

# Api de usuarios de sistema

Este documento explicita com exemplos, como utilizar os recursos disponíveis no REST API de cadastro e login de usuarios. Assim como, as formas de se realizar uma requisição e suas possíveis respostas.

## Cadastro de Usuário

### Requisição

Exemplo de Requisição cadastrar um novo usuário.

Method	URL
POST	/register_user

Header	
Content-Type	application/json

Request Body
<pre>1 { 2   "name": "marco", 3   "login": "marco-antonio", 4   "password": "marco", 5   "email": "marco.antonio.fulltime@gmail.com" 6 }</pre>

## Resposta

Como resposta, obtém-se uma mensagem de sucesso informado que usuário foi criado, e status code 201 Created (Criado).

Status	Response Body
201 Created	<div><div>JSON ▾Auth ▾QueryHeader 1Docs</div><pre>1 { 2   "name": "marco", 3   "login": "marco-antonio", 4   "password": "marco", 5   "email": "marco.antonio.fulltime@gmail.com" 6 }</pre></div>

## Requisição

Exemplo de Requisição para tentar cadastrar outro usuário com mesmo login “marco-antonio” e e-mail “marco.antonio.fulltime@gmail.com” .

Method	URL
POST	/register_user

Header	
Content-Type	application/json

Request Body
<div><div>JSON ▾Auth ▾QueryHeader 1Docs</div><pre>1 { 2   "name": "marco", 3   "login": "marco-antonio", 4   "password": "marco", 5   "email": "marco.antonio.fulltime@gmail.com" 6 }</pre></div>

## Resposta

Como resposta, obtém-se uma mensagem de erro, informando que o usuário chamado “marco-antonio” e e-mail “marco.antonio.fulltime@gmail.com” já existem.

Status	Response Body
400 Bad Request	<div><div>400 BAD REQUEST4.69 ms65 B</div><div>Preview ▾Header 4CookieTimeline</div><div><pre>1 { 2   "message": "The login 'marco-antonio' already exists." 3 }</pre></div></div> <div><div>400 BAD REQUEST5.8 ms84 BJust Now ▾</div><div>Preview ▾Header 4CookieTimeline</div><div><pre>1 { 2   "message": "The email 'marco.antonio.fulltime@gmail.com' already exists." 3 }</pre></div></div>

## Login de Usuário

### Requisição

Exemplo de Requisição de login com um usuário.

Method	URL
POST	/login

Header	
Content-Type	application/json

Request Body
<div> <div>JSON ▾</div> <div>Auth ▾</div> <div>Query</div> <div>Header</div> </div> <pre> 1 { 2   "login": "marco-antonio", 3   "password": "marco" 4 }</pre>

## Resposta

Como resposta, obtém-se uma mensagem com o token de acesso que será necessário para fazer as requisições que só podem ser feitas com login.

Status	Response Body
400	<pre> 1 { 2   "message": "User not confirmed" 3 }</pre>

## Check seu e-mail:

abra o e-mail no-reply e click no link

<input type="checkbox"/> ☆ NO-REPLY 4	Confirmação de cadastro de usuario - Click no link para confirmar seu cadastro de usuario.
---------------------------------------	--

[Click no link para confirmar seu cadastro de usuario.](#)

após clicar você sera informado que esta tudo certo.

Não seguro   marco25.pythonanywhere.com/confirmed/1	
---	--

## Obrigado

O email "marco.antonio.fulltime@gmail.com" foi confirmado com sucesso!  
Seu usuario "marco-antonio" já pode efetuar login.

## Resposta

agora que o usuario foi validado como resposta, obtém-se uma um token de acesso que será necessário para fazer as requisições que só podem ser feitas com login.

Status	Response Body
200 OK	<pre> 1 { 2   "token":    "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJmcmVzaCI6ZmFsc2UsImhhdCI6MTYyNDI5MDU    2NSwianRpiIjoIMDk1MjRkMGEtOWQyNS00YjEzLTk3OTQtZDBjYzIxN2YzMDCzIiwidHlwZSI6ImFj    Y2VzcyIsInN1YiI6MSwibmJmIjoxNjI0MjkwNTY1LCJleHAiOjE2MjQzMzM3NjV9.Mwo00VnMYlgK    QUSHFfH7IYF-ohIkKj9FLW_6m49Fwn8" 3 } </pre>

## Requisição

Exemplo de Requisição para tentar fazer login com um usuário que não existe.

Method	URL
POST	/login

Header	
Content-Type	application/json

Request Body
<div> <div>JSON ▾</div> <div>Auth ▾</div> <div>Query</div> <div>Header</div> </div> <pre> 1 { 2   "login": "usuario-inexistente", 3   "password": "senha" 4 } </pre>

## Resposta

Como resposta, obtém-se uma mensagem de erro 401 não autorizado, informando que usuário ou senha estão incorretos.

Status	Response Body
401 unauthorized	<div> <div>Preview</div> <div>Header</div> <div>Cookie</div> <div>Timeline</div> </div> <pre> 1 { 2   "message": "The username or password is incorrect." 3 }</pre>

## Logout de Usuário

### Requisição

Exemplo de Requisição para fazer logout de usuário. Envia-se o token de acesso, e esse token é invalidado.

Method	URL
POST	/logout

Header	
Authorization	Bearer {token_de_acesso}

### Resposta

Exemplo de Resposta: mensagem de sucesso informando que o usuário foi deslogado. Ao tentar usar esse token de acesso em qualquer requisição, ele não funcionará mais, a não ser que o usuário faça o login.

Status	Response Body
200 OK	<pre> 1 { 2   "message": "Logged out successfully" 3 }</pre>

# Consultar dados de Usuário

## Requisição

Exemplo de Requisição para ler dados de usuário: `/users/{user_id}`

Method	URL
GET	/users/1

Header	
Content-Type	application/json

## Resposta

Como resposta, obtém-se os dados do usuário com id “1” exceto a senha.

Status	Response Body
200 OK	<pre>1 { 2   "user_id": 1, 3   "name": "marco", 4   "login": "marco-antonio", 5   "email": "marco.antonio.fulltime@gmail.com", 6   "active": true 7 }</pre>

# Atualizar Usuário

## Requisição

Exemplo de Requisição para atualizar dados de usuário: `/users/{user_id}`

**OBS:** COMO REGRA DE NEGOCIO SO É POSSIVEL ATUALIZAR NOME E SENHA

Method	URL
PUT	/users/1

Header	
Content-Type	application/json

Request Body
<div> <div>JSON ▾</div> <div>Bearer ▾</div> <div>Query</div> </div> <pre> 1 { 2   "name": "Marco Antonio", 3   "password": "123456" 4 }</pre>

## Resposta

Como resposta, obtém-se os dados atualizados do usuário com id “1” exceto a senha.

Status	Response Body
200 OK	<pre> 1 { 2   "message": "user updated!", 3   "user": { 4     "user_id": 1, 5     "name": "Marco Antonio", 6     "login": "marco-antonio", 7     "email": "marco.antonio.fulltime@gmail.com", 8     "active": true 9   } 10 }</pre>

## Deletar Usuário

### Requisição

Exemplo de Requisição para deletar um usuário: `/users/{user_id}`

Method	URL
<b>DELETE</b>	/users/1

Header	
Authorization	Bearer {token_de_acesso}



## Resposta

Exemplos de Respostas, primeiro, a mensagem de sucesso ao deletar um usuário existente. Depois, ao tentar deletar o mesmo usuário, obtém-se o erro 404 not found, informando que o usuário não existe ou não foi encontrado

Status	Response Body
200 OK	<pre>1 { 2   "message": "user deleted." 3 }</pre>
404 Not Found	<pre>1 { 2   "message": "user not found" 3 }</pre>