# Spring MVC

Introduction

* The Spring Web MVC framework provides Model-View-Controller (MVC) architecture and ready components that can be used to develop flexible and loosely coupled web applications.
* The **Model** encapsulates the application data and in general they will consist of POJO.
* The **View** is responsible for rendering the model data and in general it generates HTML output that the client's browser can interpret.
* The **Controller** is responsible for processing user requests and building an appropriate model and passes it to the view for rendering.
* The Spring Web model-view-controller (MVC) framework is designed around a ***DispatcherServlet*** that handles all the HTTP requests and responses.
* Upon initialization of DispatcherServlet, the framework will try to load the application context from a file named **[servlet-name]-servlet.xml**
* <servlet-mapping> tag indicates what URLs will be handled by which DispatcherServlet. Here all the HTTP requests ending with **.jsp** will be handled by the DispatcherServlet.
* you can customize this file name and location by adding the servlet listener *ContextLoaderListener*
* The *<context:component-scan...>* tag will be use to activate Spring MVC annotation scanning capability which allows to make use of annotations like @Controller and @RequestMapping
* The *InternalResourceViewResolver* will have rules defined to resolve the view names