# Documentação da API

## Introdução

Esta API foi desenvolvida utilizando **Flask** e **MySQL**. Ela fornece um conjunto de operações para manipulação de usuários, armazenamento e produtos em um sistema. A API está protegida com um limite de requisições por minuto, utilizando o **Flask-Limiter**, e todos os acessos são registrados em um arquivo de log.

### Endpoints Disponíveis:

* **Login**
* **Cadastro de Usuários**
* **Consulta e Atualização de Usuários**
* **Cadastro de Armazenamento**
* **Consulta e Atualização de Armazenamento**
* **Cadastro de Produtos**
* **Consulta e Atualização de Produtos**

## Configurações Gerais

* **Porta**: 3000
* **Host**: 0.0.0.0
* **Limite de Requisições**: 5 por minuto (em todos os endpoints)

## 1. ****Login****

### Rota: /api/validar\_login

* **Método HTTP**: POST
* **Descrição**: Valida o login de um usuário a partir de seu nome de usuário e senha.

### Requisição:

{  
 "usuario": "nome\_do\_usuario",  
 "senha": "senha\_do\_usuario"  
}

### Resposta:

**200 OK**: Login bem-sucedido

{  
 "status": true,   
 "aut": valor\_autorizacao  
}

**401 Unauthorized**: Login inválido

{  
 "status": false,  
 "aut": 0  
}

**400 Bad Request**: Parâmetros inválidos ou faltantes

{  
 "status": false,  
 "aut": 0  
}

## 2. ****Cadastro de Usuário****

### Rota: /api/cadastro\_usuario

* **Método HTTP**: POST
* **Descrição**: Cadastra um novo usuário no sistema. Apenas usuários com autorização 5 podem realizar essa operação.

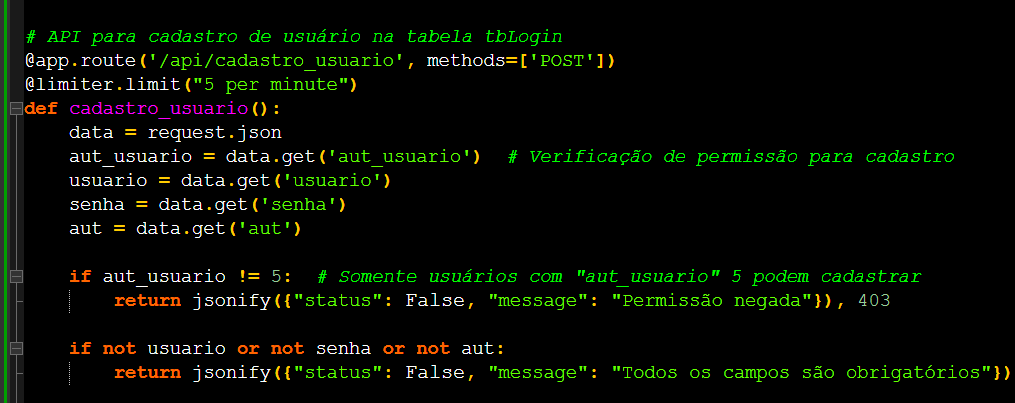
### Requisição:

{

"aut\_usuario": 5,  
 "usuario": "novo\_usuario",  
 "senha": "nova\_senha",  
 "aut": 1

}

OBS:   
 "aut\_usuario": 5, <-> Tem que vim do usuario logado  
"aut": 1 <-> Define se e master ou basico na autenticação  
  
Se aut = 1: siginifica que o usuario tera apenas permissoes simples  
Se aut = 5 siginifica que o usuario tera apenas permissoes de admininistrador

Para cadastro de usuario e preciso ser aut 5 , no codigo em python ele limita a função   


### Resposta:

**201 Created**: Usuário cadastrado com sucesso  
{  
 "status": true  
}

**403 Forbidden**: Permissão negada

{  
 "status": false,  
 "message": "Permissão negada"  
}

**403 Forbidden**: Permissão negada  
{  
 "status": false,  
 "message": "Permissão negada"  
}

**400 Bad Request**: Falta de parâmetros obrigatórios

{  
 "status": false,   
 "message": "Todos os campos são obrigatórios"  
}  
  
  
3. **Atualização de Usuário**

### Rota: /api/update\_usuario

* **Método HTTP**: PUT
* **Descrição**: Atualiza as informações de um usuário existente.

### Requisição:

{  
 "cod": 1,  
 "usuario": "usuario\_atualizado",  
 "senha": "senha\_atualizada",  
 "aut": 3  
}

OBS:   
"cod": 1, <-> Codigo do usuario a ser feito o update no banco   
usar a api “/api/consulta\_usuarios” para ver o codigo do cliente

### Resposta:

**200 OK**: Usuário atualizado com sucesso  
{   
 "status": true,  
 "message": "Usuário atualizado com sucesso"  
}

**404 Not Found**: Usuário não encontrado  
{  
 "status": false,  
 "message": "Usuário não encontrado"  
}

**400 Bad Request**: Parâmetros inválidos ou faltantes  
{  
 "status": false,  
 "message": "Todos os campos são obrigatórios"  
}

## 4. ****Consulta de Usuários****

### Rota: /api/consulta\_usuarios

* **Método HTTP**: GET
* **Descrição**: Consulta todos os usuários cadastrados. Apenas usuários com autorização 5 podem realizar essa operação.

### Parâmetro de URL:

* **aut\_usuario**: O nível de autorização do usuário que faz a requisição.  
  Exemplo: /api/consulta\_usuarios?aut\_usuario=5

### Resposta:

**200 OK**: Lista de todos os usuários  
[ {  
 "cod": 1,  
 "usuario": "usuario1",  
 "senha": "senha1",   
 "aut": 5  
 }, {  
 "cod": 2,  
 "usuario": "usuario2",  
 "senha": "senha2",  
 "aut": 3   
}]  
  
**403 Forbidden**: Permissão negada  
{  
 "status": false,  
 "message": "Permissão negada"  
}

## 5. ****Cadastro de Armazenamento****

### Rota: /api/cadastro\_armazenamento

* **Método HTTP**: POST
* **Descrição**: Cadastra um novo local de armazenamento.

### Requisição:

{  
 "desc\_local": "Novo Local"  
}

### Resposta:

**201 Created**: Armazenamento cadastrado com sucesso  
{  
 "status": true  
}

**400 Bad Request**: Parâmetro obrigatório faltante  
{  
 "status": false  
}

## 6. ****Consulta de Armazenamento****

### Rota: /api/consulta\_armazenamento

* **Método HTTP**: GET
* **Descrição**: Consulta todos os locais de armazenamento cadastrados.

### Resposta:

**200 OK**: Lista de todos os locais de armazenamento  
[ {  
 "cod": 1,  
 "desc\_local": "Local 1"  
 }, {  
 "cod": 2,  
 "desc\_local": "Local 2"  
 }]

**500 Internal Server Error**: Erro no banco de dados  
{  
 "status": false,  
 "message": "Erro no banco de dados"  
}

## 7. ****Atualização de Armazenamento****

### Rota: /api/update\_armazenamento

* **Método HTTP**: PUT
* **Descrição**: Atualiza as informações de um local de armazenamento.

### Requisição:

{  
 "cod": 1,  
 "desc\_local": "Local Atualizado"  
}

OBS: "cod": 1, codigo do local cadastrado, rodar api “consulta\_armazenamento”

### Resposta:

**200 OK**: Local de armazenamento atualizado com sucesso  
{   
"status": true,  
 "message": "Armazenamento atualizado com sucesso"  
}

**404 Not Found**: Local de armazenamento não encontrado  
{  
 "status": false,  
 "message": "Código do armazenamento não encontrado"  
}  
  
**400 Bad Request**: Parâmetros inválidos ou faltantes  
{  
 "status": false,  
 "message": "Os campos 'cod' e 'desc\_local' são obrigatórios"  
}

## 8. ****Cadastro de Produto****

### Rota: /api/cadastro\_produto

* **Método HTTP**: POST
* **Descrição**: Cadastra um novo produto no sistema.

### Requisição:

{  
 "codArmaz": 1,  
 "dscNome": "Produto 1",  
 "fornecedor": "Fornecedor X",   
 "qtdMaxima": 100,  
 "estadoFisico": "Líquido",  
 "substancias": "Substância Y",  
 "numeroCas": "123-45-6",  
 "concentracao": "10%",  
 "classsGHS": "Classe GHS 1",  
 "elemento1": 1,  
 "elemento2": 1,  
 "elemento3": 1,  
 "elemento4": 1,  
 "elemento5": 1,  
 "elemento6": 1,  
 "elemento7": 1,  
 "elemento8": 1,  
 "elemento9": 1,  
 "advertencia": "Advertência",  
 "frasePerigo": "Frase de Perigo",  
 "frasePrecaucao": "Frase de Precaução"  
}

OBS:  
os elementos e considerao 0 e 1   
se elemento = 1 o produto possui o elemento na sua composição  
se elemento = 0 o produto nao possui o elemento na sua composição  
"codArmaz": 1, Rodar antes a api consulta\_armazenemento para listar todos os fornecedores

### Resposta:

**201 Created**: Produto cadastrado com sucesso  
{   
 "status": true  
}

**400 Bad Request**: Falta de parâmetros obrigatórios  
{  
 "status": false,  
 "message": "O campo 'campo\_faltante' é obrigatório"  
}

## 9. ****Consulta de Produto****

### Rota: /api/consulta\_produto

* **Método HTTP**: GET
* **Descrição**: Consulta todos os produtos cadastrados ou um produto específico.

### Parâmetro de URL (Opcional):

* **cod**: Código do produto a ser consultado.

### Resposta:

**200 OK**: Produto(s) consultado(s) com sucesso  
[ {  
 "cod": 1,  
 "dscNome": "Produto 1",  
 "fornecedor": "Fornecedor X",  
 ...   
} ]  
  
**404 Not Found**: Produto não encontrado  
{   
"status": false,  
 "message": "Produto não encontrado"  
}

## 10. ****Atualização de Produto****

### Rota: /api/update\_produto

* **Método HTTP**: PUT
* **Descrição**: Atualiza as informações de um produto existente.

### Requisição:

{  
 "cod": 1,  
 "dscNome": "Produto Atualizado",  
 "fornecedor": "Fornecedor Y",  
 ...  
}

OBS: "cod": 1, codigo do produto rodar a consulta\_produtos para pegar o valor

### Resposta:

**200 OK**: Produto atualizado com sucesso  
{  
 "status": true,  
 "message": "Produto atualizado com sucesso"  
}  
  
**404 Not Found**: Produto não encontrado  
{  
 "status": false,  
 "message": "Código do produto não encontrado"  
}

**400 Bad Request**: Falta de parâmetros obrigatórios  
{  
 "status": false,  
 "message": "Os campos obrigatórios são 'cod' e 'dscNome'"  
}

## Conclusão

Essa API permite o gerenciamento completo de usuários, locais de armazenamento e produtos. Todos os endpoints são protegidos por um limite de requisições por minuto, e os acessos são registrados para auditoria e monitoramento.

Se houver dúvidas sobre a implementação ou uso dos endpoints, entre em contato com o administrador da API.