

## *Testando Configurações Spring Security com Tipo de Autenticação oauth2/token com Postman*

### **Gerando o Token**

O método para gerar um Token deverá ser um método POST

The screenshot shows the Postman interface for configuring a POST request. The top bar indicates the method is POST and the URL is localhost:8080/oauth/token. Below this, the 'Authorization' tab is selected, showing the 'Type' set to 'Basic Auth'. The 'Username' field is filled with 'angular' and the 'Password' field is masked with dots. A 'Show Password' checkbox is visible at the bottom.

POST	localhost:8080/oauth/token			
Authorization	Headers (2)	Body	Pre-request Script	Tests
Type	Basic Auth			
Username	angular			
Password	.....			
<input type="checkbox"/> Show Password				

O usuário e senha configurados na guia Authorization é o que está no método abaixo:  
Classe : AuthorizationServerConfig

```
@Override
    public void configure(ClientDetailsServiceConfigurer clients) throws Exception {
        clients.inMemory()
            .withClient("angular")
            .secret("@ngul@r0")
            .scopes("read", "write")
            .authorizedGrantTypes("password")
            .accessTokenValiditySeconds(1800);
    }
```

A guia Headers deverá ficar assim. Isso é gerado automaticamente pelo Postman :

Authorization ●		Headers (2)	Body ●	Pre-request Script	Tests
Key		Value			
<input checked="" type="checkbox"/>	Content-Type	application/x-www-form-urlencoded			
<input checked="" type="checkbox"/>	Authorization	Basic YW5ndWxhcjpAbmd1bEBYMA==			
New key		Value			

## Guia Body

POST ▾	localhost:8080/oauth/token			
Authorization ●	Headers (2)	Body ●	Pre-request Script	Tests
<input type="radio"/> form-data	<input checked="" type="radio"/> x-www-form-urlencoded	<input type="radio"/> raw	<input type="radio"/> binary	
Key	Value			
<input checked="" type="checkbox"/> client	angular			
<input checked="" type="checkbox"/> username	admin			
<input checked="" type="checkbox"/> password	admin			
<input checked="" type="checkbox"/> grant_type	password			
<input type="text" value="New key"/>	<input type="text" value="Value"/>			

O usuário e senha desta tela acima estão na classe :  
Classe : ResourceServerConfig e no método abaixo :

```
@Autowired
public void configure(AuthenticationManagerBuilder auth) throws Exception {
    auth.inMemoryAuthentication().withUser("admin").password("admin").roles("ROLE");
}
```

Ao clicar no botão Send deverá ter a seguinte informação. Este é o token gerado que terá validade pelo tempo conforme configurado.

```
{
  "access_token": "22426dbb-fa96-429a-8e4b-d5b3e3ef2210",
  "token_type": "bearer",
  "expires_in": 1299,
  "scope": "read write"
}
```

Testando o mapeamento /lancamentos que precisa de um Token para ser exibido.

Forma da chamada no Postman

GET ▾	localhost:8080/lancamentos
-------	----------------------------

**Guia Authorization . Pode deixar sem autenticação**

Type	No Auth ▾
------	-----------

## Guia Headers

Authorization	Headers (1)	Body	Pre-request Script	Tests
Key		Value		
<input checked="" type="checkbox"/>	Authorization	bearer 22426dbb-fa96-429a-8e4b-d5b3e3ef2210		
<input checked="" type="checkbox"/>				

Observe acima que o token tem que ser incluso conforme acima:

Se estiver tudo, certo a consulta GET do mapeamento lancamentos irá ser autorizada conforme configurado.

Atenção : Depois da palavra bearer tem um espaço.

Estando tudo certo irá ter um retorno

Vale lembrar que quando se tratar do SpringBoot Versão 2X , existe um outra configuração necessário a se fazer que deverá ter o nome de : OAuthSecurityConfig

Se por algum motivo o Token estiver errado a mensagem abaixo será exibida.

Body

Cookies

Headers (9)


Test Results

Pretty

Raw

Preview

JSON ▾



1 ▾

2

3

4

```
{  
  "error": "invalid_token",  
  "error_description": "Cannot convert access token to JSON"  
}
```