



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ**  
**ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ**  
**ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ**  
**ΥΠΟΛΟΓΙΣΤΩΝ**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Οδηγός Σχεδίασης Διαδικτυακών εφαρμογών με την χρήση Spring  
Framework**

**Πέτρος Β. Γκοντρόλης**

**Εισηγητής: Νικόλαος Ζάχαρης**

**ΑΘΗΝΑ**  
**ΣΕΠΤΕΜΒΡΙΟΣ 2019**



**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**Οδηγός Σχεδίασης Διαδικτυακών εφαρμογών με την χρήση Spring Framework**

**Πέτρος Β. Γκοντρόλης  
Α.Μ. ais0069**

**Εισηγητής:**

**Νικόλαος Ζάχαρης, Καθηγητής**

**Εξεταστική Επιτροπή:**

**Ιωάννης Έλληνας, Καθηγητής  
Αναστασία Βελώνη, Καθηγητής**

**Ημερομηνία εξέτασης 01/10/2019**



## ΔΗΛΩΣΗ ΣΥΓΓΡΑΦΕΑ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ

Ο κάτωθι υπογεγραμμένος **Πέτρος Γκοντρόλης**, του **Βασίλειου** με αριθμό μητρώου **ais0069** φοιτητής του Τμήματος Μηχανικών Η/Υ Συστημάτων Τ.Ε. του Α.Ε.Ι. Πειραιά Τ.Τ. πριν αναλάβω την εκπόνηση της Πτυχιακής Εργασίας μου, δηλώνω ότι ενημερώθηκα για τα παρακάτω:

«Η Πτυχιακή Εργασία (Π.Ε.) αποτελεί προϊόν πνευματικής ιδιοκτησίας τόσο του συγγραφέα, όσο και του Ιδρύματος και θα πρέπει να έχει μοναδικό χαρακτήρα και πρωτότυπο περιεχόμενο.

Απαγορεύεται αυστηρά οποιοδήποτε κομμάτι κειμένου της να εμφανίζεται αυτούσιο ή μεταφρασμένο από κάποια άλλη δημοσιευμένη πηγή. Κάθε τέτοια πράξη αποτελεί προϊόν λογοκλοπής και εγείρει θέμα Ηθικής Τάξης για τα πνευματικά δικαιώματα του άλλου συγγραφέα. Αποκλειστικός υπεύθυνος είναι ο συγγραφέας της Π.Ε., ο οποίος φέρει και την ευθύνη των συνεπειών, ποινικών και άλλων, αυτής της πράξης.

Πέραν των όποιων ποινικών ευθυνών του συγγραφέα σε περίπτωση που το Ίδρυμα του έχει απονείμει Πτυχίο, αυτό ανακαλείται με απόφαση της Συνέλευσης του Τμήματος. Η Συνέλευση του Τμήματος με νέα απόφαση της, μετά από αίτηση του ενδιαφερόμενου, του αναθέτει εκ νέου την εκπόνηση της Π.Ε. με άλλο θέμα και διαφορετικό επιβλέποντα καθηγητή. Η εκπόνηση της εν λόγω Π.Ε. πρέπει να ολοκληρωθεί εντός τουλάχιστον ενός ημερολογιακού δμήνου από την ημερομηνία ανάθεσης της. Κατά τα λοιπά εφαρμόζονται τα προβλεπόμενα στο άρθρο 18, παρ. 5 του ισχύοντος Εσωτερικού Κανονισμού.»



## **ΕΥΧΑΡΙΣΤΙΕΣ**

Η παρούσα πτυχιακή εργασία ολοκληρώθηκε μετά από επίμονες προσπάθειες, σε ένα ενδιαφέρον γνωστικό αντικείμενο, όπως αυτό της επεξεργασίας κειμένου. Την προσπάθειά μου αυτή υποστήριξε ο επιβλέπων καθηγητής μου, τον οποίο θα ήθελα να ευχαριστήσω.





## **ΠΕΡΙΛΗΨΗ**

Η παρούσα πτυχιακή εργασία ασχολείται με την ανάπτυξη διαδικτυακών εφαρμογών με την χρήση του Spring Framework. Συγκεκριμένα αναφέρει τι είναι το Spring Framework και πως μπορείς να το χρησιμοποιήσεις ως προγραμματιστής (developer). Ο οδηγός αυτός θα δείξει βήμα βήμα την κατασκευή μιας διαδικτυακής εφαρμογής χρησιμοποιώντας το eclipse JDK (Java Development Kit). Ο apache Tomcat θα είναι ο εξυπηρετητής (application server) για τα JSP/Servlet μας.

## **ABSTRACT**

This diploma thesis deals with the development of web applications using the Spring Framework. Specifically, it states what the Spring Framework is and how to use it as a developer. This guide will step by step build an internet application using the JDK (Java Development Kit) eclipse. Apache Tomcat will be the application server for our JSP / Servlet.

## ΠΕΡΙΕΧΟΜΕΝΑ

<b>1.</b>	<b>ΕΙΣΑΓΩΓΗ.....</b>	<b>14</b>
1.1	Τι είναι το Java EE.....	14
1.2	Τι τεχνολογίες χρησιμοποιούνται στο JEE .....	14
1.3	Που τρέχουν οι διαδικτυακές εφαρμογές μας.....	15
1.4	Τι είναι τα components.....	15
<b>2.</b>	<b>ΕΙΣΑΓΩΓΗ.....</b>	<b>16</b>
2.1	Ποιό μοντέλο προγραμματισμού χρησιμοποιούνταν παλιότερα .....	16
2.2	Γιατί αυτο το “παλιό” μοντέλο προγραμματισμού άλλαξε; .....	16
2.3	Ποιο είναι το “νέο” μοντέλο προγραμματισμού; .....	17
2.4	Λίγα λόγια για Servlet’s και JSP’s .....	18
<b>3.</b>	<b>ΕΙΣΑΓΩΓΗ.....</b>	<b>19</b>
3.1	Εργαλεία για την εκτέλεση της εφαρμογής μας .....	19
<b>4.</b>	<b>ΕΙΣΑΓΩΓΗ.....</b>	<b>20</b>
4.1	Τι είναι το Spring; .....	20
4.2	Ποιός το δημιούργησε; .....	20
4.3	Πότε κυκλοφόρησε πρώτη φορά; .....	20
4.4	Γιατί να επιλέξουμε το Spring για τις εφαρμογές μας; .....	20
<b>5.</b>	<b>ΕΙΣΑΓΩΓΗ .....</b>	<b>22</b>
5.1	Σε τι οφελεί το JDBC framework; .....	22
<b>6.</b>	<b>ΕΙΣΑΓΩΓΗ .....</b>	<b>24</b>
6.1	Dynamic web Project .....	24
6.2	MVC DispatcherServlet .....	24
6.3	MVC Controller .....	25
6.4	ApplicationContext , MVC .....	25
6.5	AutoWiring .....	25
6.6	ViewResolver.....	25

<b>7. ΒΙΒΛΙΟΓΡΑΦΙΑ .....</b>	<b>27</b>
------------------------------	-----------

## **ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ**

<b>Εικόνα 1.1: Enterprise application .....</b>	<b>13</b>
<b>Εικόνα 2.1: “Παλιό” μοντέλο προγραμματισμού .....</b>	<b>15</b>
<b>Εικόνα 2.2: “Νέο” μοντέλο προγραμματισμού .....</b>	<b>16</b>
<b>Εικόνα 4.1: Χαρακτηριστικά “modules” .....</b>	<b>20</b>
<b>Εικόνα 5.1: JDBC framework .....</b>	<b>20</b>
<b>Εικόνα 6.1: DispatcherServlet .....</b>	<b>26</b>

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

- JEE** Java Enterprise Edition
- JDK** Java Developmet Kit
- JSP** Java Server Page
- JSTL** JSP Standard Tag Library
- STS** Spring Tool Suite
- POJO** Plain Old Java Object
- AOP** Aspect Orientation Programming
- MVC** Model View Controller
- IoC** Inversion of Control
- DI** Dependency Injection
- XML** eXtensible Markup Language
- ORM** Object Relational Mapping
- JDBC** Java DataBase Connectivity
- CRUD** Create Read Update Delete
- SLF4J** Simple Logging Facade for Java

## **Εγκατάσταση απαραίτητων προγραμμάτων**

1. JDK (JavaDevelopmentKit)
2. Εγκατάσταση eclipse ή STS
3. Εγκατάσταση και ρύθμιση Tomcat
4. Εγκατάσταση Spring IDE (eclipse marketplace)
5. Εγκατάσταση Maven (M2Eclipse)
6. Εγκατάσταση MySQL WorkBench

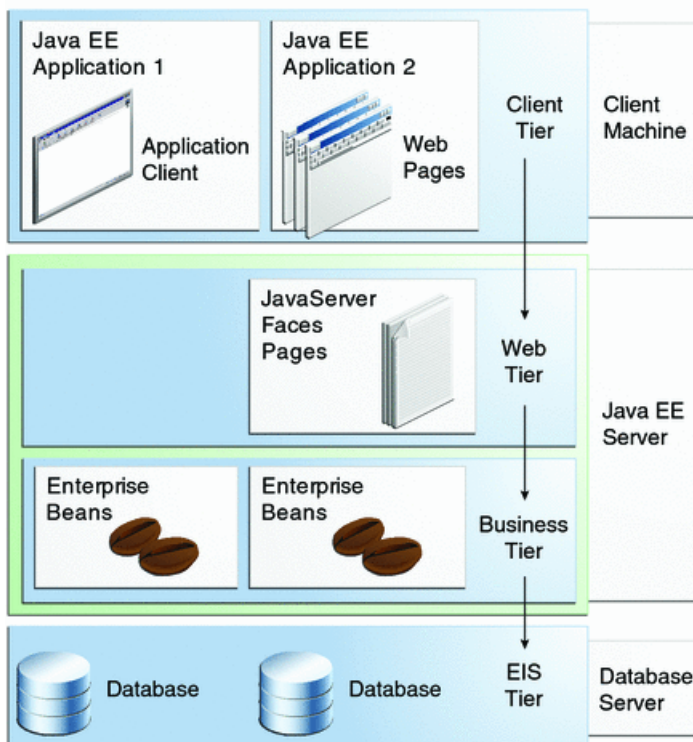
## ΚΕΦΑΛΑΙΟ 1

### ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο γίνεται εισαγωγή στο Java EE (Java Enterprise Edition).

#### 1.1 Τι είναι το Java EE

Το Java EE γνωστό και ως JEE δεν είναι μια εφαρμογή ή προϊόν αλλά μια συλλογή από τεχνολογίες και APIs που επιτρέπουν την δημιουργία “μεγάλων” εφαρμογών. Με τον όρο μεγάλες εφαρμογές εννοούμε μεγάλης κλίμακας προγράμματα που μπορούν να εξυπηρετήσουν χιλιάδες πελάτες όπως οι εφαρμογές των τραπεζών, χρηματιστηρίων κ.α. Τέτοιου είδους εφαρμογές είναι απαιτητικές και κρίσιμες για τις εταιρίες. Μια enterprise εφαρμογή θα μπορούσε να μοιάζει με την παρακάτω εικόνα.



**Εικόνα 1.1:** Enterprise application

## **1.2 Τι τεχνολογίες χρησιμοποιούνται στο JEE;**

Γενικά στο Java EE χρησιμοποιούνται πολλές τεχνολογίες, ενδεικτικά κάποιες από αυτές είναι JDBC, JSP και Servlets

## **1.3 Που τρέχουν οι διαδικτυακές εφαρμογές μας;**

Όπως αναφέρθηκε και παραπάνω ο server room είναι ο apache tomcat ο οποίος τρέχει τοπικά στο σύστημά μας. Με αυτόν τον τρόπο αποτελεί τον container για τις διαδικτυακές εφαρμογές μας δηλαδή το περιβάλλον εκείνο όπου μέσα εκτελούνται οι συνιστώσεις (components) και γίνεται και η διαχείριση- έλεγχος των υπηρεσιών (control services).

## **1.4 Τι είναι τα components;**

Οι εφαρμογές JEE αποτελούνται από components. Είναι ένα αυτόνομο λογισμικό το οποίο συνεργάζεται με άλλα και δημιουργούν όλα μαζί μία JEE εφαρμογή. Χαρακτηριστικά παραδείγματα components είναι τα JSPs και Servlets.

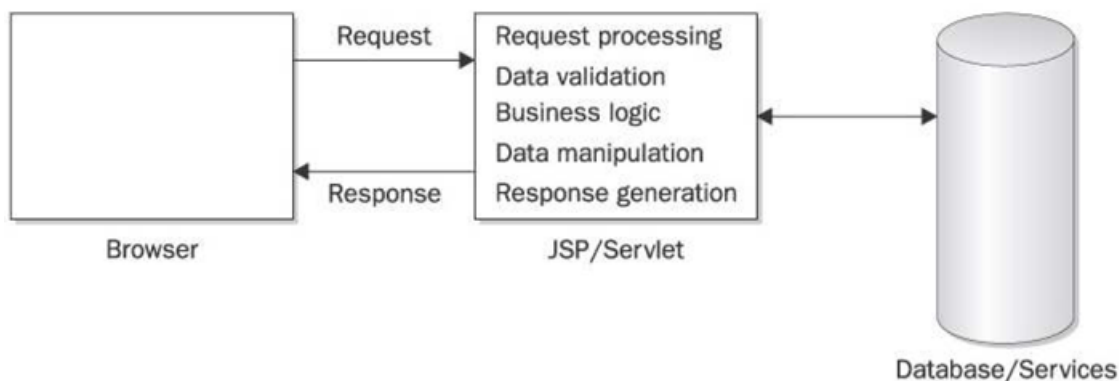
## ΚΕΦΑΛΑΙΟ 2

### ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο γίνεται περιγραφή για τα οφέλη που παρέχει η ανάπτυξη διαδικτυακών εφαρμογών με την χρήση του Spring Framework.

#### 2.1 Ποιό μοντέλο προγραμματισμού χρησιμοποιούνταν παλιότερα;

Περίληπτικά το Http request στέλνεται από τον περιηγητή μας (browser) σε μια σελίδα JSP ή αλλιώς μια δυναμική HTML σελίδα, μετά το JSP αλληλεπιδρούσε με την διαχείριση των δεδομένων (business logic) και τα επεξεργασμένα δεδομένα επέστρεφαν πίσω στο JSP που με την σειρά του τα προωθούσε στον περιηγητή μας. Παρακάτω ακολουθεί σχετική εικόνα.



Εικόνα 2.1: “Παλιό” Μοντέλο προγραμματισμού

#### 2.2 Γιατί αυτό το “παλιό” μοντέλο προγραμματισμού άλλαξε;

Χρόνο με τον χρόνο τα προγράμματα έγιναν πιο περίπλοκα, πιο απαιτητικά και το παλιό μοντέλο προγραμματισμού περιέπλεκε την γραφή του κώδικα.

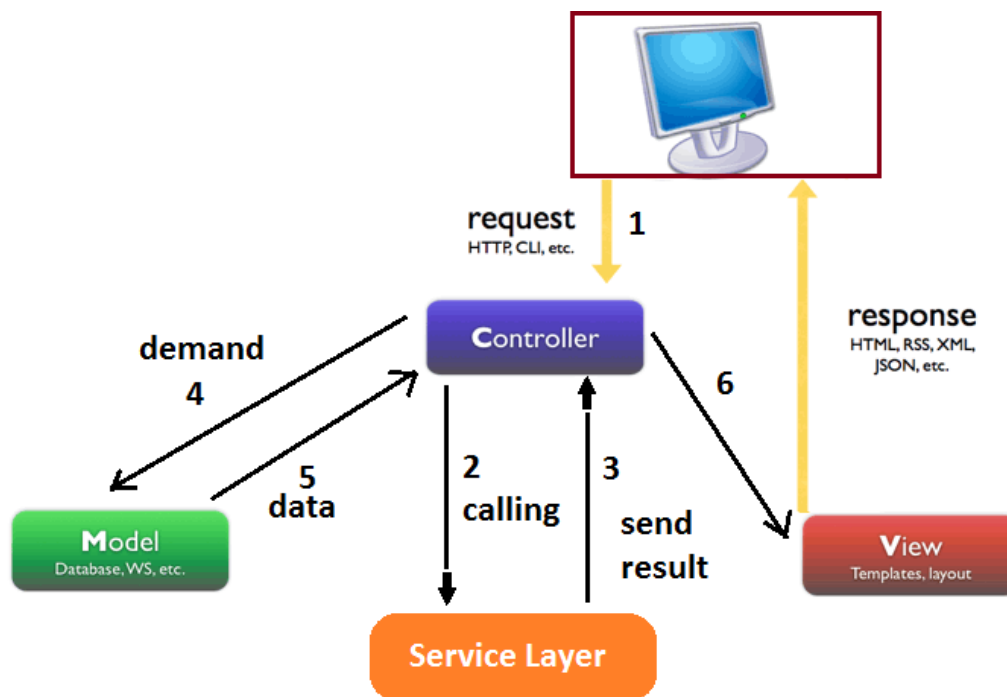


### 2.3 Ποιο είναι το “νέο” μοντέλο προγραμματισμού;

Το νέο μοντέλο προγραμματισμού ,το οποίο χρησιμοποιείται και σήμερα ήταν μια σημαντική αναβάθμιση και αποτελεί τον ενδεδειγμένο τρόπο γραφής κώδικα.

Στηρίζεται στο μοτίβο MVC (MVC design pattern) . Το MVC αποτελείται απο τρεις ενότητες το Model, το View και το Controller

- Το Model αποτελεί τα δεδομένα
- Το View είναι υπεύθυνο για την απεικόνιση των δεδομένων μας , στην ουσία αποτελείται απο JSP σελίδες της εφαρμογής μας.
- Ο Controller χειρίζεται τα αιτήματα (requests) και αποφασίζει που να τα προωθήσει, το Servlet αναλαμβάνει αυτό το ρόλο.



Εικόνα 2.2: “Νέο” Μοντέλο προγραμματισμού

### 2.4 Λίγα λόγια για Servlet's και JSP's

Το Servlet είναι μια τεχνολογία για την δημιουργία διαδικτυακών εφαρμογών. Στην ουσία είναι μια Java κλάση όπου χειρίζεται τα εισερχόμενα request και τα

εξερχόμενα response, όπως αναφέραμε και παραπάνω είναι ένα component για την δημιουργία διαδικτυακών εφαρμογών. Οι μέθοδοι που χρησιμοποιεί είναι doGet, doPost , doHead, doTrace και άλλες.

Το JSP είναι επίσης μια τεχνολογία για την δημιουργία διαδικτυακών εφαρμογών. Στην ουσία επιτρέπει στους προγραμματιστές την ενσωμάτωση java κώδικα σε σελίδες HTML με την χρήση ειδικών tags

## ΚΕΦΑΛΑΙΟ 3

### ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο γίνεται περιγραφή για τα διαθέσιμα εργαλεία ανάπτυξης εφαρμογών Spring. Τα διαθέσιμα αυτά εργαλεία είναι:

1. **Spring IDE:** Το Spring IDE είναι ένα σέτ από plugins ειδικά για το eclipse. Λέγοντας plugin εννοούμε μια προσθήκη, ένα component το οποίο προσθέτει επιπλέον χαρακτηριστικά στο υπάρχον πρόγραμμά μας. Έτσι Το Spring IDE βοηθάει τους προγραμματιστές να φτιάχνουν τα project τους πολύ πιο εύκολα καθώς υπάρχει μια πληθώρα από βοηθήματα ειδικά για Spring projects.

2. **Spring Tool Suite (STS) :** Το Spring Tool Suite είναι ένα έτοιμο προς χρήση εργαλείο (ready-to-use) με προεγκατεστημένο το Spring IDE. Περιλαμβάνει τον tomcat προεγκατεστημένο αλλά και πολλά άλλα component.

**Συμπέρασμα:** Σύμφωνα με τις παραπάνω επιλογές εάν υπάρχει μια προεγκατεστημένη έκδοση του eclipse τότε θα εγκαταστήσουμε το “Spring IDE” αλλιώς θα κάνουμε εγκατάσταση από την αρχή του “Spring Tool Suite”. Κατά την διάρκεια της διπλωματικής μου εργασίας δοκίμασα και τους 2 παραπάνω τρόπους. Τέλος απαραίτητο εργαλείο για τα project μας αποτελεί και η εγκατάσταση του Apache Maven που έγινε με την M2Eclipse plugin .

## ΚΕΦΑΛΑΙΟ 4

### ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο γίνεται σύντομη περιγραφή του Spring Framework

#### 4.1 Τι είναι το Spring;

Το Spring είναι η δημοφιλέστερη εφαρμογή Java για την δημιουργία διαδικτυακών εφαρμογών η οποία χρησιμοποιείται απο παρα πολλους προγραμματιστές.

#### 4.2 Ποιος το δημιούργησε;

Δημιουργός του Spring Framework είναι ο Rod Johnson.

#### 4.3 Πότε κυκλοφόρησε για πρώτη φορά;

Κυκλοφόρησε τον Ιούνιο του 2003 με την έκδοση 2.0 του Apache.

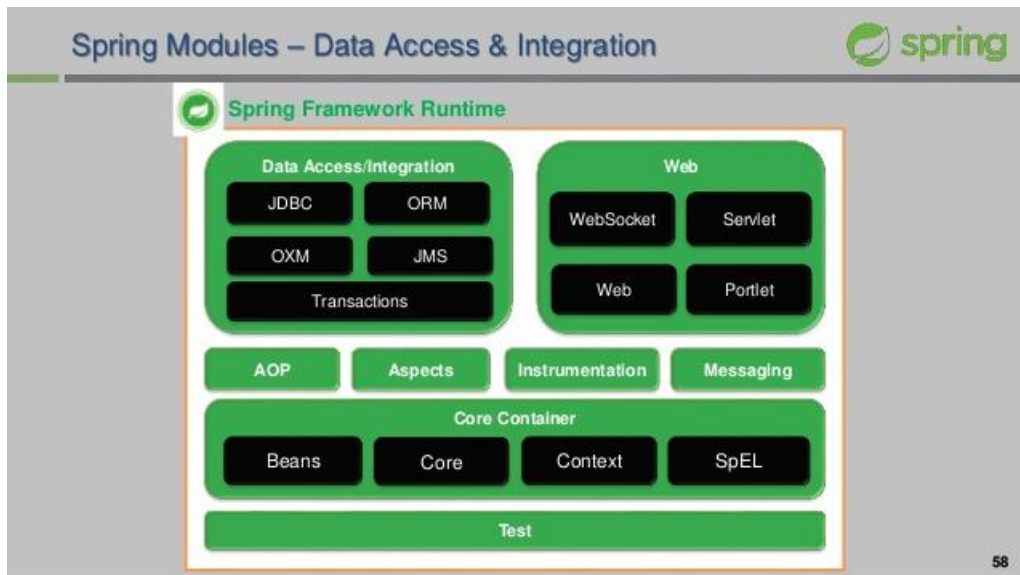
#### 4.4 Γιατί να επιλέξουμε το Spring για την δημιουργία των εφαρμογών μας;

Για να απαντήσουμε το παραπάνω ερώτημα θα αναφέρουμε τα πλεονεκτήματα που παρέχει, παρακάτω ακολουθούν τα κυριότερα:

- a. Οι προγραμματιστές αγαπούν το Spring γιατί είναι “ελαφρύ”. Τι εννοούμε με τον όρο ελαφρύ όμως; Το Spring εκπροσωπεί το μοντέλο POJO(Plain Old Java Object) έτσι οι εφαρμογές τρέχουν με τον Apache Tomcat όπου εύκολα γίνεται εγκατάσταση και δεν καταναλώνει πόρους του συστήματος.
- b. Το Spring είναι επεκτάσιμο “Modular” . Τι εννοούμε με τον όρο επεκτάσιμο; Ο προγραμματιστής έχει την ελευθερία να διαλέξει τα modules εκείνα που επιθυμεί και να απορρίψει αλλα. Με απλα λόγια η εφαρμογή γίνεται μικρότερη και πιο ευέλικτη.
- c. Το Spring εκπροσωπεί την φιλοσοφία της επαναχρησιμοποίησης του υπάρχοντος framework σε αλλες εφαρμογές.
- d. Το μοντέλο POJO κάνει το Testing ενός Spring project πολύ πιο εύκολο και γενικότερα επιτρέπει στους προγραμματιστές να αφοσιωθούν περισσότερο στο business logic.

#### 4.5 Τι είναι τα Spring Modules;

Παραπάνω αναφέραμε ότι το Spring είναι modular, έτσι ακολουθούν κάποια χαρακτηριστικά modules.



Εικόνα 4.1: Χαρακτηριστικά “modules”

Ακολουθεί περιληπτική επεξήγηση της παραπάνω εικόνας (2.3).

#### **To Core container αποτελείται απο:**

Beans και Core: Αυτα τα modules αντιπροσωπεύουν το IoC (Inversion of Control) και το DI (Dependency Injection).

Context: Οπου παρέχει τα μέσα πρόσβασης στα αντικείμενα (objects).

SpEL (Spring Language Expression): Αποτελεί την γλώσσα εκείνη για αναζήτηση (querying) και χειρισμό (manipulating) των αντικειμένων (objects).

AOP (Aspect Orientation Programming): Το AOP αφορά πράγματα όπως logging και security και άλλα.

Aspects: Παρέχει ενσωμάτωση με AspectJ

Instrumentation: Συνδέει τον application server

Messaging: Το module αυτο παρέχει εφαρμογές για μηνύματα.

**Το επίπεδο Data Access/Integration αποτελείται από :**

JDBC: Επιτρέπει την δημιουργία εφαρμογών JDBC.

ORM: Επιτρέπει την σύνδεση με το Hibernate.

OXM: Επιτρέπει το mapping των αντικειμένων και την ενσωμάτωσή τους με προγράμματα όπως το JAXB, Castor και άλλα.

JMS: Επιτρέπει την ενσωμάτωση με υπηρεσίες JMS(Java Messaging Service).

Transaction: Επιτρέπει την δημιουργία Transactional εφαρμογών.

**Το επίπεδο Web αποτελείται από:**

Το συγκεκριμένο module είναι και το πιο σημαντικό καθώς χρησιμοποιείται συνέχεια στις εφαρμογές Spring.

**Τέλος ακολουθεί το module Test:**

Το Module αυτό μας δίνει την υποστήριξη εκείνη για την δημιουργία εφαρμογών για να κάνουν unit testing και integration testing .

## ΚΕΦΑΛΑΙΟ 5

### ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο γίνεται σύντομη περιγραφή του Spring framework JDBC το οποίο αποτελεί ένα πολύ ισχυρό εργαλείο για την σύνδεση των application μας με την βάση δεδομένων

#### 5.1 Σε τι οφελεί το JDBC framework;

Το καλύτερο παράδειγμα για να δείξουμε τα οφέλη του JDBC αποτελεί ο παρακάτω πίνακας. Ο πίνακας δείχνει τις υποχρεώσεις του developer και τι αναλαμβάνει το Spring. Έτσι καταλήγουμε στο συμπέρασμα ότι τα περισσότερα πράγματα δεν χρειάζεται να τα κάνουμε εμείς.

Action	Spring	You	
Define connection parameters.		X	
Open the connection.	X		
Specify the SQL statement.		X	
Declare parameters and provide parameter values		X	
Prepare and execute the statement.	X		
Set up the loop to iterate through the results (if any).	X		
Do the work for each iteration.		X	
Process any exception.	X		
Handle transactions.	X		
Close the connection, statement and resultset.	X		

**Εικόνα 5.1: JDBC framework**

## **ΣΥΜΠΕΡΑΣΜΑ**

Το Spring JDBC είναι ένας μηχανισμός να συνδεθούμε με την βάση δεδομένων και να εκτελέσουμε SQL queries, χρησιμοποιώντας λιγότερο κώδικα.

## **ΚΕΦΑΛΑΙΟ 6**

### **ΕΙΣΑΓΩΓΗ**

Αυτο κεφάλαιο αποτελεί το manual της εφαρμογής. Γίνεται αναλυτική περιγραφή για τα βήματα που ακολούθησα για την υλοποίησή της.



## 6.1 Δημιουργία dynamic web project μετατροπή σε Maven project και προσθήκη απαραίτητων dependencies.

Μέσα στον φάκελο pom.xml βρίσκονται όλα τα dependencies του Project.

## 6.2 Προσθήκη του MVC Dispatcher Servlet.

Ο Dispatcher Servlet είναι μια κλάση που παίζει έναν βασικό ρόλο στην εφαρμογή μας, στην ουσία είναι ένας super controller ο οποίος δρομολογεί όλα τα εισερχόμενα http requests. Μέσα στον φάκελο web.xml βρίσκονται οι ρυθμίσεις του Dispatcher Servlet, ακολουθεί σχετικό παράδειγμα του κώδικα

```
<servlet>
  <servlet-name>dispatcher</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>/WEB-INF/applicationContext.xml</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>dispatcher</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping>
```

## 6.3 Δημιουργία του Spring Controller, “C κομμάτι από το MVC”

Στην εφαρμογή μου έχω φτιάξει 3 controller κλάσεις “@controller”, όλες έχουν τα κατάλληλα annotation τα οποία αποτελούν οδηγίες για τον compiler.

## 6.4 Ρύθμιση του applicationContext.xml και MVC.

Μέσα στο αρχείο applicationContext.xml έχω προσθέσει το context component scan, ακολουθεί παρακάτω :

```
<!-- Where the Component, Repositories etc. annotations are-->
<context:component-scan base-package="main.com" />
```

Αποτελεί έναν μηχανισμό να βρίσκει τα beans μέσα στον application context, βέβαια τα beans έχουν τα κατάλληλα annotation.

Επίσης έχω προσθέσει και το παρακάτω:

```
<!-- Add support for conversion, formatting and validation support -->
<mvc:annotation-driven/>
```

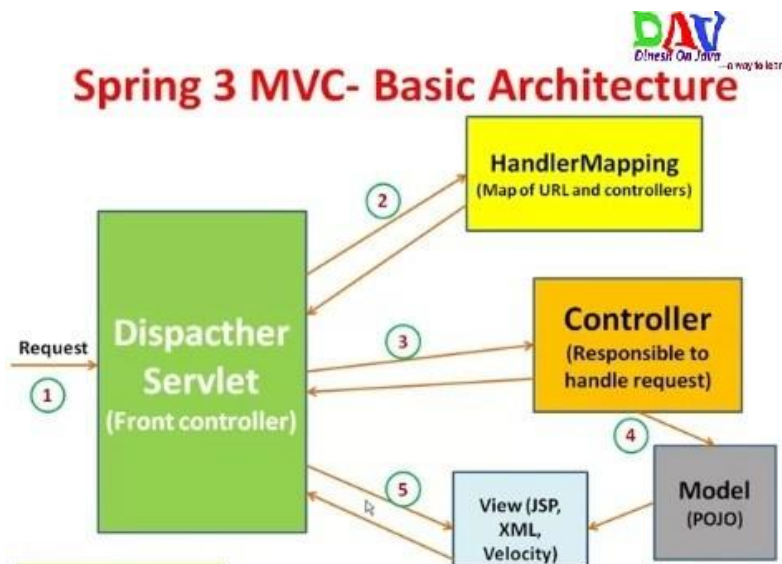
## 6.5 Λίγα λόγια για το Autowiring.

Το autowiring βοηθάει το Spring αυτόματα να καλύπτει τα bean's dependencies βρίσκοντας αλλα beans στο application context που ταιριάζουν με τα τις ανάγκες. Το ενα bean είναι depended με το αλλο και ούτω καθεξής.

## 6.6 Προσθήκη ViewResolver.

Ο dispatcher servlet χρειάζεται ενα view resolver ωστε να μεταφράζει ενα όνομα σε view, δηλαδή να επιστρέφει μια κατάληξη .jsp . Το Spring διαθέτει πληθώρα ViewResolver κλάσεων, στην εφαρμογή μου χρησιμοποίησα το InternalResourceViewResolver , στο applicationContext.xml πρόσθεσα τα παρακάτω:

```
<!-- Define Spring MVC view resolver -->
<!--Where are the jsp pages-->
<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
  <property name="prefix" value="/WEB-INF/views/" />
  <property name="suffix" value=".jsp" />
</bean>
```



Εικόνα 6.1: Dispatcher Servlet

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Spring framework Masterclass : Beginner to professional , Tim buchalka .  
<https://www.udemy.com/java-spring-framework-masterclass/learn/v4/content>
- [2] Spring Framework Master Class - Beginner to Expert , In 28 minutes .  
<https://www.udemy.com/spring-tutorial-for-beginners/learn/v4/t/lecture/8068606?start=15>
- [3] Spring MVC Essentials: A Primary Course for Java Spring , TELCOMA .  
<https://www.udemy.com/spring-mvc-essentials-java-web/learn/v4/t/lecture/7907176?start=0>
- [4] The Java Spring Tutorial: Learn Java's Popular Web Framework , john Purcel .  
<https://www.udemy.com/javaspring/learn/v4/t/lecture/372550?start=15>

**[5]** Introduction to Databases and SQL Querying , Rakesh Gopalakrishnan .

<https://www.udemy.com/introduction-to-databases-and-sql-querying/learn/v4/t/lecture/3250986?start=0>