

Matheus Campelo Cavalcante

(61) 98333-4445

matheuscampelocavalcante@hotmail.com

GitHub: github.com/1133939

Relatório - Garrafa Copo

19/03/2019

Sobre o meu projeto

O projeto CRUD está funcionando toda a parte Frontend, porém ocorrem erros e problemas no backend quando o sistema tenta fazer alguma função de modificação no banco como Inserir, Deletar e Atualizar.

Em buscas este problema não ocorre e as buscas estão funcionando.

O Backend também está pronto, porém falta alguma anotação para que os problemas acabem no insert e no update, já no delete é um pouco mais estranho.

Buscar:

CRUD Garrafa - Copo				
Buscar Garrafa:		<input type="text"/>	<input type="button" value="Buscar"/>	
#	Garrafa	Quantidade de Copos		
1	Garrafa de Vinho	2	<input type="button" value="Deletar"/>	<input type="button" value="Editar"/>
2	Garrafa de Criança	1	<input type="button" value="Deletar"/>	<input type="button" value="Editar"/>

Assim que a página carrega são buscadas todas as garrafas através do método `findAll()` e jogadas em um array listando o nome e a quantidade de filhas, através do método `length`, de cada entidade pai.

Buscar por nome:

CRUD Garrafa - Copo

Buscar Garrafa:

#	Garrafa	Quantidade de Copos		
1	Garrafa de Vinho	2	<input type="button" value="Deletar"/>	<input type="button" value="Editar"/>

Cadastrar Garrafa

Quando se faz a busca pelo nome é utilizado a cláusula LIKE no retorno da lista de objetos, envolvida de “%” assim: %param%.

Editar:

Buscar Garrafa:

#	Garrafa	Quantidade de Copos		
1	Garrafa de Vinho	2	<input type="button" value="Deletar"/>	<input checked="" type="button" value="Editar"/>
2	Garrafa de Criança	1	<input type="button" value="Deletar"/>	<input type="button" value="Editar"/>

Cadastrar Garrafa

Nome da garrafa:
Nome do copo:

Editar Garrafa

Escolha um novo nome para a garrafa:

Taça de Vinho

Taça de Cristal

Ao clicar no botão editar ele carrega o nome da garrafa no campo “input text” através da utilização da tag “placeholder”, lista todos seus filhos, dá a opção de apagar um por um e atualizar o nome da garrafa. Ao clicar em atualizar o Java recebe o objeto, utiliza o método merge() e muda o nome, porém somem os copos.

Cadastrar:

Cadastrar Garrafa

Nome da garrafa:

Nome do copo:

Editar Garrafa

Escolha um novo nome para a garrafa:

Após preencher e clicar em Add, o copo é posto em uma lista do objeto garrafa e listada em baixo limitando o máximo de 5 copos, o 6º copo não entra na lista nem é listado, e também não é aceito copo vazio na lista.

Cadastrar Garrafa

Nome da garrafa:

Nome do copo:

Copo 1: Copo 99
Copo 2: Copo 77
Copo 3: Copo 55
Copo 4: Copo 33
Copo 5: Copo 22

Editar Garrafa

Escolha um novo nome para a garrafa:

PROBLEMA: Após apertar em salvar é mandado o objeto via POST para o REST que consegue persistir, porém ele não persiste a chave estrangeira da entidade Copo se relacionando com Garrafa, fazendo com que a entidade pai fique sem filho.

Deletar

CRUD Garrafa - Copo

Buscar Garrafa:

#	Garrafa	Quantidade de Copos		
1	Garrafa de Vinho	2	<input type="button" value="Deletar"/>	<input type="button" value="Editar"/>
2	Garrafa de Criança	1	<input type="button" value="Deletar"/>	<input type="button" value="Editar"/>

Cadastrar Garrafa

Nome da garrafa:

Nome do copo:

Editar Garrafa

Escolha um novo nome para a garrafa:

Taça de Vinho

Taça de Cristal

O “Deletar” e o “Excluir Copo” são super parecidos, fazem o transporte do id que será excluído até a camada Service que manda via \$http.delete passando o id pela url, porém quando o Java recebe ele lança uma exceção.

Exceção no Console do navegador: Access to XMLHttpRequest at 'http://localhost:8080/projeto/api/copo/2' from origin 'http://localhost:9001' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource.

Exceção no Java: Unknown entity: java.lang.Long

Como eu trabalhei

Eu tinha basicamente duas opções REST para trabalhar, acabei escolhendo a opção do JavaEE ao invés do SpringBoot do projeto 5, pois foi a primeira que comecei a trabalhar e achei interessante de aprender, porém depois vi que desenvolver com SpringBoot é bem mais fácil, mas continuei na que eu tinha escolhido aprender.

Fiz métodos de busca na Dao utilizando hibernate e consultas JPQL para que eles fossem tratados posteriormente na Service com a lógica das regras de negócio. Na camada REST o envio e o recebimento dessas informações.

Fiz praticamente todo o backend, não digo todo, pois não testei tudo, então pode ou não faltar detalhes. No AngularJS, eu modifiquei a lista que já existia para que aparecesse a quantidade ao invés do nome de cada filho. Listei os filhos(copos), para facilitar a visualização durante o Cadastro de pais(garrafas), criei o cadastro a busca e o delete.

Sobre mim

Aprendi muito neste treinamento, pois eu pude olhar um código, aprender com ele, entender como uma empresa trabalha, testar, fazer funcionar... No fim, acabei me informando mais sobre as tecnologias usadas no momento, aprendi bastante sobre elas e vi um grande avanço na questão do meu aprendizado, foram 5 dias que me deram um aprendizado gigantesco.

Cronograma do projeto

16/03 - Período da tarde, começo dos estudos no código e criação da lógica na Service e das consultas na Dao.

16/03 - 17/03 - Pequenos intervalos de estudo no código e testando o funcionamento com outros bancos.

17/03 - Madrugada, término quase total do backend, integrando com o Frontend AngularJS e criação do banco de dados.

18/03 - Tarde e Noite, avanços no frontend criando a "capa" do CRUD e testando a criação de métodos para aprender mais sobre como o AngularJS funciona.

19/03 - Tarde, mais avanços no frontend começando a conversa com a camada REST e aprendendo mais sobre como trabalhar com AngularJS.

19/03 - Treinamento & Madrugada, super avanços, testando e resolvendo erros no Front e no Backend, término do projeto, produção do relatório e commit no Git.

O que eu aprendi e não sabia: NodeJS, Git, REST e aprendi muito do AngularJS.