Curso: Spring Boot com Ionic - Estudo de Caso Completo

https://www.udemy.com/user/nelio-alves

Prof. Dr. Nelio Alves

Capítulo: Ajustes finais no backend e bucket

Objetivo geral:

- Incluir detalhes finais no backend e bucket para possibilitar a construção e teste da aplicação

Expondo o header Authorization (problema de Cors)

Cors (Cross-origin resource sharing): quais recursos (ex: quais métodos HTTP? quais headers?) estarão disponíveis para requisições advindas de origens diferentes?

Sugestão: use configurações padrão. À medida em que novas necessidades forem surgindo, inclua a solução.

Referências:
https://en.wikipedia.org/wiki/Cross-origin_resource_sharing
https://upload.wikimedia.org/wikipedia/commons/c/ca/Flowchart_showing_Simple_and_Preflight_XHR.svg
https://stackoverflow.com/questions/1256593/why-am-i-getting-an-options-request-instead-of-a-get-request
https://stackoverflow.com/questions/47687774/how-to-access-headers-from-a-httpclient-response-angular-ionic
https://www.html5rocks.com/en/tutorials/cors/

Checklist:

Acrescentar no objeto de resposta: response.addHeader("access-control-expose-headers", "Authorization");

- Em JWTAuthenticationFilter, em successfulAuthentication
- Em AuthResource, no endpoint de refresh_token

Configuração de Cors no bucket

```
<?xml version="1.0" encoding="UTF-8"?>
<CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
<CORSRule>
    <AllowedOrigin>http://*</AllowedOrigin>
    <AllowedOrigin>https://*</AllowedOrigin>
    <AllowedMethod>GET</AllowedMethod>
    <MaxAgeSeconds>3000</MaxAgeSeconds>
    <AllowedHeader>Authorization</AllowedHeader>
</CORSRule>
</CORSConfiguration>
```

Imagens para categorias, produtos e clientes

(link anexo à aula)

Endpoint para buscar cliente por email

Checklist:

- Atualizar ClienteService
- Atualizar ClienteResource

Endpoints para buscar estados e cidades

Checklist:

- Criar EstadoDTO
- Em EstadoRepository, acrescentar método findAllByOrderByNome
- Criar EstadoService
- Criar endpoint para obter estados (classe EstadoResource)
- Criar CidadeDTO
- Em CidadeRepository, acrescentar método findCidades
- Criar CidadeService
- Criar endpoint para obter cidades (sugestão: criar **`/estados/{estado_id}/cidades`**)
- Em SecurityConfig, liberar acesso público aos endpoints

Padronizando formato das exceções

Campos:

```
"timestamp": 1467943353634,
"status": 415,
"error": "Unsupported Media Type",
"message": "Content type 'application/xml' not supported",
"path": "/some-resource"
```

Checklist:

- **ATENÇÃO**: trocar código de erro de validação para **422** (**IMPORTANTE**)
- Atualizar StandardError
- Atualizar ValidationError
- Atualizar ResourceExceptionHandler

Acrescentando mais produtos para testar infinity scroll

1) Declarar os novos produtos e associá-los com a categoria cat1:

```
Produto p12 = new Produto(null, "Produto 12", 10.00);
Produto p13 = new Produto(null, "Produto 13", 10.00);
Produto p14 = new Produto(null, "Produto 14", 10.00);
Produto p15 = new Produto(null, "Produto 15", 10.00);
Produto p16 = new Produto(null, "Produto 16", 10.00);
Produto p17 = new Produto(null, "Produto 17", 10.00);
```

```
Produto p18 = new Produto(null, "Produto 18", 10.00);
Produto p19 = new Produto(null, "Produto 19", 10.00);
Produto p20 = new Produto(null, "Produto 20", 10.00);
Produto p21 = new Produto(null, "Produto 21", 10.00);
Produto p22 = new Produto(null, "Produto 22", 10.00);
Produto p23 = new Produto(null, "Produto 23", 10.00);
Produto p24 = new Produto(null, "Produto 24", 10.00);
Produto p25 = new Produto(null, "Produto 25", 10.00);
Produto p26 = new Produto(null, "Produto 26", 10.00);
Produto p27 = new Produto(null, "Produto 27", 10.00);
Produto p28 = new Produto(null, "Produto 28", 10.00);
Produto p29 = new Produto(null, "Produto 29", 10.00);
Produto p30 = new Produto(null, "Produto 30", 10.00);
Produto p31 = new Produto(null, "Produto 31", 10.00);
Produto p32 = new Produto(null, "Produto 32", 10.00);
Produto p33 = new Produto(null, "Produto 33", 10.00);
Produto p34 = new Produto(null, "Produto 34", 10.00);
Produto p35 = new Produto(null, "Produto 35", 10.00);
Produto p36 = new Produto(null, "Produto 36", 10.00);
Produto p37 = new Produto(null, "Produto 37", 10.00);
Produto p38 = new Produto(null, "Produto 38", 10.00);
Produto p39 = new Produto(null, "Produto 39", 10.00);
Produto p40 = new Produto(null, "Produto 40", 10.00);
Produto p41 = new Produto(null, "Produto 41", 10.00);
Produto p42 = new Produto(null, "Produto 42", 10.00);
Produto p43 = new Produto(null, "Produto 43", 10.00);
Produto p44 = new Produto(null, "Produto 44", 10.00);
Produto p45 = new Produto(null, "Produto 45", 10.00);
Produto p46 = new Produto(null, "Produto 46", 10.00);
Produto p47 = new Produto(null, "Produto 47", 10.00);
Produto p48 = new Produto(null, "Produto 48", 10.00);
Produto p49 = new Produto(null, "Produto 49", 10.00);
Produto p50 = new Produto(null, "Produto 50", 10.00);

cat1.getProdutos().addAll(Arrays.asList(p12, p13, p14, p15, p16, p17, p18, p19, p20,
p21, p22, p23, p24, p25, p26, p27, p28, p29, p30, p31, p32, p34, p35, p36, p37, p38,
p39, p40, p41, p42, p43, p44, p45, p46, p47, p48, p49, p50));

p12.getCategorias().add(cat1);
p13.getCategorias().add(cat1);
p14.getCategorias().add(cat1);
p15.getCategorias().add(cat1);
p16.getCategorias().add(cat1);
p17.getCategorias().add(cat1);
p18.getCategorias().add(cat1);
p19.getCategorias().add(cat1);
p20.getCategorias().add(cat1);
p21.getCategorias().add(cat1);
p22.getCategorias().add(cat1);
p23.getCategorias().add(cat1);
p24.getCategorias().add(cat1);
```

```
p25.getCategorias().add(cat1);
p26.getCategorias().add(cat1);
p27.getCategorias().add(cat1);
p28.getCategorias().add(cat1);
p29.getCategorias().add(cat1);
p30.getCategorias().add(cat1);
p31.getCategorias().add(cat1);
p32.getCategorias().add(cat1);
p33.getCategorias().add(cat1);
p34.getCategorias().add(cat1);
p35.getCategorias().add(cat1);
p36.getCategorias().add(cat1);
p37.getCategorias().add(cat1);
p38.getCategorias().add(cat1);
p39.getCategorias().add(cat1);
p40.getCategorias().add(cat1);
p41.getCategorias().add(cat1);
p42.getCategorias().add(cat1);
p43.getCategorias().add(cat1);
p44.getCategorias().add(cat1);
p45.getCategorias().add(cat1);
p46.getCategorias().add(cat1);
p47.getCategorias().add(cat1);
p48.getCategorias().add(cat1);
p49.getCategorias().add(cat1);
p50.getCategorias().add(cat1);
```

2) Inserir os produtos:

```
produtoRepository.save(Arrays.asList(p12, p13, p14, p15, p16, p17, p18, p19, p20,
p21, p22, p23, p24, p25, p26, p27, p28, p29, p30, p31, p32, p34, p35, p36, p37, p38,
p39, p40, p41, p42, p43, p44, p45, p46, p47, p48, p49, p50));
```

Liberando CORS para PUT e DELETE

Referências:

https://spring.io/blog/2015/06/08/cors-support-in-spring-framework
https://docs.spring.io/spring-security/site/docs/current/reference/html/cors.html
https://spring.io/understanding/CORS

Expondo o header location nas respostas

Referências:

https://stackoverflow.com/questions/19825946/how-to-add-a-filter-class-in-spring-boot
https://gist.github.com/rgiaviti/80d50041541475d5ad7a752b53aa4eed

Aumentando o tamanho máximo permitido para upload

ATUALIZAÇÃO

Se você criou o projeto usando Spring Boot versão 2.x.x:
https://github.com/acenelio/springboot2-ionic-backend

O nome das chaves mudou:

```
spring.servlet.multipart.max-file-size=10MB
spring.servlet.multipart.max-request-size=10MB
```

Referências:

https://stackoverflow.com/questions/37540028/how-to-set-the-max-size-of-upload-file

Checklist:

- Em application.properties, fazer:

```
spring.http.multipart.max-file-size=10MB
spring.http.multipart.max-request-size=10MB
```