

# 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:

GrupoController

```
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.controller.openapi.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(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);
    }
}

```

CidadeController

```

package com.algaworks.algafood.api.controller;

import com.algaworks.algafood.api.assembler.CidadeInputDisassembler;
import com.algaworks.algafood.api.assembler.CidadeModelAssembler;
import com.algaworks.algafood.api.controller.openapi.CidadeControllerOpenApi;
import com.algaworks.algafood.api.model.CidadeModel;
import com.algaworks.algafood.api.model.input.CidadeInput;
import com.algaworks.algafood.domain.exception.EstadoNaoEncontradoException;
import com.algaworks.algafood.domain.exception.NegocioException;
import com.algaworks.algafood.domain.model.Cidade;
import com.algaworks.algafood.domain.repository.CidadeRepository;
import com.algaworks.algafood.domain.service.CadastroCidadeService;
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(value = "/cidades")
public class CidadeController implements CidadeControllerOpenApi {

    @Autowired
    private CidadeRepository cidadeRepository;

    @Autowired
    private CadastroCidadeService cadastroCidade;

    @Autowired
    private CidadeModelAssembler cidadeModelAssembler;

    @Autowired
    private CidadeInputDisassembler cidadeInputDisassembler;

    @GetMapping(produces = MediaType.APPLICATION_JSON_VALUE)
    public List<CidadeModel> listar() {
        List<Cidade> todasCidades = cidadeRepository.findAll();

        return cidadeModelAssembler.toCollectionModel(todasCidades);
    }

    @GetMapping(path =("/{cidadeId}", produces = MediaType.APPLICATION_JSON_VALUE)
    public CidadeModel buscar(@PathVariable Long cidadeId) {
        Cidade cidade = cadastroCidade.buscarOuFalhar(cidadeId);

        return cidadeModelAssembler.toModel(cidade);
    }

    @PostMapping(produces = MediaType.APPLICATION_JSON_VALUE)
    @ResponseStatus(HttpStatus.CREATED)
    public CidadeModel adicionar(@RequestBody @Valid CidadeInput cidadeInput) {
        try {
            Cidade cidade = cidadeInputDisassembler.toDomainObject(cidadeInput);

            cidade = cadastroCidade.salvar(cidade);

            return cidadeModelAssembler.toModel(cidade);
        } catch (EstadoNaoEncontradoException e) {
            throw new NegocioException(e.getMessage(), e);
        }
    }

    @PutMapping(path =("/{cidadeId}", produces = MediaType.APPLICATION_JSON_VALUE)
    public CidadeModel atualizar(@PathVariable Long cidadeId,
                                @RequestBody @Valid CidadeInput cidadeInput) {
        try {
            Cidade cidadeAtual = cadastroCidade.buscarOuFalhar(cidadeId);

            cidadeInputDisassembler.copyToDomainObject(cidadeInput, cidadeAtual);

            cidadeAtual = cadastroCidade.salvar(cidadeAtual);

            return cidadeModelAssembler.toModel(cidadeAtual);
        } catch (EstadoNaoEncontradoException e) {

```

```
        throw new NegocioException(e.getMessage(), e);
    }
}

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

}
```

## Código-fonte atualizado

[Código-fonte da aula com as dependências atualizadas](#)