

Spring Boot Training

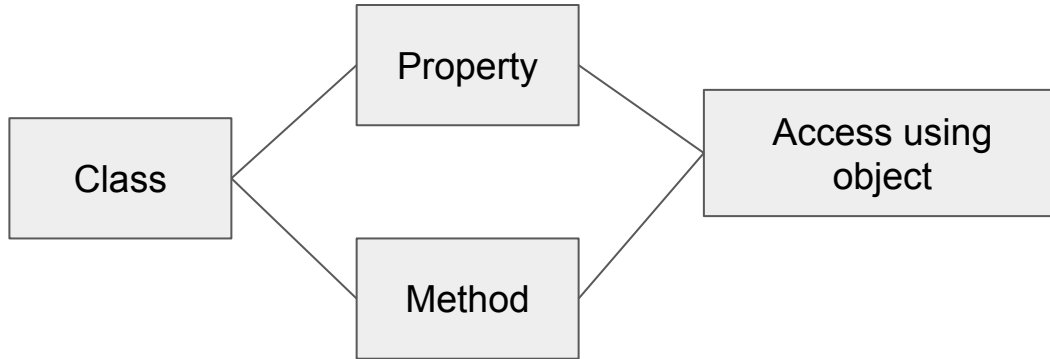
Jupiter

Pre-Test

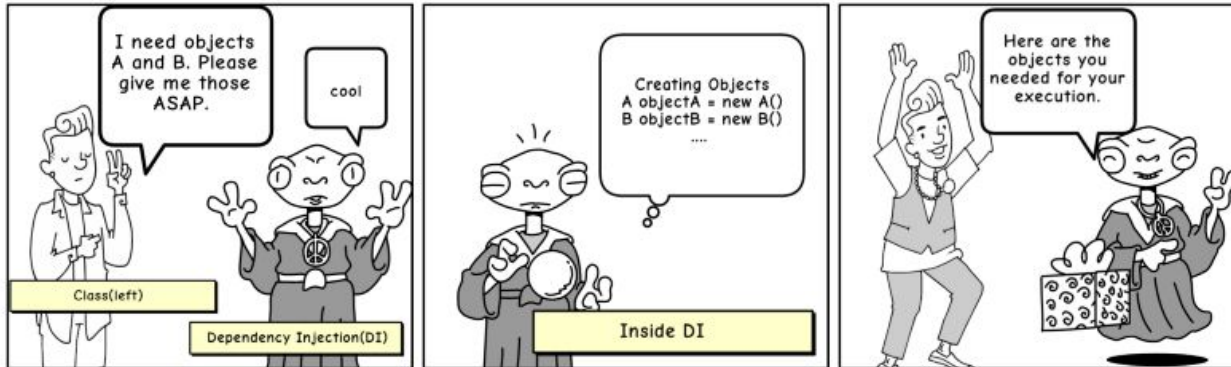
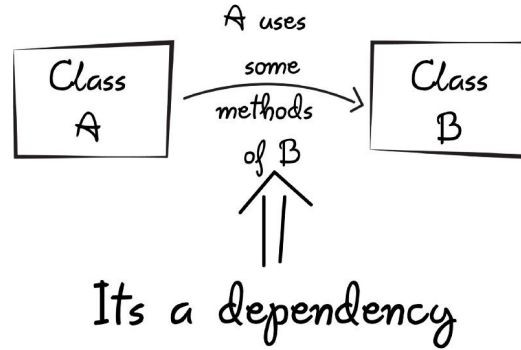
1. Apa yang anda ketahui tentang IOC atau dependency injection.
2. Apa perbedaan overloading vs overriding
3. Sebutkan dan jelaskan spring module yang anda ketahui? Minimal (3)
4. Jelaskan Perbedaan antara Spring JDBC dengan Spring JPA
5. Apa Perbedaan Web Service dengan REST API
6. Sebutkan Template Engine yang biasanya digunakan di Java EE

Kirim Jawaban email ke : jupiterc@gmail.com dengan subject:
trainingjava_namapeserta

Java Introduction



Dependency Injection



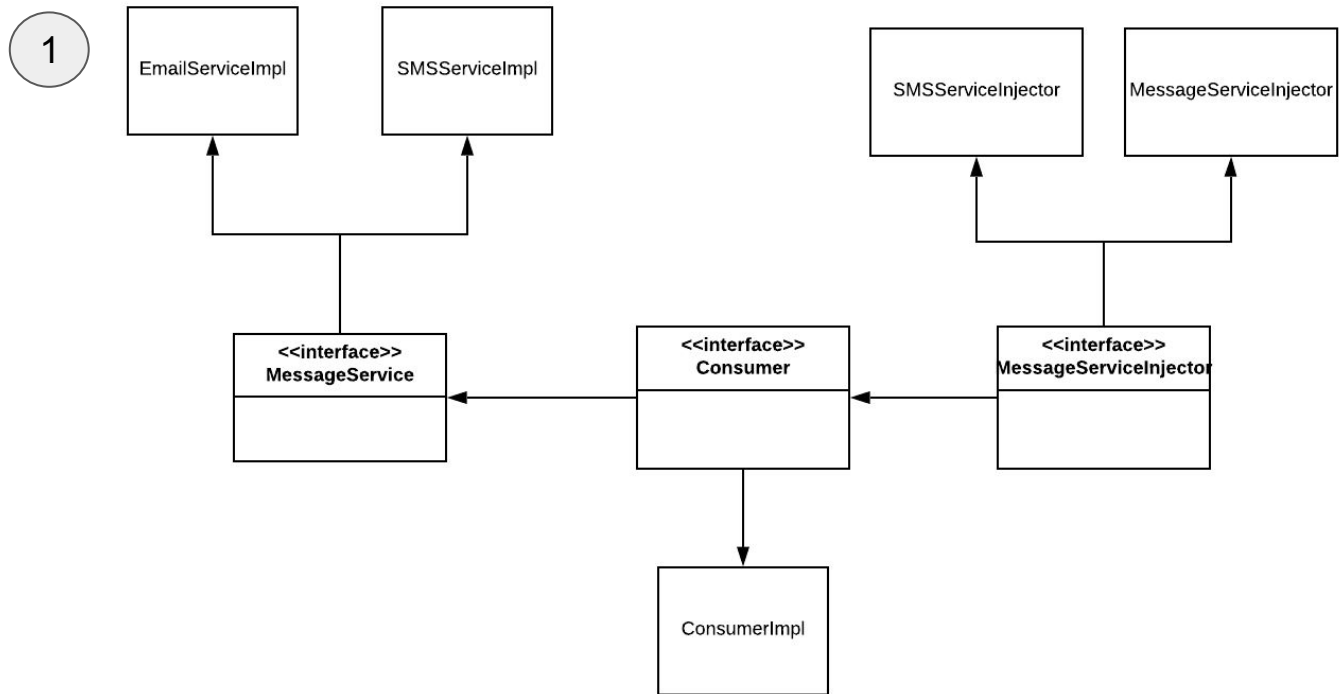
This comic was created at www.MakeBeliefsComix.com. Go there and make one now!

Dependency Injection

Dependency Injection in java requires at least following:

1. Service components should be designed with base class or interface. It's better to prefer interfaces or abstract classes that would define contract for the services.
2. Consumer classes should be written in terms of service interface.
3. Injector classes that will initialize the services and then the consumer classes.

Dependency Injection



Advantages VS Disadvantages

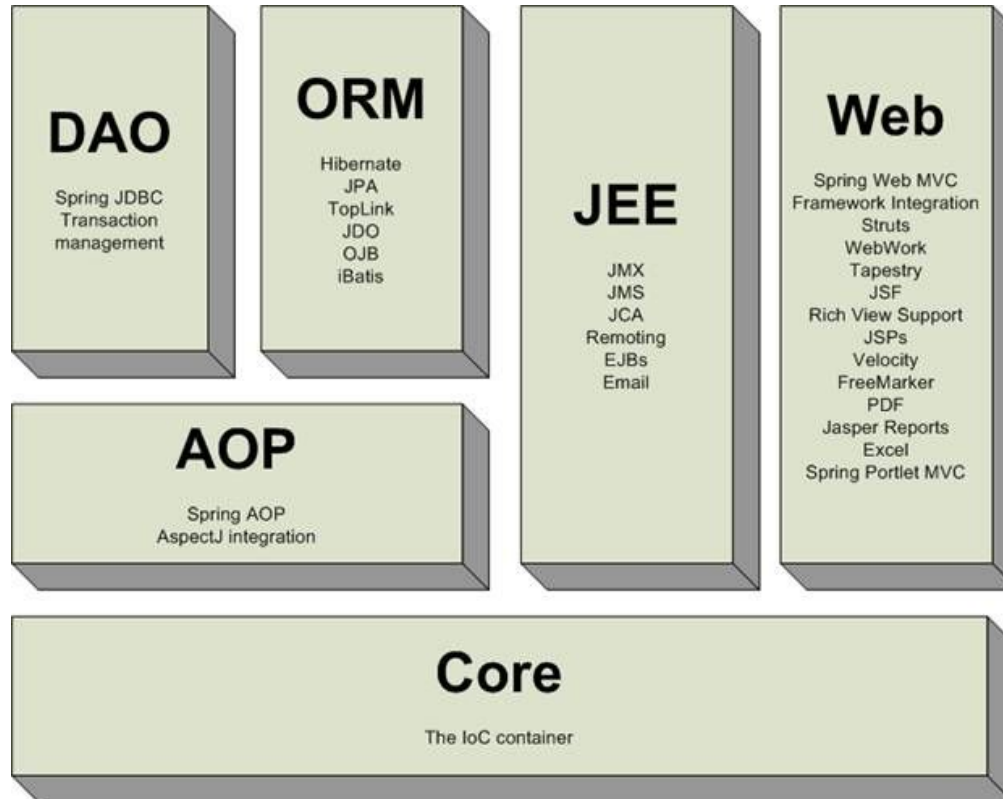
Advantages

- Separation of Concerns
- Boilerplate Code reduction in application classes because all work to initialize dependencies is handled by the injector component
- Configurable components makes application easily extendable

Disadvantages

- If overused, it can lead to maintenance issues because effect of changes are known at runtime.
- Dependency injection in java hides the service class dependencies that can lead to runtime errors that would have been caught at compile time.

Spring Framework



Initializing a Spring Boot project

Through a web-based interface

Via Spring Tool Suite

Via IntelliJ IDEA

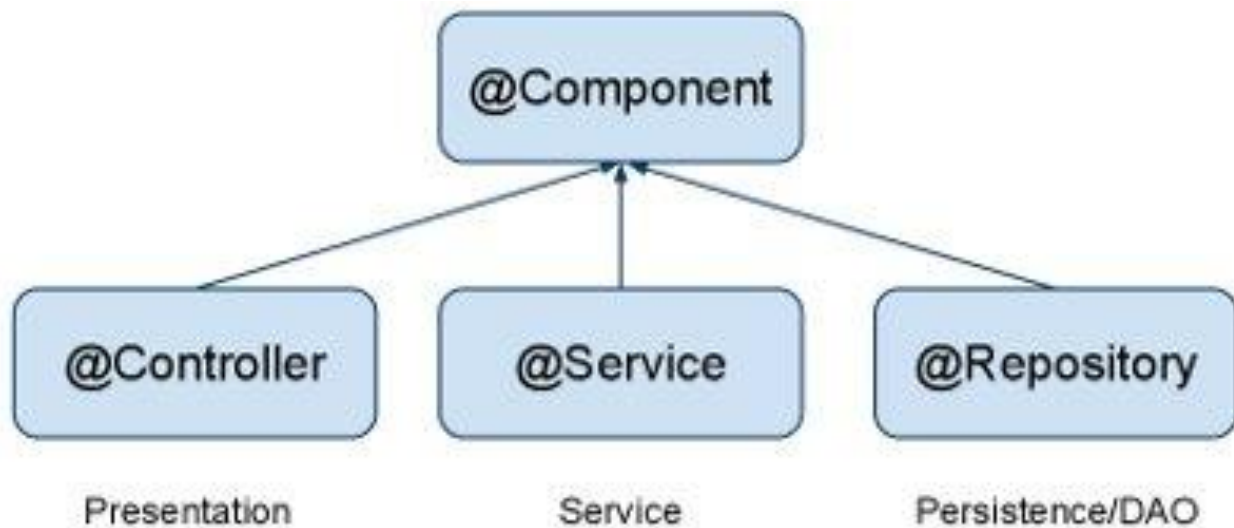
Using the Spring Boot CLI

Why Spring Boot

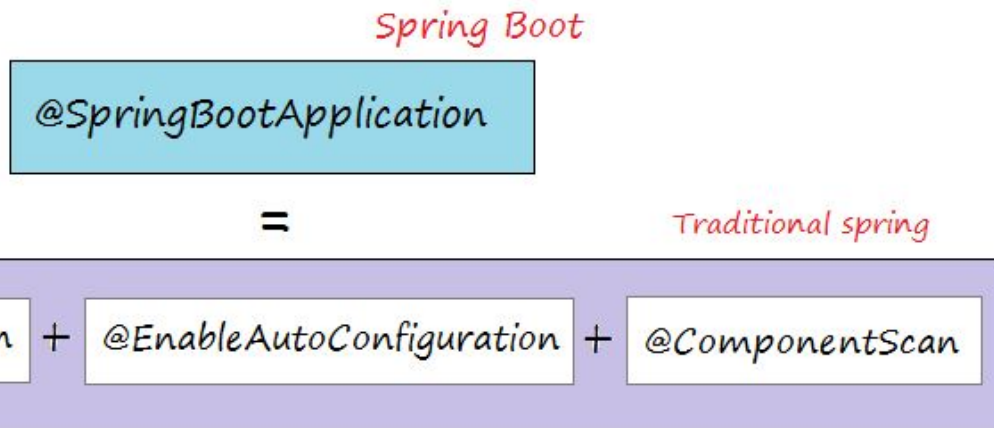
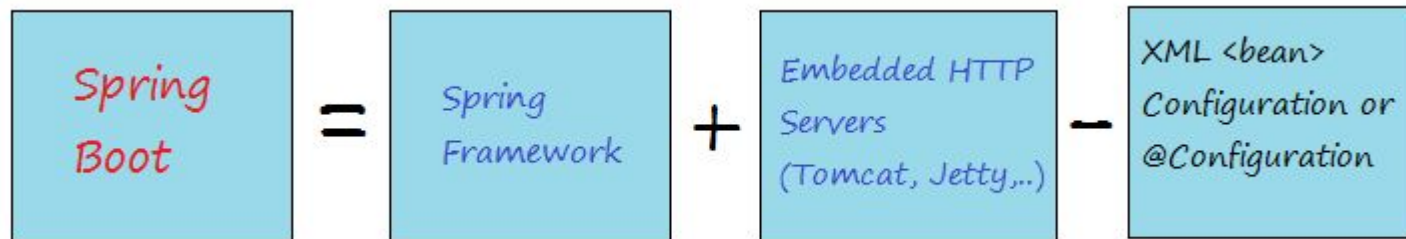
- No Clumsy XML configuration by Developer
- Use Dependency Management Tools Such As Maven and Gradle
- Help Fast Development and production ready code
- Embed Tomcat, Jetty or Undertow directly (no need deploy WAR files)
- In Memory DB
- For Documentation :

<https://docs.spring.io/spring-boot/docs/current/reference/html/index.html>

Annotation



Spring Boot



Build Simple Web using Spring + Thymeleaf



+



Model

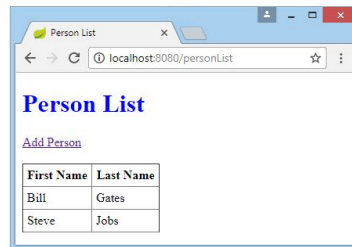
```
public class Person {  
    private String firstName;  
    private String lastName;  
}
```

List<Person> persons

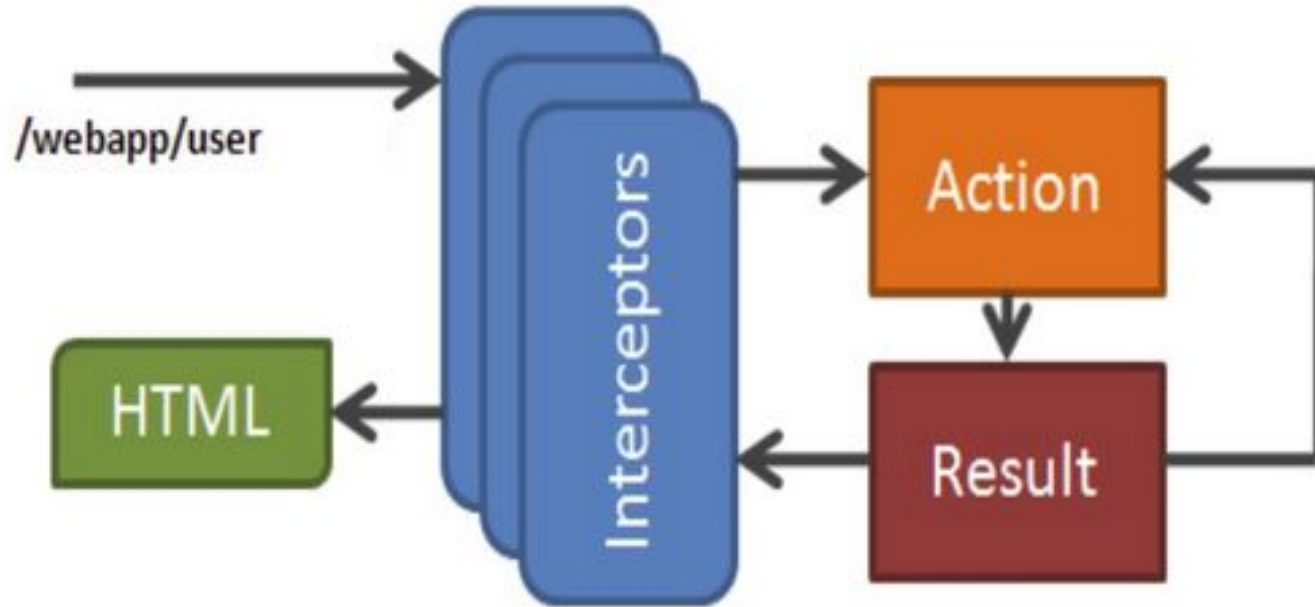
View (Thymeleaf Template)

```
<!DOCTYPE HTML>  
<html xmlns:th="http://www.thymeleaf.org">  
  <head>  
    <meta charset="UTF-8" />  
    <title>Person List</title>  
    <link rel="stylesheet" type="text/css"  
          th:href="@{/css/style.css}"/>  
  </head>  
  <body>  
    <h1>Person List</h1>  
    <a href="#addPerson">Add Person</a>  
    <br/><br/>  
    <div>  
      <table border="1">  
        <tr>  
          <th>First Name</th>  
          <th>Last Name</th>  
        </tr>  
        <tr th:each="person : ${persons}">  
          <td th:text="${person.firstName}">...</td>  
          <td th:text="${person.lastName}">...</td>  
        </tr>  
      </table>  
    </div>  
  </body>  
</html>
```

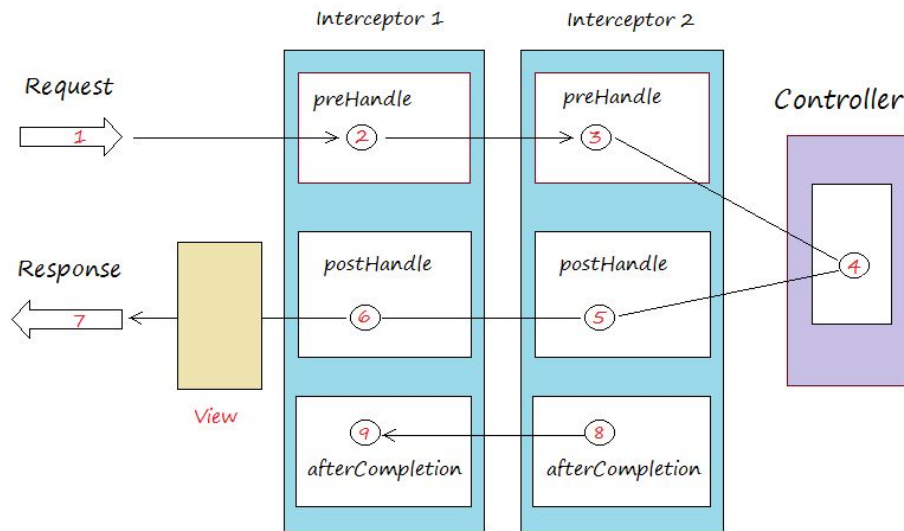
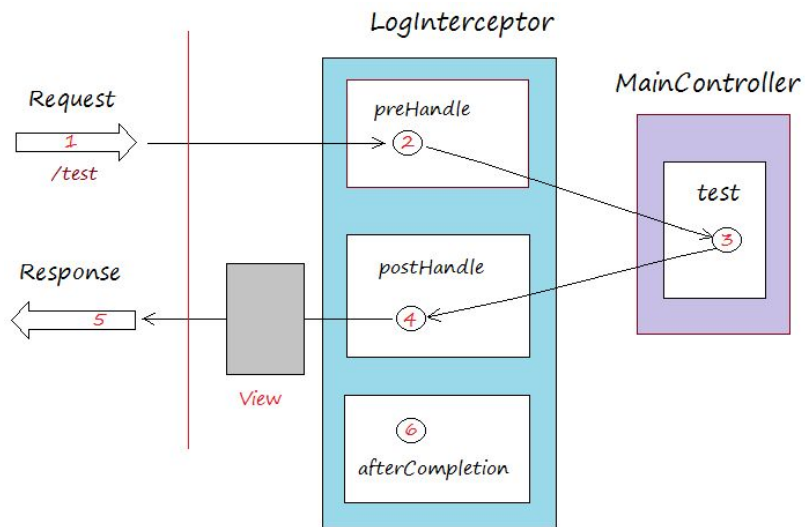
Thymeleaf Engine



Spring Boot Interceptor



Spring Boot Interceptor



Spring Boot JDBC & Spring Transactional



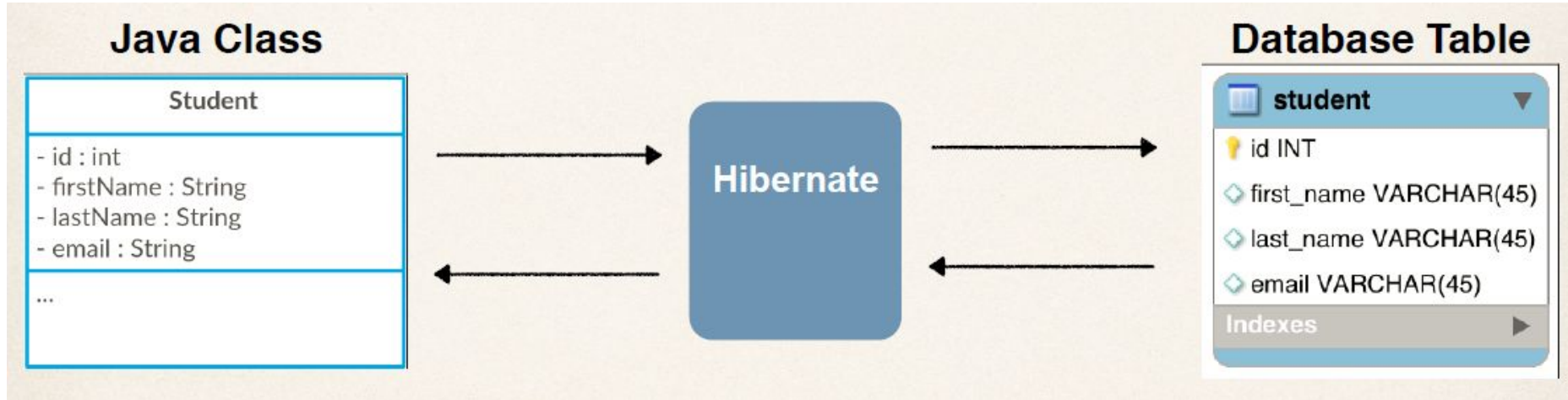
Day 4

JPA Hibernate Mapping

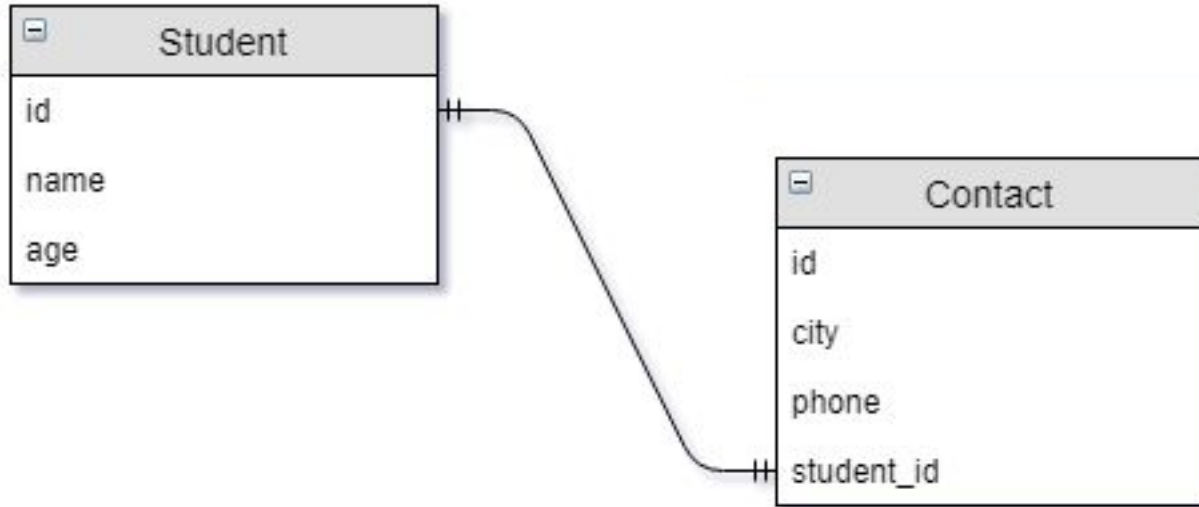
CRUD Restfull

JPA Auditing

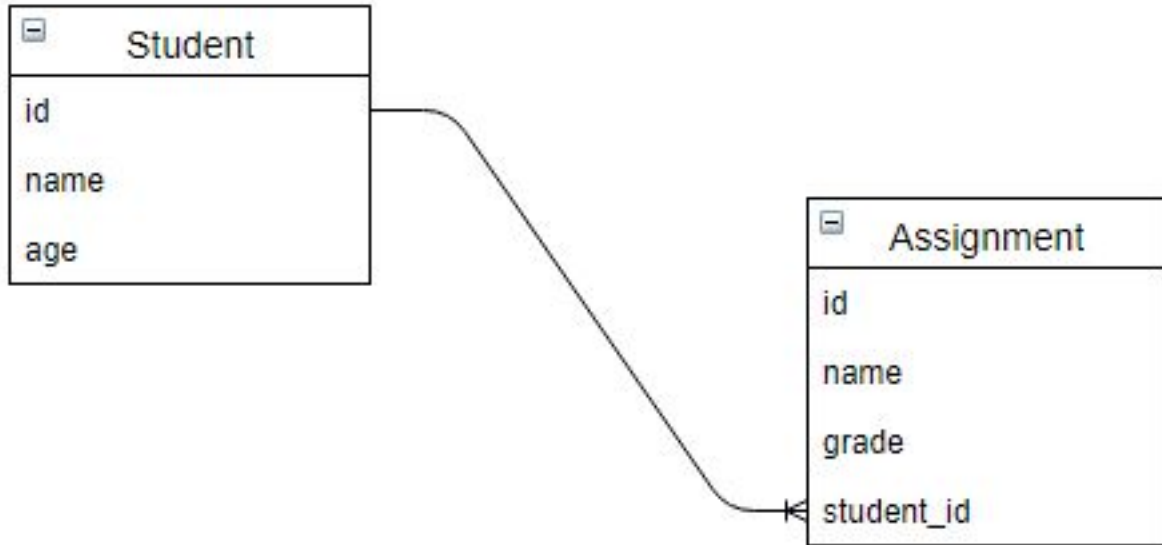
JPA Architecture



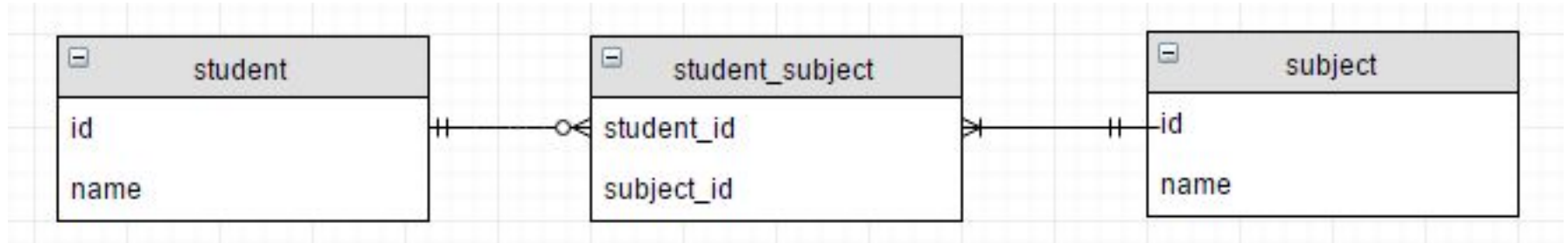
JPA Mapping One To One



JPA Mapping One To Many



JPA Mapping Many To Many



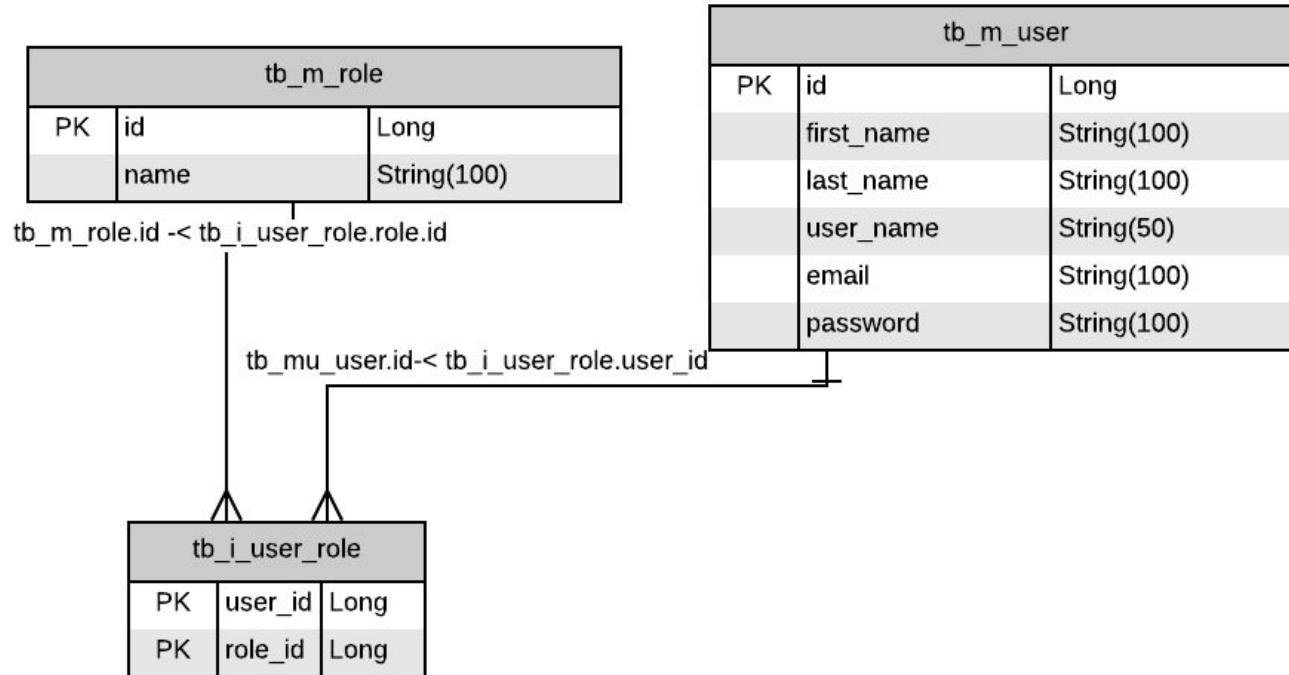
Day 05

- Spring Migration [X]

- Spring Mapping

- Spring Custom Error[X]

JPA Mapping

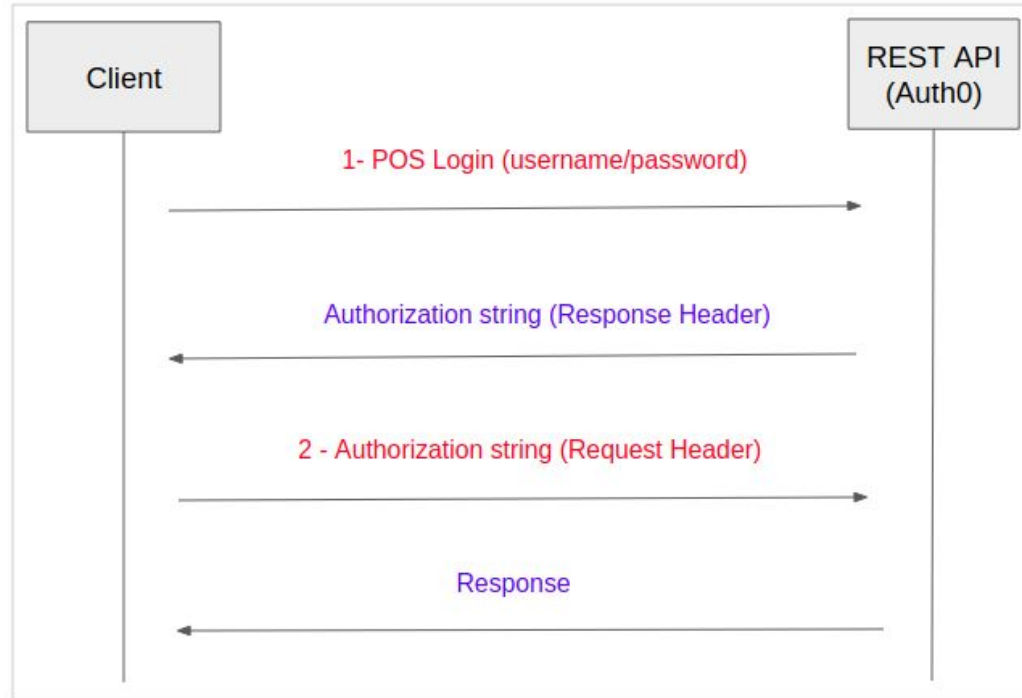


Day07

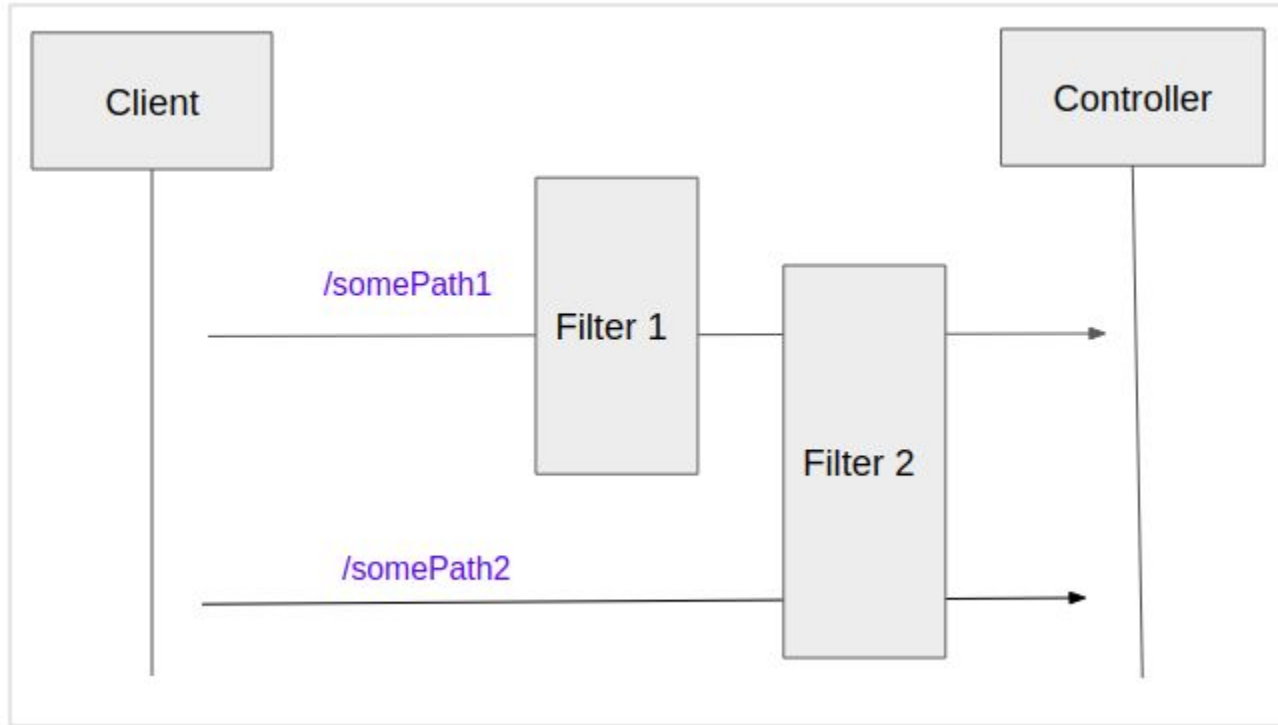
- Spring Security

- angular integration

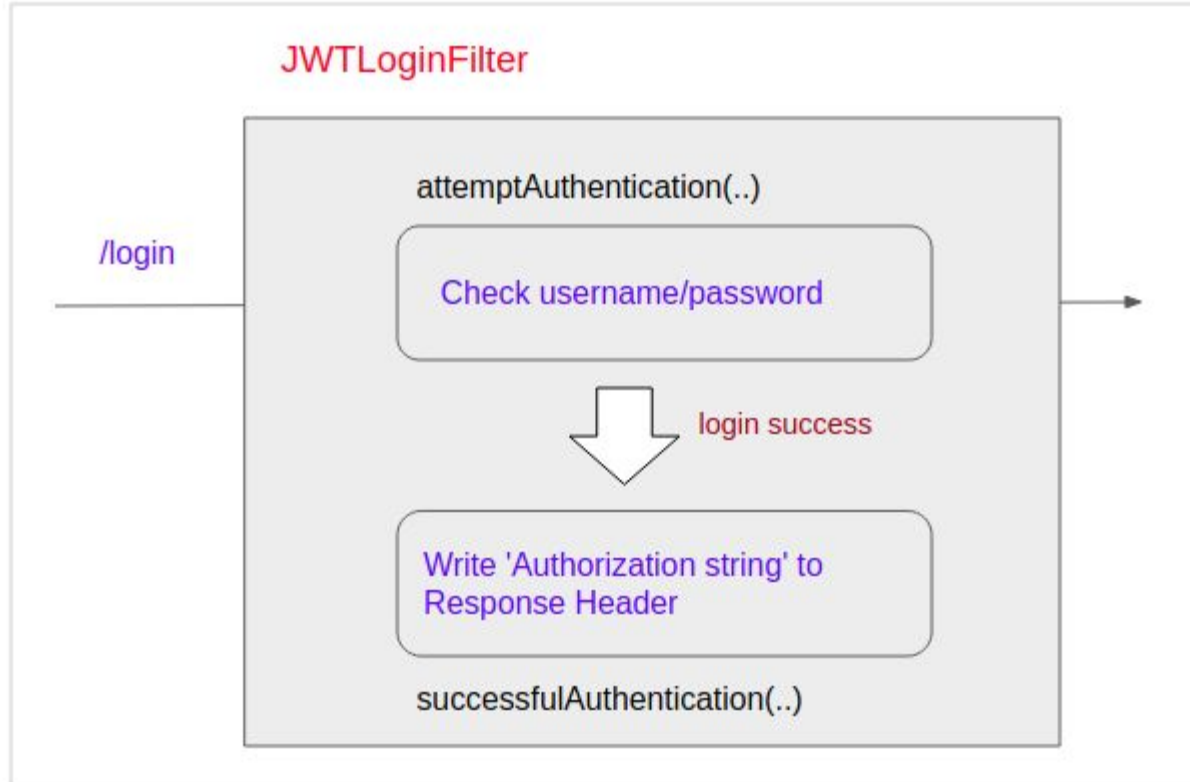
Rest API With Auth0



Rest API With Auth0



Rest API With Auth0



Rest API With Auth0

