



Testing Spring

Beginner to Guru

Spring MVC Test Overview



Spring MVC Test

- Spring MVC Test is a testing framework for testing interactions with Spring MVC controllers
- Provides Servlet API Mock objects to mock the web environment
 - **MockHttpServletRequest** - Mock of Java's HttpServletRequest
 - **MockHttpServletResponse** - Mock of Java's HttpServletResponse
- **DispatcherServlet** - Requests are routed through Spring MVC's DispatcherServlet



Spring MVC Test Configuration Modes

- Standalone Setup
 - Very light weight - Ideal for unit tests
 - Tests one controller at a time
 - Allows for testing of controller requests and responses
- WebApplicationContext Setup
 - Loads larger context of Spring Configuration
 - Tests many controllers - per configuration
 - Allows for testing of application config



Static Imports

- Spring MVC Test uses a “fluent” API via several static imports
- **MockMvcRequestBuilders.*** - Builds request
- **MockMvcResultMatchers.*** - Create assertions against response
- **MockMvcBuilders.*** - Configure and build an instance of MockMvc.



Important Differences from Container

- Spring MVC Test does not use a running Servlet container
- No network request is made (ie to port 80, or 8080)
- Key differences from running in container
- HTML is not generated, thus templates are not executed (JSP, Thymeleaf, etc)
- You can test the view (template) requested, or redirected to
 - But cannot test expected HTML to be rendered
- Spring does support testing with a running container when needed.

