Documentação da API utilizado no sistema da FastParking

Membros: Gabriel Mochiute e João Lucas.

Para que serve?

A API FastParking prove uma maneira de inserir e monitorar a chegada e saída de veículos no estacionamento também tendo opções de fazer relatórios diários, mensais e anuais.

Para utilizar API deve se entrar na pasta “API” no arquivo “index.php” pelo navegador ou por alguma linguagem de programação que possa consumir API’s

A uma lista de EndPoints que podem ser utilizados logo abaixo.

Lista de EndPoints GET’s que podem ser feitas pela URL

1. **Estadias**
    1) estadia
        Retorna todas as estadias registradas no Banco de Dados
    2) estadia/(ID)
        Retorna da Estadia com ID requisitado na URL
2. **Usuários**
    1) usuario
        Retorna todos os usuários registrados no Banco de Dados
    2) usuario/(ID)
        Retorna o usuario com o ID requisitado na URL

3. **Preços**
    1) preco
        Retorna os preços da hora inicial e horas adicionais

Lista de EndPoints POST's que podem ser utilizados

1. **estadia**
   Insere no Banco de Dados uma estadia com as informações enviadas em JSON

   Method:'POST',
   Headers: {'Content-Type': 'application/json'},
   Body: JSON. Stringify (JSON)

   As variáveis do json devem ser
   {
   "nomeDoCliente":nome,
   "placaDoVeiculo":placa,
   "dataDaEntrada":data,
   "horaDaEntrada":hora,
   "pago":boolean,
   "valor":preço
   }

2. **usuario**
   Insere no Banco de Dados um usuario com as informações enviadas em JSON

   Method:'POST',
   Headers: {'Content-Type': 'application/json'},
   Body: JSON. Stringify (JSON)

As variáveis do json devem ser
{
“nome”:nome,
“senha”:placa,
“nivelAcesso”:data
}

3. **usuario/Imagem/(ID)**

move a imagem enviada para uma pasta de fotos de usuários e atualiza no banco do usuário baseado no ID enviado, a imagem do usuario

enviado por form-data

4. **senha**

Retorna a senha enviada criptografada em md5 para verificação de usuario no login

Method:’POST’,
Headers: {'Content-Type': 'application/json'},
Body: JSON. Stringify (JSON)

As variáveis do json devem ser
{
“senha”:senha,
}

5. **preco**

altera os preços no Banco de Dados usando as informações enviadas em JSON

Method:’POST’,
Headers: {'Content-Type': 'application/json'},
Body: JSON. Stringify (JSON)

As variáveis do json devem ser
{
“nome”:nome,
“senha”:placa,
“nivelAcesso”:data
}

Lista de EndPoints PUT’s que podem ser utilizados

1. **estadia**

altera os dados de uma estadia pelo ID no Banco de Dados substituindo pelos dados enviados em JSON

Method: ‘PUT’,
Headers: {'Content-Type': 'application/json'},
Body: JSON. Stringify (JSON)

As variáveis do json devem ser
{
“idEstadia”:ID
“nomeDoCliente”:nome,
“placaDoVeiculo”:placa,
“dataDaEntrada”:data,
“horaDaEntrada”:hora,
“dataDaSaida”:data,
“horaDaSaida”:hora,

“pago”:boolean,
“valor”:preço
}

2. estadia/saída

altera os dados da parte de saída, valor e se a estadia foi paga de uma estadia pelo ID no Banco de Dados substituindo ou adicionando pelos dados enviados em JSON

Method: ’PUT’,
Headers: {'Content-Type': 'application/json'},
Body: JSON. Stringify (JSON)

As variáveis do json devem ser
{
“idEstadia”:ID
“dataDaSaida”:data,
“horaDaSaida”:hora,
“pago”:boolean,
“valor”:preço
}

3. usuário

altera os dados de um usuário pelo ID no Banco de Dados substituindo pelos dados enviados em JSON

Method: ‘PUT’,
Headers: {'Content-Type': 'application/json'},
Body: JSON. Stringify (JSON)

As variáveis do json devem ser

```
{
    “idUsuario”:ID,
    “nome”:nome,
    “senha”:senha,
    “statusUsuario”:boolean
    “nivelAcesso”:preço
}
```

4. **usuário/ativarDesativar/(id)**
   ativa ou desativa um usuário baseado no ID enviado pela URL

Lista de EndPoints DELETE’s que podem ser utilizados

1. estadia/(ID)
   Ira deletar uma estadia do Banco de Dados baseado no ID recebido.

2. usuario/(ID)
   Ira deletar um usuario do Banco de Dados baseado no ID recebido.

Os EndPonts possuem dois ou mais Status codes que eles podem retornar para informar se a requisição deu certo ou errado, será passado uma lista de status codes de cada EndPoint.

EndPoints de método GET

1. estadia e estadia/(ID)

   Caso tiver retorno dos dados o status code será 200(OK) junto com todos os dados da tabela de estadia,

porem Caso não tiver nem um retorno de dados a API irá retorna um 204(No Content).

2. usuario e usuario/(ID)

   Caso tiver retorno dos dados o status code será 200(OK) junto com todos os dados da tabela de usuários, porem Caso não tiver nem um retorno de dados a API irá retorna um 204(No Content).

3. preco

   Caso tiver retorno dos dados o status code será 200(OK) junto com todos os dados da tabela de preços, porem Caso não tiver nem um retorno de dados a API irá retorna um 204(No Content).

EndPoints de método POST

1. estadia
   - 400(Bad Request) – A API retorna esse status code e uma mensagem caso o usuario estiver enviando dados nulos para serem inseridos no banco.

   - 201(Created) – A API retorna esse status code junto com os dados enviados pelo usuario quando a inserção de uma estadia nova foi feita com sucesso.

   - 500 (Internal Server Error) – A Api retorna esse status code e uma mensagem quando ouve uma falha ao tentar inserir os dados no banco, esse erro pode ser causa caso seja enviado algum valor do tipo errado.

- 415(Unsupported Media Type) – A API retorna esse erro e uma mensagem caso seja enviado dados em um formato que não é aceito pela API.

2. usuario
   - 400(Bad Request) – A API retorna esse status code caso o usuario estiver enviando dados nulos para serem inseridos no banco.

   - 201(Created) – A API retorna esse status code junto com os dados enviados pelo usuario quando a inserção de um Usuario novo foi feita com sucesso.

   - 500 (Internal Server Error) – A Api retorna esse status code e uma mensagem quando houve uma falha ao tentar inserir os dados no banco, esse erro pode ser causa caso seja enviado algum valor do tipo errado.

3. usuario/Imagem/(ID)
   - 201(Created) – A API retorna esse status code junto com os dados enviados pelo usuario quando a criação de um novo arquivo de foto e o nome do arquivo foram inseridos na tabela do usuario requisitado.

   - 400(Bad Request) – A API retorna esse status code caso o usuario estiver enviando dados nulos para serem inseridos no banco.

- 415(Unsupported Media Type) – A API retorna esse erro e uma mensagem caso seja enviado dados em um formato que não é aceito pela API.

4. senha
   - 400(Bad Request) – A API retorna esse status code caso o usuario estiver enviando dados nulos para serem inseridos no banco.

   - 200(OK) – A API retorna esse status code junto com o dado criptografado são retornados com sucesso

   - 415(Unsupported Media Type) – A API retorna esse erro e uma mensagem caso seja enviado dados em um formato que não é aceito pela API.

EndPoints de método PUT

1. estadia e estadia/saida
   - 400(Bad Request) – A API retorna esse status code caso o usuario estiver enviando dados nulos para serem inseridos no banco.

   - 201(Created) – A API retorna esse status code junto com os dados enviados pelo usuario quando a atualização dos dados foi feita com secesso

   - 500 (Internal Server Error) – A Api retorna esse status code e uma mensagem quando ouve uma falha ao

tentar inserir os dados no banco, esse erro pode ser causa caso seja enviado algum valor do tipo errado.

- 415(Unsupported Media Type) – A API retorna esse erro e uma mensagem caso seja enviado dados em um formato que não é aceitado pela API.

2. usuario

- 400(Bad Request) – A API retorna esse status code caso o usuario estiver enviando dados nulos para serem inseridos no banco.

- 201(Created) – A API retorna esse status code junto com os dados enviados pelo usuario quando a atualização dos dados foi feita com secesso

- 500 (Internal Server Error) – A Api retorna esse status code e uma mensagem quando houve uma falha ao tentar inserir os dados no banco, esse erro pode ser causa caso seja enviado algum valor do tipo errado.

- 415(Unsupported Media Type) – A API retorna esse erro e uma mensagem caso seja enviado dados em um formato que não é aceito pela API.

3. usuario/ativarDesativar/(ID)

- 201(Created) – A API retorna esse status code junto com os dados enviados pelo usuario quando a atualização dos dados foi feita com secesso

- 500 (Internal Server Error) – A Api retorna esse status code e uma mensagem quando houve uma falha ao tentar inserir os dados no banco, esse erro pode ser causa caso seja enviado algum valor do tipo errado.

4. preco
    - 400(Bad Request) – A API retorna esse status code caso o usuario estiver enviando dados nulos para serem inseridos no banco.

    - 201(Created) – A API retorna esse status code junto com os dados enviados pelo usuario quando a atualização dos dados foi feita com secesso

    - 500 (Internal Server Error) – A Api retorna esse status code e uma mensagem quando ouve uma falha ao tentar inserir os dados no banco, esse erro pode ser causa caso seja enviado algum valor do tipo errado.

    - 415(Unsupported Media Type) – A API retorna esse erro e uma mensagem caso seja enviado dados em um formato que não é aceito pela API.

EndPoints de método DELETE

1. usuario/(ID)
    - 202(Accepted) – A API retorna esse status code junto com uma mensagem falando que o usuario com ID recebido foi deletado

- 204(No Content) – A API retorna esse status code caso não foi encontrado um usuario com o ID recebido

2. estadia/(ID)

- 202(Accepted) – A API retorna esse status code junto com uma mensagem falando que a estadia com ID recebido foi deletada

- 204(No Content) – A API retorna esse status code caso não foi encontrado uma estadia com o ID recebido