Para testar nossa API, temos alguns exemplos de payloads do tipo Json para envio de requisições via Postman ou qualquer outra ferramenta. Esse payloads se encontram no projeto feito em java que pode ser clonado à partir do seguinte repositório:  
<https://github.com/JulioBCP/TechChallenge/tree/develop>

Após clonar este repositório, vá até o seguinte caminho no projeto: car-rental\car-rental\src\test\java\com\fiap\techchallenge\carrental\payloadsJson

Note que este caminho se encontra na área de testes automatizados da API. Utilizando qualquer um dos payloads disponibilizados para estes testes manuais, poderemos fazer as seguintes requisições, conforme veremos abaixo.

Outro ponto importante são as dependências de cada classe. Precisamos primeiramente criar um usuário, portanto a primeira requisição a ser feita será do Usuário. O qual contém login e senha, nome e tipo de usuário.

**Método: POST - inserirUsuario**

localhost:8080/Usuarios

**Exemplo de Body** (json)

{

"login": "joaodoe",

"senha": "senha123",

"nome": "João da Silva",

"tipoUsuario": "CLIENTE"

}

Após termos um usuário cadastrado, podemos cadastrar todos os dados de um cliente no sistema, logo podemos fazer a seguinte requisição POST:

**Método: POST – inserirCliente**

localhost:8080/clientes

**Exemplo de Body** (json)

{

"numeroCnh": 12345678910,

"dataNascimento": "1990-01-15",

"sexo": "MASCULINO",

"cpf": 12345678910,

"telefone": 19998765362,

"endereco": {

"rua": "Rua das Flores",

"numero": "123",

"bairro": "Centro",

"cidade": "Cidade",

"estado": "UF",

"pais": "Brasil",

"CEP": "12345-678"

},

"usuario": {

"id": 1,

"login": "joaodoe",

"senha": "senha123",

"nome": "João da Silva",

"tipoUsuario": "CLIENTE"

}

}

A qualquer momento podemos também cadastrar um veículo, pois este não depende de nenhuma outra entidade.

Para cadastrarmos um veículo temos a url e o exemplo abaixo:

**Método: POST – inserirVeiculo**

localhost:8080/Veiculos

**Exemplo de Body** (json)

{

"tipoVeiculo": "HATCH",

"marca": "TOYOTA",

"modelo": "Corolla",

"anomodelo": 2022,

"anoFabricacao": 2021,

"km": 15500,

"kmManutencao": 14000,

"motor": "2.0L 4 cilindros",

"cor": "Prata",

"tracao": "T4X2",

"arCondicionado": true,

"trioEletrico": true,

"tetoSolar": false,

"combustivel": "GASOLINA",

"cabineDupla": false,

"capacidadeCarga": "500 kg"

}

Temos vários payloads de veículos como exemplo a serem cadastrados. Se precisar cadastrar mais veículo, vá a pasta outrosVeiculos para payloads extras.

Tendo um cliente cadastrado no sistema, podemos fazer uma reserva de um veículo. Para tal, temos a requisição abaixo:

**Método: POST – inserirReserva**

localhost:8080/Reservas

**Exemplo de Body** (json)

{

"cliente": {

"numeroCnh": 12345678910,

"dataNascimento": "1990-01-15",

"sexo": "MASCULINO",

"cpf": 12345678910,

"telefone": 19998765362,

"endereco": {

"rua": "Rua das Flores",

"numero": "123",

"bairro": "Centro",

"cidade": "Cidade",

"estado": "UF",

"pais": "Brasil",

"CEP": "12345-678"

},

"usuario": {

"id": 1,

"login": "joaodoe",

"senha": "senha123",

"nome": "João da Silva",

"tipoUsuario": "CLIENTE"

}

},

"veiculo": {

"id": 1,

"tipoVeiculo": "HATCH",

"marca": "TOYOTA",

"modelo": "Corolla",

"anomodelo": 2022,

"anoFabricacao": 2021,

"km": 15500,

"kmManutencao": 14000,

"motor": "2.0L 4 cilindros",

"cor": "Prata",

"tracao": "T4X2",

"arCondicionado": true,

"trioEletrico": true,

"tetoSolar": false,

"combustivel": "GASOLINA",

"cabineDupla": false,

"capacidadeCarga": "500 kg"

},

"valorReserva": 1300.0,

"dataInicio": "2023-01-15",

"dataFim": "2023-01-20"

}

Tendo uma reserva cadastrada no sistema, podemos fechar um contrato. A requisição para o contrato segue o formato abaixo:

**Método: POST – inserirContrato**

localhost:8080/Contratos

**Exemplo de Body** (json)

{

"formaPagamento": "CREDITO",

"multaDiaria": 50.0,

"reserva": {

"id": 1,

"cliente": {

"numeroCnh": 12345678910,

"dataNascimento": "1990-01-15",

"sexo": "MASCULINO",

"telefone": 19998765362,

"endereco": {

"rua": "Rua das Flores",

"numero": 123,

"bairro": "Centro",

"cidade": "Cidade",

"estado": "UF",

"pais": "Brasil",

"cep": 0

},

"usuario": {

"id": 1,

"login": "joaodoe",

"senha": "senha123",

"nome": "João da Silva",

"tipoUsuario": "CLIENTE"

},

"cpf": 12345678910

},

"veiculo": {

"id": 1,

"tipoVeiculo": "HATCH",

"marca": "TOYOTA",

"modelo": "Corolla",

"anomodelo": 2022,

"anoFabricacao": 2021,

"km": 15500,

"kmManutencao": 14000,

"motor": "2.0L 4 cilindros",

"cor": "Prata",

"tracao": "T4X2",

"arCondicionado": true,

"trioEletrico": true,

"tetoSolar": false,

"combustivel": "GASOLINA",

"cabineDupla": false,

"capacidadeCarga": "500 kg"

},

"valorReserva": 1300.0,

"dataInicio": "2023-01-15",

"dataFim": "2023-01-20"

}

}

Na saída ou retorno do veículo, a equipe de manutenção realizada o check-list do veículo, o qual deve estar vinculado ao contrato, para que em caso de necessidade de cobrança extra, esta seja feita. Para inserir um checklist no sistema, temos a seguinte requisição:

**Método: POST – inserirCheckList**

localhost:8080/checkLists

**Exemplo de Body** (json)

{

   "contrato":{

        "id": 1,

        "formaPagamento": "CREDITO",

        "multaDiaria": 50.0,

        "reserva":{

            "id": 1,

            "cliente": {

                "numeroCnh": 12345,

                "dataNascimento": "1990-01-15",

                "sexo": "MASCULINO",

                "CPF": 12345678901,

                "telefone": 987654321,

                "endereco": {

                    "rua": "Rua das Flores",

                    "numero": "123",

                    "bairro": "Centro",

                    "cidade": "Cidade",

                    "estado": "UF",

                    "CEP": "12345-678"

                },

                "usuario": {

                    "login": "joaodoe",

                    "senha": "senha123",

                    "nome": "João da Silva",

                    "tipoUsuario": "CLIENTE"

                }

            },

            "veiculo": {

                "id": 1,

                "tipoVeiculo": "HATCH",

                "marca": "TOYOTA",

                "modelo": "Corolla",

                "anomodelo": 2022,

                "anoFabricacao": 2021,

                "km": 15500,

                "kmManutencao": 14000,

                "motor": "2.0L 4 cilindros",

                "cor": "Prata",

                "tracao": "T4X2",

                "arCondicionado": **true**,

                "trioEletrico": **true**,

                "tetoSolar": **false**,

                "combustivel": "GASOLINA",

                "cabineDupla": **false**,

                "capacidadeCarga": "500 kg"

            },

            "valorReserva": 1300.0,

            "dataInicio": "2023-01-15",

            "dataFim": "2023-01-20"

        }

    },

   "abastecido": **true**,

   "pressaoPneus": 32,

   "avariasPintura": "Arranhões na porta esquerda",

   "avariasInterna": "Manchas no banco traseiro",

   "limpeza": "Interior limpo, exterior sujo",

   "km": 15500,

   "tipoCheckList": "ENTRADA"

}

É importante observar os ids de cada entidade. Nos exemplos acima, temos todas as entidades sendo criadas com id =1, porém esses ids são gerados automaticamente, portanto, se for cadastrado mais de um cliente, por exemplo, é necessário verificar a qual cliente está sendo vinculado a reserva, ou a qual reserva está sendo vinculado um contrato e assim por diante.

Com todos os cadastros realizados, caso queira buscar algum dado de qualquer entidade, temos os seguintes métodos GET.

**Método: GET - buscaUsuario**

localhost:8080/Usuarios/{id}

**Método: GET - buscaCliente**

localhost:8080/clientes/{numeroCnh}

**Método: GET - buscaVeiculo**

localhost:8080/Veiculos/{id}

**Método: GET - buscaReserva**

localhost:8080/Reservas/{id}

**Método: GET - buscaContrato**

localhost:8080/Contratos/{id}

**Método: GET - buscaCheckList**

localhost:8080/checkLists/{id}

Da mesma forma temos os métodos PUT para updates. Note que não temos update Contrato, pois em caso de necessidade de alteração, deve ser feito um novo contrato:

**Método: PUT - atualizaUsuario**

localhost:8080/Usuarios/{id}

Para a entidade Usuários só é possível alterar login e senha.

**Método: PUT - atualizaCliente**

localhost:8080/clientes/{numeroCnh}

Para a entidade Clientes só é possível alterar nome e endereço, de acordo com o DTO recebido.

**Método: PUT - atualizaVeiculo**

localhost:8080/Veiculos/{id}

Para a entidade Veiculos só é possível alterar

**Método: PUT - atualizaReserva**

localhost:8080/Reservas/{id}

Para a entidade Reservas só é possível alterar km atual e km da próxima manutenção, de acordo com o DTO recebido neste método.

**Método: PUT - atualizaCheckList**

localhost:8080/checkLists/{id}

Para a entidade CheckList é possível alterar os seguintes campos: isAbastecido, pressão dos pneus, avarias de pintura, avarias interna, limpeza e km atual.

E por fim também temos os métodos Delete, para exclusão de qualquer entidade do sistema, a partir de seu id:

**Método: DELETE - deletaUsuario**

localhost:8080/Usuarios/{id}

**Método: DELETE - deletaCliente**

localhost:8080/clientes/{numeroCnh}

**Método: DELETE - deletaVeiculo**

localhost:8080/Veiculos/{id}

**Método: GE DELETE T - deletaReserva**

localhost:8080/Reservas/{id}

**Método: DELETE - deletaContrato**

localhost:8080/Contratos/{id}

**Método: DELETE – deletaCheckList**

localhost:8080/checkLists/{id}