε μια εφαρμογή Spring πρέπει να προστεθούν τα ακόλουθα beans στο dispatcher-servlet.xml:

```
<bean id="tilesConfigurer"
      class="org.springframework.web.servlet.view.tiles3.TilesConfigurer">
    <property name="definitions">
      <list>
        <value>/WEB-INF/defs/general.xml</value>
      </list>
    </property>
  </bean>

  <bean id="viewResolver"
        class="org.springframework.web.servlet.view.UrlBasedViewResolver">
    <property name="viewClass"
              value="org.springframework.web.servlet.view.tiles3.TilesView" />
  </bean>
```

**Εικόνα 21: Τα beans του Tiles**

Πέρα από το Apache Tiles το οποίο αφορά την οργάνωση της εμφάνισης της σελίδας, χρησιμοποιήθηκε το πλαίσιο εργασίας λογισμικού Bootstrap, για το οποίο υπάρχουν οδηγίες στην ηλεκτρονική διεύθυνση: <http://getbootstrap.com/>.

Για την οργάνωση της βάσης δεδομένων χρησιμοποιήθηκε το Hibernate και το Spring Data JPA για την αποτελεσματικότερη διαχείρισή της. Τα αντίστοιχα dependencies πρέπει να συμπεριληφθούν στο pom.xml. Συνεπώς οι πίνακες της βάσης δεδομένων θα αναπαρίστανται με οντότητες (entities). Κάθε οντότητα θα επισημαίνεται με το Annotation @Entity όπου θα πρέπει να υπάρχει ένα πρωτεύον κλειδί που θα αντιστοιχίζεται με το Annotation @Id. Κάθε μεταβλητή θα πρέπει να έχει μεθόδους getters και setters.

Ενδεικτικά, δίνεται ένα μέρος της οντότητας User για γίνουν κατανοητά τα παραπάνω:

```

import javax.persistence.ManyToMany;
import javax.persistence.Table;
import javax.validation.constraints.Size;

import org.hibernate.validator.constraints.Email;

import com.quiz.annotation.UniqueUsername;

@Table(name = "spring_users")
@Entity
public class User {

    @Id
    @GeneratedValue
    private Integer id;

    @Size(min = 3, message="Username must be 3 characters or greater!")
    @Column(unique=true)
    @UniqueUsername(message="This username already exists")
    private String username;

    @Size(min = 5, message="Password must be 5 characters or greater!")
    private String password;

    @Size(min = 1, message="Email not valid")
    @Email // For email validation
    private String email;

    private boolean enabled;

    private String category;

```

Εικόνα 22: Η οντότητα User

Θα υπάρχουν ένα Service και ένα Repository για κάθε οντότητα τα οποία θα υλοποιεί την αλληλεπίδραση με τις οντότητες.

Το Service θα επισημαίνεται με το Annotation @Service και θα είναι σε ξεχωριστό package από τις οντότητες. Επιπλέον, θα υπάρχουν τα αντίστοιχα repositories για κάθε οντότητα τα οποία θα επεκτείνουν το JpaRepository.

Επίσης, στην εφαρμογή ενσωματώθηκαν δυνατότητες ασφάλειας μέσω του Spring Security. Αρχικά πρέπει να προστεθούν τα σχετικά dependencies στο pom.xml. Επιπλέον χρειάζεται το ακόλουθο filter στο αρχείο web.xml το οποίο σε περίπτωση που χρησιμοποιούνται και άλλα filter πρέπει να είναι το πρώτο (Εικόνα 23):

```

<filter>

    <filter-name>springSecurityFilterChain</filter-name>

    <filter-
class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>

</filter>

<filter-mapping>

    <filter-name>springSecurityFilterChain</filter-name>

    <url-pattern>/*</url-pattern>

</filter-mapping>

```

**Εικόνα 23: Το filter του Spring Security**

Επιπλέον, χρειάζεται η δημιουργία ενός Spring Bean Configuration File με όνομα security.xml όπου θα ορίζεται η πρόσβαση στην εφαρμογή. Για παράδειγμα μόνο ο διαχειριστής μπορεί να δει την καρτέλα Users που έχει τους εγγεγραμμένους χρήστες της εφαρμογής. Αυτό επιτυγχάνεται μέσω της εντολής:

```
<intercept-url pattern="/users*" access="hasRole('ROLE_ADMIN')" />
```

Επιπλέον, στο applicationContext.xml πρέπει να συμπεριληφθεί το αρχείο security.xml με την ακόλουθη δήλωση: <import resource="security.xml" />. Η λειτουργία της αποσύνδεσης χρήστη είναι έτοιμη στο Spring και δηλώνεται στο security.xml με την εντολή: <logout logout-url="/logout" />. Επιπλέον στο security.xml ορίζεται ένας αλγόριθμος κρυπτογράφησης για τον κωδικό πρόσβασης (password) στη βάση δεδομένων ο BCrypt. Αυτό γίνεται με τη δήλωση:

```
<password-encoder hash="bcrypt" />. Επιπλέον, για το Spring Security 4 χρειάζεται η
ακόλουθη γραμμή στο security.xml: <csrf disabled="true" />. Πρόσθετα για να μην είναι η
εφαρμογή ευάλωτη σε επιθέσεις τύπου Cross Site Scripting (XSS) στο αρχείο web.xml
προστέθηκε το ακόλουθο φίλτρο:
```

```

<context-param>

    <param-name>defaultHtmlEscape</param-name>

    <param-value>>true</param-value>

</context-param>

```

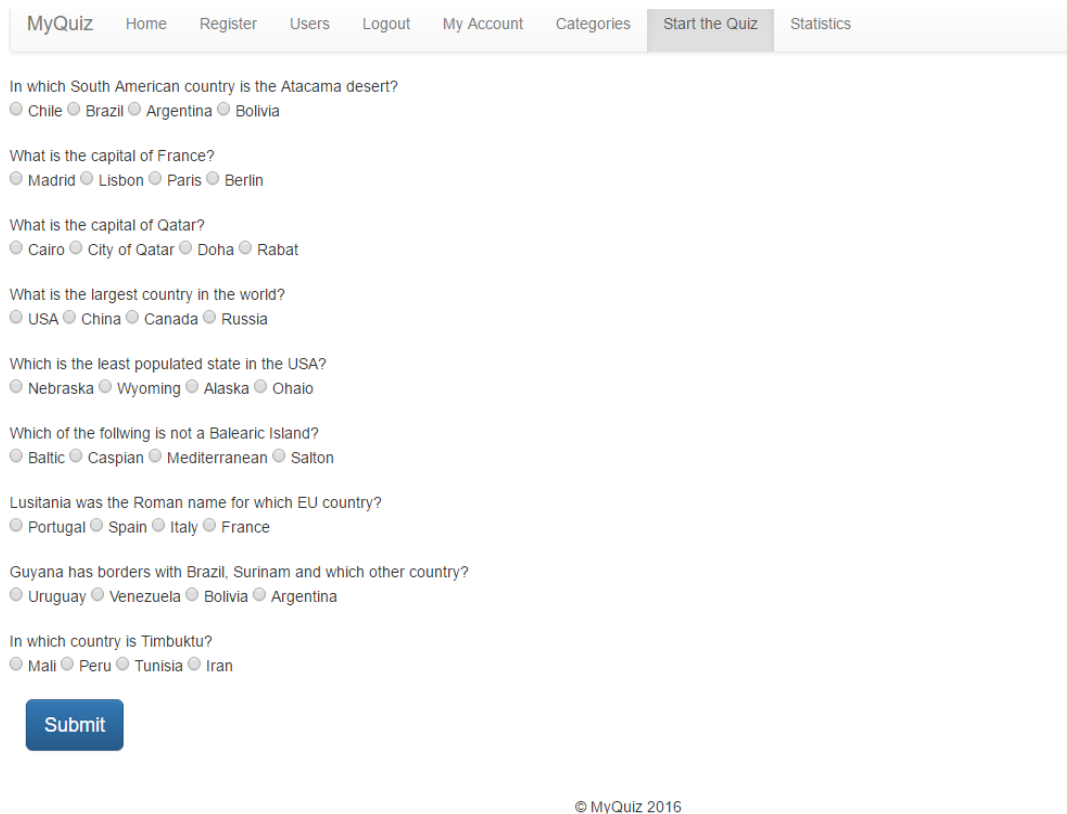
**Εικόνα 24: Προστασία της εφαρμογής από Cross Site Scripting**

Για τη φόρμα εγγραφής χρησιμοποιήθηκε επικύρωση από την πλευρά του εξυπηρετητή (server-side validation) και επικύρωση από την πλευρά του πελάτη(client-side validation). Για server-side validation χρησιμοποιήθηκε ο Hibernate Validator. Αρχικά στο dispatcher-servlet.xml προστέθηκε η γραμμή <mvc:annotation-driven /> και προστέθηκε στο pom.xml το dependency του Hibernate Validator. Συνεπώς, όπως φαίνεται και στην Εικόνα 19 το Annotation @Size με min=1 δεν επιτρέπει κενές εγγραφές και μέσω του message εμφανίζεται σχετικό μήνυμα αν υπάρξει αποτυχία στην επικύρωση της εισόδου. Στο σχετικό με το χρήστη Controller προστέθηκε το Annotation @Valid για τον χρήστη. Στο

επίπεδο του View το Spring διαθέτει το tag `<form:errors path="variable_name"/>` για την εμφάνιση σφαλμάτων επικύρωσης. Στη φόρμα επικυρώνονται όλα τα στοιχεία. Για το client side validation χρησιμοποιήθηκε το jquery validation plugin το οποίο είναι διαθέσιμο στην ιστοσελίδα:

<https://jqueryvalidation.org/>.

Τέλος, ο χρήστης μπορεί να κάνει το quiz και να δει τα αποτελέσματά του, καθώς και τα στατιστικά ανά ερώτηση. Ενδεικτικά, παρουσιάζονται η σελίδα με τις ερωτήσεις του quiz (Εικόνα 25) καθώς και των στατιστικών των ερωτήσεων (Εικόνα 26):



The screenshot shows the MyQuiz application interface. At the top, there is a navigation bar with the following links: MyQuiz, Home, Register, Users, Logout, My Account, Categories, Start the Quiz (highlighted), and Statistics. Below the navigation bar, there are ten multiple-choice questions, each with four radio button options:

- In which South American country is the Atacama desert?  
 Chile  Brazil  Argentina  Bolivia
- What is the capital of France?  
 Madrid  Lisbon  Paris  Berlin
- What is the capital of Qatar?  
 Cairo  City of Qatar  Doha  Rabat
- What is the largest country in the world?  
 USA  China  Canada  Russia
- Which is the least populated state in the USA?  
 Nebraska  Wyoming  Alaska  Ohio
- Which of the following is not a Balearic Island?  
 Baltic  Caspian  Mediterranean  Salton
- Lusitania was the Roman name for which EU country?  
 Portugal  Spain  Italy  France
- Guyana has borders with Brazil, Surinam and which other country?  
 Uruguay  Venezuela  Bolivia  Argentina
- In which country is Timbuktu?  
 Mali  Peru  Tunisia  Iran

At the bottom of the form, there is a blue button labeled "Submit". Below the form, the copyright notice "© MyQuiz 2016" is displayed.

Εικόνα 25: Η σελίδα των ερωτήσεων του quiz

MyQuiz	Home	Register	Users	Logout	My Account	Categories	Start the Quiz	Statistics
Question	Success rate							
In which South American country is the Atacama desert?	42,59%							
What is the capital of France?	11,11%							
What is the capital of Qatar?	12,73%							
What is the largest country in the world?	3,7%							
Which is the least populated state in the USA?	9,26%							
Which of the following is not a Balearic Island?	0%							
Lusitania was the Roman name for which EU country?	3,7%							
Guyana has borders with Brazil, Surinam and which other country?	0%							
In which country is Timbuktu?	3,7%							

**Εικόνα 26: Η σελίδα των στατιστικών του quiz**

## ΚΕΦΑΛΑΙΟ 6

### ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΠΙΘΑΝΕΣ ΕΠΕΚΤΑΣΕΙΣ

Παρατηρείται ότι με τη συνεχή εξέλιξη της τεχνολογίας και την ανάγκη για ομοιόμορφο και αποδοτικό κώδικα καθώς και υιοθέτηση βέλτιστων πρακτικών τα πλαίσια εργασίας εφαρμογών ιστού είναι πολύ χρήσιμα. Βοηθάνε τον προγραμματιστή να δημιουργήσει αποδοτικές και δομημένες εφαρμογές και μπορούν να συνεργαστούν με άλλα πλαίσια εργασίας εφαρμογών ιστού και άλλα στοιχεία λογισμικού.

Όσον αφορά το Struts2, έχει λιγότερες δυνατότητες από το Spring και είναι λιγότερο διαδεδομένο καθώς και πιο δύσκολο να βρεθεί πληθώρα πληροφοριών για αυτό. Ένα βασικό μειονέκτημά του είναι ότι δεν υποστηρίζει πρόσθετες δυνατότητες ασφάλειας όπως το Spring που έχει το Spring Security. Ωστόσο το Struts2 μπορεί να συνεργαστεί με το Spring κάτι το οποίο θα είχε ενδιαφέρον να υλοποιηθεί.

Το Spring την στιγμή που γράφεται η παρούσα διατριβή είναι το πιο διαδεδομένο πλαίσιο εργασίας εφαρμογών ιστού σε Java. Διαθέτει πάρα πολλά στοιχεία από τα οποία ο προγραμματιστής μπορεί να επιλέξει μόνο όσα χρειάζεται. Επιπλέον, διαθέτει πληθώρα πληροφοριών και μια πολύ μεγάλη κοινότητα χρηστών.

Μια πιθανή προέκταση της υπάρχουσας εφαρμογής quiz θα ήταν να ενσωματωθούν δυνατότητες ασφάλειας στην έκδοση που έχει υλοποιηθεί στο Struts2 π.χ. μέσω του Spring Security αφού το Struts2 μπορεί να συνεργαστεί με το Spring. Στην έκδοση που έχει υλοποιηθεί στο Spring θα μπορούσαν οι ερωτήσεις να είναι σε δύο ή περισσότερες σελίδες ανάλογα με την προτίμηση του χρήστη. Τέλος και στις δύο εφαρμογές θα μπορούσαν να προστεθούν παραπάνω από μια κατηγορίες και να εμφανίζονται στο χρήστη μόνο οι σχετικές με την κατηγορία ερωτήσεις.



## ΒΙΒΛΙΟΓΡΑΦΙΑ

1. «An approach to specifying software frameworks», L. Murray, D. Carrington, P. Strooper, 2004
2. «Framework- DocForge », <http://docforge.com/wiki/Framework>, Τελευταία πρόσβαση στις 08/09/2015
3. «Object Oriented Application Frameworks», M. Fayad, D.C. Schmidt, 1997
4. «Mac App», <http://c2.com/cgi/wiki?MacApp>, Τελευταία πρόσβαση στις 16/09/2015
5. «About the Dabo Application Framework», <http://dabodev.com/>, Τελευταία πρόσβαση στις 17/09/2015
6. «Jt - Java Pattern Oriented Framework», <http://www.ibm.com/developerworks/library/ws-designpattern/>, Τελευταία πρόσβαση στις 17/09/2015
7. «Fiz: A Component Framework for Web Applications», J.K. Ousterhout, 2009
8. «Web application framework - DocForge», [http://docforge.com/wiki/Web\\_application\\_framework](http://docforge.com/wiki/Web_application_framework), Τελευταία πρόσβαση στις 08/09/2015
9. «What is object-relational mapping (ORM)? - Definition from WhatIs.com», <http://searchwindevelopment.techtarget.com/definition/object-relational-mapping>, Τελευταία πρόσβαση στις 25/09/2015
10. «Coralbits S.L.», <http://www.coralbits.com/libonion/>, Τελευταία πρόσβαση στις 24/09/2015
11. «KORE.io - A fast SPDY capable web server for web development in C», <https://kore.io/doc/>, Τελευταία πρόσβαση στις 25/09/2015
12. « KClone Embedded Web Server – Overview», <http://www.koanlogic.com/kclone/index.html>, Τελευταία πρόσβαση στις 27/09/2015
13. «ESP Web Framework», <https://embedthis.com/esp/>, Τελευταία πρόσβαση στις 27/09/2015
14. «CppCMS — High Performance C++ Web Framework», <http://cppcms.com/wikip/en/page/main>, Τελευταία πρόσβαση στις 27/09/2015
15. «Wt, C++ Web Toolkit», <http://www.webtoolkit.eu/wt>, Τελευταία πρόσβαση στις 28/09/2015
16. «Tntnet homepage», <http://www.tntnet.org/>, Τελευταία πρόσβαση στις 28/09/2015
17. «TreeFrog Framework | High-speed C++ MVC Framework for Web Application», <http://www.treefrogframework.org/documents/introduction>, Τελευταία πρόσβαση στις 28/09/2015
18. «Platinum C++ Framework»,<http://www.pt-framework.org/>, Τελευταία πρόσβαση στις 29/09/2015
19. «CFWheels - an open source CFML framework inspired by Ruby on Rails», <http://cfwheels.org/>, Τελευταία πρόσβαση στις 03/11/2015
20. «Home - Fusebox», <http://www.fusebox.org/>, Τελευταία πρόσβαση στις 15/11/2015
21. «Home - FarCry Core 7.0», <https://www.farcrycore.org/>, Τελευταία πρόσβαση στις 15/11/2015
22. «vibe.d», <http://vibed.org/>, Τελευταία πρόσβαση στις 14/11/2015
23. «Happstack», <http://www.happstack.com/page/view-page-slug/1/happstack>, Τελευταία πρόσβαση στις 28/12/2015

24. «Web/Frameworks - HaskellWiki»,<https://wiki.haskell.org/Web/Frameworks#Happstack>, Τελευταία πρόσβαση στις 28/12/2015
25. «Snap Framework: a web framework for Haskell»,<http://snapframework.com/>, Τελευταία πρόσβαση στις 28/12/2015
26. «Welcome to the Apache Struts project», <http://struts.apache.org/>, Τελευταία πρόσβαση στις 28/12/2015
27. «apache/struts · GitHub», <https://github.com/apache/struts>, Τελευταία πρόσβαση στις 28/12/2015
28. «Spring Tutorial - javatpoint», <http://www.javatpoint.com/spring-tutorial>, Τελευταία πρόσβαση στις 10/01/2016
29. «Spring Framework», <https://projects.spring.io/spring-framework/>, Τελευταία πρόσβαση στις 10/01/2016
30. «James Shore: Dependency Injection Demystified», <http://www.jamesshore.com/Blog/Dependency-Injection-Demystified.html>, Τελευταία πρόσβαση στις 11/01/2016
31. «ASPECT ORIENTED PROGRAMMING», Palo Alto Research Center 2003, Τελευταία πρόσβαση στις 11/01/2016
32. «Play Framework - Build Modern and Scalable Web Apps with Java and Scala», <https://www.playframework.com/>, Τελευταία πρόσβαση στις 11/01/2016
33. «The Grails Framework», <https://grails.org/>, Τελευταία πρόσβαση στις 10/02/2016
34. «Ember.js - A framework for creating ambitious web applications», <http://emberjs.com/>, Τελευταία πρόσβαση στις 10/02/2016
35. «AngularJS — Superheroic JavaScript MVW Framework», <https://angularjs.org/>, Τελευταία πρόσβαση στις 10/02/2016
36. «Meteor», <https://www.meteor.com/>, Τελευταία πρόσβαση στις 10/02/2016
37. «Scalatra | A tiny, Sinatra-like web framework for Scala», <http://scalatra.org/>, Τελευταία πρόσβαση στις 11/02/2016
38. «Scalatra in Action», Dave Hrycyszyn, Stefan Ollinger και Ross A. Baker, 2016, Manning Publications
39. «Catalyst | Perl MVC web application framework», <http://www.catalystframework.org/>, Τελευταία πρόσβαση στις 11/02/2016
40. «Mojolicious - Perl real-time web framework», <http://mojomolicious.org/>, Τελευταία πρόσβαση στις 11/02/2016
41. «Laravel - The PHP Framework For Web Artisans», <https://laravel.com/>, Τελευταία πρόσβαση στις 11/02/2016
42. «GitHub - laravel/laravel: A PHP Framework For Web Artisans», <https://github.com/laravel/laravel>, Τελευταία πρόσβαση στις 11/02/2016
43. «Building Web Applications from Scratch with Laravel - Envato Tuts+ Code Tutorial»,<http://code.tutsplus.com/tutorials/building-web-applications-from-scratch-with-laravel--net-25517>, Τελευταία πρόσβαση στις 11/02/2016
44. «Symfony, High Performance PHP Framework for Web Development», <https://symfony.com>, Τελευταία πρόσβαση στις 11/02/2016

45. «Learn symfony: a Beginner's Tutorial», <http://www.sitepoint.com/symfony-beginners-tutorial/>, Τελευταία πρόσβαση στις 11/02/2016
46. «CodeIgniter Web Framework», <http://www.codeigniter.com/>, Τελευταία πρόσβαση στις 11/02/2016
47. «CakePHP v3.2 - the rapid development php framework.», <http://cakephp.org/>, Τελευταία πρόσβαση στις 11/02/2016
48. «The Web framework for perfectionists with deadlines | Django», <https://www.djangoproject.com/>, Τελευταία πρόσβαση στις 12/02/2016
49. «The Web Framework that scales with you. TurboGears2 Website 3.0 documentation», <http://www.turbogears.org/>, Τελευταία πρόσβαση στις 13/02/2016
50. «web2py Web Framework», <http://www.web2py.com/>, Τελευταία πρόσβαση στις 13/02/2016
51. «Ruby on Rails», <http://rubyonrails.org/>, Τελευταία πρόσβαση στις 13/02/2016
52. «Sinatra», <http://www.sinatrarb.com/>, Τελευταία πρόσβαση στις 13/02/2016
53. «GitHub - soveran/cuba: Rum based microframework for web development», <https://github.com/soveran/cuba>, Τελευταία πρόσβαση στις 13/02/2016
54. «GitHub - rtomayko/tilt: Generic interface to multiple Ruby template engines», <https://github.com/rtomayko/tilt>, Τελευταία πρόσβαση στις 13/02/2016
55. «ASP.NET | The ASP.NET Site», <http://www.asp.net/>, Τελευταία πρόσβαση στις 13/02/2016
56. «Struts 2 Tutorial», [http://www.tutorialspoint.com/struts\\_2/index.htm](http://www.tutorialspoint.com/struts_2/index.htm), Τελευταία πρόσβαση στις 24/03/2016
57. «Introduction to Struts 2», <https://app.pluralsight.com/library/courses/struts2/table-of-contents>, Τελευταία πρόσβαση στις 24/03/2016
58. «From a birds eye», <https://struts.apache.org/birdseye.html>, Τελευταία πρόσβαση στις 24/03/2016
59. «spring Tutorial», <http://www.tutorialspoint.com//spring/index.htm>, Τελευταία πρόσβαση στις 16/05/2016
60. «Spring Framework», <http://projects.spring.io/spring-framework/>, Τελευταία πρόσβαση στις 16/05/2016
61. «Spring Framework Reference Documentation», <http://docs.spring.io/spring/docs/current/spring-framework-reference/htmlsingle/#overview-modules>, Τελευταία πρόσβαση στις 16/05/2016
62. «Apache Tiles- Home», <https://tiles.apache.org/>, Τελευταία πρόσβαση στις 16/05/2016