OAUTH-2 SETUP FOR APEX REST API'S

ABSTRACT

Below documents will guide to setup the oauth-2 configuration to secure the rest API.

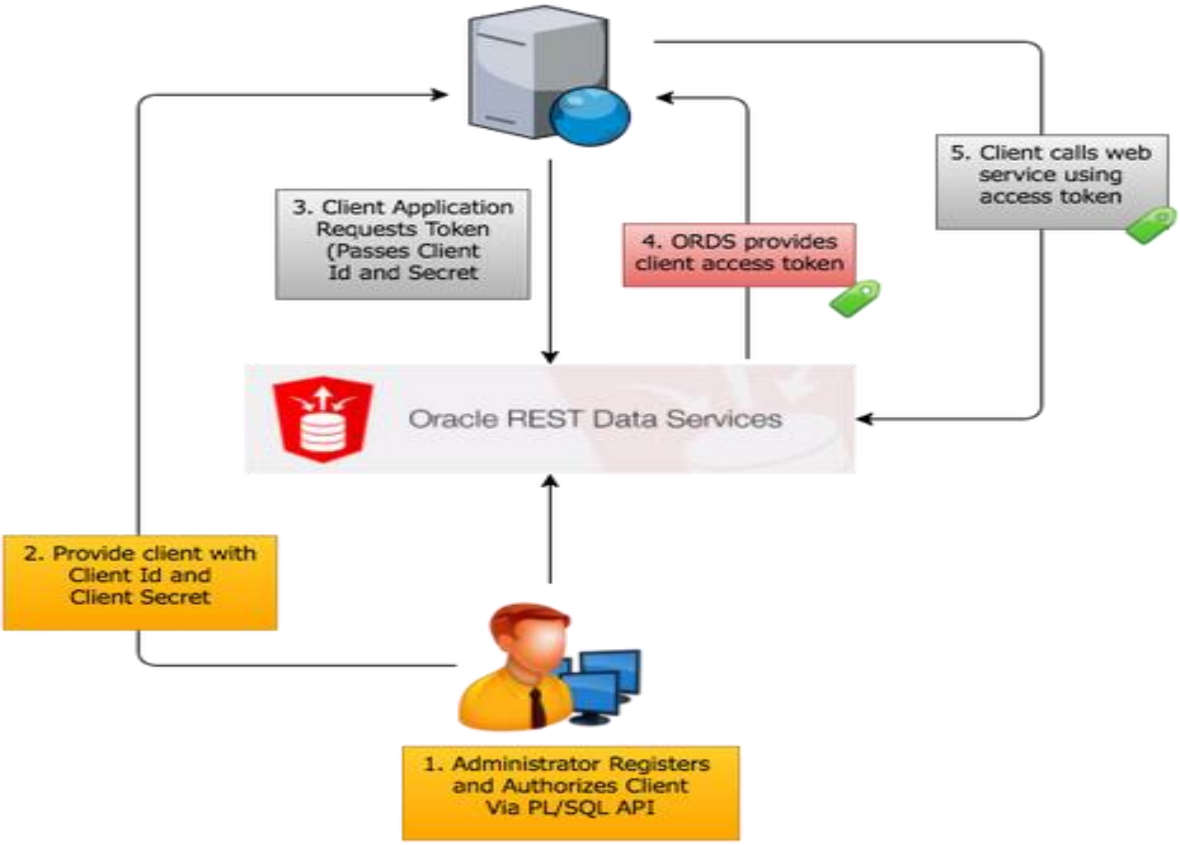
Rajkumar L @ 4i Apps

OAuth: Client Credentials

The client credentials flow is a two-legged process that seems the most natural to me as I mostly deal with server-server communication, which should have no human interaction. For this flow we use the client credentials to return an access token, which is used to authorize calls to protected resources. The example steps through the individual calls, but in reality, it would be automated by the application.

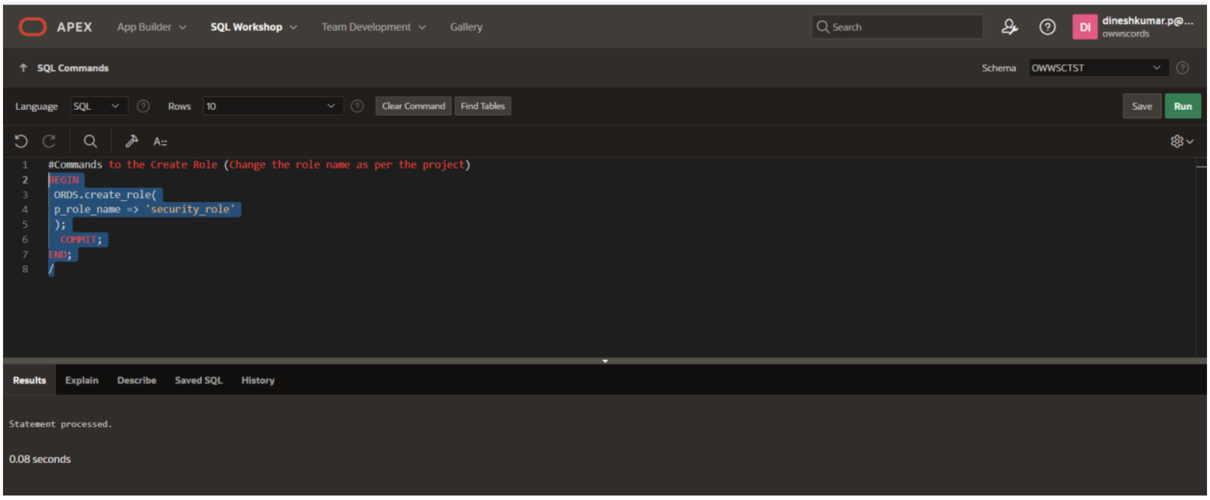
This flow involves the following high level steps:

1. You develop a REST service in ORDS and secure it. You then register a 'client' application with ORDS.
2. You provide the client application (usually to the administrator) the client id and client secret generated from the client registration. The client application stores the client id and client secret securely on their server.
3. The client application makes a call to a special ORDS URL passing their client id and client secret.
4. ORDS responds with an authorization token. By default, the token will remain active for 3,600 seconds (1 Hour).
5. Once the token is obtained, the client calls the appropriate service, passing the token to gain access to the service.


3. Client Application Requests Token (Passes Client Id and Secret
4. ORDS provides client access token
5. Client calls web service using access token
Oracle REST Data Services
2. Provide client with Client Id and Client Secret
1. Administrator Registers and Authorizes Client Via PL/SQL API

By default, a newly created web service is unsecured. Follow these steps to create a privilege and associate it with a service in order to secure it.

- Enabled ORDS in schema 'REST' using ORDS.ENABLE_SCHEMA.
- Created a module called 'hr' using ORDS.DEFINE_MODULE.



```
#Commands to the Create Role (Change the role name as per the project)
BEGIN
 ORDS.create_role(
 p_role_name => 'security_role'
 );
  COMMIT;
END;
/
```

```
Statement processed.

0.08 seconds
```

#Commands to the Create Role (Change the role name as per the project)

BEGIN

 ORDS.create_role(

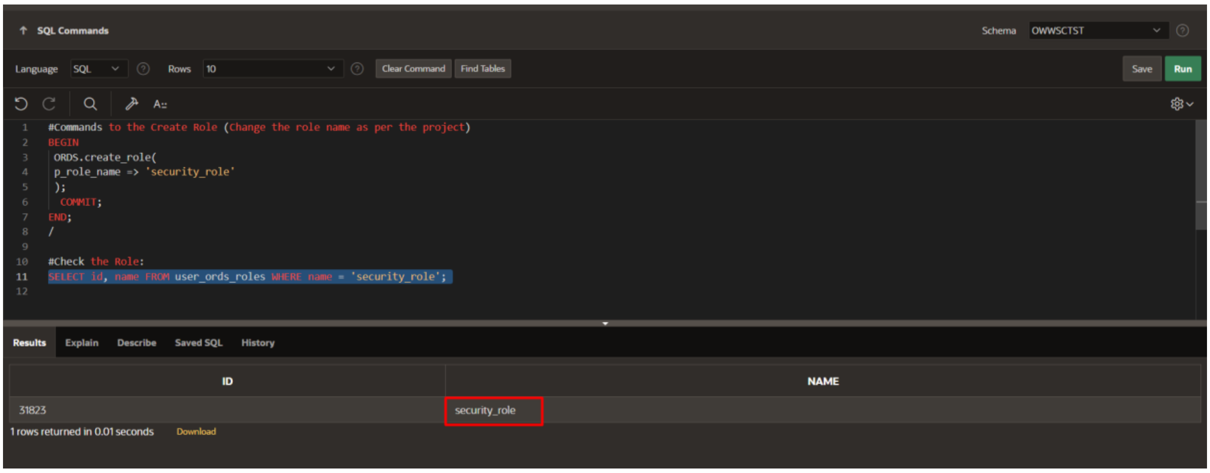
 p_role_name => 'security_role'

 );

  COMMIT;

END;

/



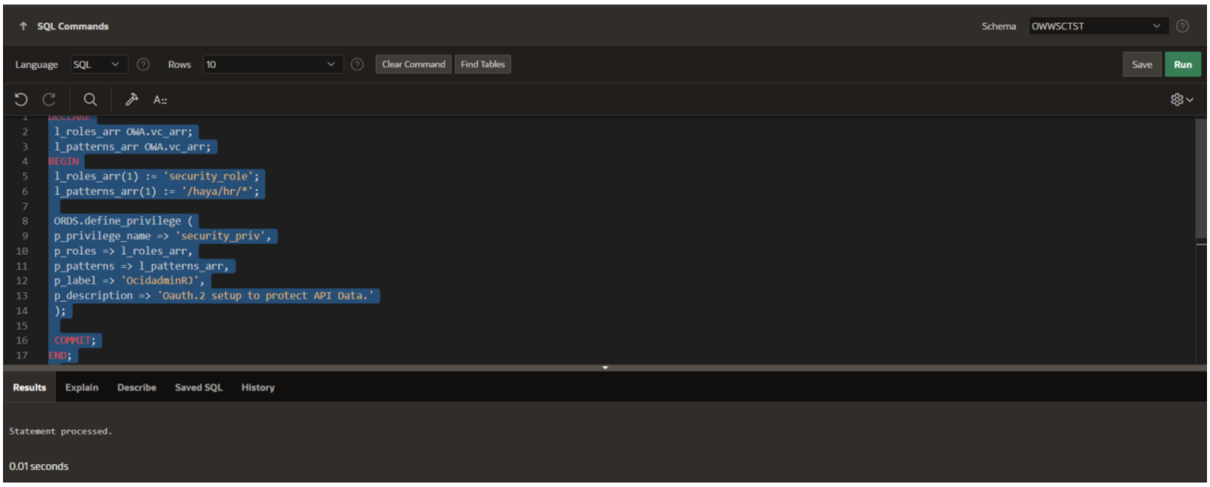
```
#Commands to the Create Role (Change the role name as per the project)
BEGIN
 ORDS.create_role(
 p_role_name => 'security_role'
 );
  COMMIT;
END;
/

#Check the Role:
SELECT id, name FROM user_ords_roles WHERE name = 'security_role';
```

ID	NAME
31823	security_role

1 rows returned in 0.01 seconds

#Check the Role:

```
SELECT id, name FROM user_ords_roles WHERE name = 'security_role';
```

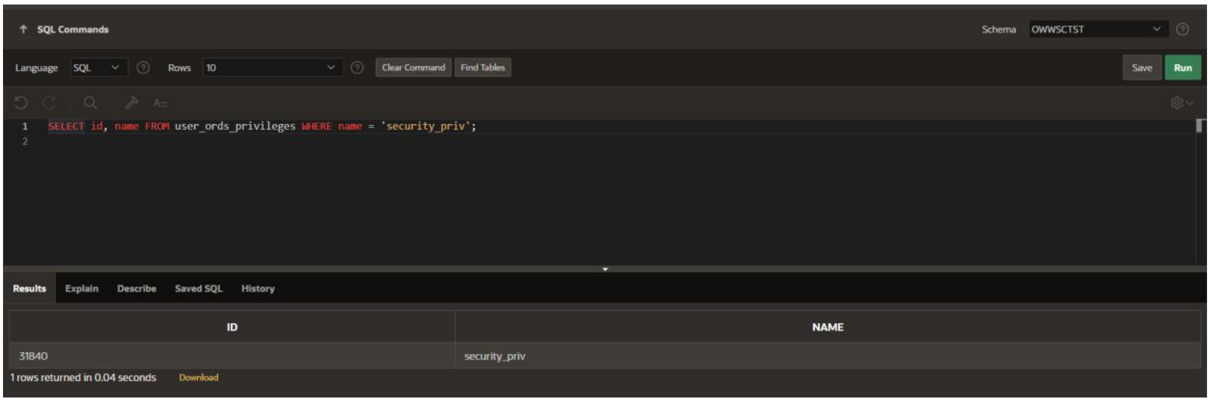


```
DECLARE
 l_roles_arr OWA.vc_arr;
 l_patterns_arr OWA.vc_arr;
BEGIN
 l_roles_arr(1) := 'security_role';
 l_patterns_arr(1) := '/haya/hr/*';

 ORDS.define_privilege (
 p_privilege_name => 'security_priv',
 p_roles => l_roles_arr,
 p_patterns => l_patterns_arr,
 p_label => 'OcidadminRJ',
 p_description => 'Oauth.2 setup to protect API Data.'
 );

 COMMIT;
END;
```

```
Statement processed.

0.01 seconds
```

#Create the Privilege and define the base url pattern to protect and map the privilege to the role using the below command:

```
DECLARE
 l_roles_arr OWA.vc_arr;
 l_patterns_arr OWA.vc_arr;
BEGIN
 l_roles_arr(1) := 'security_role';
 l_patterns_arr(1) := '/haya/hr/*';
 ORDS.define_privilege (
 p_privilege_name => 'security_priv',
 p_roles => l_roles_arr,
 p_patterns => l_patterns_arr,
 p_label => 'OcidadminRJ',
 p_description => 'Oauth.2 setup to protect API Data.'
 );

 COMMIT;
END;
/
```



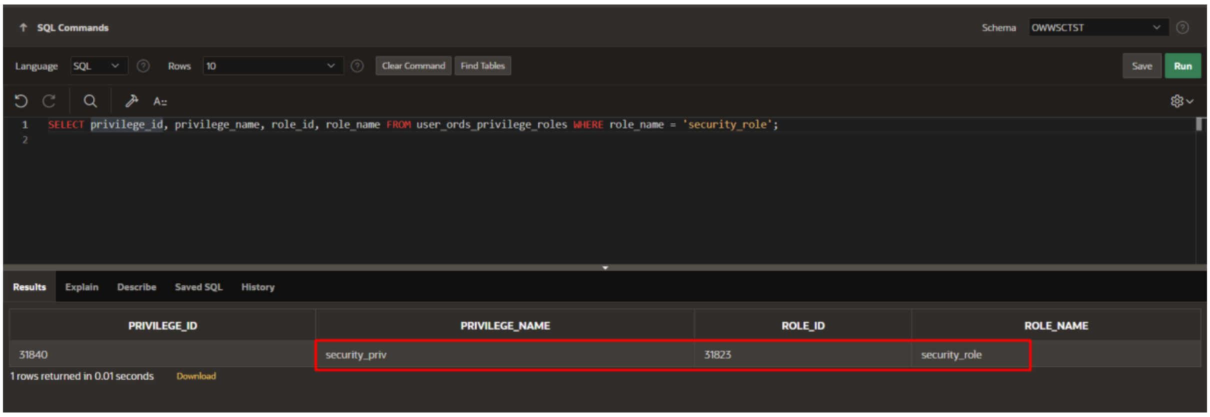
```
SELECT id, name FROM user_ords_privileges WHERE name = 'security_priv';
```

ID	NAME
31840	security_priv

1 rows returned in 0.04 seconds

#Check the Privilege

SELECT id, name FROM user_ords_privileges WHERE name = 'security_priv';



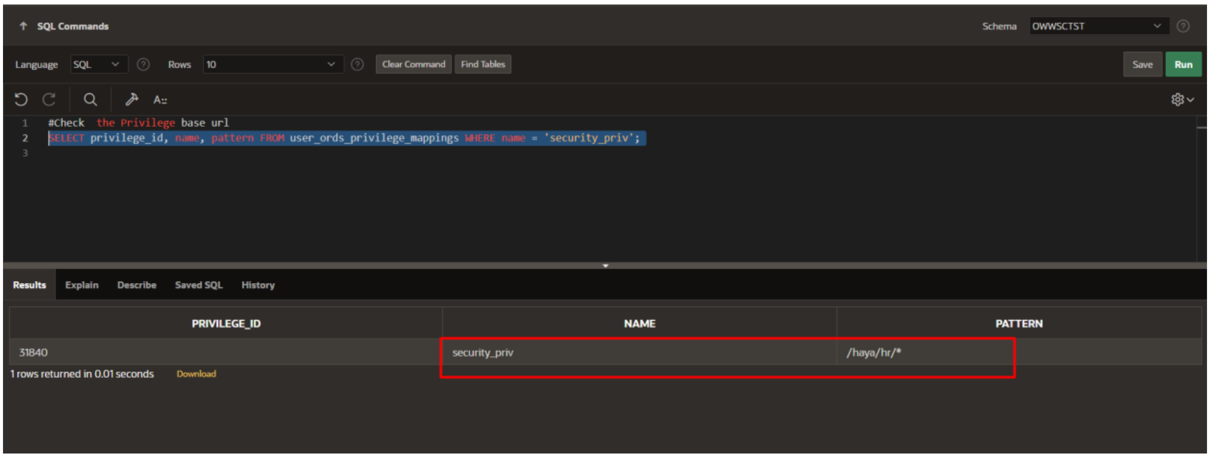
```
SELECT privilege_id, privilege_name, role_id, role_name FROM user_ords_privilege_roles WHERE role_name = 'security_role';
```

PRIVILEGE_ID	PRIVILEGE_NAME	ROLE_ID	ROLE_NAME
31840	security_priv	31823	security_role

1 rows returned in 0.01 seconds

#Check the Privilege and role mapping:

SELECT privilege_id, privilege_name, role_id, role_name FROM user_ords_privilege_roles WHERE role_name = 'security_role';



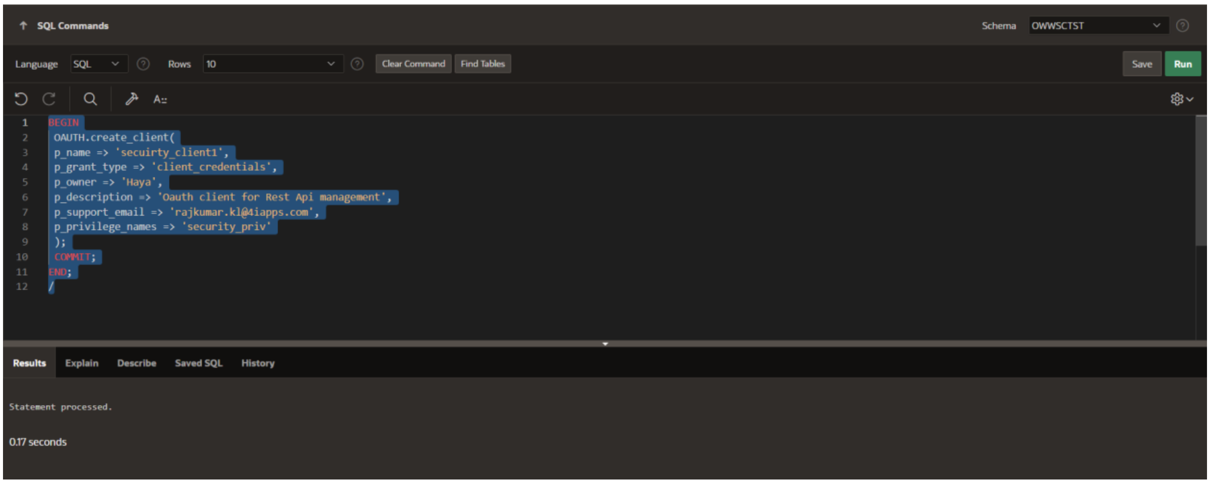
```
#Check  the Privilege base url
SELECT privilege_id, name, pattern FROM user_ords_privilege_mappings WHERE name = 'security_priv';
```

PRIVILEGE_ID	NAME	PATTERN
31840	security_priv	/haya/hr/*

1 rows returned in 0.01 seconds

#Check the Privilege base url

SELECT privilege_id, name, pattern FROM user_ords_privilege_mappings WHERE name = 'security_priv';

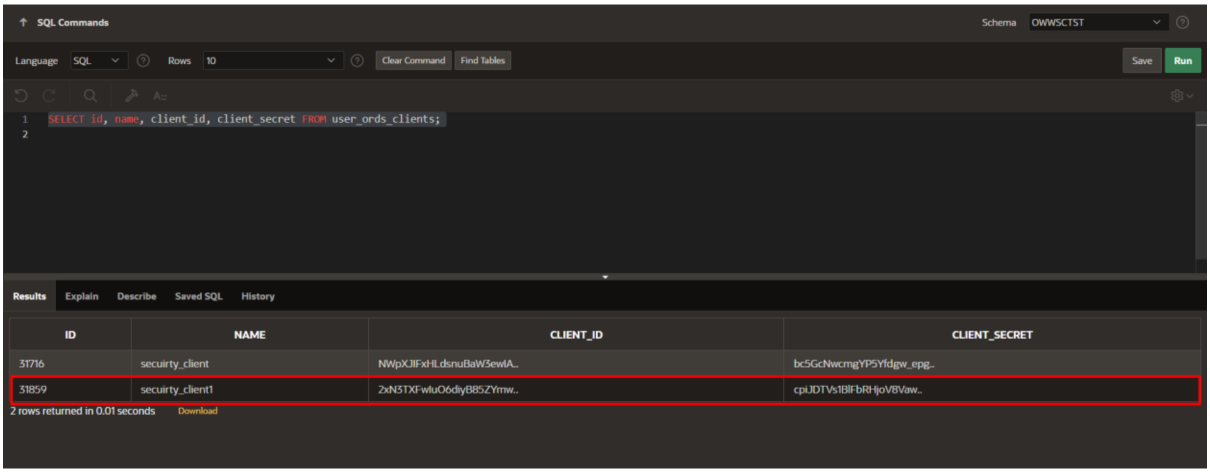


```
BEGIN
 OAUTH.create_client(
 p_name => 'secuirty_client1',
 p_grant_type => 'client_credentials',
 p_owner => 'Haya',
 p_description => 'Oauth client for Rest Api management',
 p_support_email => 'rajkumar.kl@4iapps.com',
 p_privilege_names => 'security_priv'
 );
 COMMIT;
END;
/
```

```
Statement processed.

0.17 seconds
```

#Check the Oauth client:

```
BEGIN
 OAUTH.create_client(
 p_name => 'secuirty_client',
 p_grant_type => 'client_credentials',
 p_owner => 'Haya',
 p_description => 'Oauth client for Rest Api management',
 p_support_email => 'rajkumar.kl@4iapps.com',
 p_privilege_names => 'security_priv'
 );
 COMMIT;
END;
/
```



Schema: OWWSCTST

Language: SQL

Rows: 10

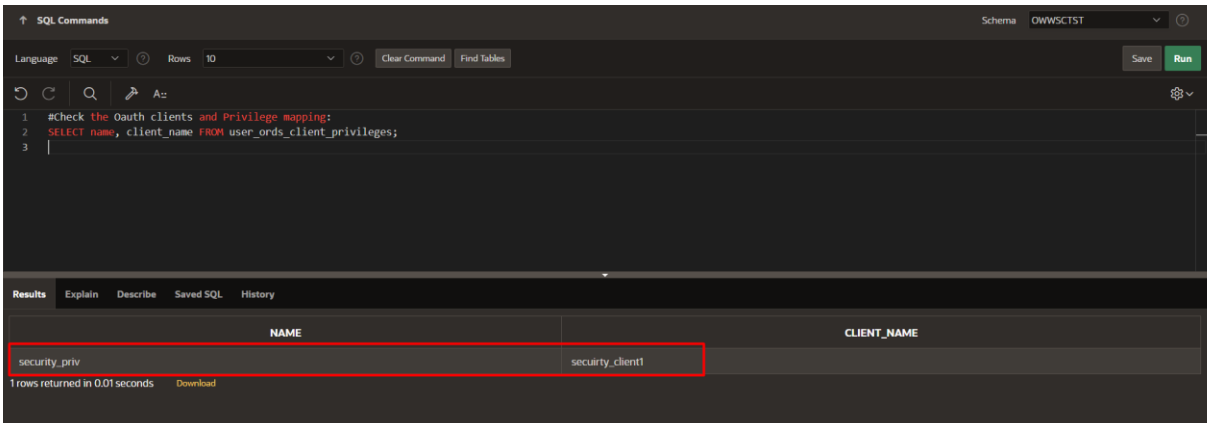
```
SELECT id, name, client_id, client_secret FROM user_ords_clients;
```

ID	NAME	CLIENT_ID	CLIENT_SECRET
31716	secuirty_client	NWpXJIFxHLdsnuBaW3ewlA..	bc5GcNwcmgYP5Yfdgw_epg..
31859	secuirty_client1	2xN3TXFwluO6diyB85ZYmw..	cpiJDTVs1BlFbRHjoV8Vaw..

2 rows returned in 0.01 seconds

#Check the Oauth clients

SELECT id, name, client_id, client_secret FROM user_ords_clients;



Schema: OWWSCTST

Language: SQL

Rows: 10

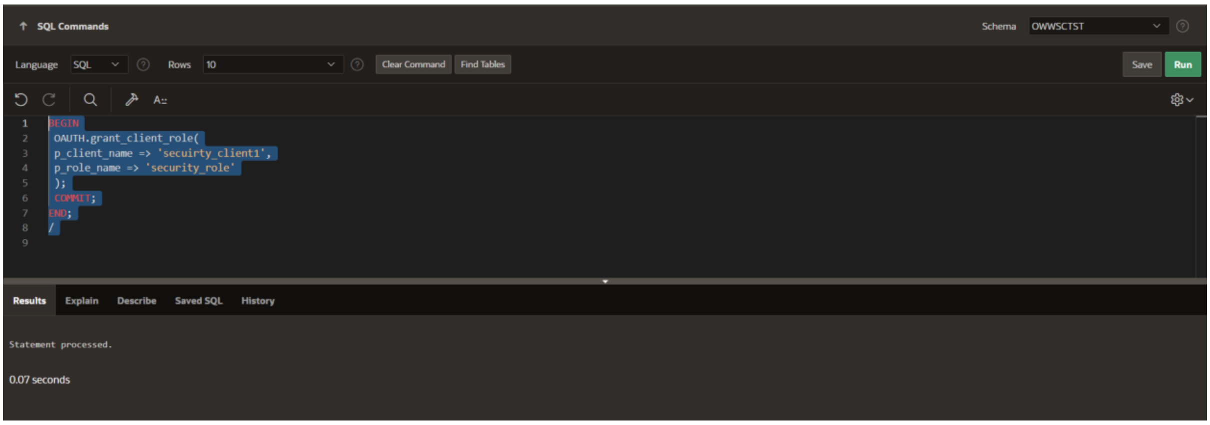
```
#Check the Oauth clients and Privilege mapping:
SELECT name, client_name FROM user_ords_client_privileges;
```

NAME	CLIENT_NAME
security_priv	secuirty_client1

1 rows returned in 0.01 seconds

#Check the Oauth clients and Privilege mapping:

SELECT name, client_name FROM user_ords_client_privileges;



Schema: OWWSCTST

Language: SQL

Rows: 10

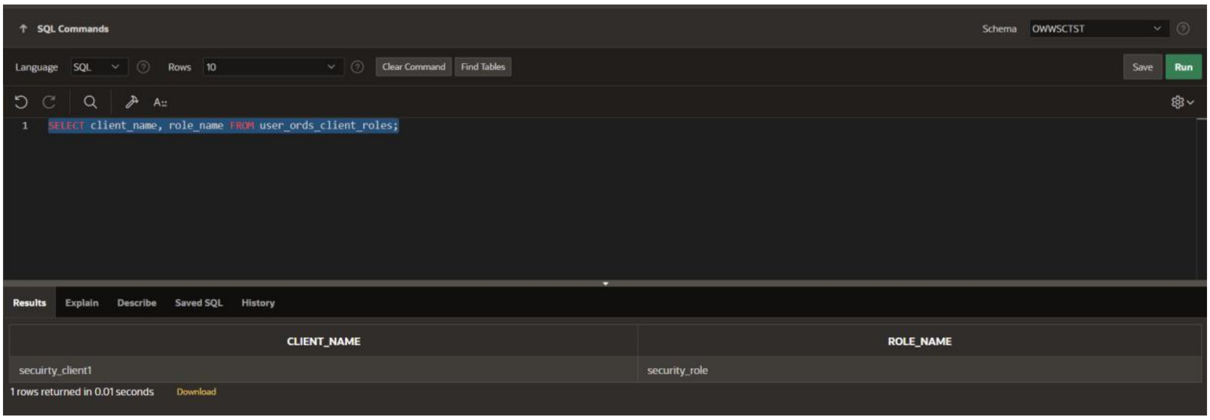
```
BEGIN
 OAUTH.grant_client_role(
 p_client_name => 'secuirty_client1',
 p_role_name => 'security_role'
 );
 COMMIT;
END;
/
```

```
Statement processed.
```

0.07 seconds

#Grant client role to clients:

```
BEGIN
 OAUTH.grant_client_role(
 p_client_name => 'secuirty_client1',
 p_role_name => 'security_role'
 );
 COMMIT;
END;
/
```



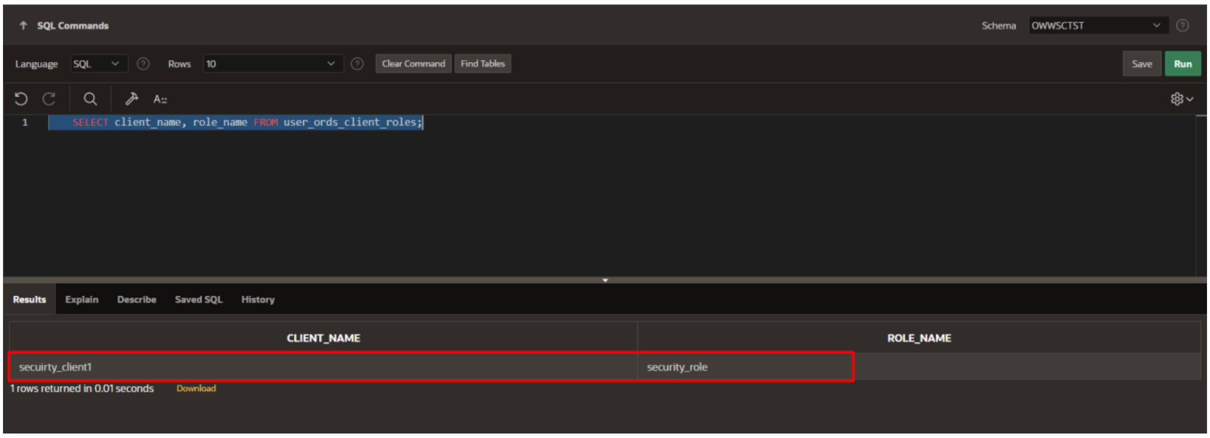
```
SELECT client_name, role_name FROM user_ords_client_roles;
```

CLIENT_NAME	ROLE_NAME
secuirty_client1	security_role

1 rows returned in 0.01 seconds

# Check the Oauth clients and Role mapping:

SELECT client_name, role_name FROM user_ords_client_roles;



```
SELECT client_name, role_name FROM user_ords_client_roles;
```

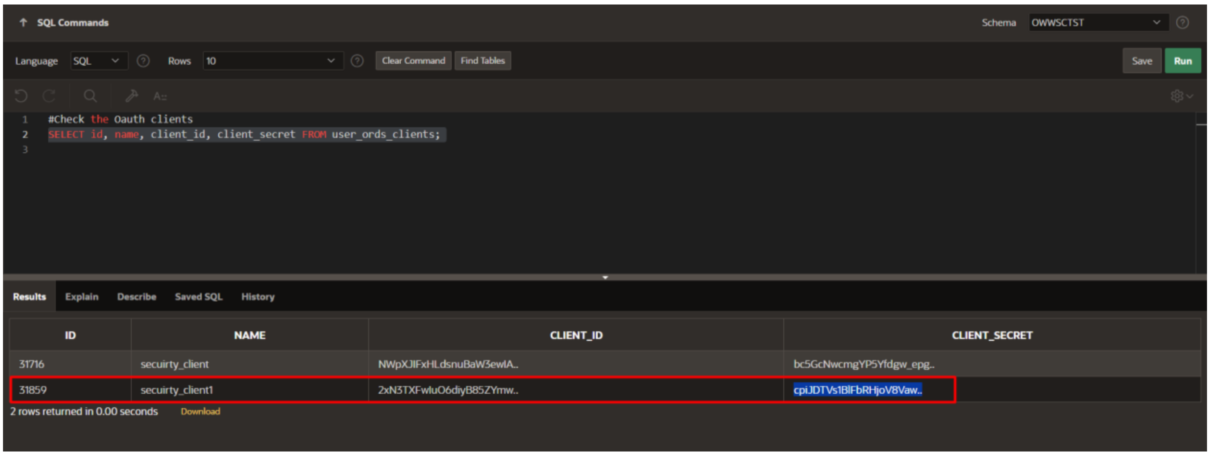
CLIENT_NAME	ROLE_NAME
secuirty_client1	security_role

1 rows returned in 0.01 seconds

Note down the Oauth clients:

Client ID: 2xN3TXFwIuO6diyB85ZYmw..

Client Secret: cpiJDTVs1BlFbRHjoV8Vaw..

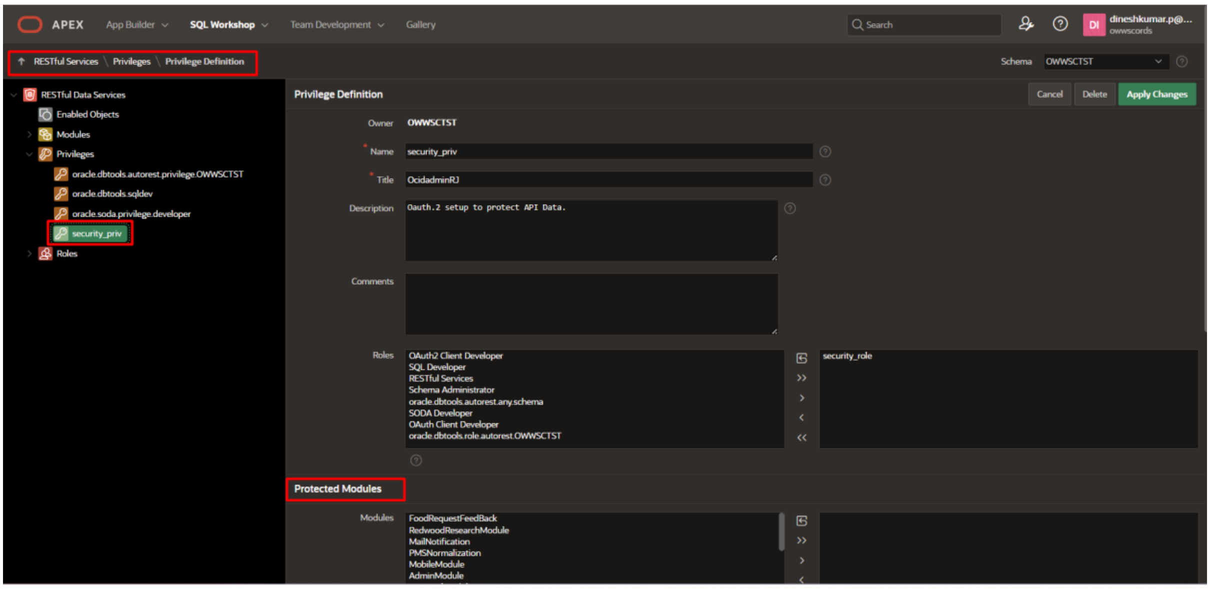


```
#Check the Oauth clients
SELECT id, name, client_id, client_secret FROM user_ords_clients;
```

ID	NAME	CLIENT_ID	CLIENT_SECRET
31716	secuirty_client	NWpXJIFxHLdsnuBaW3ewlA..	bc5GcNwcmgYP5Yfdgw_epg..
31859	secuirty_client1	2xN3TXFwluO6diyB85ZYmw..	cpiJDTVs1BlFbRHjoV8Vaw..

2 rows returned in 0.00 seconds

Now protect the module with the newly created privilege.



Owner	OWWSCTST
* Name	security_priv
* Title	OcidadminRJ
Description	Oauth.2 setup to protect API Data.
Comments	

Roles:

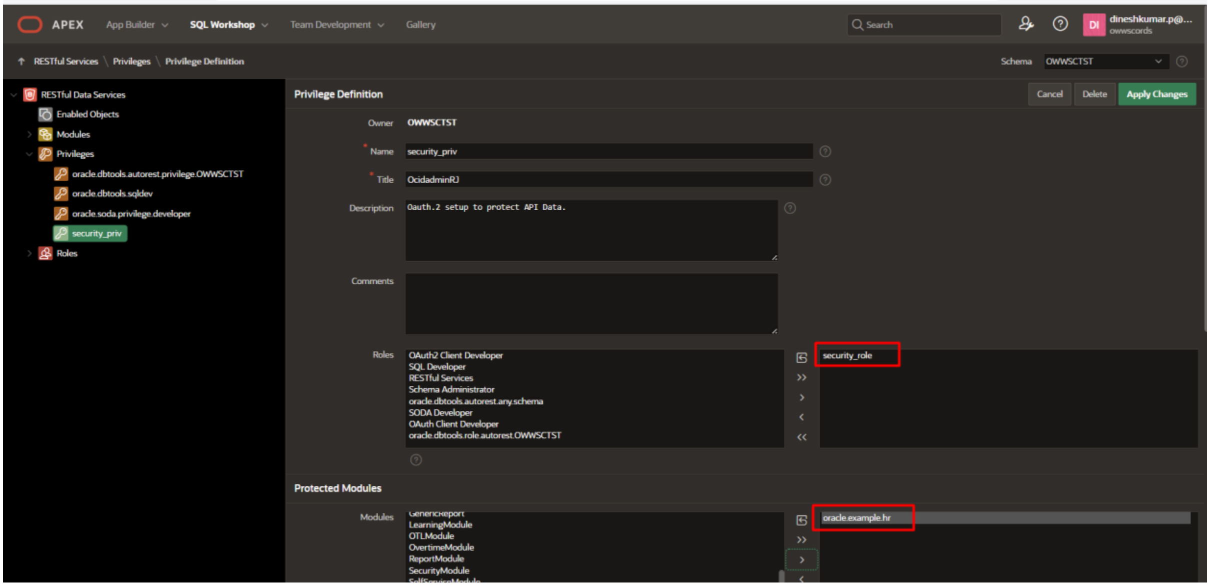
OAuth2 Client Developer
SQL Developer
RESTful Services
Schema Administrator
oracle.dbtools.autorest.any.schema
SODA Developer
OAuth Client Developer
oracle.dbtools.role.autorest.OWWSCTST

Selected: security_role

Protected Modules

Modules:

FoodRequestFeedBack
RedwoodResearchModule
MailNotification
PMSNormalization
MobileModule
AdminModule



Schema: OWWSCTST

Privilege Definition

Owner: OWWSCTST

* Name: security_priv

* Title: OcidadminRJ

Description:

```
Oauth.2 setup to protect API Data.
```

Comments:

Roles:
OAuth2 Client Developer
SQL Developer
RESTful Services
Schema Administrator
oracle.dbtools.autorest.any.schema
SODA Developer
OAuth Client Developer
oracle.dbtools.role.autorest.OWWSCTST

Selected: security_role

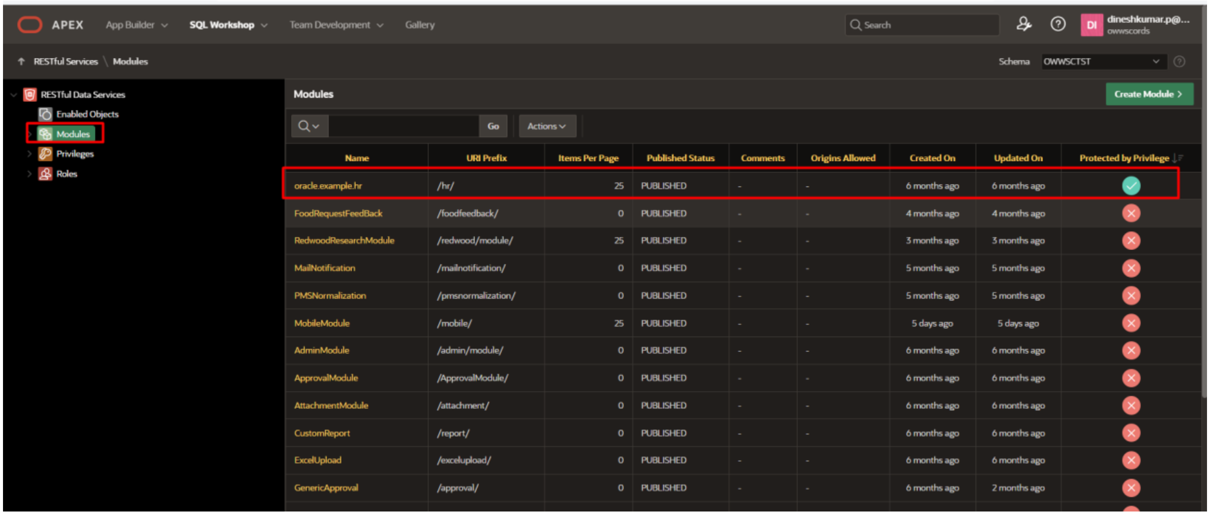
Protected Modules

Modules:
GenericReport
LearningModule
OTLModule
OvertimeModule
ReportModule
SecurityModule
SelfServiceModule

Selected: oracle.example.hr

Click save apply changes.

Check the module is protected status.

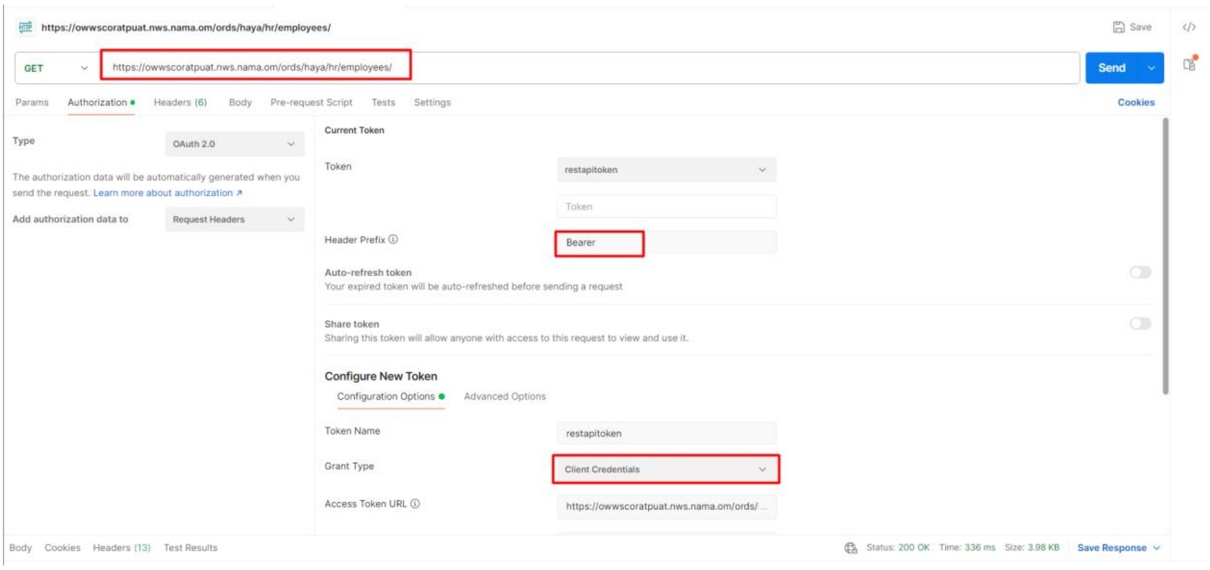


Schema: OWWSCTST

Modules

Name	URI Prefix	Items Per Page	Published Status	Comments	Origins Allowed	Created On	Updated On	Protected by Privilege
oracle.example.hr	/hr/	25	PUBLISHED	-	-	6 months ago	6 months ago	✓
FoodRequestFeedBack	/foodfeedback/	0	PUBLISHED	-	-	4 months ago	4 months ago	✗
RedwoodResearchModule	/redwood/module/	25	PUBLISHED	-	-	3 months ago	3 months ago	✗
MailNotification	/mailnotification/	0	PUBLISHED	-	-	5 months ago	5 months ago	✗
PMSNormalization	/pmsnormalization/	0	PUBLISHED	-	-	5 months ago	5 months ago	✗
MobileModule	/mobile/	25	PUBLISHED	-	-	5 days ago	5 days ago	✗
AdminModule	/admin/module/	0	PUBLISHED	-	-	6 months ago	6 months ago	✗
ApprovalModule	/ApprovalModule/	0	PUBLISHED	-	-	6 months ago	6 months ago	✗
AttachmentModule	/attachment/	0	PUBLISHED	-	-	6 months ago	6 months ago	✗
CustomReport	/report/	0	PUBLISHED	-	-	6 months ago	6 months ago	✗
ExcelUpload	/excelupload/	0	PUBLISHED	-	-	6 months ago	6 months ago	✗
GenericApproval	/approval/	0	PUBLISHED	-	-	6 months ago	2 months ago	✗

Use postman to validate the Oauth.



https://owwscoratpuat.nws.nama.om/ords/haya/hr/employees/

GET https://owwscoratpuat.nws.nama.om/ords/haya/hr/employees/

Type: OAuth 2.0

The authorization data will be automatically generated when you send the request. Learn more about authorization ↗

Add authorization data to: Request Headers

Current Token

Token: restapitoken

Header Prefix: Bearer

Auto-refresh token
Your expired token will be auto-refreshed before sending a request

Share token
Sharing this token will allow anyone with access to this request to view and use it.

Configure New Token

Configuration Options ● Advanced Options

Token Name: restapitoken

Grant Type: Client Credentials

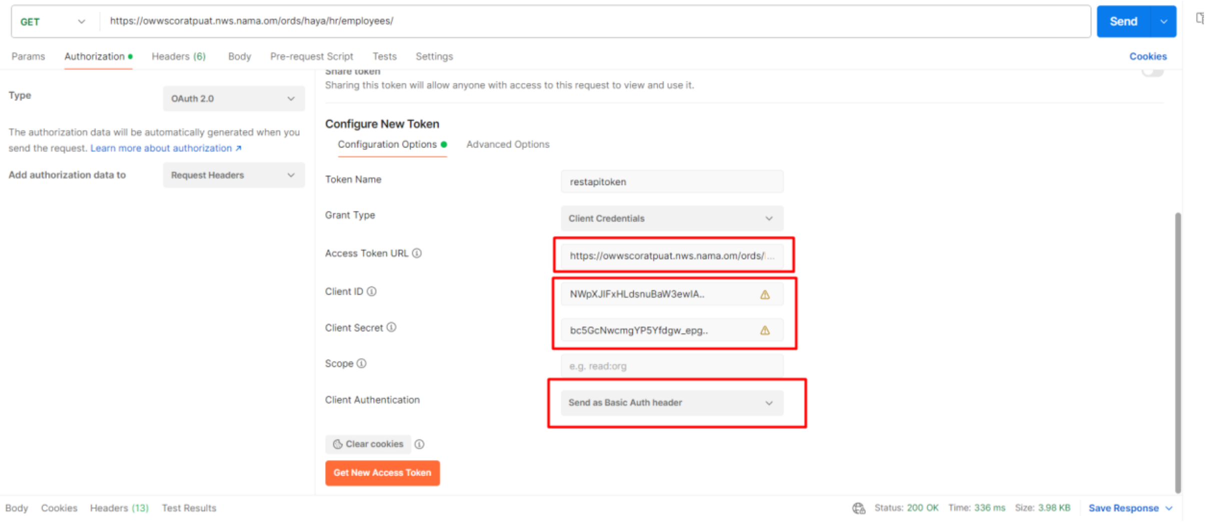
Access Token URL: https://owwscoratpuat.nws.nama.om/ords/...

Status: 200 OK Time: 336 ms Size: 3.98 KB

Token URL:

Sample: https://<apex-hostname>/ords/<schema_alias>/oauth/token

https://owwscoratpuat.nws.nama.om/ords/haya/oauth/token



GET https://owwscoratpuat.nws.nama.om/ords/haya/hr/employees/

Type: OAuth 2.0

The authorization data will be automatically generated when you send the request. Learn more about authorization ↗

Add authorization data to: Request Headers

Share token
Sharing this token will allow anyone with access to this request to view and use it.

Configure New Token

Configuration Options ● Advanced Options

Token Name: restapitoken

Grant Type: Client Credentials

Access Token URL: https://owwscoratpuat.nws.nama.om/ords/...

Client ID: NWpXJIFxHLdsnuBaW3ewlA..

Client Secret: bc5GcNwcmgYP5Yfdgw_epg..

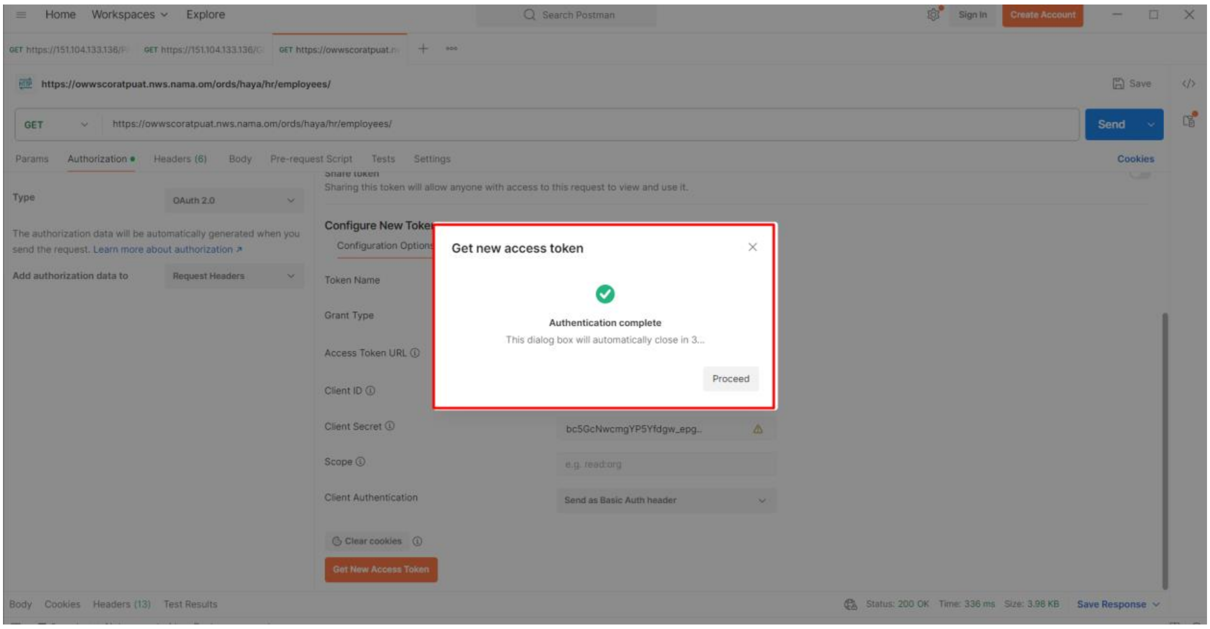
Scope: e.g. read:org

Client Authentication: Send as Basic Auth header

Status: 200 OK Time: 336 ms Size: 3.98 KB

Obtaining An Authorization Token

Before the client application can call our secured service, they will first need to obtain an authorization token. This is done by calling a special ORDS REST service '.../oauth2/token'. By default, the token expires after 3,600 seconds (or 1 Hour).



Home
Workspaces
Explore
Search Postman
Sign In
Create Account
GET https://151.104.133.136/
GET https://151.104.133.136/
GET https://owwscoratpuat.n

https://owwscoratpuat.nws.nama.om/ords/haya/hr/employees/

GET https://owwscoratpuat.nws.nama.om/ords/haya/hr/employees/

Share token
Sharing this token will allow anyone with access to this request to view and use it.

Type: OAuth 2.0

The authorization data will be automatically generated when you send the request. Learn more about authorization ↗

Add authorization data to: Request Headers

Configure New Token

Token Name

Grant Type

Access Token URL ⓘ

Client ID ⓘ

Client Secret ⓘ: bc5GcNwcmgYP5Yfdgw_epg..

Scope ⓘ: e.g. read:org

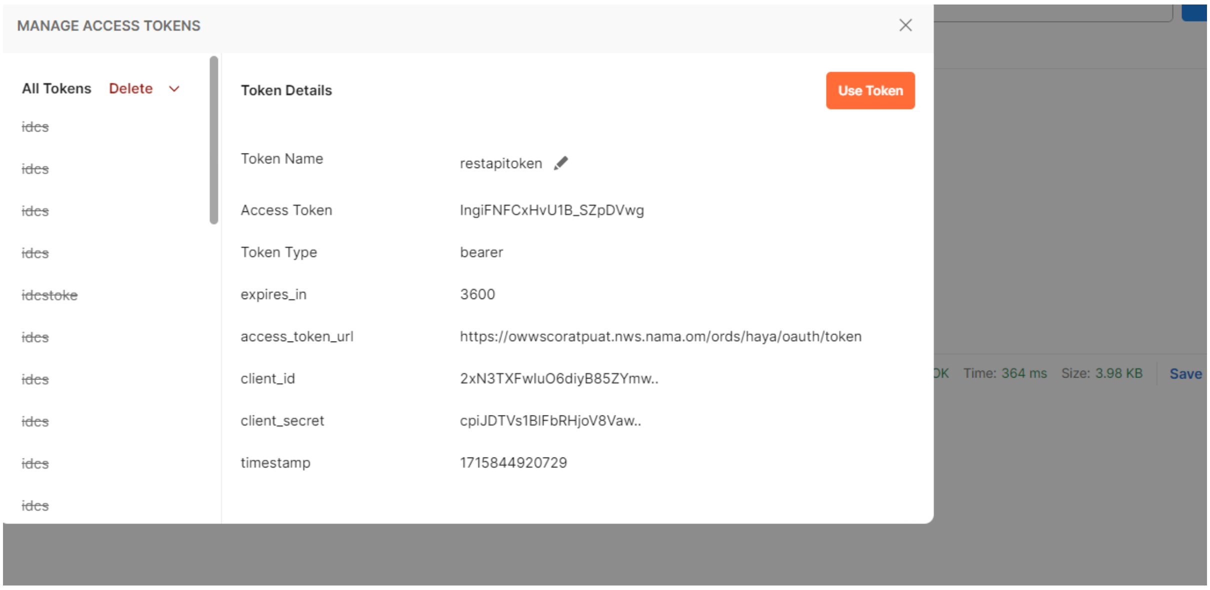
Client Authentication: Send as Basic Auth header

Get new access token

Authentication complete

This dialog box will automatically close in 3...

Status: 200 OK Time: 336 ms Size: 3.98 KB



MANAGE ACCESS TOKENS

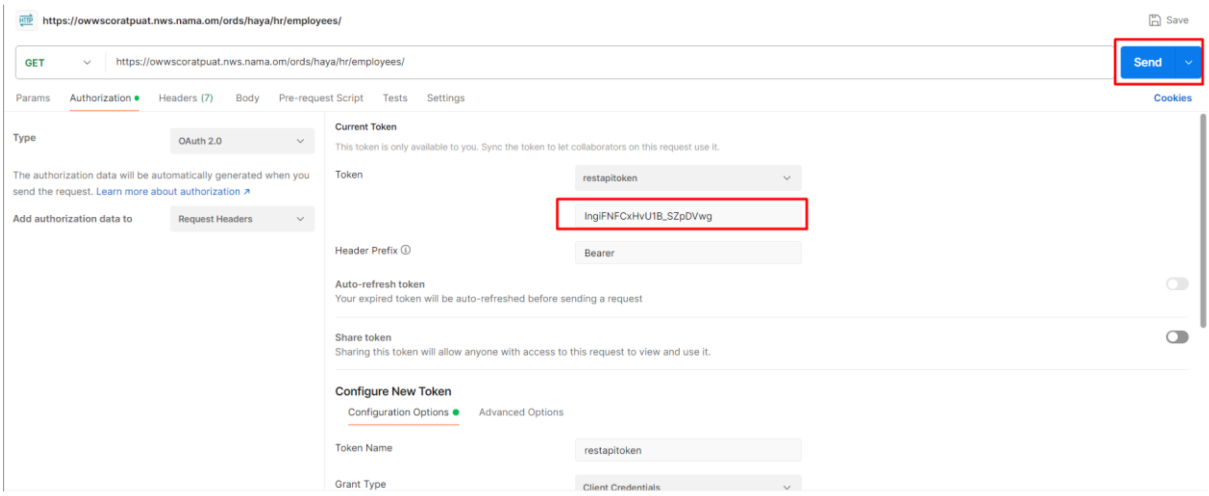
~~ides~~
~~ides~~
~~ides~~
~~ides~~
~~idestoke~~
~~ides~~
~~ides~~
~~ides~~
~~ides~~
~~ides~~

Token Details

Token Name	restapitoken
Access Token	IngiFNFCxHvU1B_SZpDVwg
Token Type	bearer
expires_in	3600
access_token_url	https://owwscoratpuat.nws.nama.om/ords/haya/oauth/token
client_id	2xN3TXFwluO6diyB85ZYmw..
client_secret	cpiJDTVs1BIFbRHjoV8Vaw..
timestamp	1715844920729

OK Time: 364 ms Size: 3.98 KB Save

Use the token.



https://owwscoratpuat.nws.nama.om/ords/haya/hr/employees/

GET https://owwscoratpuat.nws.nama.om/ords/haya/hr/employees/

Type: OAuth 2.0

The authorization data will be automatically generated when you send the request. Learn more about authorization ↗

Add authorization data to: Request Headers

Current Token

This token is only available to you. Sync the token to let collaborators on this request use it.

Token: restapitoken

IngiFNFCxHvU1B_SZpDVwg

Header Prefix ⓘ: Bearer

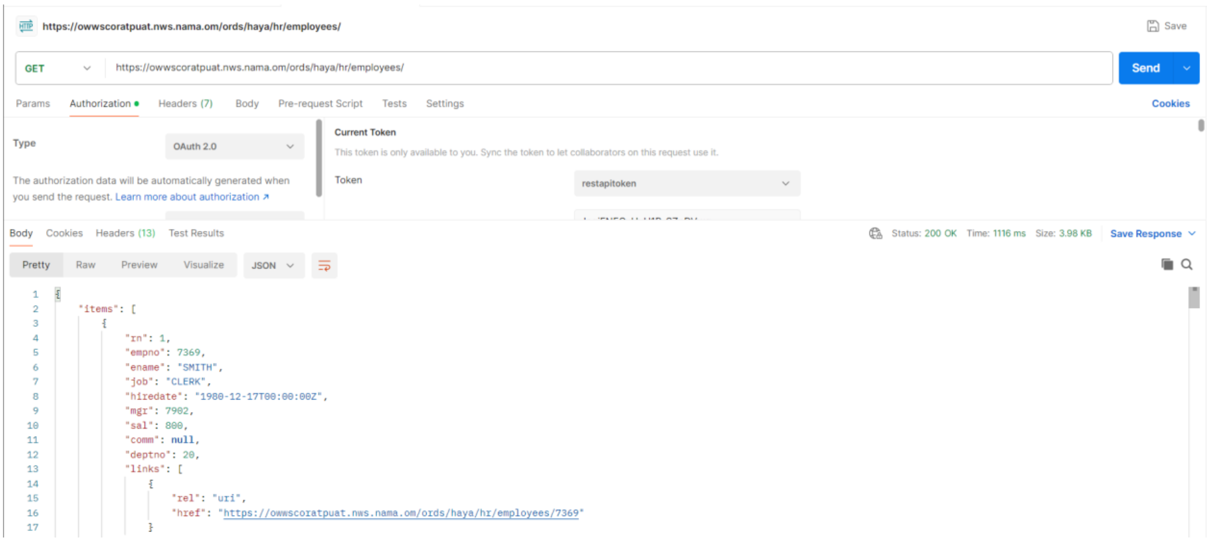
Auto-refresh token
Your expired token will be auto-refreshed before sending a request

Share token
Sharing this token will allow anyone with access to this request to view and use it.

Configure New Token

Token Name: restapitoken

Grant Type: Client Credentials



https://owwscoratpuat.nws.nama.om/ords/haya/hr/employees/

GET https://owwscoratpuat.nws.nama.om/ords/haya/hr/employees/

Type: OAuth 2.0

The authorization data will be automatically generated when you send the request. Learn more about authorization ↗

Current Token

This token is only available to you. Sync the token to let collaborators on this request use it.

Token: restapitoken

Status: 200 OK Time: 1116 ms Size: 3.98 KB

```
{
    "items": [
        {
            "rn": 1,
            "empno": 7369,
            "ename": "SMITH",
            "job": "CLERK",
            "hiredate": "1980-12-17T00:00:00Z",
            "mgr": 7902,
            "sal": 800,
            "comm": null,
            "deptno": 20,
            "links": [
                {
                    "rel": "uri",
                    "href": "https://owwscoratpuat.nws.nama.om/ords/haya/hr/employees/7369"
                }
```