Usando o Client Credentials

Usando o Client Credentials

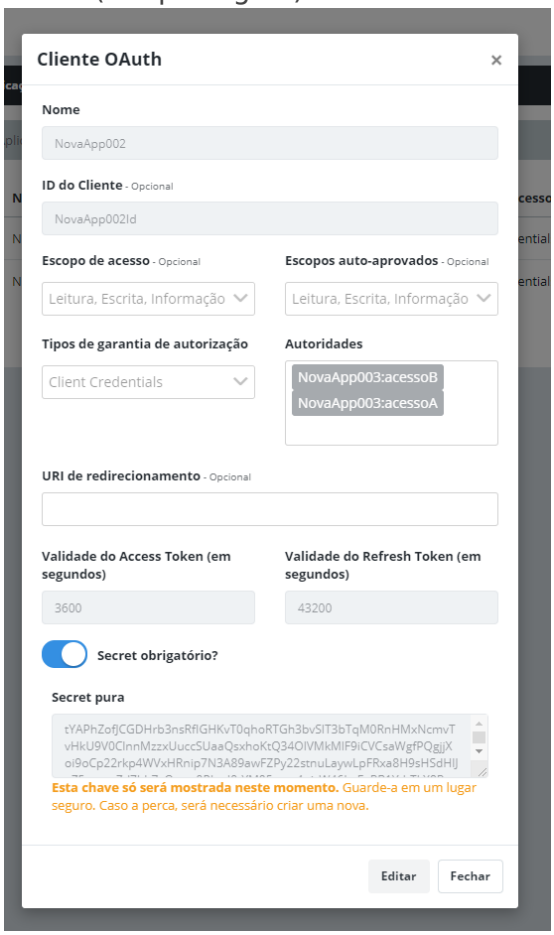
Usando o Client Credentials

Neste tutorial vamos aprender o seguinte:

1. Configuração no Cerberus (Authorization Server);
2. Obtendo um token para acessar a API protegida (Client);
3. Criando uma API protegida (Resource Server);

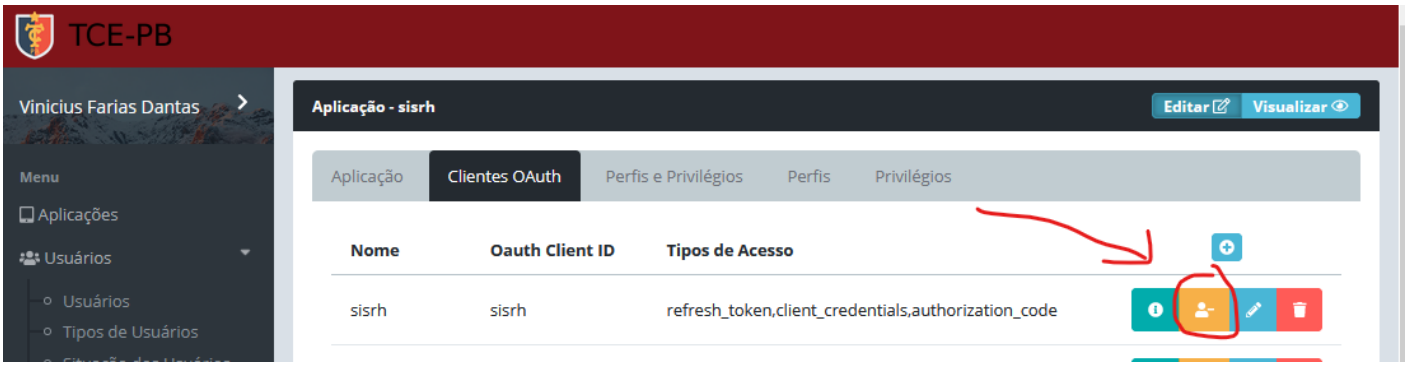
Configuração no Cerberus (Authorization Server)

1. No Cerberus, logado como Admin, vá em Aplicações;
2. Cadastre uma nova aplicação, por exemplo, nome "NovaApp001", descrição "Aplicação para testar o Client Credentials Grant Type";
3. Ainda no cadastro da nova aplicação, clique no botão Editar, em seguida vá na aba Clientes OAuth, e clique no botão azul com o símbolo de soma;
4. Preencha o dialog Adicionar Cliente OAuth da maneira abaixo. Perceba que em Tipos de garantia de autorização Client Credentias foi selecionado, e que em Autoridades definimos duas: NovaApp003:acessoA e NovaApp003:acessoB
   Atenção: utilizamos as Autoridades para gerenciar o acesso aos recursos do Resource Sever (API protegida).



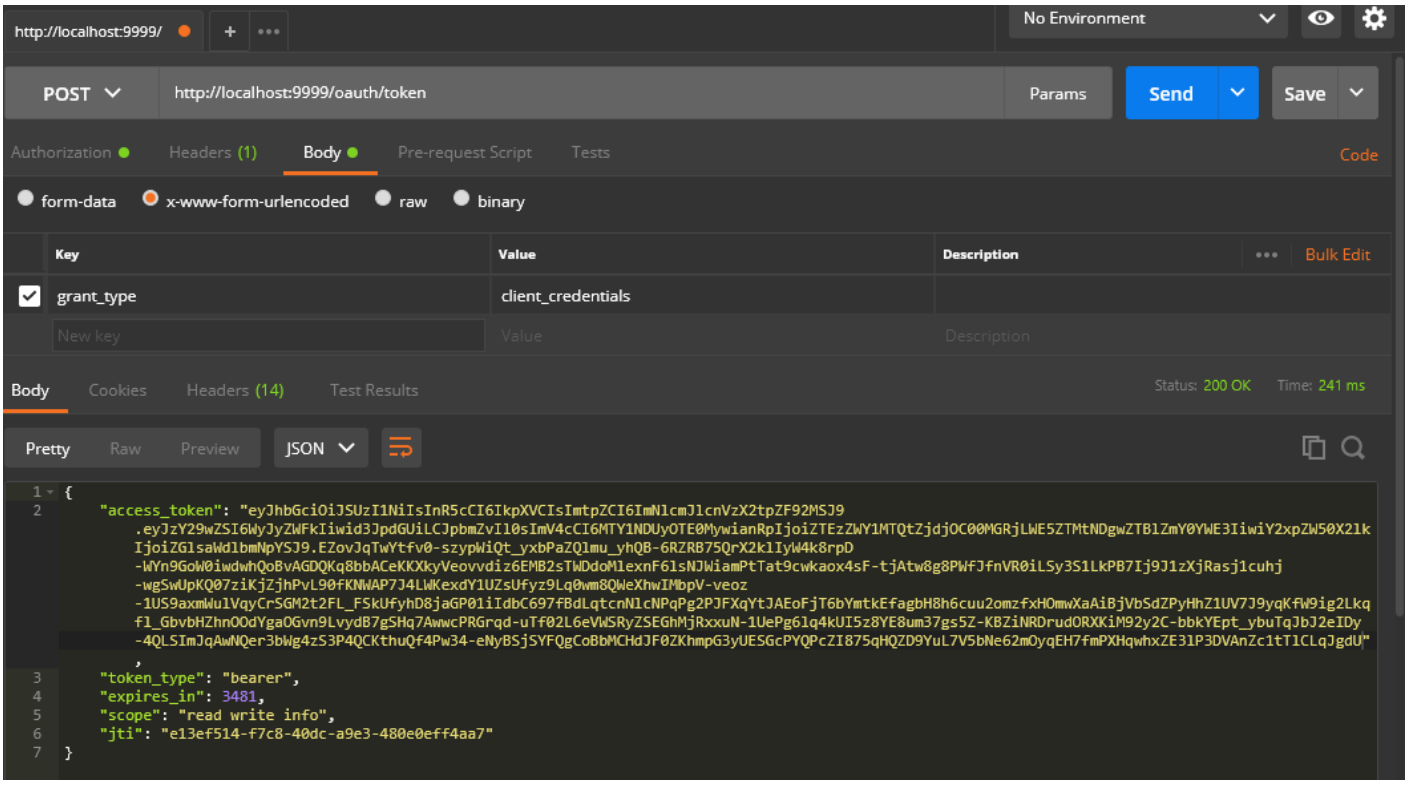
5. Ao clicar no botão Salvar, anote o Secret pura. O ID do Cliente e a Secret pura são o user e o password que usaremos para obter um token.

ATENÇÃO: caso um Cliente OAuth seja editado, lembre-se de apertar no botão *Revogar Tokens da Aplicação* (botão laranja na listagem de *Clientes OAuth* no detalhamento de uma Aplicação), isso irá remover o respectivo token da tabela *oauth_access_token* do banco *cerberus*, para um novo ser gerado no próximo request de token do cliente

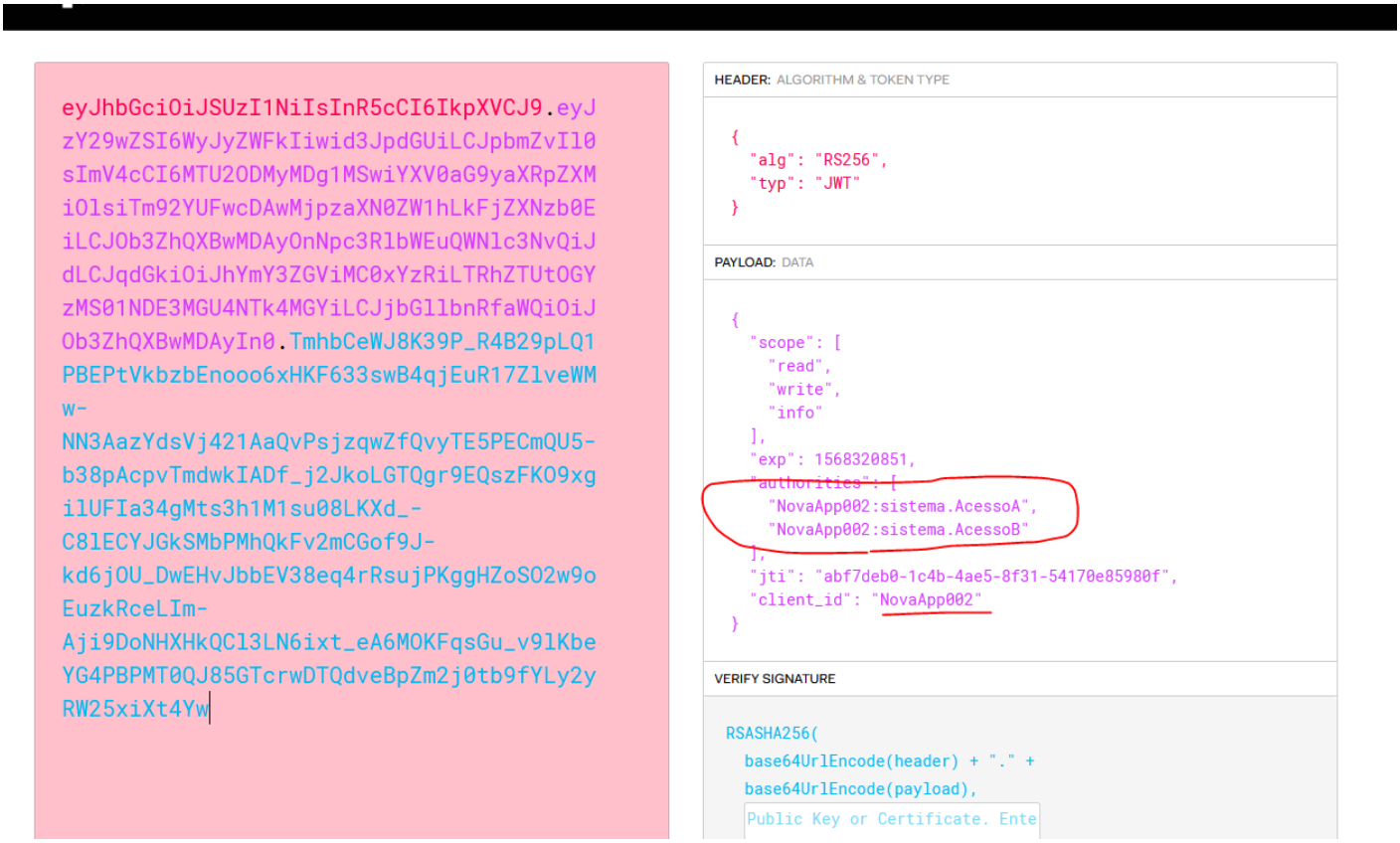


Obtendo um token para acessar a API protegida (Client).

Obtemos um token fazendo um request ao Cerberus, para isso usaremos os seguintes dados: será um POST utilizando o Authorization do tipo Basic Auth usando como Username e Password o ID do Cliente(por exemplo: **diligencia**) e a Secret pura (**é a secret gerada na hora que foi criada ao cadastrar no Cerberus o Client Oauth**). Definimos um header Content-Type=application/x-www-form-urlencoded e como Body definimos grant_type=client_credentials. O retorno dessa request conterá o access token.

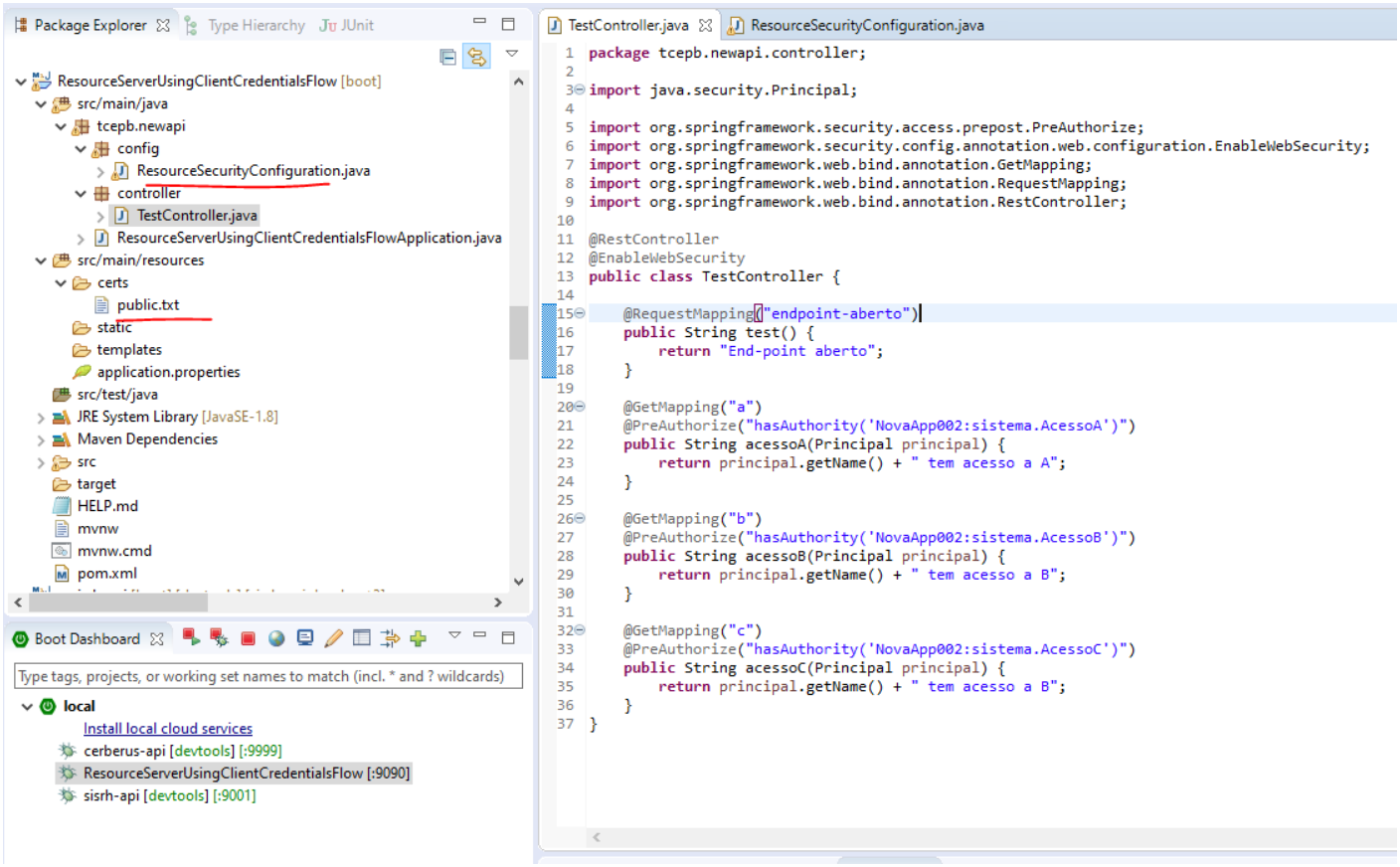


Usando o site JWT para decodificar o token obtido, temos o seguinte:



Criando uma API protegida (Resource Server)

O resource server normalmente é uma API, em anexo temos o boilerplate de uma que pode ser reutilizada como base. Ela basicamente é uma adaptação do sisrh-api, onde a classe de configuração principal é a tcepb.newapi.config.ResourceSecurityConfiguration que basicamente é uma adaptação da classe do sisrh-api tcepb.rh.config.ResourceConfig. Copiamos também a chave publica utilizada sisrh-api, ela esta localizada em src/main/resources/certs/public.txt



Referências:

1. Spring Boot + OAuth 2 Client Credentials Grant - Hello World Example. URL: https://www.javainuse.com/spring/springboot-oauth2-client-grant >. Acessado em: 12/09/2019;