# API RESTful

# Desenvolvida em Python utilizando o Micro-Framework FLASK e hospedada em servidor de Cloud Computing, EC2 da AWS.

Este documento tem por objetivo demonstrar as funcionalidades e modo de utilização dos recursos disponíveis da REST API. Foi desenvolvida por Fernando Emerson, candidato à vaga de estágio em Cloud Public da Claranet.

Para se conectar à API, o usuário deve realizar um primeiro cadastro e confirmá-lo por e-mail. Após o login, uma chave token de autenticação é gerada automaticamente e através dela será possível utilizar os recursos da API. A chave token não é fixa, a cada nova sessão do usuário uma nova chave é criada. Referente ao armazenamento das informações da API, são mantidas em arquivo de banco de dados SQLite.

Para consultar os dados que o teste prático solicita, é necessário realizar o cadastro na API, confirmar a conta acessando o link que será enviado para o email cadastrado, fazer o login e utilizar o token que será gerado para acessar através da URL /Fernando – (mais detalhes na sessão “consulta de dados”)

URL: http://ec2-18-218-189-237.us-east-2.compute.amazonaws.com/

## Cadastro de usuário

Exemplo de Requisição para cadastrar um novo usuário.

|  |  |
| --- | --- |
| **METODO** | **URL** |
| **POST** | /register |

|  |  |
| --- | --- |
| **Header** |  |
| **Content-Type** | **Application/json** |

|  |
| --- |
| **Request Body** |
| {  "login": "usuario",      "password": "senha",      "email": "email"  } |

**Resposta**

Feita a requisição com o método POST, será retornada uma mensagem informando que o usuário foi criado, seguido pelo status code correspondente (201).

|  |  |
| --- | --- |
| **Status** | **Response Body** |
| **201 Created** | {      "message": "Usuário criado com sucesso!"  } |

**Exceções**

1. A senha deve conter no mínimo 8 dígitos, caso contrário a seguinte mensagem será exibida

|  |  |
| --- | --- |
| **Status** | **Response Body** |
| **400** Bad Request | {  "message": "Senha muito fraca, por favor digite uma senha com pelo menos 8 caracteres."  } |

1. Caso o login que o cliente digitou já exista no banco de dados

|  |  |
| --- | --- |
| **Status** | **Response Body** |
| **400** Bad Request | {      "message": "Este login já existe, por favor escolha um diferente.”  } |

1. Caso o e-mail que o cliente digitou já exista no banco de dados

|  |  |
| --- | --- |
| **Status** | **Response Body** |
| **400** Bad Request | {      "message": "O E-mail digitado já existe"  } |

1. Caso não tenha sido preenchido o campo e-mail

|  |  |
| --- | --- |
| **Status** | **Response Body** |
| **400** Bad Request | {      "message": "O campo 'e-mail' precisa ser preenchido"  } |

1. O E-mail é verificado através de Expressões regulares, e se inválido retornará a seguinte mensagem

|  |  |
| --- | --- |
| **Status** | **Response Body** |
| **400** Bad Request | {      "message": "O e-mail é inválido, digite um e-mail válido e tente novamente.  } |

## Login de Usuário

Exemplo de Requisição para logar com um usuário.

|  |  |
| --- | --- |
| **METODO** | **URL** |
| **POST** | /login |

|  |  |
| --- | --- |
| **Header** |  |
| **Content-Type** | **Application/json** |

|  |
| --- |
| **Request Body** |
| {  "login": "joao",      "password": "123456789",      "email": "joao@gmail.com"  } |

**Resposta**

Retornará uma mensagem contendo o token de acesso que será necessário para fazer as requisições.

O Token expira a cada nova sessão do usuário.

|  |  |
| --- | --- |
| **Status** | **Response Body** |
| **200 OK** | {      "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJmcmVzaCI6ZmFsc2UsImlhdCI6MTYzOTY4NTYxOCwianRpIjoiYzg2ODFkZGEtMTM1My00NjRlLTgwOWEtM2NiM2QyNzBlNjMzIiwidHlwZSI6ImFjY2VzcyIsInN1YiI6MSwibmJmIjoxNjM5Njg1NjE4LCJleHAiOjE2Mzk2ODY1MTh9.OSRXRRncVvDbWDCCOqW\_PGnDp2ZQpncON9GVXBdwoLE"  } |

**Exceções**

1. Login ou senha que não existem

|  |  |
| --- | --- |
| **Status** | **Response Body** |
| **401** unauthorized | {      "message": "O usuário ou a senha está errado"  } |

1. Conta não foi verificada

|  |  |
| --- | --- |
| **Status** | **Response Body** |
| **401** Bad Request | {      "message": "É preciso verificar a conta para se conectar à API"  } |

## 

## Consultar dados

|  |  |
| --- | --- |
| **METHOD** | **URL** |
| **GET** | /Fernando |

|  |  |
| --- | --- |
| **Header** | |
| Content-Type | application/json |
| Authorization | Bearer {token\_de\_acesso} |

**Resposta**

|  |  |
| --- | --- |
| **Status** | **Response Body** |
| **200 OK** | {      "Nome": "Fernando Emerson",      "Idade": 22,      "Telefone": "(21)98931-9729",      "Status Code": "200 OK "  } |

**Exceções**

1. Token incorreto

|  |  |
| --- | --- |
| **Status** | **Response Body** |
| **422** UNPROCESSABLE ENTITY | {      "message": "Token inválido"  } |

1. Token expirou

|  |  |
| --- | --- |
| **Status** | **Response Body** |
| **401** unauthorized | {      "msg": "Token expirou"  } |

## Logout

|  |  |
| --- | --- |
| **METHOD** | **URL** |
| **POST** | /logout |

|  |  |
| --- | --- |
| **Header** | |
| Content-Type | application/json |
| Authorization | Bearer {token\_de\_acesso} |

|  |
| --- |
| **Request Body** |
| {  “message”: “Você foi deslogado!”  } |