<https://docs.google.com/spreadsheets/d/10cNfYi5PRYWT2m7h1OxoxWlUs_gecL5XJLbEMRB4NeA/edit#gid=1752438153>

Week 8 (Spring & Spring Boot)

Spring is a Open Source Java Based Back End Framework.

Spring is used to create Enterprise Level Java Applications.

Enterprise Applications – Applications which handles Lot of users, Lot of Access Methods, [WhatsApp, Twitter, LinkedIn, Gmail, --- Web Interface (web.whatsapp.com, mobile App, download & install App for WhatsApp desktop]

Banking App (Security, High Availability, Load Balancing) – Enterprise Applications will have these properties.

Spring has many modules. [Core, Aspect [AOP], Web [MVC], Data [JPA,ORM] ]

Spring uses two important Design Patterns (IoC – Inversion of Control, DI – Dependency Injection )

Spring will manage the life cycle of beans. Create object and inject at the run time (@Autowired)

Types of Injection ( Constructor Injection [CI], Setter Injection [SI]) – Object initialization happens

Scope of the Beans [ Request, Application, Session, Singleton, Prototype]

Spring Boot – Spring Boot is a way of creating Spring Application.

Spring Boot – will help you to easily create spring based application.

The main Challenge in Spring framework is, xml configuration file (applicationContext.xml, spring-config.xml) XML is case & space sensitive.

Spring is called as Framework of Frameworks (Bcos It supports other frameworks like struts, Hibernate….)

SpringBoot is called as Opinionated Framework. (It can automatically configure based on pom.xml file and application.properties file)

Configuring Spring Boot

1. Using application.properties (Set of pre-defined key-value pairs)
2. Using Annotations
3. Using pom.xml dependencies

SpringBoot Annotations

1. @SpringBootApplication [@AutoConfigure, @ComponentScan, @

SpringBoot helps to create REST based/SOAP based Web Service

API /Web Service – Help to execute some methods using URI

URI – Uniform Resource Identifier

URL – Uniform Resource Locator

//Setter Injection

Employee emp = new Employee();

emp.setId(10);

emp.setEmpName(“ABC”);

<bean id=”emp” class=”com.revature.Employee”>

<property name=”id” value=10/>

<property name=”empName” value=”ABC”/>

</bean>

//Example for Constructor Injection

Employee emp1 = new Employee(11,”XYZ”);

<bean id=”emp1” class=”com.revature.Employee”>

<constructor-arg index=”0” value=11/>

< constructor-arg index=”1” value=”XYZ”/>

</bean>

@Bean

Public Employee getEmployee() {

return new Employee();

}

@Autowired

Employee e;

@Entity

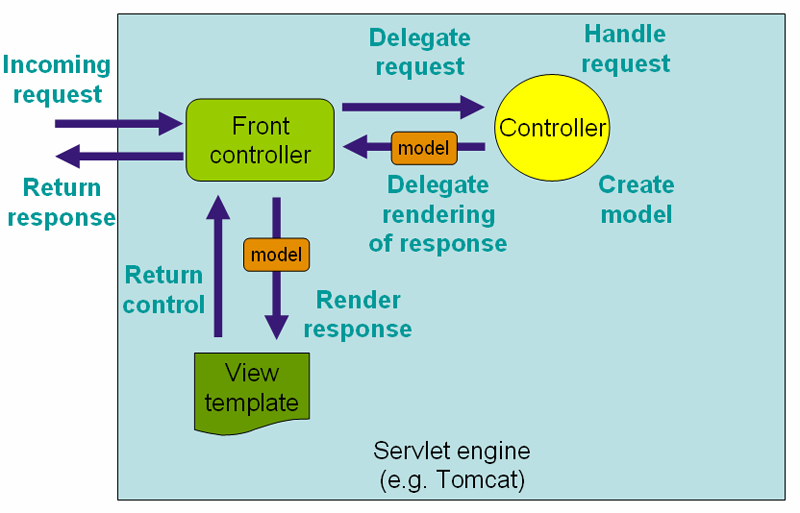
Q: What is front Controller in Spring MVC

A: Dispatcher Servlet

Spring MVC ( Model View Controller)

Spring Web Module

1. Adding spring Servlet details to web.xml (Entry point of Web Application)
2. Spring Servlet (Front Controller – Dispatcher Servlet) <url- mapping> \*.html
3. @Controller class (Will have mapping url for the View, and also provide ModelAndView Object)
4. View Resolver (prefix & suffix)
5. @Controller
6. @RequestMapping("/appointments")



@RequestParam – Used to Read the data from Request Header parameter

@PathVariable – Used to read the data from URL

Spring MVC Annotations

* Request Handling
  + @Controller
  + @RestController
* Data Binding
  + @RequestBody
  + @PathVariable
  + @RequestParam
  + @ResponseBody
* RequestMapping
  + @RequestMapping
  + @GetMapping
  + @PostMapping
  + @PutMapping
  + @DeleteMapping
* Data Transformation
  + the consumes and produces attributes of the RequestMapping annotations.