Contents

[***1.*** **Features of Spring** 1](#_Toc496651075)

[***2.*** **SpringBoot** 1](#_Toc496651076)

[***3.*** **@EnableAutoConfiguration and @ComponentScan** 1](#_Toc496651077)

[***4.*** **Add property file** 2](#_Toc496651078)

[***5.*** **Error Page** 2](#_Toc496651079)

1. **@RestController**

A convenience annotation that is itself annotated with @Controller and @ResponseBody

1. @Controller is used to mark classes as Spring MVC Controller
2. @RestController is a convenience annotation that does nothing more than adding @Controller and @ResponseBody annotation
3. @RestContrller also converts the response to JSON/XML automatically.
4. **REST**
5. REST stands for Representation State Transfer
6. It’s an architectural style which can be used to design web services, that can be consumed from a variety of clients. The core idea is that rather than using complex mechanism such as SOAP to connect between machines, simple HTTP is used to make calls among them.

In Rest based design, resources are being manipulated using a common set of verbs.

* To Create a resource : HTTP POST should be used
* To Retrieve a resource : HTTP GET should be used
* To Update a resource : HTTP PUT should be used
* To Delete a resource : HTTP DELETE should be used

1. **RestController**

This is what our REST API does:

* GET request to /api/user/ returns a list of users
* GET request to /api/user/1 returns the user with ID 1
* POST request to /api/user/ with a user object as JSON creates a new user
* PUT request to /api/user/3 with a user object as JSON updates the user with ID 3
* DELETE request to /api/user/4 deletes the user with ID 4
* DELETE request to /api/user/ deletes all the users

1. **Add property file**
2. Configure the port and context-path for our app.
3. Property/yml files are commonly located inside src/main/resources.
4. Why to use application.yml file ?

: By default , Spring-Boot will use no context-path, and the default port would be 8080, means our application would be available at localhost:8080. But we can overwrite these properties by declaring them in application.yml or application.properties file

**src/main/resources/application.yml**

|  |
| --- |
| server:    port: 8080    contextPath: /SpringBootStandAloneExample |

**If you had preferred .properties file, application.properties corresponding to above .yml would be**

|  |
| --- |
| server.port: 8080  server.contextPath: /SpringBootStandAloneExample |

1. **Error Page**

By default, Spring Boot installs a ‘whitelabel’ error page that is shown in browser client if you encounter a server error. You can override that page, based upon the templating technology you are using. For freemarker, you can create a page with name ‘error.ftl’ which would be shown in case an error occurred.