

Gustavo Ferreira Silva

Documentação do processo seletivo Voxus

Regras

Entrega deve ser feita em até 48 horas: 12/06/2020 às 13:00h.

Enviar para: recrutamento@voxus.tv e nathalia.ogura@voxus.tv

Artefatos à serem entregues: Documentação do processo, Códigos de implementação do desafio e manual de utilização da aplicação desenvolvida.

Desafio

Sistema para postar e armazenar pagamentos, deverá manipular conjunto de Dados contendo Título, Valor, Data, uma Taxa Externa e Observações.

Regras de negócio

Título: deve ter entre 5 e 100 caracteres

Valor: deve ser numérico

Data: deve estar no formato YYYY-MM-dd

Taxa Externa: 5% no valor do pagamento deve ser calculada automaticamente

Observações: Não é obrigatório o preenchimento

Requisitos do sistema

1. Inserção de pagamentos
2. Armazenar pagamentos em um banco de dados
3. Listagem de todos os pagamentos
4. Edição dos pagamentos listados
5. Exclusão dos pagamentos listados
6. Inserir pagamentos via arquivo .xls (Assumindo que o arquivo sempre venha formatado corretamente do usuário)

Desenvolvimento das funcionalidades

Server side

Criar API REST para persistir dados usando **Express** (módulo node responsável pelo servidor configurado na porta 3000), **MySQL** (como banco de dados), **body-parser** (para traduzir as requisições HTTP), **cors** (para dar permissão de requisições vindas de qualquer lugar).

Validação de data e título são realizadas no lado do servidor com funções específicas. Já a taxa é calculada no back End da aplicação, sempre que um cliente inserir um pagamento a taxa é enviada como nula, o que garante que o cliente possa ver a taxa no momento de sua inserção.

A persistência de dados é realizada seguindo os padrões REST, onde a inserção é feita pelo método POST, a listagem pelo GET, a alteração pelo PUT e deleção pelo DELETE.

Client side

O front End foi feito utilizando um template bootstrap pronto, para facilitar a criação do layout.

Na aplicação o cliente pode acessar a página de cadastro e inserir novos pagamentos, também tem botões para os pagamentos inseridos que servem para abrir a página de edição (onde ela é carregada com as informações do registro selecionado, usando seu id para buscá-lo no banco e exibir as informações

O cliente também seleciona o arquivo .xlsx e a aplicação transforma ele em json e envia ao servidor para realizar a persistência dos seus dados.

Maiores desafios

Bom, o maior desafio foi integrar os dois lados do sistema, eu nunca havia feito uma API, o conhecimento de back End que eu tinha era relativo à trabalho acadêmicos utilizando JAVA para integrar com MySQL usando o jdbc.

Como nos últimos meses eu fiz muitos cursos da plataforma de cursos online da Alura, muitos deles relativos a desenvolvimento web, já tinha uma ideia de como começar a construir uma API e que caminhos eu iria tomar para fazer a persistência de dados, também sabia como utilizar os padrões REST para realizar as requisições. Mas mesmo assim tive que fazer uma pesquisa mais aprofundada e longa a respeito de temas como: integrar o banco de dados usando node, desenvolver uma API seguindo os padrões REST com node, como consumir essa API usando fetch e também como transformar um .xlsx em um json para realizar o cadastro de pagamentos utilizando ele.