18.24. Resolução do desafio

1 - Alterando SpringFoxConfig

Primeiro, vamos alterar o método apiDocket, adicionando novas Tags

```
.tags(new Tag("Cidades", "Gerencia as cidades"),
        new Tag("Grupos", "Gerencia os grupos de usuários"),
        new Tag("Cozinhas", "Gerencia as cozinhas"),
        new Tag("Formas de pagamento", "Gerencia as formas de pagamento"));
```

2 - Alterando FormaPagamentoInput

Vamos adicionar a anotação @ApiModelProperty acima da propriedade "descricao"

```
@ApiModelProperty(example = "Cartão de crédito", required = true)
@NotBlank
private String descricao;
```

3 - Alterando FormaPagamentoModel

Agora, vamos adicionar a mesma anotação nas propriedades de nosso modelo

```
@ApiModelProperty(example = "1")
private Long id;

@ApiModelProperty(example = "Cartão de crédito")
private String descricao;
```

4 - Criando FormaPagamentoControllerOpenApi

```
@Api(tags = "Formas de pagamento")
public interface FormaPagamentoControllerOpenApi {

    @ApiOperation("Lista as formas de pagamento")
        public  ResponseEntity<List<FormaPagamentoModel>>  listar(ServletWebRequest
request);

    @ApiOperation("Busca uma forma de pagamento por ID")
    @ApiResponses({
          @ApiResponse(code = 400, message = "ID da forma de pagamento inválido",
response = Problem.class),
          @ApiResponse(code = 404, message = "Forma de pagamento não encontrada",
response = Problem.class)
    })
    public ResponseEntity<FormaPagamentoModel> buscar(
            @ApiParam(value = "ID de uma forma de pagamento", example = "1")
            Long formaPagamentoId,

            ServletWebRequest request);

    @ApiOperation("Cadastra uma forma de pagamento")
    @ApiResponses({
        @ApiResponse(code = 201, message = "Forma de pagamento cadastrada"),
    })
    public FormaPagamentoModel adicionar(
             @ApiParam(name = "corpo", value = "Representação de uma nova forma de
pagamento")
```

```
            FormaPagamentoInput formaPagamentoInput);

    @ApiOperation("Atualiza uma cidade por ID")
    @ApiResponses({
        @ApiResponse(code = 200, message = "Forma de pagamento atualizada"),
          @ApiResponse(code = 404, message = "Forma de pagamento não encontrada",
response = Problem.class)
    })
    public FormaPagamentoModel atualizar(
            @ApiParam(value = "ID de uma forma de pagamento", example = "1")
            Long formaPagamentoId,

                @ApiParam(name = "corpo", value = "Representação de uma forma de
pagamento com os novos dados")
            FormaPagamentoInput formaPagamentoInput);

    @ApiOperation("Exclui uma forma de pagamento por ID")
    @ApiResponses({
        @ApiResponse(code = 204, message = "Forma de pagamento excluída"),
          @ApiResponse(code = 404, message = "Forma de pagamento não encontrada",
response = Problem.class)
    })
    public void remover(Long formaPagamentoId);
}
```

5 - Alterando FormaPagamentoController

Vamos alterar FormaPagamentoController para que o mesmo implemente a interface FormaPagamentoControllerOpenApi

```
@RestController
@RequestMapping(path = "/formas-pagamento", produces =
MediaType.APPLICATION_JSON_VALUE)
public class FormaPagamentoController implements FormaPagamentoControllerOpenApi {
    //...
}
```

Atualizando para o SpringFox 3.0 e Open API 3

Este documento irá te auxiliar a fazer esta aula com a versão 3.0.0 do Spring Fox e suas dependências.

Observação: Este documento é apenas para os que querem usar as versões mais recentes das dependências contidas nesta aula.

Evitando um NullPointerException

Ao adicionar o `produces = MediaType.APPLICATION_JSON_VALUE` na Annotation `@RequestMapping` nos Controllers de Grupo ou Cidade, o SpringFox 3 nos apresentará um NullPointerException. Isso é um problema conhecido que ainda não foi corrigido nessa biblioteca. Acontece devido a alguns métodos das Controllers terem o retorno `void`, como os de `DELETE`, que somado ao `produces = MediaType.APPLICATION_JSON_VALUE` gera um NullPointerException.

Sendo assim, é necessário adicionar o `produces` em cada um dos métodos, com exceção daqueles que retornam `void`, ao invés de adicioná-los na Controller.

Deixando as implementações da seguinte forma:

```
package com.algaworks.algafood.api.controller;

import com.algaworks.algafood.api.assembler.GrupoInputDisassembler;
import com.algaworks.algafood.api.assembler.GrupoModelAssembler;
import com.algaworks.algafood.api.openapi.controller.GrupoControllerOpenApi;
import com.algaworks.algafood.api.model.GrupoModel;
import com.algaworks.algafood.api.model.input.GrupoInput;
import com.algaworks.algafood.domain.model.Grupo;
import com.algaworks.algafood.domain.repository.GrupoRepository;
import com.algaworks.algafood.domain.service.CadastroGrupoService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.MediaType;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseStatus;
import org.springframework.web.bind.annotation.RestController;

import javax.validation.Valid;
import java.util.List;

@RestController
@RequestMapping(path = "/grupos")
public class GrupoController implements GrupoControllerOpenApi {

	@Autowired
	private GrupoRepository grupoRepository;

	@Autowired
	private CadastroGrupoService cadastroGrupo;

	@Autowired
	private GrupoModelAssembler grupoModelAssembler;

	@Autowired
	private GrupoInputDisassembler grupoInputDisassembler;

	@GetMapping(produces = MediaType.APPLICATION_JSON_VALUE)
	public List<GrupoModel> listar() {
		List<Grupo> todosGrupos = grupoRepository.findAll();

		return grupoModelAssembler.toCollectionModel(todosGrupos);
	}

	@GetMapping(path = "/{grupoId}", produces =
MediaType.APPLICATION_JSON_VALUE)
	public GrupoModel buscar(@PathVariable Long grupoId) {
		Grupo grupo = cadastroGrupo.buscarOuFalhar(grupoId);

		return grupoModelAssembler.toModel(grupo);
	}

	@PostMapping(produces = MediaType.APPLICATION_JSON_VALUE)
```

```
	@ResponseStatus(HttpStatus.CREATED)
	public GrupoModel adicionar(@RequestBody @Valid GrupoInput grupoInput) {
		Grupo grupo = grupoInputDisassembler.toDomainObject(grupoInput);

		grupo = cadastroGrupo.salvar(grupo);

		return grupoModelAssembler.toModel(grupo);
	}

	@PutMapping(path = "/{grupoId}", produces = MediaType.APPLICATION_JSON_VALUE)
	public GrupoModel atualizar(@PathVariable Long grupoId,
			@RequestBody @Valid GrupoInput grupoInput) {
		Grupo grupoAtual = cadastroGrupo.buscarOuFalhar(grupoId);

		grupoInputDisassembler.copyToDomainObject(grupoInput, grupoAtual);

		grupoAtual = cadastroGrupo.salvar(grupoAtual);

		return grupoModelAssembler.toModel(grupoAtual);
	}

	@DeleteMapping("/{grupoId}")
	@ResponseStatus(HttpStatus.NO_CONTENT)
	public void remover(@PathVariable Long grupoId) {
		cadastroGrupo.excluir(grupoId);
	}

}
```

Código-fonte atualizado

Código-fonte da aula com as dependências atualizadas