**API Fast Parking**

API criada para fins estudantis para o trabalho proposto pelos professores Fernando Leonid e Marcel Neves do curso de Desenvolvimento de Sistemas para o Senai Jandira.

Sendo implementado os métodos POST, GET, PUT e DELETE, ou seja, todos os métodos necessários para um CRUD.

A API foi desenvolvida com o intuito de criar um sistema capaz de controlar a entrada e saída de clientes, gerar comprovantes e relatórios de rendimentos de um Estacionamento para a empresa FastParking**.**

**Utilização**

Configurar os arquivos para receberem os dados do banco de dados já pronto através de uma função:

function conexaoMysql ()

{

/\*Variáveis para conexão com o BD\*/

$server = (string) "localhost";

$user = (string) "root";

$password = (string) "\*senhaDoBanco\*";

$dataBase = (string) "\*nomeDoBanco\*";

/\*Cria a conexão com o BD MySQL\*/

if ($conexao = @mysqli\_connect($server, $user, $password, $dataBase))

return $conexao;

else

return false;

}

**XAMPP**

* Instalar o [XAMPP](https://www.apachefriends.org/pt_br/download.html)
* Ir até a pasta htdocs e baixar o projeto, por padrão ele é instalado em 'C:\xampp\htdocs\
* Abrir o navegador e digitar localhost ou o ip 127.0.0.1
* Será listado as pastas existentes, procurar por fastPark e acessar o arquivo index dentro da mesma.

**Banco de dados**

* No diretório fastParking/bd/conexaoMysql.php se localiza as informações sobre o banco de dados.



EndPoints – Método GET

|  |  |
| --- | --- |
| Titulo | Método GET para recebimento de todos os dados do cliente. |
| URL | /cliente |
| Método | GET |
| Resposta de sucesso | Código: 200 { "status":"success" } |
| Resposta de erro | Código: 204 {“status”: ”no content”} |

|  |  |
| --- | --- |
| Titulo | Método GET para recebimento dos dados de cliente pela ID |
| URL | /cliente/id |
| Método | GET |
| Resposta de sucesso | Código: 200 { "status":"success" } |
| Resposta de erro | Código: 204 {“status”: ”no content”} |

|  |  |
| --- | --- |
| Titulo | Método GET para recebimento do rendimento diário |
| URL | /relatorioDiario |
| Método | GET |
| Resposta de sucesso | Código: 200 { "status":"success" } |
| Resposta de erro | Código: 204 {“status”: ”no content”} |

|  |  |
| --- | --- |
| Titulo | Método GET para recebimento do rendimento mensal |
| URL | /relatorioMensal |
| Método | GET |
| Resposta de sucesso | Código: 200 { "status":"success" } |
| Resposta de erro | Código: 204 {“status”: ”no content”} |

|  |  |
| --- | --- |
| Titulo | Método GET para recebimento do rendimento anual |
| URL | /relatorioAnual |
| Método | GET |
| Resposta de sucesso | Código: 200 { "status":"success" } |
| Resposta de erro | Código: 204 {“status”: ”no content”} |

EndPoint – Método POST

|  |  |
| --- | --- |
| Titulo | Método POST para enviar dados |
| URL | /cliente |
| Método | POST |
| Resposta de sucesso | Código: 201 { "status":"success" } |
| Resposta de erro | Código: 204 {“status”: ”no content”} |
| Resposta de erro | Código: 400 {“status”: “Dados enviados não podem não podem ser nulos”} |
| Resposta de erro | Código: 400 {“status”: “Falha ao inserir os dados”} |
| Resposta de erro | Código : 400 {“status”: “Erro no contente-type da mensagem”} |

No método POST é obrigatório que o dado enviado seja em JSON, exemplo: {

        "nomeCliente": "Alfredo",

        "numeroPlacaVeiculo": "BCD1234",

        "dataHorarioEntrada": current\_timestamp(),

        "idVaga": "1",

        "idValor": "1",

        "diferenca": "1",

    }

EndPoints – Método Put

|  |  |
| --- | --- |
| Titulo | Método PUT para atualização de dados na saída do cliente |
| URL | /cliente/id |
| Método | PUT |
| Resposta de sucesso | Código: 200 { "status":"success" } |
| Resposta de erro | Código: 400 {“status”: ”fail”} |

EndPoints – Método DELETE

|  |  |
| --- | --- |
| Titulo | Método DELETE PARA EXCLUIR |
| URL | /cliente/id |
| Método | DELETE |
| Resposta de sucesso | Código: 200 { "status":"success" } |
| Resposta de erro | Código: 400 {“status”: ”fail”} |