

INSTRUÇÕES PARA EXECUÇÃO

Requisitos:

Java-jdk 11

Banco de dados: Postgresql 11.14

Inicio:

Banco de dados Postgres:

username=**postgres**

password=**postgres**

Criar uma base de dados com nome de RH.

Subir a aplicação com o comando:

```
Java -jar regjob-api-0.0.1-SNAPSHOT.jar
```

Para Testar os EndPoints gerar um token jwt na seguinte URL;

<http://localhost:8080/kaspper/oauth/token>

Obs: caso for testar com o Postman Authorization Basic Auth:

Username: angular

Password: @ngul@r

▶ NOVOTOKEN Examples 0 ▾ BUILD

POST ▾

http://localhost:8080/kaspper/oauth/token

Send ▾

Params Authorization ● Headers (10) Body ● Pre-request Script Tests Settings C

TYPE

Basic Auth ▾

The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)

Username

angular

Password

@ngul@r

☒ Show Password

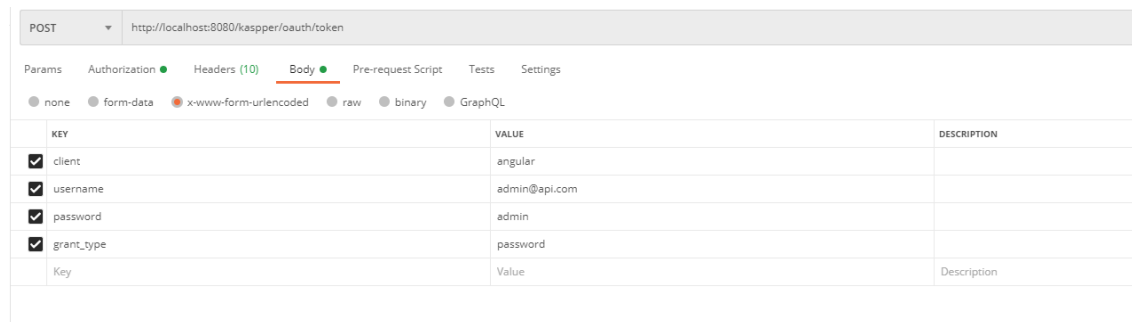
Body:

Client: angular

Username: admin@api.com (este usuário é usuário do banco de dados da tabela usuario)

Password: admin

Grant_type: password



POST http://localhost:8080/kaspper/oauth/token

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

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

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> client	angular	
<input checked="" type="checkbox"/> username	admin@api.com	
<input checked="" type="checkbox"/> password	admin	
<input checked="" type="checkbox"/> grant_type	password	
Key	Value	Description

Para cadastrar uma vaga nova utilizar o endpoint.

localhost:8080/kaspper/vaga/cadastrar

authorization Token: (utilizar o token gerado no endpoint acima)

no Body utilizar o json abaixo:

```
{
  "descricao": "desenvolvedor back-end",
  "tipo": "CLT",
  "salario": 1200.00
}
```

Gerar uma Lista de todas as vagas:

Authorization token:

localhost:8080/kaspper/vaga/listar

Para cadastrar Candidato:

localhost:8080/kaspper/candidato/cadastrar

no Body utilizar o json abaixo:

```
{
  "nome": "Fulano da silva",
  "perfil": "JUNIOR",
  "anexo": null,
  "urlAnexo": null,
  "salario": 10000.00
}
```

Obs: O upload de currículo como foi solicitado foi implementado com uma conta no servidor S3 da amazon, para testar o usuário deverá ter uma conta na Amazon e colocar suas credenciais no application.yml da aplicação, da seguinte forma:

regjobapi.s3.access-key-id= [sua_key_id](#)
regjobapi.s3.secret-access-key= [seu_access_key](#)

Authorization: Token

localhost:8080/kaspper/candidato/anexo

Para gerar uma lista de todos os Candidatos:

Authorization token:

localhost:8080/kaspper/candidato/listar

Cadastrar Candidato em uma vaga:

localhost:8080/kaspper/candidatovaga/cadastrar

json:

```
{
  "candidato":{
    "id":1
  },
  "vaga":{
    "id":1
  }
}
```

Pesquisar todos os candidatos de uma determinada vaga pelo Id da vaga:

Authorization: Token

localhost:8080/kaspper/candidatovaga/pesquisar/vagalId/1

Pesquisar todas as vagas de um determinado candidato, pelo id do candidato:

Authorization: Token

localhost:8080/kaspper/candidatovaga/pesquisar/candidatId/1