重写SwaggerResourcesProvider

```
@Component
@Primary
public class GatewaySwaggerResourcesProvider implements SwaggerResourcesProvider {

    private final RouteLocator routeLocator;

    public GatewaySwaggerResourcesProvider(RouteLocator routeLocator) {
        this.routeLocator = routeLocator;
    }

    @Override
    public List<SwaggerResource> get() {
        List<SwaggerResource> resources = new ArrayList<>();
        List<Route> routes = routeLocator.getRoutes();

        for (Route route:routes) {
            resources.add(swaggerResource(route.getId(),
route.getFullPath().replace("**", "v2/api-docs")));
        }
        return resources;
    }

    private SwaggerResource swaggerResource(String name, String location) {
        SwaggerResource swaggerResource = new SwaggerResource();
        swaggerResource.setName(name);
        swaggerResource.setLocation(location);
        swaggerResource.setSwaggerVersion("2.0");
        return swaggerResource;
    }
}
```

Swagger2配置类

```
@Configuration
@EnableSwagger2
public class Swagger2Config {
    @Bean
    public Docket buildDocket() {
        return new Docket(DocumentationType.SWAGGER_2)
                .apiInfo(buildApiInf()) // .apiInfo(apiInfo())
                .select()
                .apis(RequestHandlerSelectors.basePackage(""))// 需要生成文档的包的位置
                .paths(PathSelectors.any())
                .build();
    }

    private ApiInfo buildApiInf() {
        return new ApiInfoBuilder()
                .title("接口api")
                .description("Zuul+Swagger2构建 APIs")
                .termsOfServiceUrl("http://www.wearenobug.club")
                .contact(new Contact("nobug", "http://www.wearenobug.club",
"http://www.wearenobug.club"))
                .version("1.0")
                .build();
    }
}
```

Swagger静态文件重定向

```
@Configuration
@EnableWebMvc
public class ApplicationExceptionAdapter extends WebMvcConfigurerAdapter {

    @Override
    public void addResourceHandlers(ResourceHandlerRegistry registry) {
        registry.addResourceHandler("swagger-ui.html")
                .addResourceLocations("classpath:/META-INF/resources/");
        registry.addResourceHandler("/webjars/**")
                .addResourceLocations("classpath:/META-INF/resources/webjars/");
    }

}
```

在微服务中配置Swagger2启动类

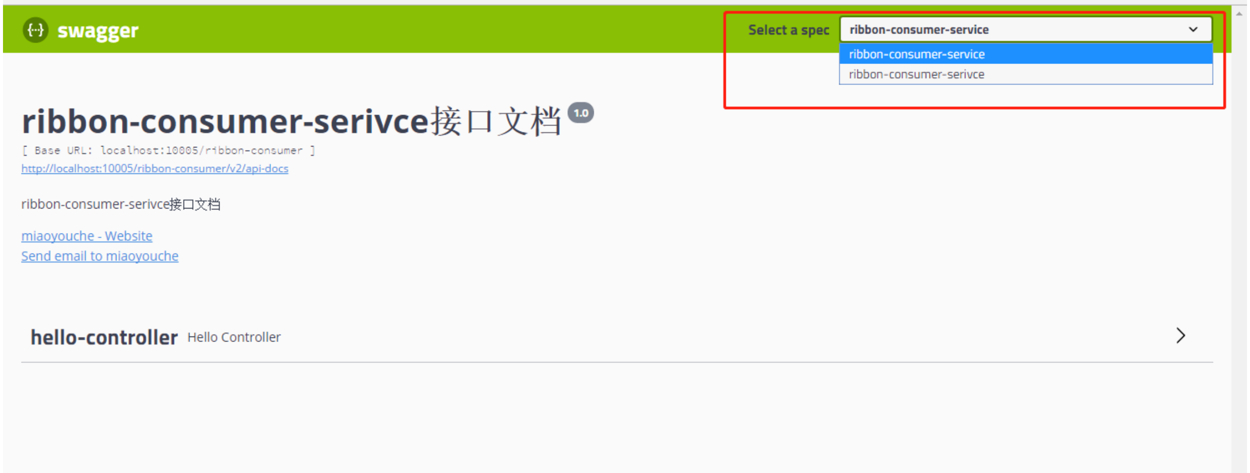
```
@ConditionalOnClass(value = {Swagger.class})
@Configuration
@EnableSwagger2
public class SwaggerConfig {
    @Value("${spring.application.name}")
    private String applicationName;
    @Bean
    public Docket createRestApi() {
        return new Docket(DocumentationType.SWAGGER_2)
                .apiInfo(apiInfo())
                .select()
                .apis(RequestHandlerSelectors.basePackage("com.nobug.ribbon.comsumer"))
                .paths(PathSelectors.any())
                .build()
                .globalOperationParameters(parameters());
    }

    private ApiInfo apiInfo() {
        return new ApiInfoBuilder()
                .title(applicationName+"接口文档")
                .description(applicationName+"接口文档")
                .contact(new Contact("nobug", "http://www.wearenobug.clbu",
"http://www.wearenobug.clbu"))
                .version("1.0")
                .build();
    }
    private List<Parameter> parameters() {
        ParameterBuilder parameterBuilder = new ParameterBuilder();
        List<Parameter> parameters = new ArrayList<>();
        parameterBuilder.name("Authorization")
                .description("Authorization")
                .modelRef(new ModelRef("string"))
                .parameterType("header")
                .required(false).build();
        parameters.add(parameterBuilder.build());
        return parameters;
    }

}
```

效果展示

红框内可以选择不同的微服务



ribbon-consumer-serivce接口文档 1.0

[Base URL: localhost:10005/ribbon-consumer]

http://localhost:10005/ribbon-consumer/v2/api-docs

ribbon-consumer-serivce接口文档

miaoyouche - Website

Send email to miaoyouche

hello-controller Hello Controller