Announcement Microservice development using swagger:

Announcement Title

Announcement desc

Announcement date

pom.xml for swagger:

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

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.announcement</groupId>

<artifactId>springboot-swagger-test</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.4.1.RELEASE</version>

</parent>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>io.springfox</groupId>

<artifactId>springfox-swagger2</artifactId>

<version>2.4.0</version>

</dependency>

<dependency>

<groupId>io.springfox</groupId>

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

<version>2.4.0</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

springapplication.java:

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

package com.announcement.swaggertest;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class Announcement {

public static void main(String[] args) {

SpringApplication.run(Announcement.class, args);

}

}

Controller:

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

package com.announcement.swaggertest;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RestController;

@RestController

public class AnnouncementController {

@RequestMapping(method = RequestMethod.GET, value = "/api/announcement")

public String sayHello() {

return "Swagger Announcement title desc date ";

}

}

Enable swagger:

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

package com.announcement.swaggertest;

import org.springframework.context.annotation.Bean;

import org.springframework.context.annotation.Configuration;

import com.google.common.base.Predicate;

import springfox.documentation.builders.ApiInfoBuilder;

import springfox.documentation.service.ApiInfo;

import springfox.documentation.spi.DocumentationType;

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

import springfox.documentation.swagger2.annotations.EnableSwagger2;

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

import static com.google.common.base.Predicates.or;

@Configuration

@EnableSwagger2

public class SwaggerConfig {

@Bean

public Docket postsApi() {

return new Docket(DocumentationType.SWAGGER\_2).groupName("public-api")

.apiInfo(apiInfo()).select().paths(postPaths()).build();

}

private Predicate<String> postPaths() {

return or(regex("/api/posts.\*"), regex("/api/announcement.\*"));

}

private ApiInfo apiInfo() {

return new ApiInfoBuilder().title("announcement API")

.description("announcement API reference for developers")

.termsOfServiceUrl("http://announcement.com")

.contact("announcement@gmail.com").license("announcement License")

.licenseUrl("announcement@gmail.com").version("1.0").build();

}

}

Now go to http://localhost:8080/swagger-ui.html.

listed all announcement microservice details in swagger

Pagination of announcemnet:

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

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

pom.xml:

----------

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.announcement</groupId>

<artifactId>SpringBootAnnouncment</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>jar</packaging>

<name>SpringBootAnnouncment</name>

<description>Demo project for Spring Boot</description>

<parent>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-parent</artifactId>

<version>1.4.1.RELEASE</version>

<relativePath /> <!-- lookup parent from repository -->

</parent>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>

<java.version>1.8</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>com.h2database</groupId>

<artifactId>h2</artifactId>

</dependency>

<dependency>

<groupId>org.apache.tomcat.embed</groupId>

<artifactId>tomcat-embed-jasper</artifactId>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<optional>true</optional>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>

Springbootapplication.java:

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

package com.announcement;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.EnableAutoConfiguration;

import org.springframework.boot.autoconfigure.SpringBootApplication;

import org.springframework.web.bind.annotation.RestController;

@RestController

@EnableAutoConfiguration

@SpringBootApplication

public class SpringBootAnnouncement {

public static void main(String[] args) {

SpringApplication.run(SpringBootAnnouncement.class, args);

}

}

AnnouncementModel:

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

package com.announcement.model;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Announcement {

@GeneratedValue(strategy = GenerationType.AUTO)

@Id

private long id;

private String name;

private String desc;

private String date;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getDesc() {

return desc;

}

public void setDesc(String desc) {

this.desc = desc;

}

@Override

public String toString() {

return "Announcement [id=" + id + ", name=" + name + ", desc=" + desc + ",date=" + date + "]";

}

}

RestController:

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

package com.announcement.controllers;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.data.domain.Page;

import org.springframework.data.domain.Pageable;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.RestController;

import com.javainuse.data.announcementRepository;

import com.javainuse.model.announcement;

@RestController

public class AnnouncementController {

@Autowired

private announcementRepository announcementRepository;

@RequestMapping(value = "/listPageable", method = RequestMethod.GET)

Page<announcement> announcementPageable(Pageable pageable) {

return announcementRepository.findAll(pageable);

}

}

DB Repository:

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

package com.announcement.data;

import org.springframework.data.jpa.repository.JpaRepository;

import com.announcement.model.announcement;

public interface announcementRepository extends JpaRepository<announcement

, Long> {

}

We now use the H2-console to insert data in H2 database. Go to localhost:8080/h2-console/login.do.

Testing USING junit:

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

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

package com.announcemnet.test;

import org.junit.Test;

import org.junit.runner.RunWith;

import org.springframework.boot.test.context.SpringBootTest;

import org.springframework.test.context.junit4.SpringRunner;

@RunWith(SpringRunner.class)

@SpringBootTest

public class SpringBootannouncemnetTests {

@Test

public void contextLoads() {

}

}

import static org.springframework.test.web.servlet.request.MockMvcRequestBuilders.get;

import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.content;

import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.jsonPath;

import static org.springframework.test.web.servlet.result.MockMvcResultMatchers.status;

import org.junit.Before;

import org.junit.Test;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.test.web.servlet.MockMvc;

import org.springframework.test.web.servlet.setup.MockMvcBuilders;

import org.springframework.web.context.WebApplicationContext;

public class TestWebApp extends SpringBootannouncemnetTests {

@Autowired

private WebApplicationContext webApplicationContext;

private MockMvc mockMvc;

@Before

public void setup() {

mockMvc = MockMvcBuilders.webAppContextSetup(webApplicationContext).build();

}

@Test

public void testAnnouncement() throws Exception {

mockMvc.perform(get("/announcemnet")).andExpect(status().isOk())

.andExpect(content().contentType("application/json;charset=UTF-8"))

.andExpect(jsonPath("$.name").value("name")).andExpect(jsonPath("$.description").value("desc"))

.andExpect(jsonPath("$.date").value("date")));

}