

Bộ môn: Kỹ Thuật Phần Mềm

Lập trình www Java



Spring MVC App – All Java Configuration



- » Instead of configuring Spring MVC app using XML
 - **web.xml**
 - **spring-mvc-demo-servlet.xml**
- » Configure the Spring MVC app with Java code

No XML!

1. Add **Maven dependencies** for **Spring MVC Web App**
2. Create **Spring App Configuration** (`@Configuration`)
3. Create **Spring Dispatcher Servlet Initializer**
4. Develop **Spring controller**
5. Develop JSP view page (optional)



Setup Dev Environment - pom.xml



```
<!-- https://mvnrepository.com/artifact/com.microsoft.sqlserver/mssql-jdbc -->
<dependency>
    <groupId>com.microsoft.sqlserver</groupId>
    <artifactId>mssql-jdbc</artifactId>
    <version>8.4.1.jre11</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>5.4.16.Final</version>
</dependency>
<!-- https://mvnrepository.com/artifact/com.mchange/c3p0 -->
<dependency>
    <groupId>com.mchange</groupId>
    <artifactId>c3p0</artifactId>
    <version>0.9.5.2</version>
</dependency>
```

MS SQL Server
connection
driver

Hibernate
driver

Connection
Pool driver

Setup Dev Environment - pom.xml



```
<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-c3p0 -->
```

```
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-c3p0</artifactId>
    <version>5.2.12.Final</version>
</dependency>
<!-- https://mvnrepository.com/artifact/com.mchange/mchange-commons-java -->
<dependency>
    <groupId>com.mchange</groupId>
    <artifactId>mchange-commons-java</artifactId>
    <version>0.2.11</version>
</dependency>
```

Connection
Pool driver

Connection
Pool driver

Setup Dev Environment - pom.xml



```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>5.2.2.RELEASE</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-tx</artifactId>
    <version>5.2.2.RELEASE</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.spring
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-orm</artifactId>
    <version>5.2.2.RELEASE</version>
</dependency>
```

Spring MVC

Spring Transaction

Spring Object
Relational
Mapping

Setup Dev Environment - pom.xml



```
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>jstl</artifactId>
    <version>1.2</version>
</dependency>
<!-- https://mvnrepository.com/artifact/javax.servlet:jstl:1.2 -->
<dependency>
    <groupId>javax.servlet</groupId>
    <artifactId>javax.servlet-api</artifactId>
    <version>3.0.1</version>
</dependency>
<!-- https://mvnrepository.com/artifact/javax.servlet-api:3.0.1 -->
<dependency>
    <groupId>javax.servlet.jsp</groupId>
    <artifactId>javax.servlet.jsp-api</artifactId>
    <version>2.3.1</version>
</dependency>
```

Step 1: Customize Maven Build



Need to customize Maven Build
Since we are not using web.xml

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-war-plugin</artifactId>
  <version>2.3</version>
  <configuration>
    <failOnMissingWebXml>false</failOnMissingWebXml>
  </configuration>
</plugin>
```

Step 2: Create Spring App Configuration



XML config

Web.xml

Spring-mvc-demo-servlet.xml

Java config

Spring
@Configuration

Spring Dispatcher
Servlet Initialize

Step 2: Create Spring App Configuration



File:DemoAppConfig.java

```
@Configuration  
@EnableWebMvc  
@ComponentScan(basePackages="com.se.security.demo")  
public class DemoAppConfig {  
    // define a bean for ViewResolver  
  
    @Bean  
    public ViewResolver viewResolver() {  
        InternalResourceViewResolver viewResolver =  
            new InternalResourceViewResolver();  
        viewResolver.setPrefix("/WEB-INF/view/");  
        viewResolver.setSuffix(".jsp");  
        return viewResolver; }  
    . . .
```

Provides similar support to
<mvc:annotation-driven /> in
XML

View Resolver
Configs

WEB-INF/view/ show-student-list .jsp

Step 3: Create Spring Dispatcher Servlet Initializer



- » Spring MVC provides support for **web app initialization**

AbstractAnnotationConfigDispatcherServletInitializer

- » TO DO list:
 - Extend this **abstract base class**
 - Override **required methods**
 - Specify **servlet mapping** and **location** of your app config

Step 3: Create Spring Dispatcher Servlet Initializer



File: MySpringMvcDispatcherServletInitializer.java

```
public class MySpringMvcDispatcherServletInitializer extends AbstractAnnotationConfigDispatcherServletInitializer {  
    @Override  
    protected Class<?>[] getRootConfigClasses() {  
        // TODO Auto-generated method stub  
        return null;    }  
    @Override  
    protected Class<?>[] getServletConfigClasses(){  
        return new Class[] { DemoAppConfig.class };    }  
    @Override  
    protected String[] getServletMappings() {  
        return new String[] { "/" };    }  
  
<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/spring-mvc-crud-demo-servlet.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>
```

Step 4: Develop Spring controller



File: DemoController.java

```
@Controller  
public class DemoController {  
    @GetMapping("/")  
    public String showHome() {  
        return "home";  
    }  
}
```



/WEB-INF/view/ home .jsp

Step 4: Develop JSP view page



File: home.jsp

```
<html>
  <head>
    <title>SE Company Home Page</title>
  </head>
  <body>
    <h2> SE Company Home Page</h2>
    <hr>
    Welcome to the SE company home page!
  </body>
</html>
```



Questions