

Date : 24/03/2021

Spring Boot 9AM

Mr. RAGHU

Swagger UI Configuration

=> POSTMAN Tool is used for testing our rest controllers.
This is fully manual tool, we should enter all
details : URL, MethodType, Param, JSON Input,
Header Details..etc.

We should listout all URLs in App, if it is small App
then it is fine. If app is big (having 40 controllers
with multiple methods-save,update,delete,get..etc)

In this case checking all URL, types, Input bit complex
and time consuming process.

=> Swagger is a advanced test tool compared to POSTMAN.
It provides easy UI for testing all RestControllers
in project.

=> If we use swagger, need not to enter URL,
choose MethodType manually.
We need to enter only input data.

====Swagger Configuration in Project====

S#1 Define one Spring Boot project with RestControllers

S#2 Add SpringFox(vendor name) Swagger Dependencies
in project:

springfox-swagger2,
springfox-swagger-ui

S#3 Define onetime configurartion for Swagger UI

- a. Create Empty Docket object (swagger UI screen)
- b. Select RestControllers
- c. provide basePackage (RestController common package)
for scanning of classes.
- d. Provide one common path for all restcontroller
(which you want to show at swagger, which you dont
want to show)
- e. Build full Docket(UI)

S#4 Run Application and Enter URL as:

http://localhost:8086/swagger-ui.html

```
<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>

*) onetime config code

```
package in.nareshit.raghu.config;
```

```
import org.springframework.context.annotation.Bean;
```

```
import org.springframework.context.annotation.Configuration;
```

```
import springfox.documentation.builders.PathSelectors;
```

```
import springfox.documentation.builders.RequestHandlerSelectors;
```

```
import springfox.documentation.spi.DocumentationType;
```

```
import springfox.documentation.spring.web.plugins.Docket;
```

```
import springfox.documentation.swagger2.annotations.EnableSwagger2;
```

```
@Configuration
```

```
@EnableSwagger2
```

```
public class SwaggerConfig {
```

```
    @Bean
```

```
    public Docket createDocket() {
```

```
        return new Docket(DocumentationType.SWAGGER_2)
```

```
            .select()
```

```
            .apis(RequestHandlerSelectors.basePackage("<restcontroller package name>"))
```

```
            .paths(PathSelectors.regex("/common path for all rest controllers"))
```

```
            .build();
```

```
    }
```

```
}
```

```
-----  
http://localhost:8086/swagger-ui.html
```

*) At RestController method level: @ApiOperation

This annotation is used to provide alias name

instead of method name at swagger UI

*) RestController class level: @Api(value = "SAMPLE",description = "EMPLOYEE OPERATIONS")

To provide alias name for className displayed at Swagger UI

*) ApiInfo : Additional Information to be displayed at
Swagger UI

===Full code=====

```
package in.nareshit.raghu.config;
```

```
import org.springframework.context.annotation.Bean;
```

```
import org.springframework.context.annotation.Configuration;
```

```
import static springfox.documentation.builders.PathSelectors.regex;
```

```
import static
```

```
springfox.documentation.builders.RequestHandlerSelectors.basePackage;
```

```
import java.util.Collections;
```

```
import springfox.documentation.service.ApiInfo;
```

```
import springfox.documentation.service.Contact;
```

```

import springfox.documentation.spi.DocumentationType;
import springfox.documentation.spring.web.plugins.Docket;
import springfox.documentation.swagger2.annotations.EnableSwagger2;

@Configuration
@EnableSwagger2
public class SwaggerConfig {

    @Bean
    public Docket createDocket() {
        return new Docket(DocumentationType.SWAGGER_2)
            .select()

        .apis(basePackage("in.nareshit.raghu.rest"))
            .paths(regex("/rest.*"))
            .build()
            .apiInfo(apiInfo())
            ;
    }

    private ApiInfo apiInfo() {
        return new ApiInfo(
            "WELCOME TO APP", //title
            "SAMPLE DESCRIPTION", //description
            "3.0", //version
            "http://nareshit.in", //URL
            new Contact("RAGHU",
                "https://www.facebook.com/groups/thejavatemple",
                "javabyraghu@gmail.com"), //developer information
            "Nit Licence", "http://nareshit.in",
            //company information
            Collections.EMPTY_LIST); //clients
        information
    }
}

```