



Esta documentação visa esclarecer o funcionamento e os principais pontos para total aproveitamento da mesma.

A principal funcionalidade é prover informações para o front end que irá consumir essas informações..

Esta api trabalha com segurança via Token de acesso, não sendo uma api pública.

Ela recebe as requisições em JSON e as respostas também são retornadas em JSON.

A api sempre trará um retorno com tipo do retorno e a resposta, o tipo pode ser erro ou sucesso, em caso de sucesso retorna a informação solicitada, em caso de erro retorna uma mensagem especificando o erro.

Vejamos as funcionalidades e seus endpoints.

Login

O login é realizado através do endpoint, onde são informados o usuário e senha cadastrados e caso tenha sucesso ela retorna o id do usuário.

localhos:8080/api/usuarios/login

método: POST

The screenshot displays a REST client interface with the following components:

- Request Section:**
 - Method:** POST
 - URL:** `{{URL}}/api/usuarios/login`
 - Body:** A JSON object with the following fields:

```
1 {
2   "email": "andre@magazineaziul.com.br",
3   "senha": "123mudar"
4 }
```
- Response Section:**
 - Body:** A JSON object with the following fields:

```
1 {
2   "tipo": "sucesso",
3   "resposta": {
4     "id": 34
5   }
6 }
```

Listar todos usuários

A listagem de usuários se faz através do endpoint

localhos:8080/api/usuarios/listar

método: GET

The screenshot displays a REST client interface. At the top, the method is set to **GET** and the URL is `{{URL}}/api/usuarios/listar`. Below the URL bar, there are tabs for **Params**, **Authorization**, **Headers (7)**, **Body** (which is selected), **Pre-request Script**, **Tests**, and **Settings**. Under the **Body** tab, there are radio buttons for **none** (selected), **form-data**, **x-www-form-urlencoded**, **raw**, **binary**, and **GraphQL**. On the right side, it says "This request".

Below the request configuration, there is a section for the response. It has tabs for **Body** (selected), **Cookies**, **Headers (7)**, and **Test Results**. Under the **Body** tab, there are buttons for **Pretty** (selected), **Raw**, **Preview**, **Visualize**, **JSON** (with a dropdown arrow), and a button with a red icon. The response body is shown in a code editor with line numbers 40 to 57. It contains a JSON array with two user objects.

```
40 {
41   "id": 9,
42   "cargo": 3,
43   "unidade": 1,
44   "regiao": 1,
45   "nome": "Ronaldinho Gaucho",
46   "email": "ronaldinho.gaucho@magazineaziul.com.br",
47   "senha": "$2y$10$.VSjodecoSYaJM70gZwFD.TP/2Jw9jjwAqq4GREDHJnIfapL1Axu0"
48 },
49 {
50   "id": 10,
51   "cargo": 3,
52   "unidade": 2,
53   "regiao": 1,
54   "nome": "Roberto Firmino",
55   "email": "roberto.firmino@magazineaziul.com.br",
56   "senha": "$2y$10$1nNHUR.t3ec2qHgNQtmu30F9lyrNTlVU/Cp4JanyGNHFhkFiFkjD."
57 },
```

Adicionar novo usuário

A inclusão de um novo usuário se faz através do endpoint abaixo e enviando os seguintes parâmetros:

```
"cargo": int,  
"unidade": int,  
"regiao": int  
"nome": string  
"email": string  
"senha": string
```

Após o cadastro ser concluído com sucesso a api irá retorno o id do novo usuário.

localhos:8080/api/usuarios/cadastrar
método: POST

The screenshot displays a REST client interface. At the top, a dropdown menu shows 'POST' and the URL '{{URL}}/api/usuarios/cadastrar'. Below this, a tab bar includes 'Params', 'Authorization', 'Headers (9)', 'Body' (which is selected), 'Pre-request Script', 'Tests', and 'Settings'. Under the 'Body' tab, a radio button menu shows 'none', 'form-data', 'x-www-form-urlencoded', 'raw' (selected), 'binary', 'GraphQL', and 'JSON'. The main area contains a JSON request body with the following content:

```
1 {  
2   ... "cargo": 4,  
3   ... "unidade": 7,  
4   ... "regiao": 2,  
5   ... "nome": "Leonel Messi",  
6   ... "email": "leonel@magazineaziul.com.br",  
7   ... "senha": "123mudar"  
8 }
```

Below the request body, another tab bar shows 'Body' (selected), 'Cookies', 'Headers (7)', and 'Test Results'. Under the 'Body' tab, a radio button menu shows 'Pretty' (selected), 'Raw', 'Preview', 'Visualize', 'JSON', and a refresh icon. The main area displays the JSON response in a pretty-printed format:

```
1 {  
2   "tipo": "sucesso",  
3   "resposta": {  
4     "id": 34  
5   }  
6 }
```

Adicionar nova venda

A inclusão de uma nova venda é realizada através do endpoint abaixo e enviando os seguintes parâmetros:

cliente: string,
produto: string,
valor: decimal
vendedor: int
lat: string
lon: string

Após o cadastro ser concluído com sucesso a api irá retorno o id da nova venda

localhos:8080/api/vendas/inserir
método: POST

The screenshot displays a REST client interface with the following details:

- Method:** POST
- URL:** {{URL}}/api/vendas/inserir/
- Body Tab:** Selected, showing a JSON payload:

```
1 {  
2   "cliente": "Jack Barbosa",  
3   "produto": "Jogo de Panelas",  
4   "valor": 632.89,  
5   "vendedor": 32,  
6   "lat": "-16.673126240814387",  
7   "lon": "-49.25248826354209"  
8 }  
9
```
- Response Tab:** Selected, showing the JSON response:

```
1 {  
2   "tipo": "sucesso",  
3   "resposta": {  
4     "id_inserido": "32"  
5   }  
6 }
```

Relatório de vendas por cargo

Neste endpoint da api deverá ser passado o id do usuário para que a api analise qual seu cargo e nível de acesso que o usuário poderá usufruir.

localhos:8080/api/vendas/relatorio/26

método: GET

GET [{{URL}}/api/vendas/relatorio/7](#)

Params Authorization Headers (7) Body Pre-request Script Tests Settings

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1  {
2    "tipo": "sucesso",
3    "resposta": [
4      {
5        "data": "21/12/2022 às 23:48",
6        "cliente": "Francisca Alves",
7        "produto": "Geladeira",
8        "valor": "3223.40",
9        "vendedor": "Kevin",
10       "unidade": "Rio de Janeiro",
11       "regiao": "Sudeste",
12       "unid_prox": "Belo Horizonte",
13       "roaming": "S",
14       "id": 29
15     },
16     {
17       "data": "21/12/2022 às 22:32",
18       "cliente": "Jonas Campos",
19       "produto": "Armário de Cozinha",
20       "valor": "2135.60",
21       "vendedor": "Gabriel",
22       "unidade": "Vitória",
23       "regiao": "Sudeste",
24       "unid_prox": "Belo Horizonte",
25       "roaming": "S",
26       "id": 25
27     },
28     {
29       "data": "20/12/2022 às 00:48",
30       "cliente": "Muricy Ramalho",
31       "produto": "Bola de Basquete",
32       "valor": "435.20",
33       "vendedor": "Hugo",
34       "unidade": "São Paulo",
35       "regiao": "Sudeste",
36       "unid_prox": "São Paulo"
```

Relatório de vendas por unidade

Neste endpoint ela irá trazer todos as somas de vendas por unidade para que seja utilizada na produção do mapa, conforme material enviado.

localhos:8080/api/vendas/relatorio

método: GET

GET	▼	{{URL}}/api/vendas/relatorio/
-----	---	-------------------------------

Params Authorization Headers (7) Body Pre-request Script Tests Settings

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON ▼

```
1  [
2    "tipo": "sucesso",
3    "resposta": [
4      [
5        {
6          "id": 1,
7          "unidade": "Porto Alegre",
8          "lat": "-30.048750057541955",
9          "lon": "-51.228587422990806",
10         "total_vendas": "391.20"
11       }
12     ],
13     [
14       {
15         "id": 2,
16         "unidade": "Florianopolis",
17         "lat": "-27.55393525017396",
18         "lon": "-48.49841515885026",
19         "total_vendas": "6533.40"
20       }
21     ],
22     [
23       {
24         "id": 3,
25         "unidade": "Curitiba",
26         "lat": "-25.473704465731746",
27         "lon": "-49.24787198992874",
28         "total_vendas": "1023.40"
29       }
30     ],
31   ]
32 ]
```


Vendas detalhes

Aqui a api irá trazer os detalhes das vendas realizadas, para isso será necessário informar o id da venda.

localhos:8080/api/vendas/detalhes

método: POST

The screenshot displays a REST client interface with the following components:

- Request Section:**
 - Method:** POST
 - URL:** {{URL}}/api/vendas/detalhes/
 - Body Type:** JSON
 - Body Content:**

```
1 {
2   "venda": 29
3 }
```
- Response Section:**
 - Body:** Pretty
 - Response Content:**

```
1 {
2   "tipo": "sucesso",
3   "resposta": [
4     {
5       "data": "21/12/2022 às 23:48",
6       "cliente": "Francisca Alves",
7       "produto": "Geladeira",
8       "valor": "3223.40",
9       "vendedor": "Kevin",
10      "unidade": "Rio de Janeiro",
11      "regiao": "Sudeste",
12      "unid_prox": "Belo Horizonte",
13      "roaming": "S",
14      "id": 29
15    }
16  ]
17 }
```

Filtros de vendas

Este endpoint é onde pode ser feito as buscas pelos diversos filtros, sempre respeitando o grau hierárquico dos cargos.

Aqui você pode ver os parâmetros utilizados para filtrar:

- Entredatas, utilizando as datas de início e fim.
- Vendedor
- Unidade
- Região

Para que seja respeitado o grau hierárquico, o id do usuário deve ser passado no endpoint.

localhos:8080/api/vendas/filtrar/6

método: POST

POST ▼ `{{URL}}/api/vendas/filtrar/6`

Params Authorization Headers (9) Body

☐ none ☐ form-data ☐ x-www-form-urlencoded

1	
2	<code>"vendedor":40</code>
3	

POST ▼ `{{URL}}/api/vendas/filtrar/6`

Params Authorization Headers (9)

☐ none ☐ form-data ☐ x-www-form-urlencoded

1	
2	<code>"dtInicio":"19/12/2022",</code>
3	<code>"dtFim":"20/12/2022"</code>
4	

POST ▼ `{{URL}}/api/vendas/filtrar/6`

Params Authorization Headers (9) Body

☐ none ☐ form-data ☐ x-www-form-urlencoded

1	
2	<code>"unidade":6</code>
3	

POST ▼ `{{URL}}/api/vendas/filtrar/6`

Params Authorization Headers (9)

☐ none ☐ form-data ☐ x-www-form-urlencoded

1	
2	<code>"regiao":2</code>
3	