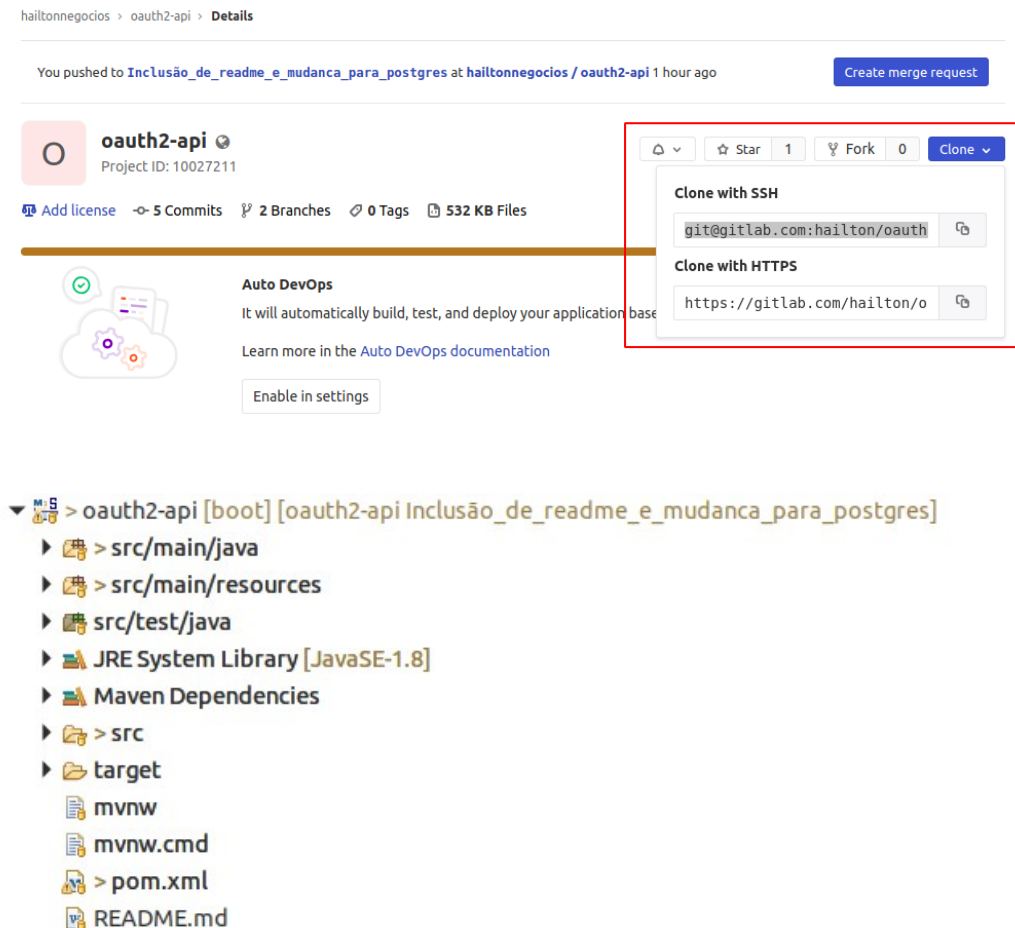


Tutorial para executar o projeto demo

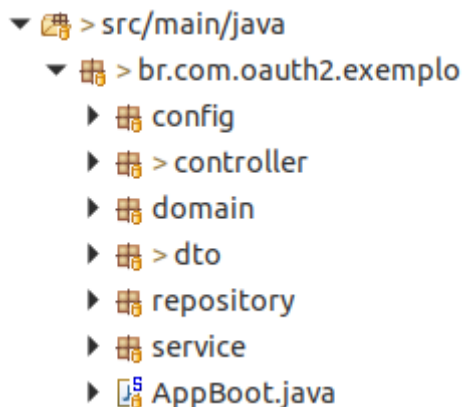
1) Clonar o projeto na URL abaixo :

<https://gitlab.com/hailton/oauth2-api>

2) Após clonar o projeto, incluir o projeto no Eclipse ou Spring Tool Suite

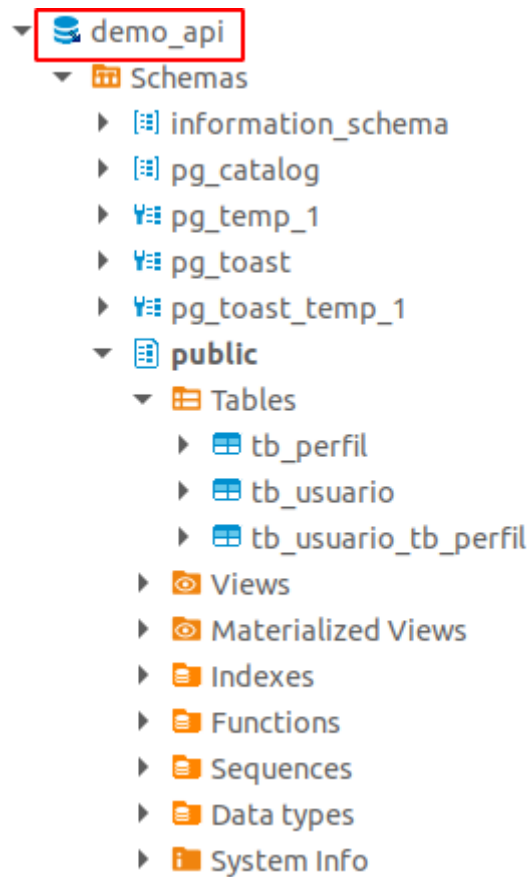


Estrutura de pastas do projeto :

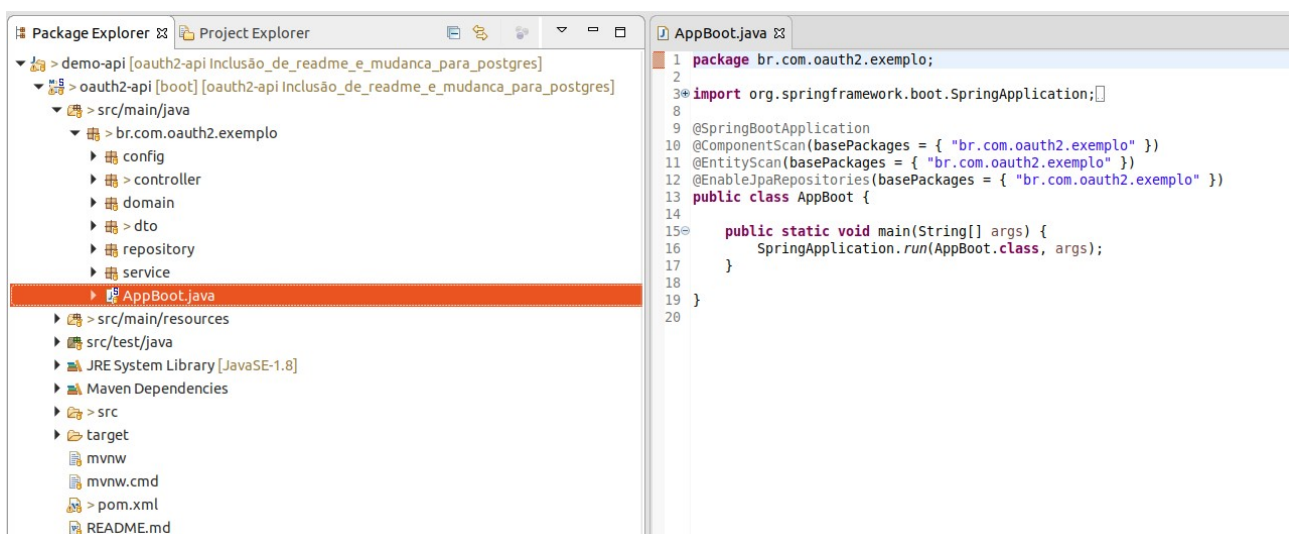


3) Após importar o projeto na IDE, criar um banco no postgres com o nome “demo-api”

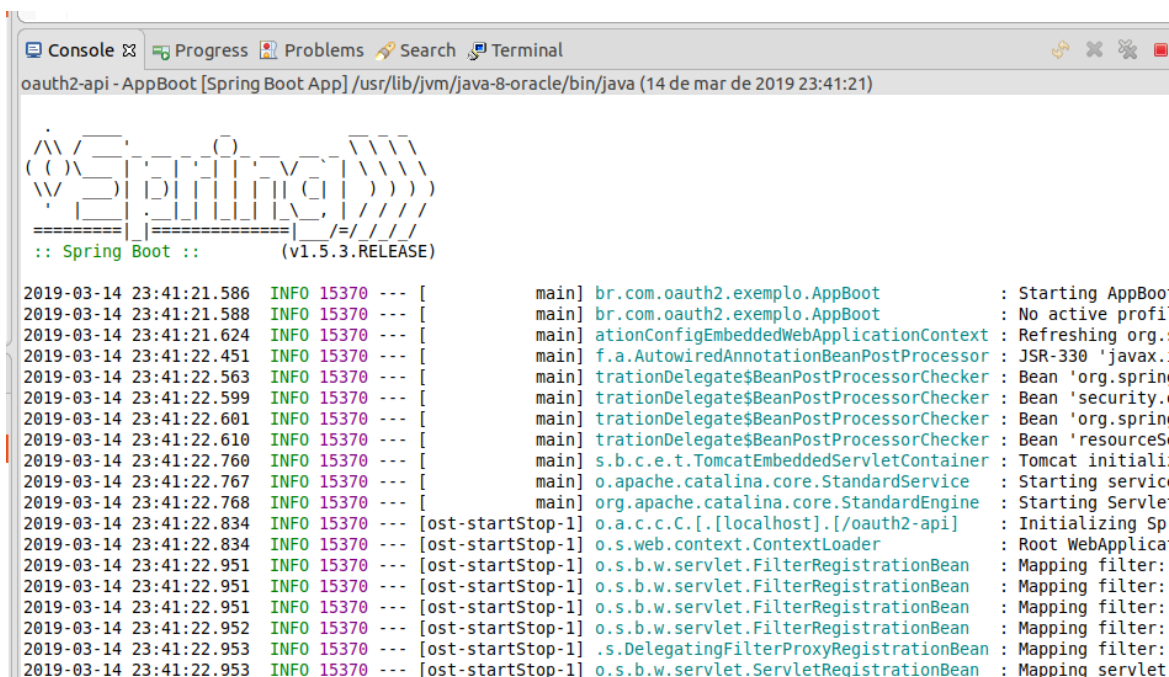
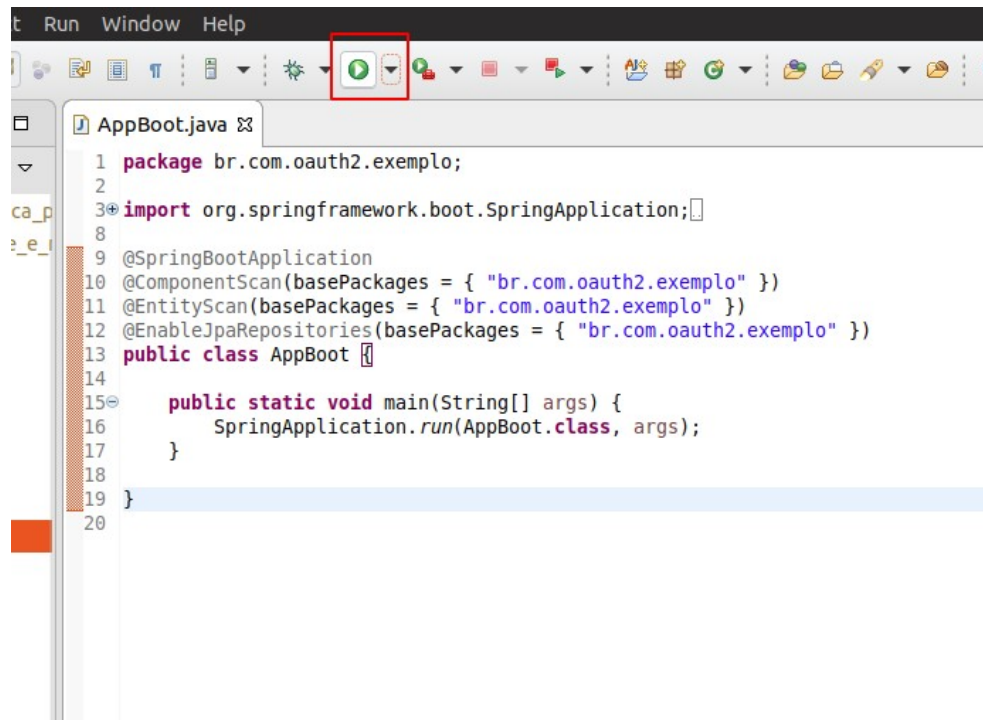
Obs : Não precisa criar as tabelas, pois o Hibernate criará para você ao startar a aplicação



4) Após a importação do projeto e a criação do banco de dados, você já pode rodar a classe do Springboot para startar a aplicação :

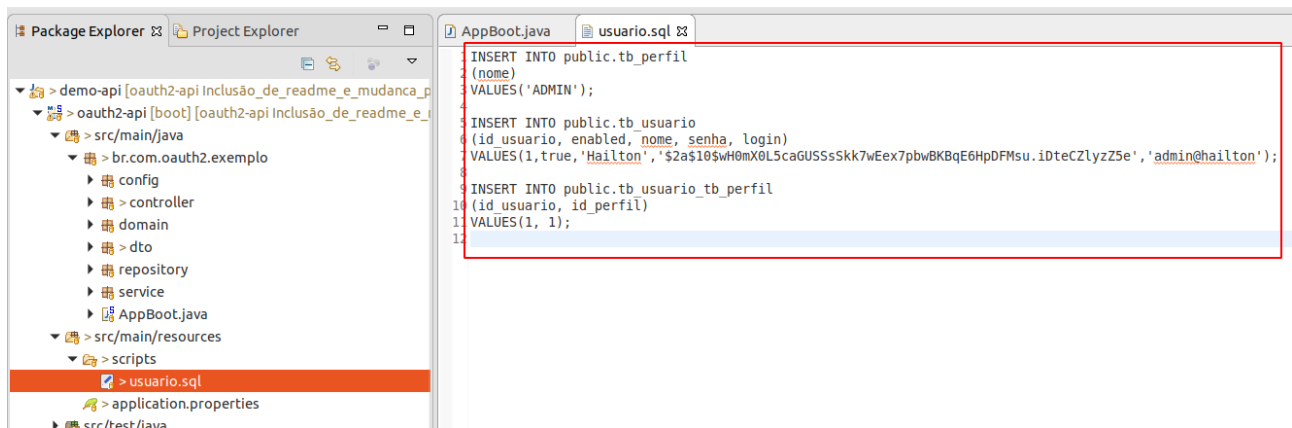
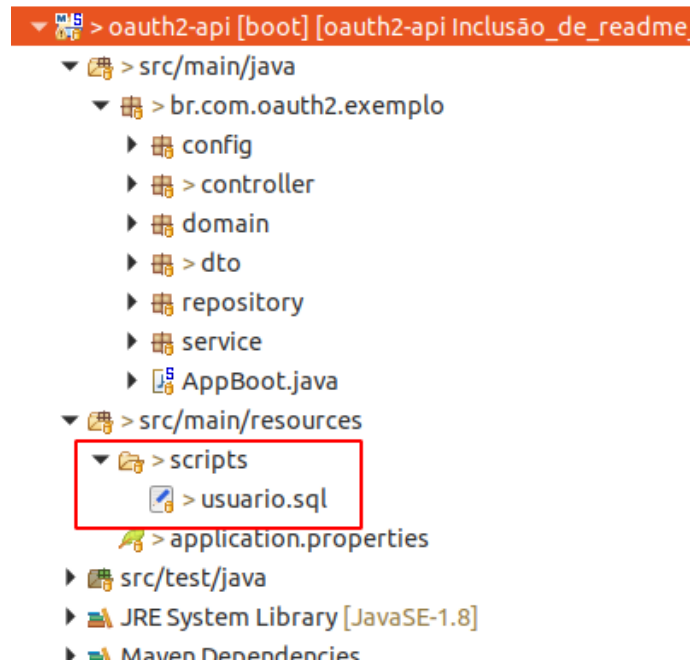


5) Para executar basta colocar o cursor do mouse sobre a classe e rodar o Run as ou clicar no botão superior com o ícone verde para iniciar :



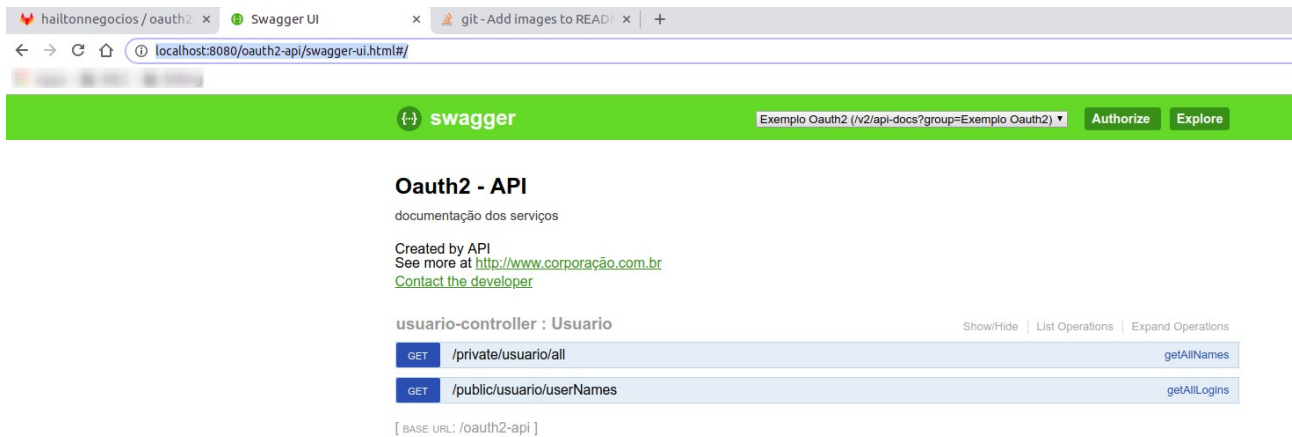
Execute o script dentro do projeto após o start da aplicação, nesse momento as tabelas já serão criadas no banco.

Se estiver criado, execute o script demo para ter alguma carga no banco:



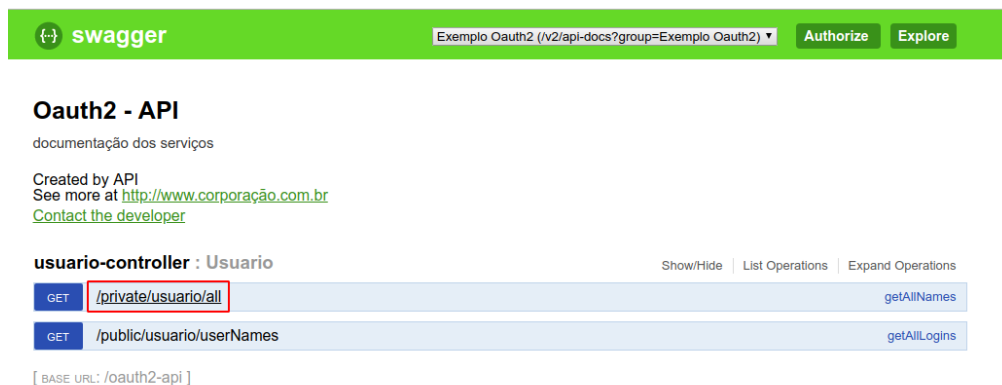
6) Após executar os passos anteriores acesse a URL abaixo :

<http://localhost:8080/oauth2-api/swagger-ui.html>



7) No projeto foi incluído dois tipos de endpoint, um privado com autenticação OAuth2 e outro publico sem a necessidade de autenticação.

Endpoint Privado :



Como é privado, click no botão Authorize para incluir as credenciais

Exemplo OAuth2 (/v2/api-docs?group=Exemplo OAuth2) **Authorize** Explore

Available authorizations

OAuth2.0

Token URL: /oauth2-api/oauth/token
flow: password

Please input username and password for password flow authorization

Username:
Password:

Setup client authentication.

Type: **Basic auth** ▼
ClientId:
Secret:

API requires the following scopes. Select which ones you want to grant to Swagger UI.

Scopes are used to grant an application different levels of access to data on behalf of the end user. Each API may declare one or more scopes. [Learn how to use](#)

☐ read
read only

☐ write
read and write

Authorize
Cancel

Digite as credenciais demos abaixo :

Essa primeira credencia é o usuario criado no banco de dados

username: [admin@hailton](#)

password : 123456

Essa segunda credencial esta no arquivo de propertie dentro do projeto

clientId : admin

secret : 123456

read : X


write: X

Após a autenticação o botão ficará azul, assim você poderá executar o consumo :

usuario-controller : Usuario

Show/Hide | List Operations | Expand Operations

GET /private/usuario/all getAllNames

Response Class (Status 200)
OK 

Model | **Example Value**

```
[
  {
    "nome": "string",
    "userName": "string"
  }
]
```

Response Content Type */* ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

Retorno da api private / usuario / all

Response Body

```
[
  {
    "nome": "Hailton",
    "userName": "admin"
  }
]
```

Endpoint Publico

Oauth2 - API

documentação dos serviços

Created by API
See more at <http://www.corporação.com.br>
[Contact the developer](#)

usuario-controller : Usuario

Show/Hide | List Operations | Expand Operations

GET	/private/usuario/all	getAllNames
GET	/public/usuario/userNames	getAllLogins

Response Class (Status 200)

OK

Model | Example Value

```
[  
  "string"  
]
```

Response Content Type */* ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

Como é publico então não precisa de autenticação, basta clicar direto no botão try it out

GET /public/usuario/userNames getAllLogins

Response Class (Status 200)
OK

Model | Example Value

```
[  
  "string"  
]
```

Response Content Type */* ▼

Response Messages

HTTP Status Code	Reason	Response Model	Headers
401	Unauthorized		
403	Forbidden		
404	Not Found		

Try it out!

Retorno da api /public / usuario / userNames

Response Body

```
[  
  "admin"  
]
```

Response Code