

Documentação RESTAPI Wishlist

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

1. Consultar produtos

Requisição

Requisição para listar todos os produtos cadastrados no sistema.

Method	URL
GET	/produtos/

Resposta

Como resposta, obtém-se uma lista de todos os produtos cadastrados:

Status	Response Body
200 OK	<pre>[{"product_id": 1, "product": "Cooler Master Cooler líquido de CPU MasterLiquid ML360", "description": "MasterFan MF120 Halo: Ventiladores RGB endereçáveis de 2ª geração com 24 LEDs individuais ARGB.", "link": "link", "photo": "photo", "bought": false }, {"product_id": 2, "product": "RYZEN 7 5800X", "description": "8-CORE, 16-THREADS, 3.8GHZ (4.7GHZ TURBO), CACHE 36MB, AM4, ", "link": "https://www.pichau.com.br/processador-amd-ryzen-7-5800x-8-core-16-threads-3-8ghz-4-7ghz-turbo-cache-36mb-am4-100", "photo": "https://media.pichau.com.br/media/catalog/product/cache/2f958555330323e505eba7ce930bdf27/1/0/100-100000063woif_1.jpg", "bought": false }]</pre>

Requisição

Requisição para visualizar os dados de um produto específico. Faz-se um GET de `/produto/int:product_id`

Method	URL
GET	/produto/1

Resposta

Como resposta, obtém-se um JSON com os dados do produto requisitado.

Status	Response Body
200 OK	<pre>{ "product_id": 1, "product": "Cooler Master Cooler líquido de CPU MasterLiquid ML360", "description": "MasterFan MF120 Halo: Ventiladores RGB endereçáveis de 2ª geração com 24 LEDs individuais ARGB.", "link": "link", "photo": "photo", "bought": false }</pre>

Requisição

Requisição exemplo de quando o usuário pesquisar por um produto que não existe.

Method	URL
GET	/produto/id_invalido

Resposta

Como resposta, obtém-se uma mensagem de erro, dizendo que o produto não foi encontrado.

Status	Response Body
404 Not Found	<pre>{ "message": "Product not found." }</pre>

Requisição

Requisição exemplo para realizar a pesquisa de um produto aleatório.

Method	URL
GET	/produto/random/

Resposta

Como resposta, é retornado um JSON de um produto aleatório.

Status	Response Body
200 OK	<pre>{ "product_id": 1, "product": "Cooler Master Cooler líquido de CPU MasterLiquid ML360", "description": "MasterFan MF120 Halo: Ventiladores RGB endereçáveis de 2ª geração com 24 LEDs individuais ARGB.", "link": "link", "photo": "photo", "bought": false }</pre>

2. Cadastro de Usuário

Requisição

Exemplo de Requisição cadastrar um novo usuário.

Method	URL
POST	/cadastro/

Header	
Content-Type	application/json

Request Body

```
1 {  
2   "login": "leandro",  
3   "senha": "1234"  
4 }
```

Resposta

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

Status	Response Body
201 Created	<pre>{ "message": "User created successfully!" }</pre>

Resposta: Usuário já existe

Quando o campo login já existe no sistema, é retornada a seguinte mensagem:

Status	Response Body
400 Bad Request	<pre>{ "message": "the login 'ricardo' already exists." }</pre>

3. Login de Usuário

Requisição

Exemplo de Requisição para logar com um usuário.

Method	URL
POST	/login/

Header	
Content-Type	application/json

Request Body

```
1 {}
2 "login": "leandro",
3 "senha": "1234"
4 {}
```

Resposta

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

Status	Response Body
200 OK	<pre>{ "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9 .eyJpYXQiOjE1NTYzMjEyOTUsIm5iZiI6MTUwNjc1MTI5NSwic2VybmFtbiI6ImVudGl0eSI6MiwiZmVudCJpZGVudGl0eSI6MTIwMzhbHNLCJ0eXB1IjoiyWYjZXNlbnQ9Sye -OVi681bp0eeTBu0_yEmIsObE-EBLNBtdxLaRg" }</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
<pre>{ "login": "usuario_que_nao_existe", "senha": "asdf" }</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	<pre>{ "message": "The username or password is incorrect." }</pre>

4. Inserir produto

Requisição

Exemplo de Requisição para inserir um novo produto: **/produto/**

Method	URL
POST	/produto/

Header	
Content-Type	application/json
Authorization	Bearer {token_de_acesso}

Request Body
<pre>{ "product": "Cooler Master Cooler líquido de CPU MasterLiquid ML360", "description": "MasterFan MF120 Halo: Ventiladores RGB endereçáveis de 2ª geração com 24 LEDs individuais ARGB.", "link": "link", "photo": "photo" }</pre>

Resposta

O sistema retorna o produto criado.

Status	Response Body
201 Created	<pre>{ "product_id": 1, "product": "Cooler Master Cooler líquido de CPU MasterLiquid ML360", "description": "MasterFan MF120 Halo: Ventiladores RGB endereçáveis de 2ª geração com 24 LEDs individuais ARGB.", "link": "link", "photo": "photo", "bought": false }</pre>

5. Atualizar produto

Requisição

Exemplo de Requisição para atualizar produto existente: `/produto/int:product_id`

Method	URL
PUT	/produto/1

Header	
Content-Type	application/json
Authorization	Bearer {token_de_acesso}

Request Body
<pre>{ "product": "Cooler Master Cooler líquido de CPU MasterLiquid ML360", "description": "MasterFan MF120 Halo Ventiladores RGB endereçáveis de 2ª geração com 24 LEDs individuais ARGB.", "link": "link", "photo": "https://img.terabyteshop.com.br/produto/g/" }</pre>

Resposta

Exemplos de Respostas: Primeiro, a mensagem de sucesso ao atualizar dados do produto, e status code 200 OK. Na segunda, a mensagem retornada quando o token utilizando para autenticação não é mais válido.

Status	Response Body
200 OK	<pre> { "product_id": 1, "product": "Cooler Master Cooler líquido de CPU MasterLiquid ML360", "description": "MasterFan MF120 Halo: Ventiladores RGB endereçáveis de 2ª geração com 24 LEDs individuais ARGB.", "link": "link", "photo": "https://img.terabyteshop.com.br/produto/g/", "bought": false } </pre>
401 Unauthorized	<pre> { "msg": "Token has expired" } </pre>

Requisição

Exemplo de Requisição para atualizar um produto para o status de “comprado” (variável boolean: bought), não é necessário inserir dados, apenas o id na requisição: **/comprei/int:product_id**

Method	URL
PUT	/comprei/2

Header	
Content-Type	application/json
Authorization	Bearer {token_de_acesso}

Resposta

É retornado o produto em que a variável bought foi alterada.

Status	Response Body
200 OK	<pre>{ "product_id": 2, "product": "RYZEN 7 5800X", "description": "8-CORE, 16-THREADS, 3.8GHZ (4.7GHZ TURBO), CACHE 36MB, AM4,", "link": "https://www.pichau.com.br/ processador-amd-ryzen-7-5800x-8-core-16-threads-3-8ghz-4-7ghz-turbo-cache-36mb-am4-100", "photo": "https://media.pichau.com.br/media/catalog/product/cache/2f958555330323e505eba7ce930bdf27/1/ 0/100-100000063wof_1.jpg", "bought": true }</pre>

6. Deletar produto

Requisição

Exemplo de Requisição para deletar um produto: **/produto/int:product_id**

Method	URL
DELETE	/produto/3

Header	
Authorization	Bearer {token_de_acesso}

Resposta

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

Status	Response Body
200 OK	<pre>1 { 2 "message": "Product deleted." 3 }</pre>

404 Not Found	<pre>{ "message": "Product not found." }</pre>
---------------	--

7. Logout de Usuário

Requisição

Exemplo de Requisição para fazer logout de usuário. O token de acesso do usuário 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 novamente.

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

8. Consultar dados de Usuário

Requisição

Exemplo de Requisição para ler dados de usuário: `/usuarios/int:user_id`

Method	URL
GET	/usuarios/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>{ "user_id": 1, "login": "teste" }</pre>

9. Deletar Usuário

Requisição

Exemplo de Requisição para deletar um usuário: `/usuarios/int:user_id`

Method	URL
DELETE	/usuarios/2

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. No terceiro exemplo, o cliente enviou um token de autorização expirado.

Status	Response Body
200 OK	<pre>{ "message": "user deleted." }</pre>
404 Not Found	<pre>{ "message": "user not found." }</pre>
401 Unauthorized	<pre>{ "msg": "Token has expired" }</pre>