

# Mobile Store Check List

TopGun Program

Fresher Academy - 2018



# Day 1



1. Understand Spring MVC Architecture

2. Integrate Spring MVC with Hibernate

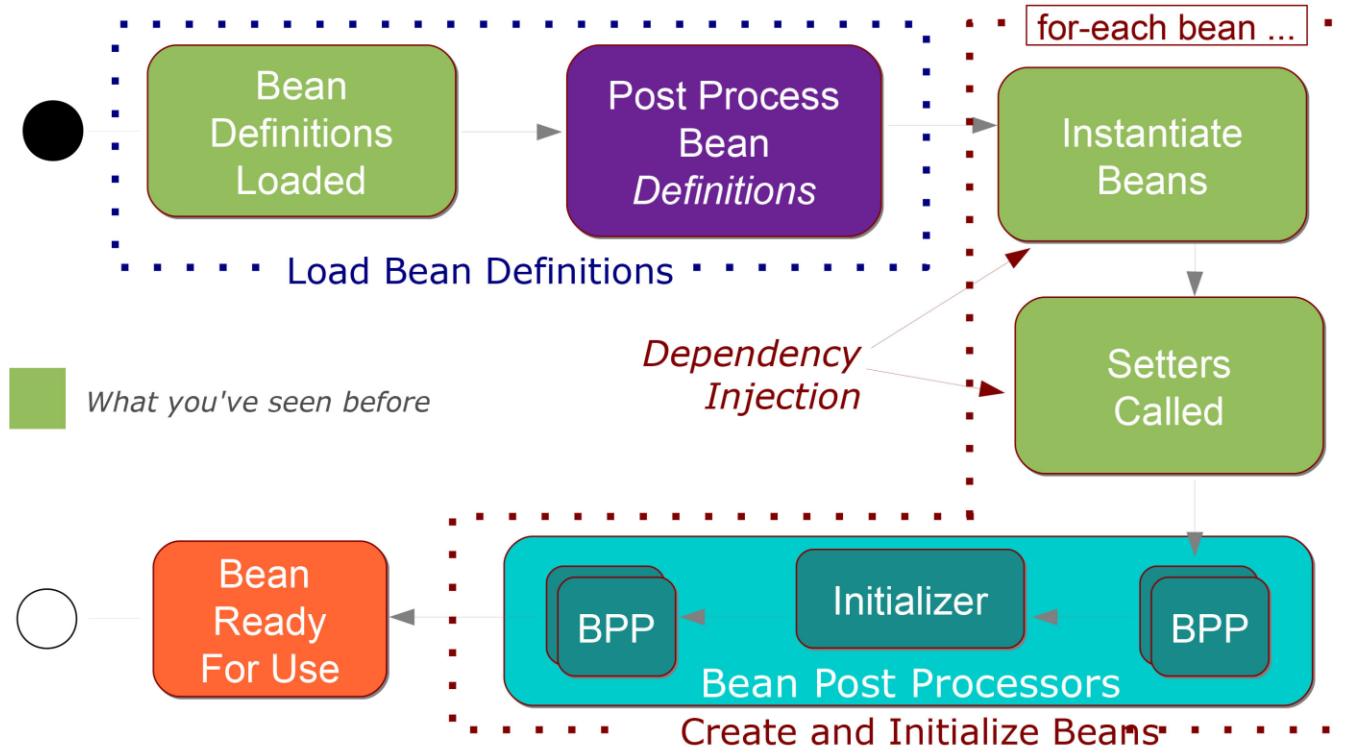
3. Spring Project Structure

# 1. Spring MVC Architecture

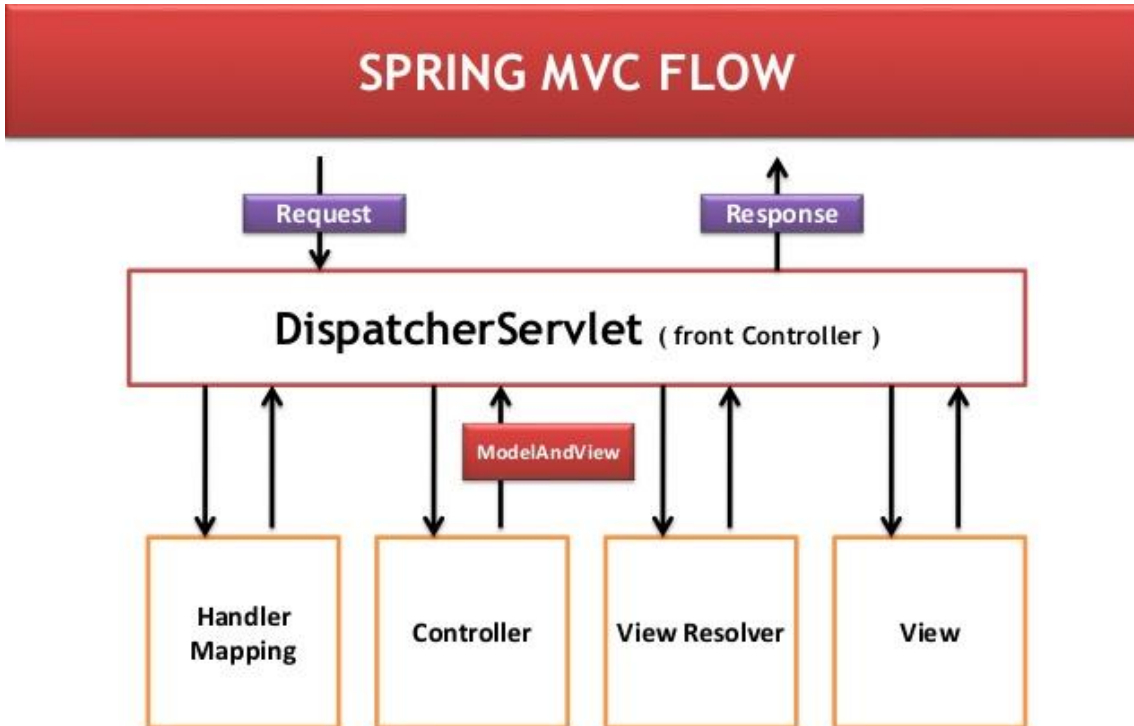
# Spring Bean Life Cycle

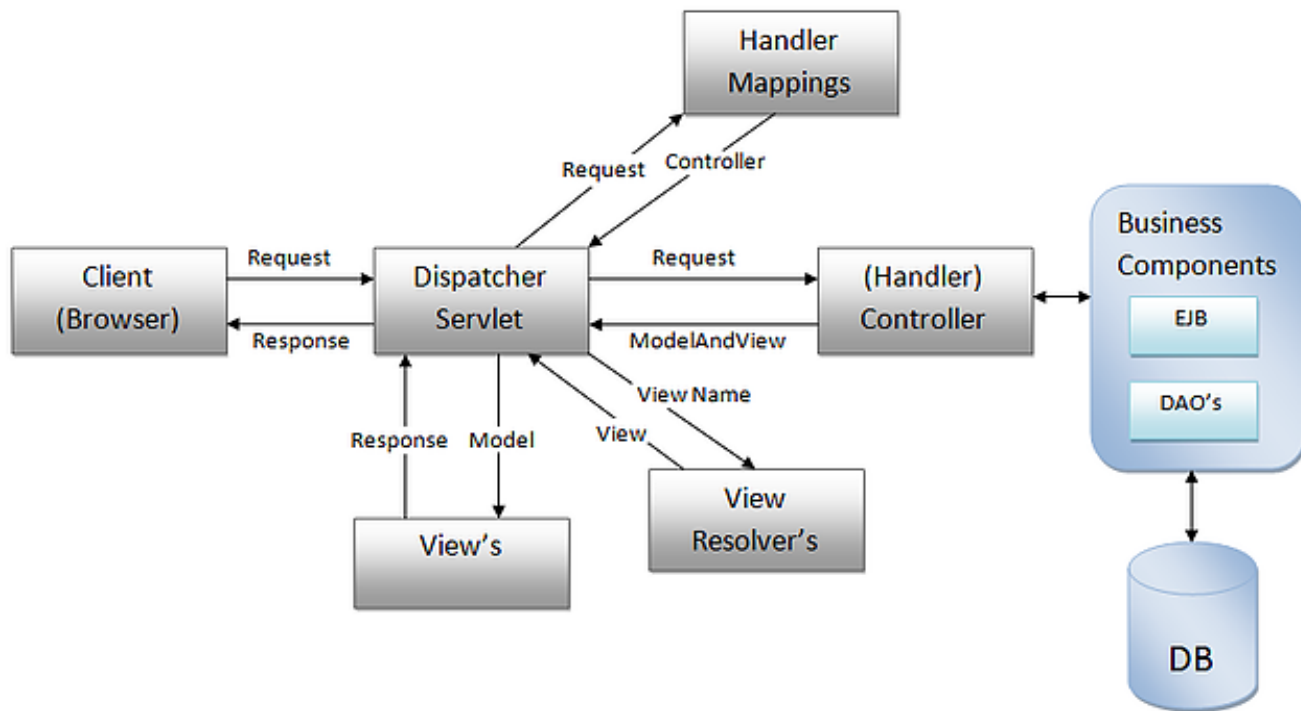


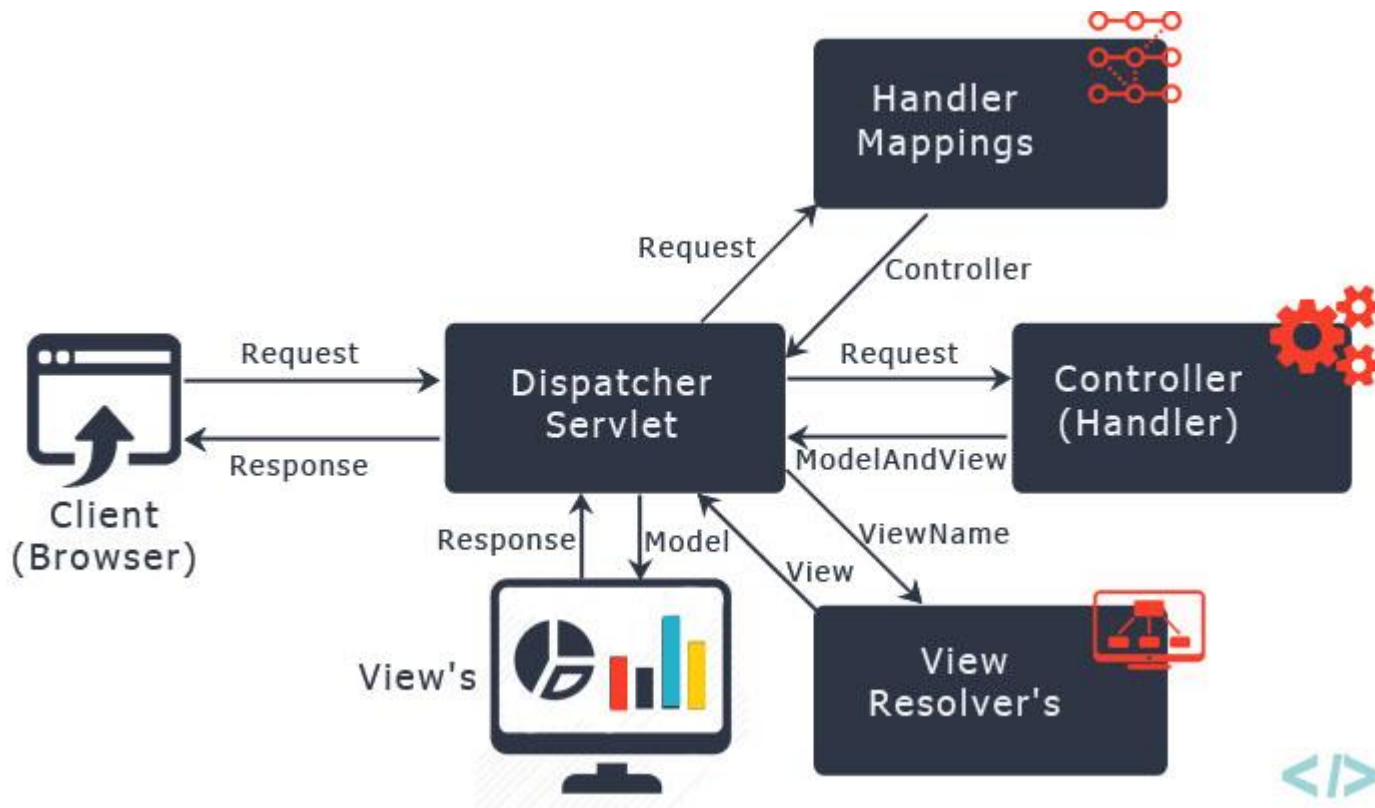
## Bean Initialization Steps



# Spring MVC Architecture

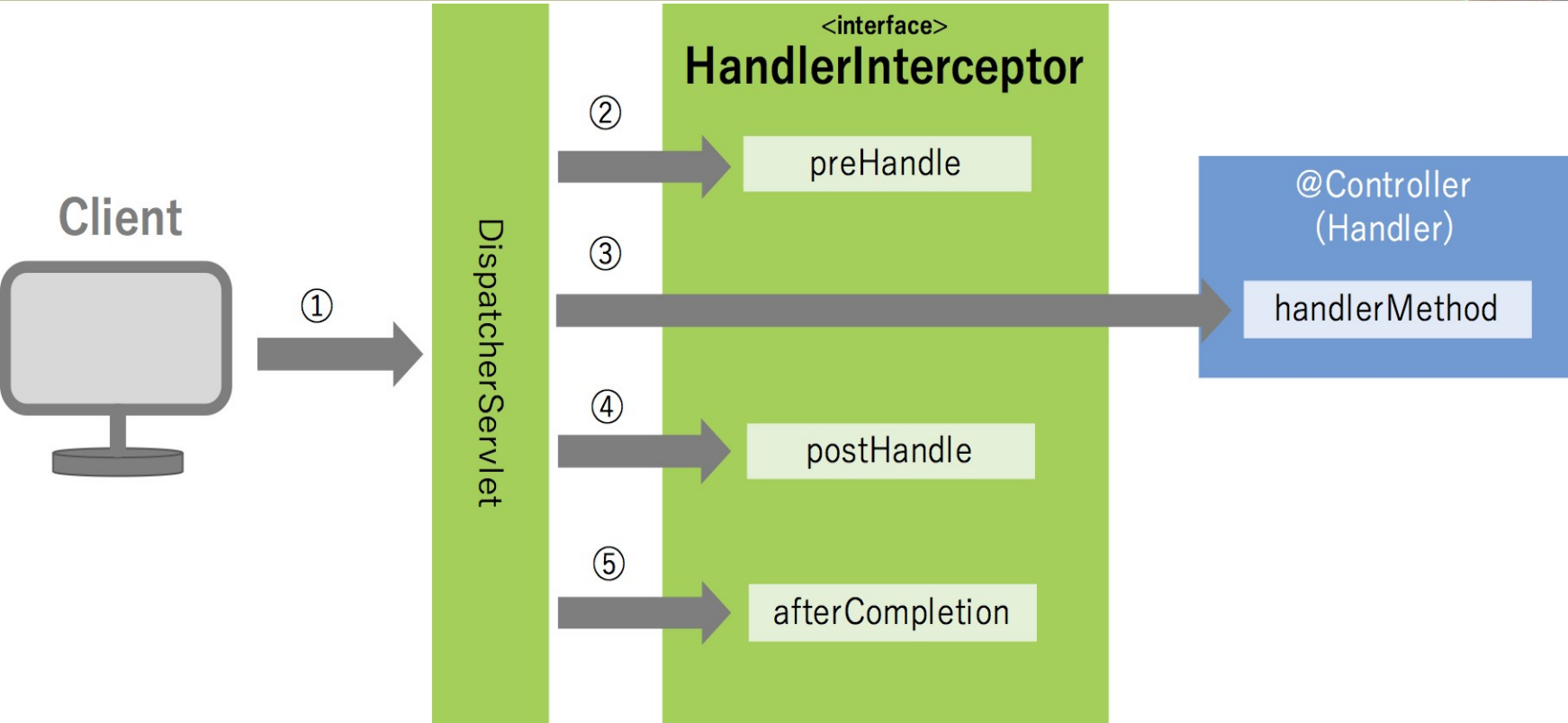






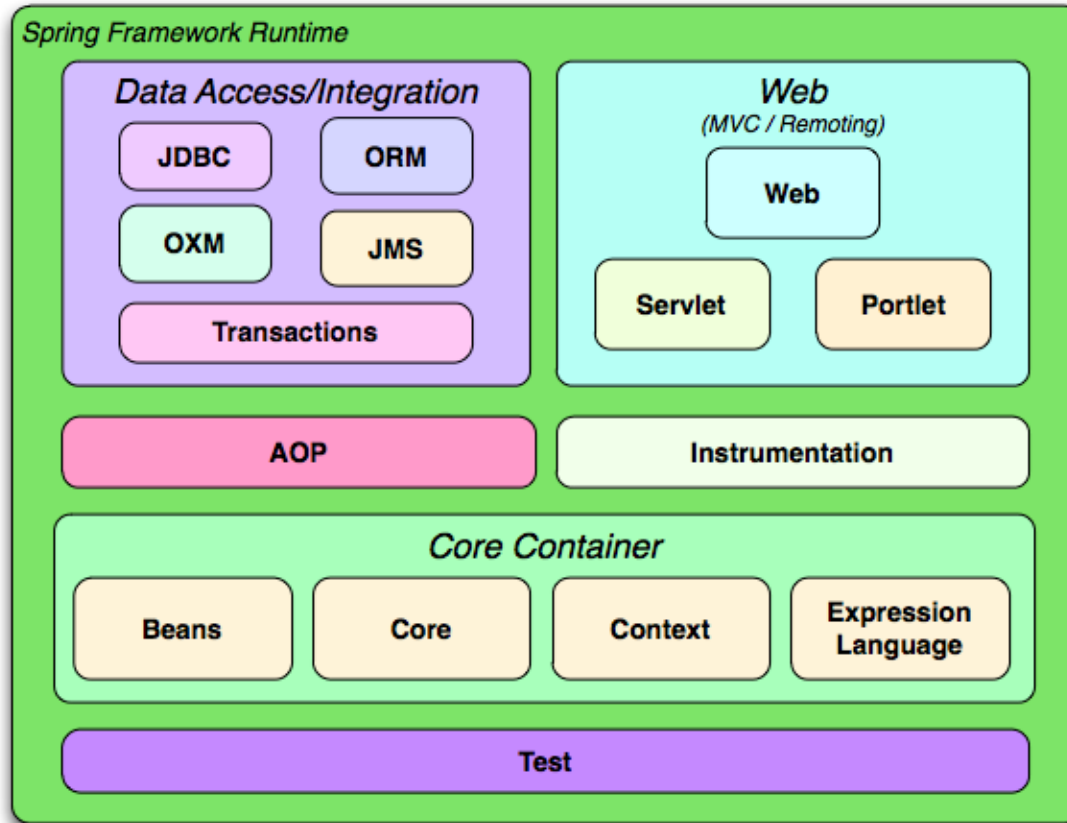


# Interceptor

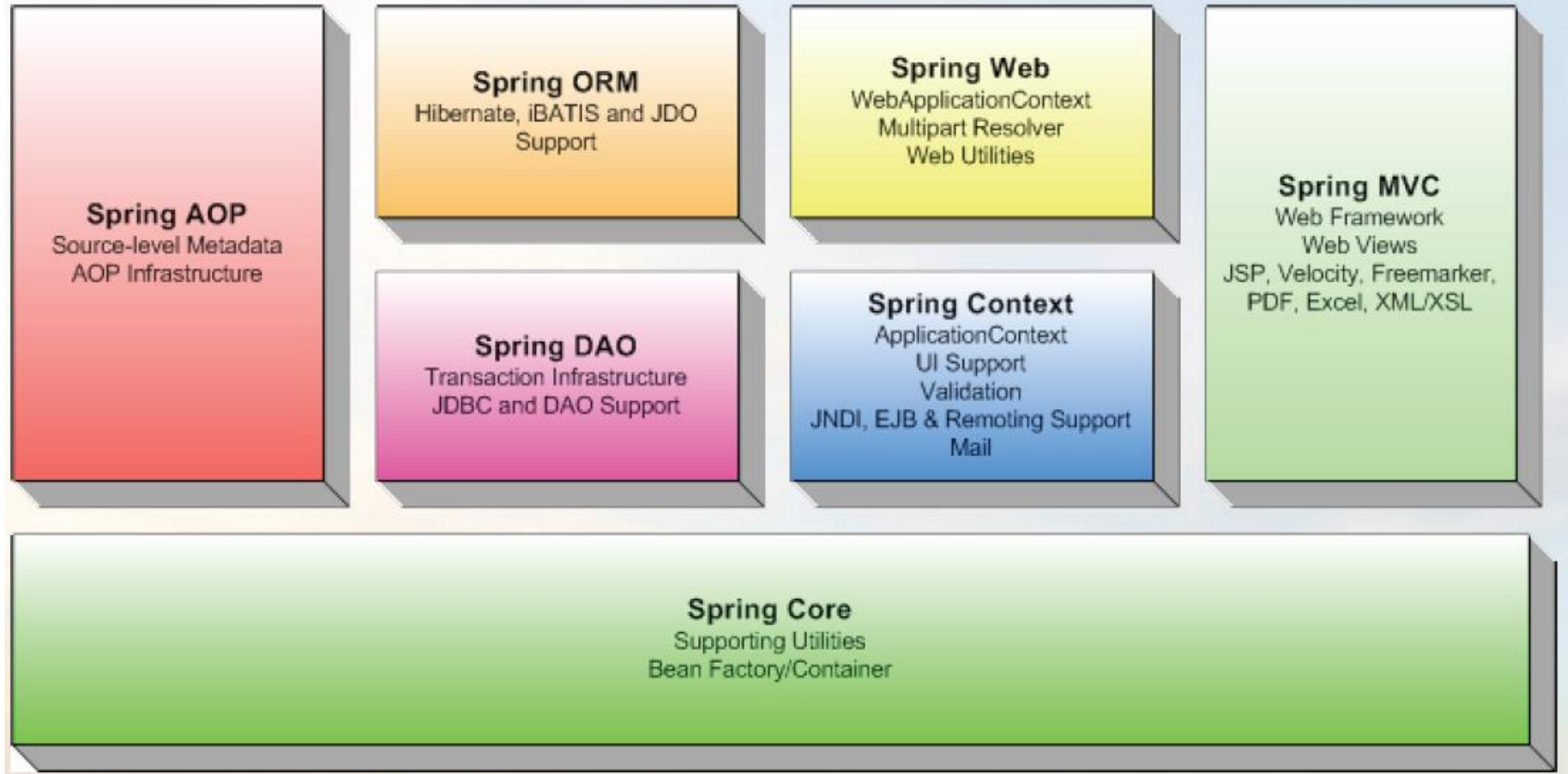




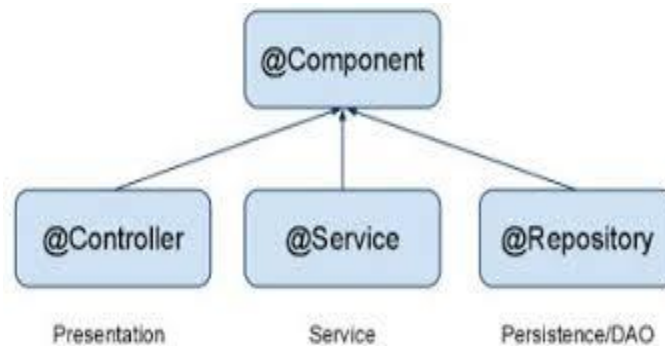
# Spring Modules



# Spring Modules



# Spring Annotations



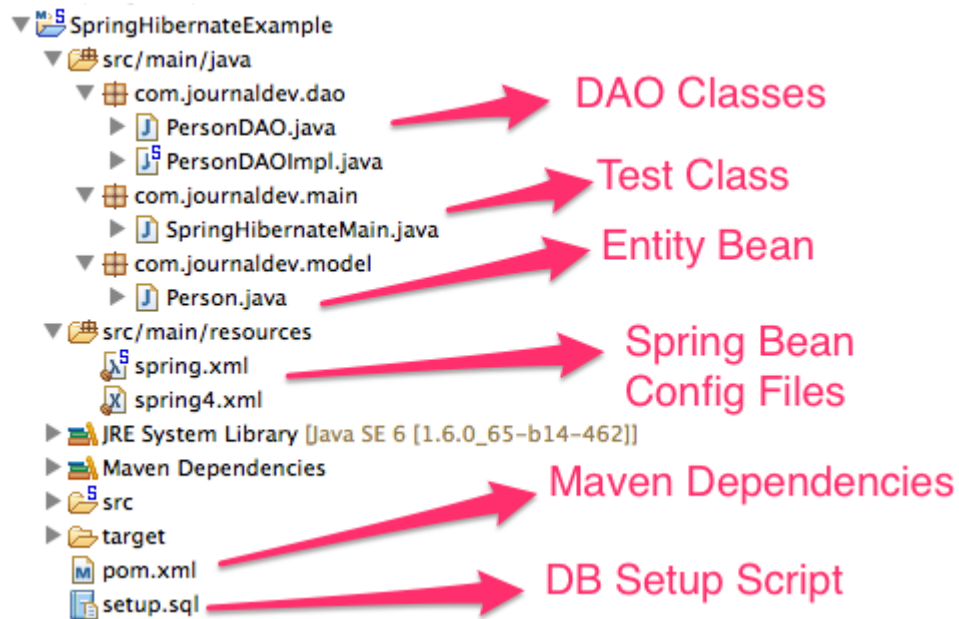
# Spring MVC Annotations

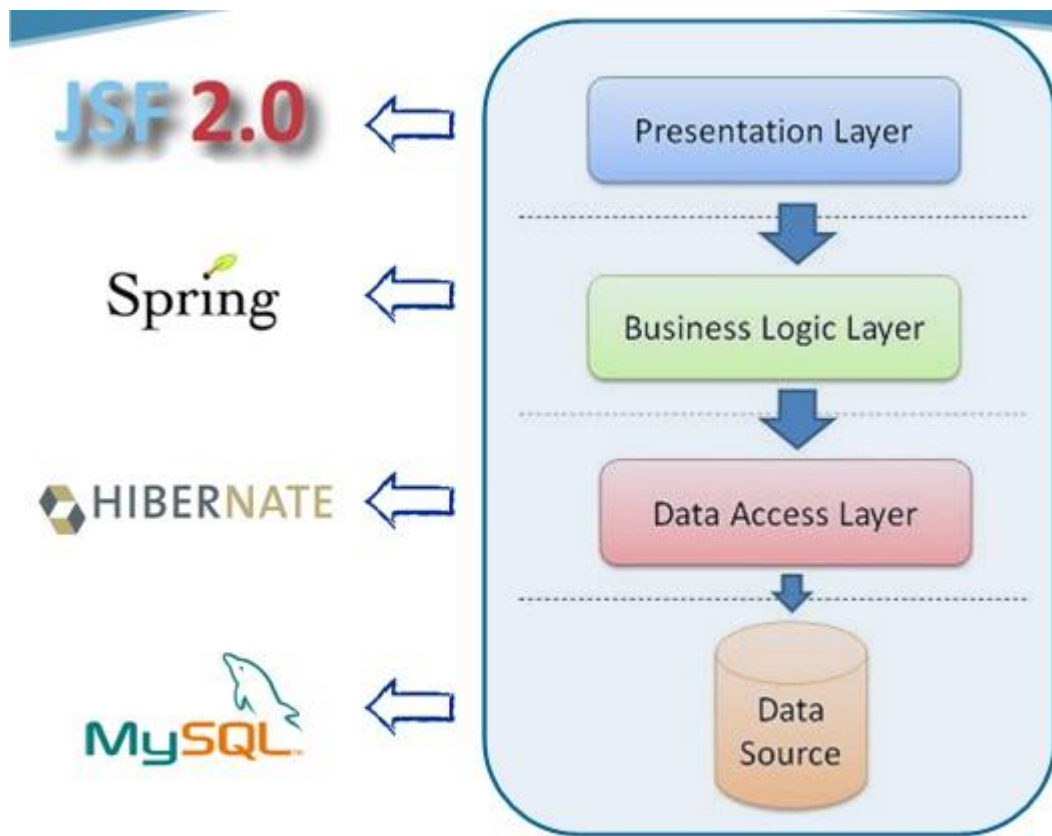


## Spring MVC Annotations

- **@Controller** – class stereotype for controller classes so they can be found and mapped
- **@SessionAttributes** – list model attributes to be stored in the session (command object)
- **@RequestMapping** – class/method mapping to requests (mode, parameters, etc.)
- **@RequestParam** – bind method params to request params
- **@ModelAttribute** – bind method params or return values to model attributes
- **@InitBinder** – method to setup binder for putting form submission into command obj

## 2. Integrate Spring with Hibernate






















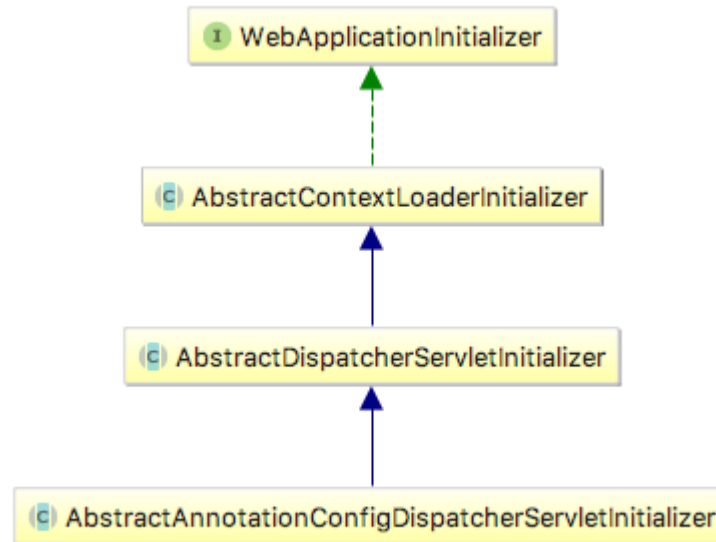


# 3. Spring project structure



- ✓  SpringMvcHibernateAnnotation
  - ✓  src/main/java
    - >  com.fa.edu.springmvchibernate.config
    - >  com.fa.edu.springmvchibernate.controller
    - >  com.fa.edu.springmvchibernate.dao
    - >  com.fa.edu.springmvchibernate.dao.impl
    - >  com.fa.edu.springmvchibernate.model
    - >  com.fa.edu.springmvchibernate.service
    - >  com.fa.edu.springmvchibernate.service.impl
  - ✓  src/main/resources
    -  application.properties
  - >  src/test/java
  - >  JRE System Library [jre1.8.0\_152]
  - >  Maven Dependencies
  - >  src
  - >  target
  -  pom.xml

# WebApplicationInitializer



# WebApplicationInitializer



```
package com.fa.edu.springmvchibernate.config;

import javax.servlet.ServletContext;

public class WebInitializer implements WebApplicationInitializer {

    public void onStartUp(ServletContext servletContext)
        throws ServletException {
        AnnotationConfigWebApplicationContext ctx = new AnnotationConfigWebApplicationContext();
        ctx.register(WebAppConfig.class);
        servletContext.addListener(new ContextLoaderListener(ctx));

        ctx.setServletContext(servletContext);

        Dynamic servlet = servletContext.addServlet("dispatcher", new DispatcherServlet(ctx));
        servlet.addMapping("/");
        servlet.setLoadOnStartup(1);
    }
}
```

# WebAppConfig



```
@Configuration
@ComponentScan("com.fa.edu.springmvchibernate")
@EnableWebMvc
@EnableTransactionManagement
@PropertySource("classpath:application.properties")
public class WebAppConfig {
    private static final String PROPERTY_NAME_DATABASE_DRIVER = "db.driver";
    private static final String PROPERTY_NAME_DATABASE_PASSWORD = "db.password";
    private static final String PROPERTY_NAME_DATABASE_URL = "db.url";
    private static final String PROPERTY_NAME_DATABASE_USERNAME = "db.username";

    private static final String PROPERTY_NAME_HIBERNATE_DIALECT = "hibernate.dialect";
    private static final String PROPERTY_NAME_HIBERNATE_SHOW_SQL = "hibernate.show_sql";
    private static final String PROPERTY_NAME_ENTITYMANAGER_PACKAGES_TO_SCAN = "entitymanager.packages.to.scan";

    @Resource
    private Environment env;

    public DataSource dataSource() {}

    public LocalSessionFactoryBean sessionFactory() {}

    private Properties hibProperties() {}

    public HibernateTransactionManager transactionManager() {}

    public UrlBasedViewResolver setupViewResolver() {}
}
```

# Happy Coding!



thank you!



Fpt Software