



LAB- OAuth2

Step 1 -- Setting up Authorization Server

**In this step, you will be working on
11-05-oauth2-auth-resource-server-start project**

**In this step , you will be setting up authorization server to generate tokens
we will be using InMemory token store.**

- 1) Open AuthorizationServerConfig.java and complete TODO-1 and TODO-2
- 2) Inside AuthorizationServerConfig.java , write the below code

```
@Autowired
private AuthenticationManager authManager;

@Override
public void configure(AuthorizationServerEndpointsConfigurer endpoints) throws
Exception {
    endpoints.authenticationManager(authManager);
}
```

- 3) Inside AuthorizationServerConfig.java , configure Client details in memory as shown below :

```
@Override
public void configure(ClientDetailsServiceConfigurer clients) throws Exception {
    clients.inMemory().withClient("way2learn").secret("way2learn").authorizedGrantTypes("password").scopes("read,write,trust");
}
```

- 4) Open WebSecurity Config and complete all the TODOs
- 5) configure server.port inside application.properties as 9090

Open Application.java and run it.

- 6) Now open postman and select POST request and give the url as
localhost:8080/oauth/token

Select Basic Auth and give credentials and click on update request.



Select params button and give below request params.

```
grant_type=password  
username=siva  
password=siva
```

Make a request and observe that you get the json which contains access_token

Step 2 -- Setting up Resource Server

**In this step, you will be modifying
11-05-oauth2-auth-resource-server-start project**

- 1) Open ResourceServerConfig.java and complete all TODOs
- 2) Open MyController.java and complete all TODOs
- 3) Now make request to /hello and observe that access is denied .
- 4) Now get the access token by passing admin credentials.
Use this access token to retest /hello.
You should be allowed to access the resource.

Step 3 -- OAuth2 Client

In this step, you will be modifying 11-06-oauth2-client-start project

- 1) Open OAuthClientApplication.java and complete TODO-1, TODO-2 and TODO-3
- 2) Open application.properties and observe that security is disabled for /clienthello
- 3) Observe execute method and its @RequestMapping .

Run the application and give a request to <http://localhost:8080/clienthello>



Step 4 -- Setting up Authorization Server using JDBC Token Store

In this step, you will be working on
11-07-oauth2-auth-server-jdbcstore-start

- 1) Open Schemaold.sql in src/main/resources and observe the DDLs
- 2) Create a database with name oauth and execute the DDLs present in schemaold.sql on that database
- 3) Open application.yml and observe that we have already configured datasource related configuration for you.

4) Open AuthorizationServerConfig.java and complete all the TODOs

Now run AuthServer.java and make a request to get access token to the following URL from POSTMAN.

localhost:9090/oauth/token?grant_type=password&username=admin&password=admin

Make sure that you select Basic Authentication and give username/password as way2learn/way2learn

You should get Access token in json response.

Now go to database and see that there is entry in oauth_access_token table for this generated accesstoken

Step 5 -- Setting up Resource Server using JDBC Token Store

In this step, you will be working on
11-08-oauth2-resourceserver-jdbcstore-start

- 1) When ever request comes to resource server, it expects access_token . If access_token is not present in request, it will not allow access to the resource.

If access_token is present in request, this token has to be validated.

There are 2 options for verifying the tokens.

First Option : Using RemoteTokenServices

You have to configure a bean of type RemoteTokenServices by completing TODO-1 in ResourceServerConfig.java

Configure resourceId for the resource server by completing TODO-2



Start OAuthResourceApplication.java and make a request to **http://localhost:7070/resource/hello?access_token= the generated access token in previous step**

You should be granted access.

Second Option : Pointing the resource server to the same token store as the one used by authorization server

Comment out the tokenService bean and complete TODO -3

Start OAuthResourceApplication.java and make a request to **http://localhost:7070/resource/hello?access_token= the generated access token in previous step**

You should be granted access.

CONGRATULATIONS YOU KNOW HOW TO SECURE YOUR REST API USING OAUTH2