

Swagger

Dependencies

```
<dependency>
  <groupId>io.springfox</groupId>
  <artifactId>springfox-swagger2</artifactId>
  <version>2.9.2</version>
  <scope>compile</scope>
</dependency>

<dependency>
  <groupId>io.springfox</groupId>
  <artifactId>springfox-swagger-ui</artifactId>
  <version>2.9.2</version>
  <scope>compile</scope>
</dependency>
```

Pierwsze dependency pozwala na skorzystanie z podstawowej wersji, tzn. pojawia się adres **v2/api-docs**, gdzie są zebrane wszystkie dostępne endpointy.

Drugie dependency to przedstawienie end pointów w postaci bardziej przyjaznej pod adresem **/swagger-ui.html**.

Konfiguracja

```
@Configuration
@EnableSwagger2
public class SwaggerConfig {
    @Bean
    public Docket docs(){
        return new Docket(DocumentationType.SWAGGER_2)
            .select()
            .apis(RequestHandlerSelectors.basePackage("pl.idzikdev.Xcom.rest"))
            .paths(PathSelectors.ant("/v1/**"))
            .build()
            .apiInfo(getApiInfo());
    }
}
```

w apiInfo można umieścić wiele informacji o naszym api. Najlepiej jest użyć tzw buildera.

```
private ApiInfo getApiInfo() {
    return new ApiInfoBuilder()
```

Adnotacje

Można opisywać klasy (modele), metody i pola.

Przykłady

```
@ApiModel(value = "Alien")
public class AlienEntity {
```

```
@ApiModelProperty(value = "time units which alien can spend")
String rank;
```

```
@ApiOperation(value = "find alien by id", notes = "the method find alien by id")
@RequestMapping(path =("/{id}", method = RequestMethod.GET, produces =
MediaType.APPLICATION_JSON_UTF8_VALUE)
public ResponseEntity<AlienEntity> getAlienById(
    @ApiParam(value = "unique id of alien") @PathVariable String id
)
```