**11.3. Spring MVC Configuration – Overview**

**Spring MVC Configuration**:

Spring MVC configuration is probably the hardest part of getting started with Spring MVC. Here we follow step-by-step process to configure Spring MVC.

**Spring MVC Configuration Process - Part 1**:

Add configurations to file: **WEB-INF/web.xml**

1. Configure Spring MVC DispatcherServlet
2. Set up URL mappings to Spring MVC Dispatcher Servlet

**Spring MVC Configuration Process - Part 2**:

Add configurations to file: **WEB-INF/spring-mvc-demo-servlet.xml**

1. Add support for Spring component scanning
2. Add support for conversion, formatting and validation
3. Configure Spring MVC View Resolver

**Step 1: Configure Spring DispatcherServlet**:

**File: web.xml**:

<web-app>

<servlet>

<!-- Dispatcher Servlet -->

<servlet-name>dispatcher</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet

</servlet-class>

<!-- Servlet reference-->

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/spring-mvc-demo-servlet.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

</web-app>

**Step 2: Set up URL mappings to Spring MVC Dispatcher Servlet**:

Basically, we want to do is tell the system, “Hey, for any URL pattern coming in, I’d like for you to pass it off to the DispatcherServlet.”

So in this case our URL pattern’s going to be slash, meaning all web request coming in

**File: web.xml**:

<web-app>

<servlet>

<servlet-name>dispatcher</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet

</servlet-class>

...

</servlet>

<servlet-mapping>

<servlet-name>dispatcher</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

11.3. Spring MVC Configuration - Overview