

A empresa Rental solicitou que você crie a API que contém os dados de todos os seus clientes, veículos, aluguéis etc. da empresa (utilize as modelagens criadas no exercício 1.2 - locadora de banco de dados).

Obs.: os clientes deverão ter nome e sobrenome separados em dois campos isolados.

Para isto, você deverá ser capaz de:

Banco de Dados

Script 01 :

- criar o banco de dados M\_Rental/T\_Rental;
- criar todas as tabelas de acordo com as modelagens;

Script 02 :

- inserir os dados seguindo as modelagens;

Script 03 :

- selecionar todos os registros;

Backend:

- Criar um projeto chamado Senai.Rental.WebApi;

- Criar os domínios correspondentes;

- Criar as interfaces correspondentes que contenham as ações de: listar, buscar por id, deletar, atualizar e inserir;

- Criar os repositórios correspondentes à interface;

- Criar os controllers contendo os métodos(verbos) de um CRUD básico para aluguéis, veículos e clientes.

Por exemplo:

GET	/api/aluguéis
-----	---------------

POST	/api/aluguéis
------	---------------

GET	/api/aluguéis/{id}
-----	--------------------

PUT	/api/aluguéis/{id}
-----	--------------------

DELETE	/api/aluguéis/{id}
--------	--------------------

Postman:

Você deverá criar uma coleção no postman chamada:

Senai.Rental.Manha/Senai.Rental.Tarde com os endpoints correspondentes ao item acima;

Logo, você terá 5 itens dentro da pasta de cada Controller com os respectivos verbos;

GitHub:

Você deverá criar um repositório ou adicionar ao seu repositório de backend existente:

- o projeto criado

- os scripts do banco de dados

  - senai-rental-01-DDL

  - senai-rental-02-DML

  - senai-rental-03-DQL

- exportar a coleção do postman e disponibilizar essa coleção dentro do repositório.

Desafios Extras:

- Incluir data de nascimento para os clientes;

- Criar um endpoint chamado /api/clientes/buscar/{nome} passando como parâmetro o nome do cliente e realizando a determinada busca no banco;
- Criar um endpoint chamado /api/clientes/nomescompletos que na saída do json, o nome e o sobrenome venham na mesma chave. Ex.: {  
"nomeCompleto" : "Catarina Strada" };
- Fazer com que o nome do cliente do backend seja obrigatório e mostre uma mensagem de erro, caso o usuário não envie.
- Criar APENAS UM endpoint para que seja listado os nomes dos clientes em ordem crescente ou decrescente. Da seguinte maneira:  
GET /api/clientes/ordenacao/{ordem}  
GET /api/clientes/ordenacao/asc ou  
GET /api/clientes/ordenacao/desc  
Caso o usuário informe uma ordenação que não exista, retornar uma mensagem de erro e devolver o erro 400 BadRequest).