

## **Documentação da API**

Trabalho apresentado ao professor Anderson Barboza da Cruz, na disciplina Programação Orientada a Objetos em Java, pelas(os) alunos(as), Alana Karine, Davi Nascimento, Gustavo Rodrigues e Jamil Salomão.

Juiz de Fora, 08 de Outubro de 2024

**Sumário**

## 1. Introdução

## 2. Endpoints

### 2.1. Rotas de Usuário

### 2.2. Rotas de Vagas

### 2.3. Rotas de Inscrição

### 2.4. Rotas de Status do Processo Seletivo

## 3. Observações Finais

## **1. Introdução**

O objetivo dessa documentação é principalmente demonstrar como utilizar a API e fornecer uma visão geral das rotas disponíveis para o sistema do projeto, métodos HTTP, necessidade de autenticação, formatos de corpo de requisição, com possíveis respostas.

## **2. Endpoints**

Primeiramente para utilizar, inicie o servidor Flask pelo arquivo `app.js` no

source do projeto, após isso você receberá uma mensagem semelhante a essa logo abaixo indicando que o servidor está pronto:

**Server is running on <http://localhost:3001>**

**LEMBRETE:** Para utilizar baixe o aplicativo Postman ou Insomnia.

**Postman:** <https://www.postman.com/downloads/>

**Insomnia:** <https://insomnia.rest/download>

#### **Autenticação:**

- A maioria dos endpoints requer que o usuário esteja autenticado. A autenticação é feita usando JWT.

#### **Headers:**

- `Content-Type: application/json`

## **2.1. Rotas de Usuário**

### **1. Pegar Dados do Usuário:**

- **URL:** /api/pegar-dados-usuario

- **Método:** GET
- **Autenticação:** Requerida
- **Resposta:**
  - **200** - Sucesso;
  - **500 Erro Interno do Servidor** – Erro.

## 2. Criar Usuário:

- **URL:** /api/criar-usuario
- **Método:** POST
- **Corpo:**

```
{
  "email": "qualquercoisa1833@gmail.com",
  "senha": "1234",
  "nome_usuario": "Seu nome",
  "cpf": "12345678902",
  "telefone": "1234567891",
  "endereco": "Rua Exemplo, 123",
  "dataNascimento": "1992-02-02",
  "sexo": "M"
}
```
- **Resposta:**
  - **200** – Usuário criado com sucesso;
  - **400 Solicitação Inválida** – Campos obrigatórios ausentes;
  - **500 Erro Interno do Servidor** - Mensagem de erro.

## 3. Login:

- **URL:** /api/login
- **Método:** POST

- **Corpo:**

```
{
  "email": "projectsgithub860@gmail.com",
  "senha": "1234"
}
```
- **Resposta:**
  - **200** – Login efetuado com sucesso;
  - **400 Solicitação Inválida** – Campos obrigatórios ausentes;
  - **500 Erro Interno do Servidor** - Mensagem de erro.

#### 4. Logout de Usuário:

- **URL:** /api/logout
- **Método:** POST
- **Autenticação:** Requerida
- **Resposta:**
  - **200** – Logout efetuado com sucesso;
  - **500 Erro Interno do Servidor** - Mensagem de erro.

#### 5. Requisitar Troca de Senha:

- **URL:** /api/requisitar-troca-senha
- **Método:** POST
- **Corpo:**

```
{
  "email": "qualquercoisa1833@gmail.com"
}
```
- **Resposta:**
  - **200** – Token de troca de senha enviado para o e-mail;
  - **400 Solicitação Inválida** - Campos obrigatórios ausentes;
  - **500 Erro Interno do Servidor** - Mensagem de erro.

#### 6. Redefinir Senha:

- **URL:** /api/redefinir-senha
- **Método:** POST

- **Corpo:**

```
{
  "email": "qualquercoisa1833@gmail.com",
  "nova_senha": "123456",
  "token": "pyGlc7savbE"
}
```
- **Resposta:**
  - **200** – Senha redefinida com sucesso;
  - **400 Solicitação Inválida** - Campos obrigatórios ausentes;
  - **500 Erro Interno do Servidor** - Mensagem de erro.

#### 7. Atualizar Senha para Usuário ou Admin:

- **URL:** /api/atualizar-para-usuario-ou-admin/:id\_usuario/:admin
- **Método:** PUT/
- **Autenticação:** Requerida
- **Resposta:**

#### 8. Atualizar Dados:

- **URL:** /api/atualizar-dados/:id\_usuario
- **Método:** PUT
- **Autenticação:** Requerida
- **Corpo:**

```
{
  "novo_email": "",
  "nova_senha": "",
  "novo_telefone": "",
  "novo_endereco": ""
}
```
- **Resposta:**

## 2.2. Rotas de Vagas

### 1. Pegar Setores dos Cargos:

- **URL:** /api/pegar-setores-cargos
- **Método:** GET
- **Autenticação:** Requerida
- **Resposta:**
  - **200** – Setores e Cargos recuperados com sucesso;
  - **500 Erro Interno do Servidor** - Mensagem de erro.

## 2. Pegar todas as Vagas:

- **URL:** /api/pegar-todas-vagas
- **Método:** GET
- **Autenticação:** Requerida
- **Resposta:**
  - **200** – Sucesso;
  - **500 Erro Interno do Servidor** - Mensagem de erro.

## 3. Criar Vaga:

- **URL:** /api/criar-vaga
- **Método:** POST



- **Autenticação:** Requerida

- **Corpo:**

```
{  
  "setor": "TI",  
  "cargo": "ESTÁGIO TI",  
  "nome_vaga": "Vaga de estágio para TI",  
  "status": "ABERTA",  
  "descricao_vaga": "É uma vaga dedica para o setor de TI, buscamos profissionais capacitados",  
  "salario": 1770.77,  
  "quantidade_vagas": 5,  
  "data_encerramento": "2024-09-21"  
}
```

- **Resposta:**

- **200** – Vaga criada com sucesso;
- **400 Solicitação Inválida** - Campos obrigatórios ausentes;
- **500 Erro Interno do Servidor** - Mensagem de erro.

#### 4. Atualizar Vaga:

- **URL:** /api/atualizar-vaga/:id
- **Método:** PUT

- **Autenticação:** Requerida

- **Corpo:**

```
{  
  "nome_vaga": "Vaga de estágio para TI",  
  "status": "ABERTA",  
  "descricao_vaga": "É uma vaga dedica para o setor de TI, buscamos profissionais  
capacitados",  
  "salario": 1770.77,  
  "quantidade_vagas": 5,  
  "data_encerramento": "2024-09-20"  
}
```

- **Resposta:**

- **200** – Vaga atualizada com sucesso;
- **400 Solicitação Inválida** - Campos obrigatórios ausentes;
- **500 Erro Interno do Servidor** - Mensagem de erro.

#### 5. Atualizar Status Vaga:

- **URL:** /api/atualizar-status-vaga/:id
- **Método:** PUT

- **Autenticação:** Requerida

- **Corpo:**

```
{  
  "nome_vaga": "Vaga de estágio para TI",  
  "status": "ABERTA",  
  "descricao_vaga": "É uma vaga dedica para o setor de TI, buscamos profissionais  
    capacitados",  
  "salario": 1770.77,  
  "quantidade_vagas": 5,  
  "data_encerramento": "2024-09-20"  
}
```

- **Resposta:**

- **200** – Vaga atualizada com sucesso;
- **400 Solicitação Inválida** - Campos obrigatórios ausentes;
- **500 Erro Interno do Servidor** - Mensagem de erro.

## 2.3 Rotas de Inscrição

### 1. Pegar Currículo:

- **URL:** /api/pegar-curriculo/:id\_usuario
- **Método:** GET
- **Autenticação:** Requerida
- **Resposta:**
  - **500 Erro Interno do Servidor** - Mensagem de erro.

### 2. Salvar Inscrição de Currículo:

- **URL:** /api/salvar-inscricao-curriculo/:id\_usuario/:id\_vaga
- **Método:** POST

- **Autenticação:** Requerida
- **Resposta:**
  - **200** – Inscrição salva com sucesso e currículo salvo;
  - **500 Erro Interno do Servidor** - Mensagem de erro;
  - **500 Erro Interno do Servidor** - Mensagem de erro.

### 3. Mostrar Inscrições de Usuário:

- **URL:** /api/mostrar-inscricoes-usuario/:id\_usuario
- **Método:** GET
- **Autenticação:** Requerida
- **Resposta:**

### 4. Forms Respondido:

- **URL:** /api/forms-respondido/:id\_status\_processo\_seletivo
- **Método:** PUT
- **Autenticação:** Requerida
- **Resposta:**

## 2.4 Rotas de Status do Processo Seletivo

### 1. Pegar Etapa das Vagas:

- **URL:** /api/pegar\_vagas\_etapas
- **Método:** GET
- **Autenticação:** Requerida
- **Resposta:**

### 2. Pegar Todos Status do Processo Seletivo:

- **URL:** /api/pegar\_todos\_status\_processo\_seletivo
- **Método:** GET

- **Autenticação:** Requerida
- **Resposta:**

### 3. Pegar Todos Usuários:

- **URL:** /api/pegar\_todos\_usuarios
- **Método:** GET
- **Autenticação:** Requerida
- **Resposta:**

### 4. Salvar Status do Processo Seletivo:

- **URL:** /api/salvar\_status\_processo\_seletivo
- **Método:** POST
- **Autenticação:** Requerida
- **Corpo:**

```
{  
  "vaga": "Vaga de estágio para TI",  
  "etapa": "FORMS INICIAL",  
  "status_processo": "EM ANDAMENTO",  
  "forms_respondido": false  
}
```
- **Resposta:**

### 5. Atualizar Status do Processo Seletivo:

- **URL:** /api/atualizar\_status\_processo\_seletivo/:id\_status\_processo\_seletivo
- **Método:** PUT

- **Autenticação:** Requerida

- **Corpo:**

```
{  
  "nome_usuario": "Seu nome",  
  "email_usuario": "qualquercoisa1833@gmail.com",  
  "data_entrevista": "0000-00-00 00:00:00",  
  "data_conclusao": "0000-00-00 00:00:0",  
  "status_processo": "ATIVO",  
  "perfil": "",  
  "observacao": "",  
  "forms_respondido": false,  
  "avaliacao_forms": "IDEAL"  
}
```

- **Resposta:**

### 3. Observações finais

Algumas observações finais vão para as opções do CORS, github do projeto e

alguns sites de ajuda para Methods HTTP e Status HTTP:

### **CORS Options:**

As permissões de CORS estão configuradas para permitir solicitações do <http://localhost:3000> (pode ser mudado para outras origens) com credenciais:

```
CORS(app, resources={ r"/" : { "origins" : "" } })
```

### **Projeto no GitHub:**

Para acessar o repositório do projeto no GitHub, visite [Project-Selection-Process-Control](#).

### **Mais Informações:**

Para mais informações sobre os status HTTP, consulte [MDN HTTP Status](#).

Para mais informações sobre os métodos HTTP, consulte [MDN HTTP Methods](#).