|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing icon  Description automatically generated | | Curso TeSP em  Programação de Sistemas de Informação  SIS | |
|  |  | |  |
| **Ano Letivo 2021/2022** | **Av. Periódica** | | **2º Ano, 1º Semestre** |

|  |
| --- |
| **PROJETO DE SERVIÇOS E INTEROPERABILIDADE DE SISTEMAS** |

**Relatório de Acompanhamento do**

**Ciclo de Desenvolvimento de Software do Projeto em Serviços e Interoperabilidade de Sistemas**



|  |  |  |
| --- | --- | --- |
| **Turno**: PL1 | **Grupo**: 1 | **Docente**: Mário Fernandes |
| **Nº 2202415** | Andreia Agostinho | |
| **Nº 2201126** | Diogo Pereira | |
| **Nº 2201131** | Gonçalo Ferreira | |

Instituto Politécnico de Leiria

Torres Vedras

Curso Técnico Superior Profissional de Programação de Sistemas de Informação

10 de Janeiro de 2022

Andreia Agostinho

Diogo Pereira

Gonçalo Ferreira

**Índice**

[1 Introdução 4](#_Toc91760867)

[2 Endpoints 5](#_Toc91760868)

[2.1 Cars 5](#_Toc91760869)

[2.1.1 Métodos de Implementação 5](#_Toc91760870)

[2.2 Companies 10](#_Toc91760871)

[2.2.1 Métodos de Implementação 10](#_Toc91760872)

[2.3 CompaniesList 13](#_Toc91760873)

[2.3.1 Métodos de Implementação 13](#_Toc91760874)

[2.4 Contributors 14](#_Toc91760875)

[2.4.1 Métodos de Implementação 14](#_Toc91760876)

[2.5 Login 17](#_Toc91760877)

[2.5.1 Métodos de Implementação 17](#_Toc91760878)

[2.6 Repairs 18](#_Toc91760879)

[2.6.1 Métodos de Implementação 18](#_Toc91760880)

[2.7 Schedules 21](#_Toc91760881)

[2.7.1 Métodos de Implementação 21](#_Toc91760882)

[2.8 Signup 24](#_Toc91760883)

[2.8.1 Métodos de Implementação 24](#_Toc91760884)

[2.9 User 25](#_Toc91760885)

[2.9.1 Métodos de Implementação 25](#_Toc91760886)

[2.10 Informações 28](#_Toc91760887)

[Conclusão 29](#_Toc91760888)

# Introdução

No âmbito da unidade curricular de Serviços e Interoperabilidade de Sistemas do 1º Semestre do 2º ano do Curso Técnico Superior Profissional de Programação de Sistemas de Informação do Instituto Politécnico de Leiria, foi criado o projeto apelidado de “CarBuddy”, que consiste no desenvolvimento de uma plataforma de gestão de todo o tipo de veículos, desde as suas reparações às suas manutenções.

A nossa plataforma consiste numa combinação de três componentes, sendo eles um Website, uma Aplicação Móvel e uma API.

Assim, para a presente unidade curricular, o objetivo é o desenvolvimento da API, sendo esta uma API REST, para a unidade curricular de Plataformas de Sistemas de Informação, o objetivo é a criação e desenvolvimento do Website e para a unidade curricular de Acesso Móvel a Sistemas de Informação, o objetivo é a criação de uma aplicação móvel.

No website, é onde ocorre a gestão de contas de utilizador, onde irão ser vistos/registados os veículos, o seu estado e as suas reparações.

Na aplicação, é onde cada cliente tem listadas todas as reparações e os respetivos veículos, sendo também possível realizar todas as ações do website na ótica do cliente.

Por fim, para que haja uma interligação de dados entre ambos os componentes referidos anteriormente, procedemos ao desenvolvimento da API REST.

O objetivo, é também que mesmo a nível internacional, se possam registar e verificar o histórico de reparações, centralizando assim todos os dados.

O projeto é comum entre ambas as disciplinas, mas as fases que traçam este projeto são divididas pelas três, conforme a sua pertinência.

Podemos referir também o facto de que o projeto tem como objetivo geral a consolidação de todo o conhecimento adquirido ao longo do curso.

# Endpoints

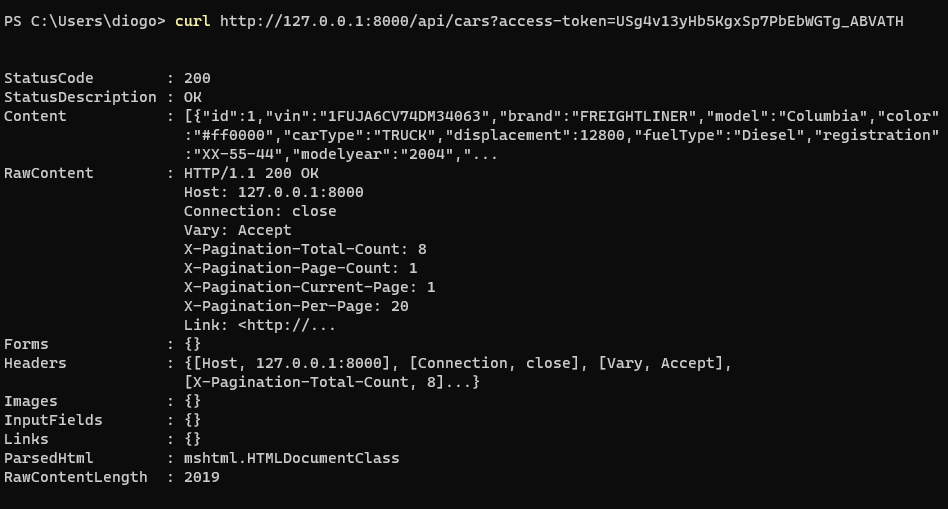
## Cars

### Métodos de Implementação

**Método GET**

Este método serve para obter a listagem de todos os veículos registados na plataforma e apenas o administrador consegue utilizar este método.

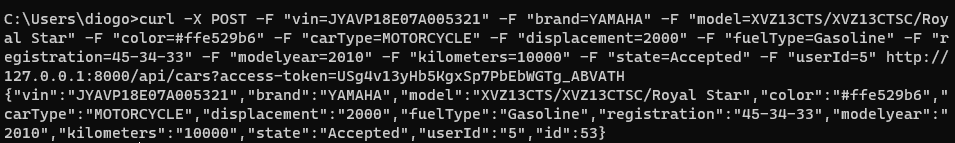
Relativamente ao Token, este é variável consoante o utilizador.



**URL:** curl <http://127.0.0.1:8000/api/cars?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método POST Default**

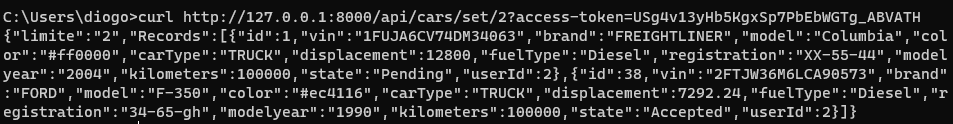
Este método serve para adicionar um novo veículo.



**URL:** curl -X POST -F "vin=JYAVP18E07A005321" -F "brand=YAMAHA" -F "model=XVZ13CTS/XVZ13CTSC/Royal Star" -F "color=#ffe529b6" -F "carType=MOTORCYCLE" -F "displacement=2000" -F "fuelType=Gasoline" -F "registration=45-34-33" -F "modelyear=2010" -F "kilometers=10000" -F "state=Accepted" -F "userId=5" <http://127.0.0.1:8000/api/cars?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método SET**

Este método serve para mostrar uma quantidade específica de dados, neste caso, todos os veículos até a um certo limite definido pelo utilizador.



**URL:** curl <http://127.0.0.1:8000/api/cars/set/2?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método TOTAL**

Este método só pode ser executado pelo administrador e serve para obter o número total de veículos registados na plataforma.



**URL:** curl <http://127.0.0.1:8000/api/cars/total?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método TOTAL USER**

Este método serve para obter o número total de veículos registados na plataforma de um dado utilizador, através da utilização do Access Token.



**URL:** curl <http://127.0.0.1:8000/api/cars/totaluser?access-token=hg95zPz_0_IKUJVLRnf07jwT7cgE_ad2>

**Método DELETE**

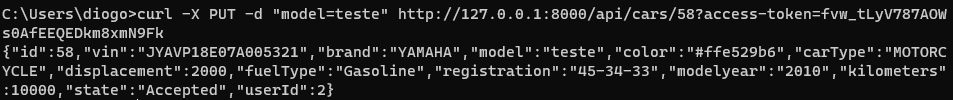
Este método só pode ser executado pelo administrador e serve para eliminar o veículo selecionado pelo mesmo.



**URL:** curl -X DELETE <http://127.0.0.1:8000/api/cars/58?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método PUT**

Este método serve para atualizar os dados de um veículo selecionado pelo utilizador.



**URL:** curl -X PUT -d "model=teste" <http://127.0.0.1:8000/api/cars/58?access-token=fvw_tLyV787AOWs0AfEEQEDkm8xmN9Fk>

**Método DELETE Custom**

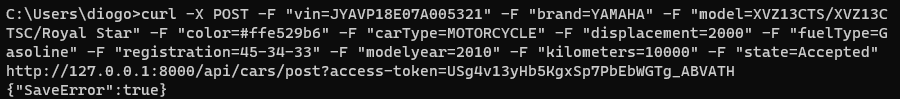
Este método só pode apagar registos de veículos quando o utilizador mencionado no Access Token é o proprietário do mesmo. Este método retorna 1 quando apaga o registo com sucesso e 0 quando o pedido falha.



**URL:** curl -X DELETE <http://127.0.0.1:8000/api/cars/deleted/61?access-token=3YEr44AJGN0CZHIXyvqKp9vYffzw0u1J>

**Método POST Custom**

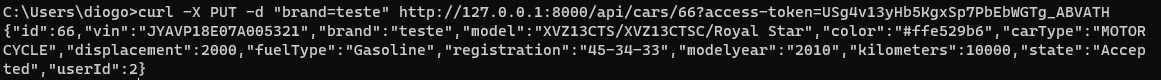
Neste método é necessário ser administrador, visto que é uma custom action.



**URL:** curl -X POST -F "vin=JYAVP18E07A005321" -F "brand=YAMAHA" -F "model=XVZ13CTS/XVZ13CTSC/Royal Star" -F "color=#ffe529b6" -F "carType=MOTORCYCLE" -F "displacement=2000" -F "fuelType=Gasoline" -F "registration=45-34-33" -F "modelyear=2010" -F "kilometers=10000" -F "state=Accepted" <http://127.0.0.1:8000/api/cars/post?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método PUT Custom**

Este método serve para editar os dados de um veículo selecionado pelo utilizador. O veículo só pode ser atualizado se o utilizador for o proprietário.



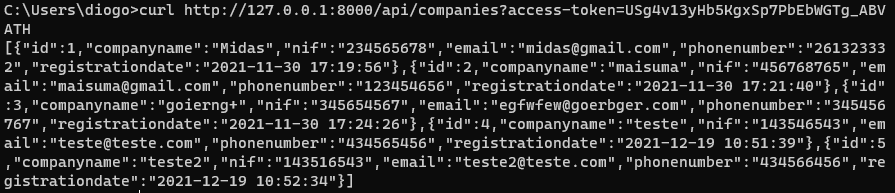
**URL:** curl -X PUT -d "brand=teste" <http://127.0.0.1:8000/api/cars/66?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

## Companies

### Métodos de Implementação

**Método GET**

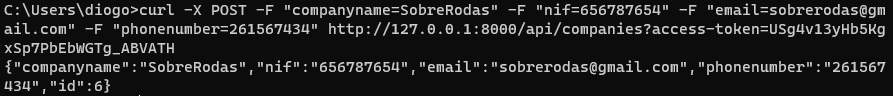
Este método serve para obter uma lista completa de todas as empresas registadas na plataforma e apenas pode ser executado pelo administrador.



**URL:** curl <http://127.0.0.1:8000/api/companies?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método POST**

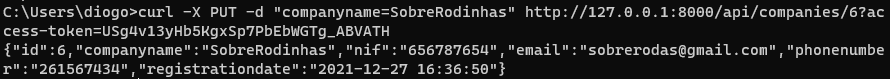
Este método serve para registar uma nova empresa na plataforma e apenas pode ser executado pelo administrador.



**URL:** curl -X POST -F "companyname=SobreRodas" -F "nif=656787654" -F "email=sobrerodas@gmail.com" -F "phonenumber=261567434" <http://127.0.0.1:8000/api/companies?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método PUT**

Este método serve para editar um registo de uma empresa na plataforma e apenas pode ser executado pelo administrador.

****

**URL:** curl -X PUT -d "companyname=SobreRodinhas" <http://127.0.0.1:8000/api/companies/6?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método DELETE**

Este método serve para eliminar um registo de uma empresana plataforma e apenas pode ser executado pelo administrador.

****

**URL:** curl -X DELETE <http://127.0.0.1:8000/api/companies/6?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método Total**

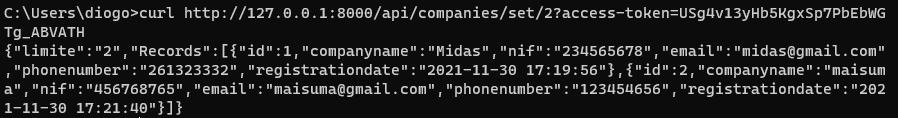
Este método serve para obter o número total de empresas registadas na plataforma e apenas pode ser executado pelo administrador.



**URL:** curl <http://127.0.0.1:8000/api/companies/total?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método SET**

Este método serve para obter uma lista até a um dado limite das empresas registadas na plataforma e apenas pode ser executado pelo administrador.

****

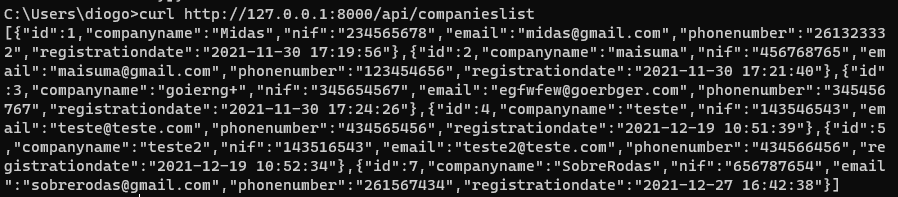
**URL:** curl <http://127.0.0.1:8000/api/companies/set/2?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

## CompaniesList

### Métodos de Implementação

**Método GET**

Este método serve para obter uma lista completa de todas as empresas registadas na plataforma e pode ser executado por qualquer utilizador, incluído o guest. Não requer Access Token.



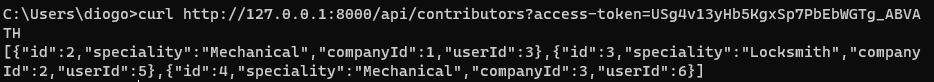
**URL:** curl <http://127.0.0.1:8000/api/companieslist>

## Contributors

### Métodos de Implementação

**Método GET**

Este método serve para obter uma lista completa de todos os colaboradores registados na plataforma e apenas pode ser executado pelo administrador.

****

**URL** curl <http://127.0.0.1:8000/api/contributors?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método POST**

Este método serve para registar um novo colaborador na plataforma e apenas pode ser executado pelo administrador.



**URL:** curl -X POST -F "speciality=Painter" -F "companyId=1" -F "userId=2" <http://127.0.0.1:8000/api/contributors?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método PUT**

Este método serve para editar um registo de um colaborador na plataforma e apenas pode ser executado pelo administrador.



**URL:** curl -X PUT -d "speciality=Electrician" <http://127.0.0.1:8000/api/contributors/5?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método DELETE**

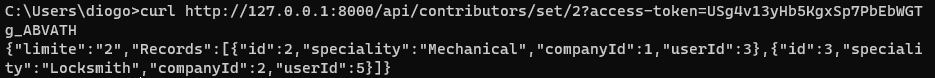
Este método serve para eliminar um registo de um colaboradorna plataforma e apenas pode ser executado pelo administrador.

****

**URL:** curl -X DELETE <http://127.0.0.1:8000/api/contributors/5?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método SET**

Este método serve para obter uma lista até a um dado limite dos colaboradores registados na plataforma e apenas pode ser executado pelo administrador.



**URL:** curl <http://127.0.0.1:8000/api/contributors/set/2?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método TOTAL**

Este método serve para obter o número total de colaboradores registados na plataforma e apenas pode ser executado pelo administrador.

****

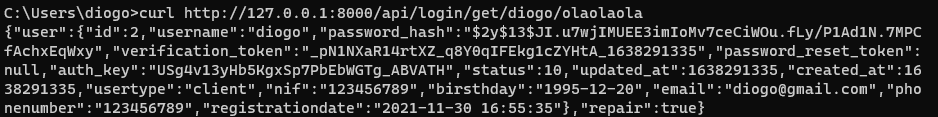
**URL:** curl <http://127.0.0.1:8000/api/contributors/total?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

## Login

### Métodos de Implementação

**Método GET**

Este método serve para validar a autenticação dos utilizadores e apenas pode ser executado pelo utilizador indicado pelo Access Token.



**URL:** curl <http://127.0.0.1:8000/api/login/get/diogo/olaolaola>

## Repairs

### Métodos de Implementação

**Método GET – Administrador**

Este método serve para obter uma lista completa de todas as reparações registadas na plataforma e apenas pode ser executado pelo administrador.

Uma imagem com texto

Descrição gerada automaticamente

**URL:** curl <http://127.0.0.1:8000/api/repairs?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método HISTORY**

Este método serve para obter uma lista completa de todas as reparações de um determinado veículo registado na plataforma e apenas pode ser executado pelo proprietário do veículo.

Uma imagem com texto

Descrição gerada automaticamente

**URL:** curl <http://127.0.0.1:8000/api/repairs/history/1?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método REPAIR CONTRIBUTOR**

Este método serve para retornar todas as reparações de um determinado colaborador e só pode ser executado por este mesmo colaborador.

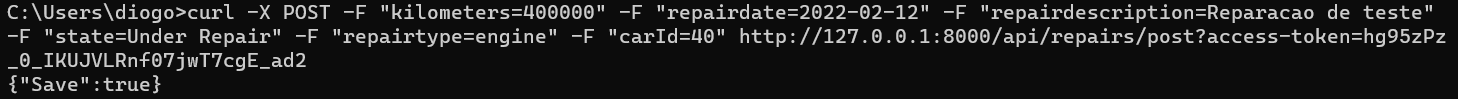
Uma imagem com texto, garrafa, captura de ecrã

Descrição gerada automaticamente

**URL:** curl <http://127.0.0.1:8000/api/repairs/repaircontributor?access-token=hg95zPz_0_IKUJVLRnf07jwT7cgE_ad2>

**Método POST**

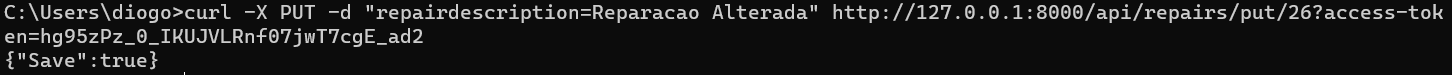
Este método serve para registar uma nova reparação na plataforma e apenas pode ser executado por um colaborador.

****

**URL:** curl -X POST -F "kilometers=400000" -F "repairdate=2022-02-12" -F "repairdescription=Reparacao de teste" -F "state=Under Repair" -F "repairtype=engine" -F "carId=40" <http://127.0.0.1:8000/api/repairs/post?access-token=hg95zPz_0_IKUJVLRnf07jwT7cgE_ad2>

**Método PUT**

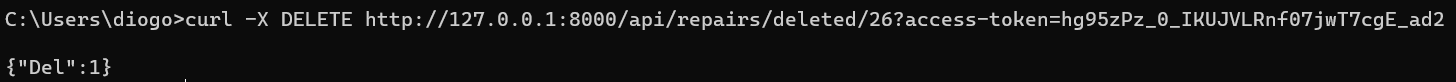
Este método serve para alterar uma reparação na plataforma e apenas pode ser executado pelo colaborador que a criou.



**URL:** curl -X PUT -d "repairdescription=Reparacao Alterada" <http://127.0.0.1:8000/api/repairs/put/26?access-token=hg95zPz_0_IKUJVLRnf07jwT7cgE_ad2>

**Método Deleted**

Este método serve para apagar uma reparação na plataforma e apenas pode ser executado pelo colaborador que a criou.



**URL:** curl -X DELETE <http://127.0.0.1:8000/api/repairs/deleted/26?access-token=hg95zPz_0_IKUJVLRnf07jwT7cgE_ad2>

## Schedules

### Métodos de Implementação

**Método TOTAL**

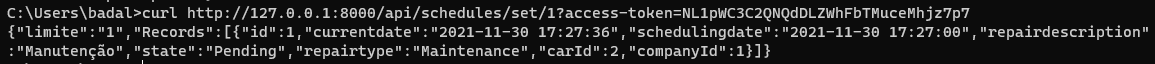
Este método serve para obter o número total de agendamentos registados na plataforma e apenas pode ser executado pelo administrador.



**URL:** curl <http://127.0.0.1:8000/api/schedules/total?access-token=NL1pWC3C2QNQdDLZWhFbTMuceMhjz7p7>

**Método SET**

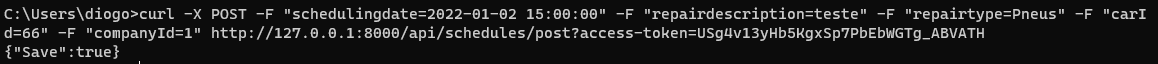
Este método serve para obter uma lista até a um dado limite dos agendamentos registados na plataforma e apenas pode ser executado pelo administrador.



**URL:** curl <http://127.0.0.1:8000/api/schedules/set/1?access-token=NL1pWC3C2QNQdDLZWhFbTMuceMhjz7p7>

**Método POST**

Este método serve para adicionar um novo agendamento na plataforma e pode ser executado por todos os utilizadores à exceção do guest.



**URL:** curl -X POST -F "schedulingdate=2022-01-02 15:00:00" -F "repairdescription=teste" -F "repairtype=Pneus" -F "carId=66" -F "companyId=1" <http://127.0.0.1:8000/api/schedules/post?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método DELETE CLIENT**

Este método serve para eliminar um registo de um agendamento na plataforma e apenas pode ser executado pelo responsável da marcação do agendamento.



**URL:** curl -X DELETE <http://127.0.0.1:8000/api/schedules/deleteclient/3?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método DELETE CONTRIBUTOR**

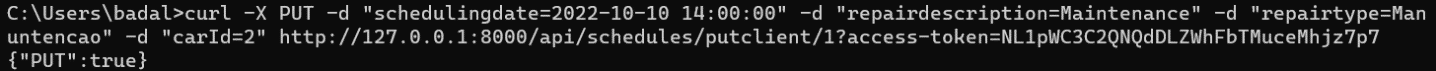
Este método serve para eliminar um registo de um agendamento na plataforma e apenas pode ser executado pelo responsável da empresa do agendamento.



**URL:** curl -X DELETE <http://127.0.0.1:8000/api/schedules/deletecontributor/2?access-token=USg4v13yHb5KgxSp7PbEbWGTg_ABVATH>

**Método PUT CLIENT**

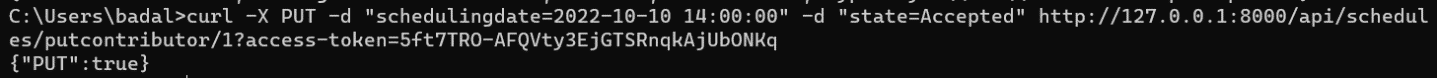
Este método serve para editar um registo de um agendamento na plataforma e apenas pode ser executado pelo responsável da marcação do agendamento.

****

**URL:** curl -X PUT -d "schedulingdate=2022-10-10 14:00:00" -d "repairdescription=Maintenance" -d "repairtype=Manuntencao" -d "carId=2" <http://127.0.0.1:8000/api/schedules/putclient/1?access-token=NL1pWC3C2QNQdDLZWhFbTMuceMhjz7p7>

**Método PUT CONTRIBUTOR**

Este método serve para editar um registo de um agendamento na plataforma e apenas pode ser executado pelo responsável da empresa do agendamento.



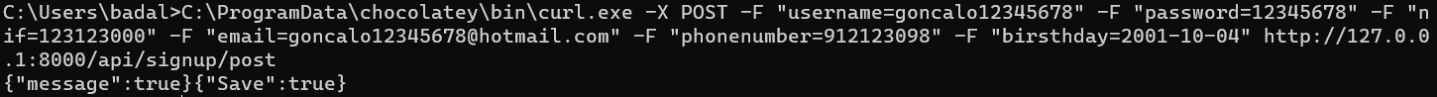
**URL:** curl -X PUT -d "schedulingdate=2022-10-10 14:00:00" -d "state=Accepted" <http://127.0.0.1:8000/api/schedules/putcontributor/1?access-token=5ft7TRO-AFQVty3EjGTSRnqkAjUbONKq>

## Signup

### Métodos de Implementação

**Método POST**

Este método serve para registar qualquer tipo de utilizador, não sendo obrigatório o Access Token.



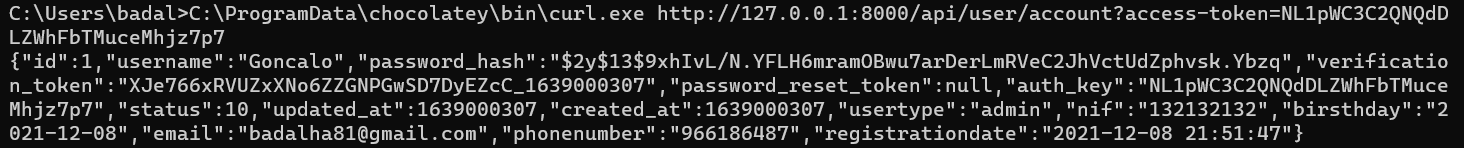
**URL:** curl -X POST -F "username=goncalo12345678" -F "password=12345678" -F "nif=123123000" -F "email=goncalo12345678@hotmail.com" -F "phonenumber=912123098" -F "birsthday=2001-10-04" <http://127.0.0.1:8000/api/signup/post>

## User

### Métodos de Implementação

**Método ACCOUNT**

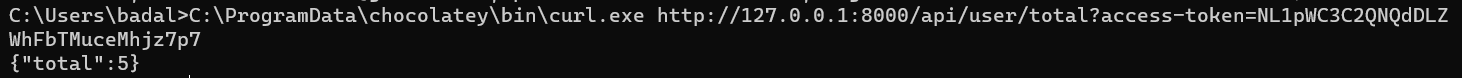
Este método serve para obter os dados da conta do utilizador associado ao Token utilizado no pedido. Este método requer Token e não está disponível para utilizadores guest.



**URL:** curl <http://127.0.0.1:8000/api/user/account?access-token=NL1pWC3C2QNQdDLZWhFbTMuceMhjz7p7>

**Método TOTAL**

Este método serve para obter o número total de utilizadores registados na plataforma e apenas pode ser executado pelo administrador.



**URL:** curl <http://127.0.0.1:8000/api/user/total?access-token=NL1pWC3C2QNQdDLZWhFbTMuceMhjz7p7>

**Método SET**

Este método serve para obter uma lista até a um dado limite dos utilizadores registados na plataforma e apenas pode ser executado pelo administrador.

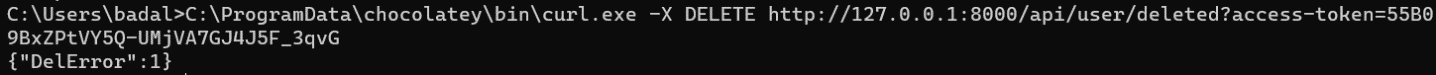
Uma imagem com texto

Descrição gerada automaticamente

**URL:** curl <http://127.0.0.1:8000/api/user/set/2?access-token=NL1pWC3C2QNQdDLZWhFbTMuceMhjz7p7>

**Método DELETED - CUSTOM**

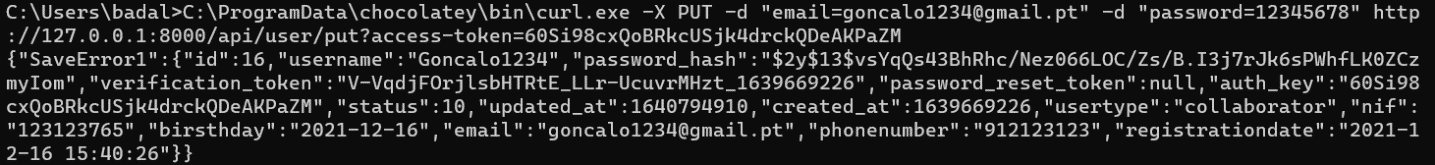
Este método serve para eliminar um registo de um utilizadorna plataforma e pode ser executado por qualquer utilizador, exceto os guest, visto que não possuem Access Token. Não requer o Id do utilizador, visto que o método utiliza o Access Token do utilizador em questão.



**URL:** curl -X DELETE <http://127.0.0.1:8000/api/user/deleted?access-token=55B09BxZPtVY5Q-UMjVA7GJ4J5F_3qvG>

**Método PUT - CUSTOM**

Este método serve para editar um registo de um utilizador na plataforma e pode ser executado por todos os utilizadores, exceto o guest.



**URL:** curl -X PUT -d "email=goncalo1234@gmail.pt" -d "password=12345678" <http://127.0.0.1:8000/api/user/put?access-token=60Si98cxQoBRkcUSjk4drckQDeAKPaZM>

## Informações

Alguns dos exemplos dos métodos referidos anteriormente necessitam, previamente, das credencias de autenticação, neste caso o access token do utilizador. Para esse efeito, para voltar a executar cada um desses métodos, o access token terá de ser alterado para um token de um utilizador que já se encontre registado na base de dados do projeto.

**Exemplo:** curl -X PUT -d "email=goncalo1234@gmail.pt" -d "password=12345678" [http://127.0.0.1:8000/api/user/put?access-token=**60Si98cxQoBRkcUSjk4drckQDeAKPaZM**](http://127.0.0.1:8000/api/user/put?access-token=60Si98cxQoBRkcUSjk4drckQDeAKPaZM)

**Explicação:** A zona a negrito do URL deve ser alterada para um novo token, salvo se, esse token estiver associado a um user já registado na base de dados.

O mesmo acontece com alguns métodos que necessitem do campo ID de um dado objeto. Por exemplo, alguns métodos de PUT ou Delete precisam do ID do objeto que queremos alterar, e mais uma vez esse ID terá que ser alterado para o ID de um objeto registado nessa tabela especifica.

**Exemplo:** curl -X PUT -d "schedulingdate=2022-10-10 14:00:00" -d "state=Accepted" [http://127.0.0.1:8000/api/schedules/putcontributor/**1**?access-token=5ft7TRO-AFQVty3EjGTSRnqkAjUbONKq](http://127.0.0.1:8000/api/schedules/putcontributor/1?access-token=5ft7TRO-AFQVty3EjGTSRnqkAjUbONKq)

**Explicação:** A zona a negrito do URL deve ser alterada para um novo ID, salvo se, esse ID estiver associado a um objeto desse tipo já registado na base de dados.

## Conclusão

Com a elaboração deste projeto, cujo tema seria a criação de uma API onde o objetivo é fornecer dados à nossa aplicação web e aplicação android, conseguimos desenvolver a prática da programação orientada a objetos e a sistemas REST.

A nível de dificuldades podemos referir, que inicialmente, não sabíamos por onde começar. Com o passar do tempo, e das aulas da UC de SIS, e após algumas das fichas práticas, fomos ganhando prática no desenvolvimento de API’s REST, sendo também bastante útil toda a documentação da YII framework. Bem perto do fim do projeto, sentimentos dificuldades também, na implementação da tecnologia Mosquitto, pois não sabíamos bem onde a utilizar.

Enquanto grupo, o trabalho foi sempre divido e muito bem gerido entre nós, podendo assim, dizer que ambos os três elementos do grupo trabalharam de igual modo. O grupo fica contente por conseguir concluir assim um projeto com esta grandeza, pois sentimos evolução pessoal e profissional ao realizar este projeto.