Swagger - Implementation Steps

Step-01: New Git branch (Using IDE)

Step-02: Add Springfox Dependencies to pom.xml

Step-03: Create SwaggerConfig file

Step-04: Adding API Info to modify header part of our documentation

Step-05: Restrict scope of swagger document generation using API Basepackages @ paths

Step-06: Auto populate documentation for JSR-303 Validations

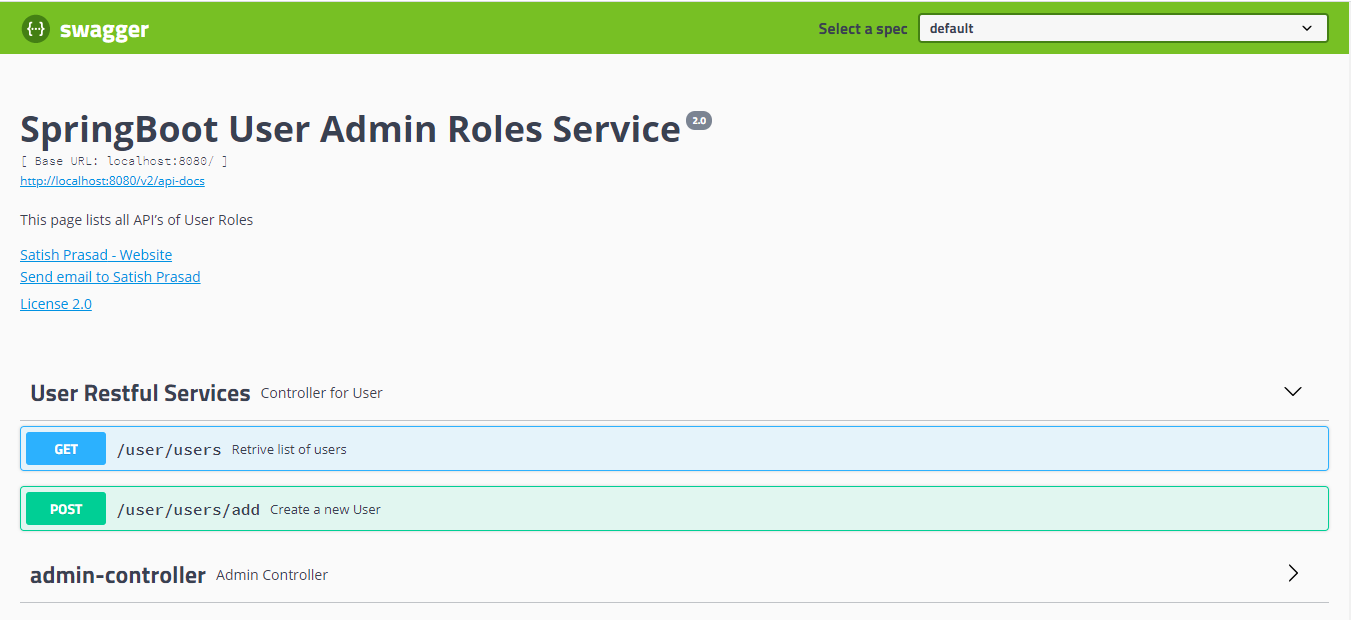
Step-07: Adding Swagger Core Annotations to Model classes

Step-08: Adding Swagger Core Annotations to Controller classes

Spring Boot – Swagger Integration

* Documenting REST API is very important primarily from API consumes point of view.
* API Documentation helps consumers to understand and implement their client applications without any confusion and also by avoiding costly mistakes.
* One of the most popular API Documentation specifications is OpenApi, formerly known as Swagger.
* Swagger allows us to describe API properties either using JSON or YAML metadata.
* Swagger also provides a web UI which transforms the JSON metadata to a nice HTML documentation.
* Swagger UI can also be used as a REST client.
* Swagger integration with Spring Framework can be implemented using SpringFox dependencies.

**Swagger UI:**



REST API Documentation using Swagger2 in Spring Boot

Step-00: Introduction

Step-01: New Git branch (using IDE)

* Git Branch name : 12-SpringBoot –Swagger –APIDocumentation
* Create a new local branch

-----------------------------------------------------------------------------------------------------------------------------------------

Step - 02: Add Springfox Dependencies to pom.xml and Restart Embedded Tomcat

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-swagger2</artifactId>

<version>2.9.2</version>

</dependency>

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-swagger-ui</artifactId>

<version>2.9.2</version>

</dependency>

Step – 03: Create Swagger file

* Annotate it with @Configuration
* Annotate it with @EnableSwagger2
* Create a Docket bean and annotate with @Bean
* Swagger Metadata URL
  + //Swagger Metadata: <http://localhost:8080/v2/api-docs>
  + Verify in readable format in SWAGGER online editor <https://editor.swagger.io/>
  + Fix Optional<User> responses.
* Swagger UI URL

//Swagger UI URL: <http://localhost:8080/swagger-ui.html>

---------------------------------------------------------------------------------------------------------------------------------------

Step – 04: Adding API info to modify header part of our documentation

* Create a new class “ApiInfo”.
* Update the Docket bean with this ApiInfo.

---------------------------------------------------------------------------------------------------------------------------------------

Step – 05: Restrict scope of swagger document generation using API Basepackages & Paths

* Update base package in RequestHandlerSelectors.basePackages
* Update PathSelectors.ant(“/users/\*\*) to limit to Specific paths.

------------------------------------------------------------------------------------------------------------------------------------------

Step – 06 : Auto populate documentation for JSR-303 Validations

* JSR-303 Spec: <https://beanvalidation.org/1.0/spec/>
* Add Dependency in pom.xml and restart Embedded Tomcat

<dependency>

* <groupId>io.springfox</groupId>
* <artifactId>springfox-bean-validators</artifactId>
* <version>2.9.2</version>
* </dependency>

-Add “@Import(BeanValidatorPluginsConfiguration.class)” on top of swagger configuration

-Verify Models in Swagger UI

Step – 07: Adding Swagger Core annotations to Model class

* Core Annotations documentation
* Core Annotations for Model class
  + Class Level:
    - @ApiModel(description = “Mode to create a new user”)
* Field Level: notes, required, position , unordered without position

@ApiModelProperty(notes = " Auto generated unique id ", required = **true**,position = 1)

@ApiModelProperty(notes = "username should be flname", example = "Sprasad",required = **false**, position = 2)

-----------------------------------------------------------------------------------------------------------------------------------------

Step – 08: Adding Swagger Core Annotations to Controller classes

* Controller Level
  + @Api(tags = "User Restful Services", value = "UserController", ~~description~~ = "Controller for User")
* Method Level:
  + @ApiOperation(value = "Retrive list of users")
  + @ApiOperation(value = "Create a new User")
* Parameter Level:
  + @ApiParam("User information for a new user to be created.")
* Produces

--------------------------------------------------------------------------------------------------------------------------------------

Step – 09: Commit & Push code via IDE