

Swagger

Swagger— это спецификация для описания и документирования REST API. Он определяет формат веб-сервисов REST, включая URL, ресурсы, методы и т. д. Swagger будет генерировать документацию из кода приложения и обрабатывать также часть рендеринга.

Добавление зависимостей

```
<dependency>
  <groupId>io.springfox</groupId>
  <artifactId>springfox-boot-starter</artifactId>
  <version>3.0.0</version>
</dependency>
```

```
@SpringBootApplication
@EnableWebMvc
public class SwaggerApplication {

    public static void main(String[] args) {
        SpringApplication.run(SwaggerApplication.class, args);
    }
}
```

Конфигурационный класс

```
@Configuration
public class SwaggerConfig {

    @Bean
    public Docket api() {
        return new Docket(DocumentationType.SWAGGER_2)
            .apiInfo(getApiInfo())
            .select(new ApiSelectorBuilder())
            .apis(RequestHandlerSelectors.basePackage("com.example.swagger.controller"))
            .paths(PathSelectors.any())
            .build();
    }

    private ApiInfo getApiInfo() {
        Contact contact = new Contact(name: "Example project", url: "http://google.com", email: "example@ya.ru");
        return new ApiInfoBuilder()
            .title("Example Api Title")
            .description("Example Api Definition")
            .version("1.0.0")
            .license("Apache 2.0")
            .licenseUrl("http://www.yandex.ru")
            .contact(contact)
            .build();
    }
}
```

Аннотации

@Api: украсить весь класс и описать роль контроллера
@ApiOperation: описать метод класса или интерфейс
@ApiParam: описание одного параметра
@ApiModel: использовать объекты для получения параметров
@ApiModelProperty: при получении параметров с объектом, опишите поле объекта
@ApiResponse: 1 описание ответа HTTP
@ApiResponses: общее описание ответа HTTP.
@ApiIgnore: используйте эту аннотацию, чтобы игнорировать этот API
@ApiError: информация, возвращаемая при возникновении ошибки
@ApiImplicitParam: параметр запроса
@ApiImplicitParams: несколько параметров запроса

Аннотации

- **@Operation** - Описывает операцию или обычно метод HTTP для определенного пути.
- **@Parameter** - Представляет один параметр в операции OpenAPI.
- **@RequestBody** - Представляет тело запроса в операции
- **@ApiResponse** - Представляет ответ в операции
- **@Tag** - Представляет теги для операции или определения OpenAPI.
- **@Server** - Представляет серверы для операции или для определения OpenAPI.
- **@Callback** - Описывает набор запросов
- **@Link** - Представляет возможную ссылку времени разработки для ответа.
- **@Schema** - Позволяет определять входные и выходные данные.
- **@ArraySchema** - Позволяет определять входные и выходные данные для типов массивов.
- **@Content** - Предоставляет схему и примеры для определенного типа мультимедиа.
- **@Hidden** - Скрывает ресурс, операцию или свойство

Контроллер и модель

```
@RestController
@Api(description = "REST API for person")
public class MyController {
    @Autowired
    PersonDao personDao;

    @GetMapping("/getAll")
    @ApiOperation("Get list of person")
    @ApiResponses(value = {
        @ApiResponse(code = 200, message = "OK"),
        @ApiResponse(code = 404, message = "The resource not found")}])
    public List<Person> getAll() {
        return personDao.getAllPerson();
    }
}
```

```
@ApiModelProperty("Person object")
public class Person {

    @ApiModelProperty(notes = "Name of person", required = true)
    private String name;

    @ApiModelProperty(notes = "Age of person", required = true)
    private int age;
}
```

http://127.0.0.1:8080/swagger-ui/index.html#/



Select a definition

default

Example Api Title 1.0.0

[Base URL: 127.0.0.1:8080/]
<http://127.0.0.1:8080/v2/api-docs>

Example Api Definition

Example project - Website
Send email to Example project
Apache 2.0

my-controller REST API for person

DELETE /delete/{id} Удаление person

GET /getAll Get list of person

POST /post Создание нового person

Models



Описание метода getAll()

GET /getAll Get list of person [Try it out](#)

Parameters

No parameters

Responses Response content type: */*

Code	Description
200	OK Example Value Model <pre>[{ "age": 0, "name": "string" }]</pre>
401	Unauthorized
403	Forbidden

Результат выполнения getAll()

Curl

```
curl -X GET "http://127.0.0.1:8080/getAll" -H "accept: */*"
```

Request URL

```
http://127.0.0.1:8080/getAll
```

Server response

Code

Details

200

Response body

```
[
  {
    "name": "Ivan",
    "age": 25
  },
  {
    "name": "Ily",
    "age": 26
  },
  {
    "name": "Vasy",
    "age": 22
  },
  {
    "name": "Pety",
    "age": 19
  },
  {
    "name": "misha",
    "age": 24
  }
]
```

Download

Описание модели

Example Api Title 1.0.0

[Base URL: 127.0.0.1:8080/]

<http://127.0.0.1:8080/v2/api-docs>

Example Api Definition

[Example project - Website](#)

[Send email to Example project](#)

Apache 2.0

my-controller REST API for person



Models



```
Person object ▾ {  
  age*           integer($int32)  
                 Age of person  
  
  name*          string  
                 Name of person  
  
}
```

Спасибо