



Spring Web

Aula 7 – Open Api (Swagger)

Sumário

- Open Api (Swagger)

Open API (Swagger)

- O Open Api se trata da automatização da geração de especificações das APIs REST em JSON;
- É composto de um arquivo de configuração, que pode ser definido em YAML ou JSON;
- O ponto mais importante é o arquivo de configuração. É nele que a API é documentada.
- Existem algumas bibliotecas do Java que facilitam este processo. No caso de uma aplicação **Spring**, a melhor opção é a biblioteca **Open API**.

Dependências

- <https://springdoc.org/>

```
<dependency>
  <groupId>org.springdoc</groupId>
  <artifactId>springdoc-openapi-ui</artifactId>
  <version>1.6.8</version>
</dependency>
```

Configuração

```
@Configuration
public class OpenApiConfig {

    @Bean
    public OpenAPI springShopOpenAPI() {
        String securitySchemeName = "bearerAuth";
        return new OpenAPI()
            .info(new Info().title("Pessoa API")
                .description("Pessoa API documentação")
                .version("v1.0.0")
                .license(new License().name("Apache 2.0").url("http://springdoc.org")))
            .addSecurityItem(new SecurityRequirement().addList(securitySchemeName))
            ;
    }
}
```

Configuração

```
@Hidden
@RestController
public class OpenApiController {

    // @Operation(hidden = true)
    @GetMapping(value = "/")
    public void index(HttpServletResponse response) throws IOException {
        response.sendRedirect("/swagger-ui/index.html");
    }
}
```



Vamos praticar!



Configuração

- Anotações de Métodos na Controller: (podem ser criadas interfaces para receberem estas docs)

```
@Operation(summary = "listar pessoas", description = "Lista todas as pessoas do banco")
@ApiResponses(
    value = {
        @ApiResponse(responseCode = "200", description = "Retorna a lista de pessoas"),
        @ApiResponse(responseCode = "403", description = "Você não tem permissão para acessar este recurso"),
        @ApiResponse(responseCode = "500", description = "Foi gerada uma exceção")
    }
)
@GetMapping
public ResponseEntity<List<Pessoa>> list() {
    List<Pessoa> lista = pessoaService.list();
    return ResponseEntity.ok(lista);
}
```


Configuração

- Anotações de Atributos (DTOs):

```
@NotEmpty
@NotBlank
@Schema(description = "Nome da Pessoa")
private String nome;
```

Exercício #1

- Configurar no seu projeto **pessoa-api** e documentar a API;
- **/pessoa**
 - Colocar anotações de documentação em todos os campos;
 - Colocar validações (caso não houver).
- Testar a API via página.

Task

- Documentar as APIs:
 - **/contato**
 - **/endereco**
- Colocar anotações de documentação em todos os campos;
- Colocar validações caso não houver;
- Testar a API via página.

Task #2 Grupo (opcional)

- Configurar o Open API no seu projeto e implementar toda a documentação dos endpoints que forem expostos (controllers e DTOs).

Obrigado!

DBC

DIGITAL BUSINESS COMPANY®



 /dbc.company

 /dbccompany

 /dbccompany.com.br

 /company/dbc-company