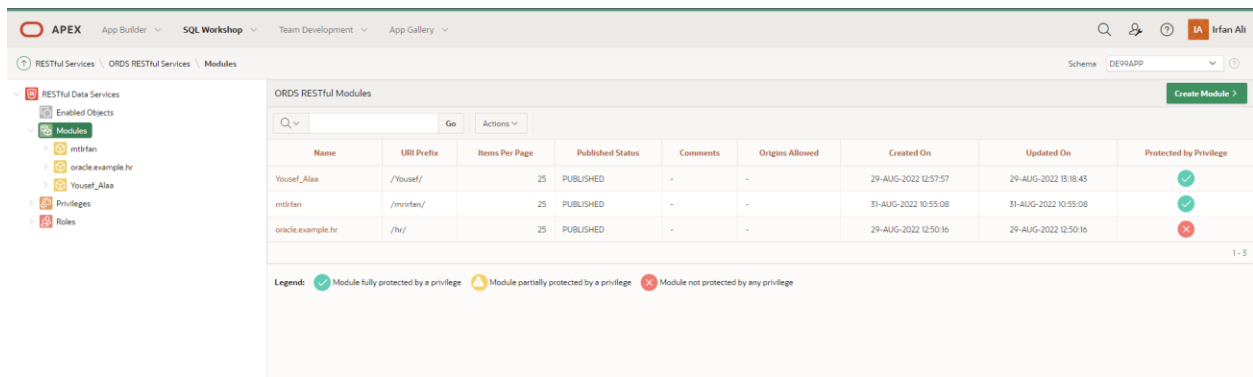


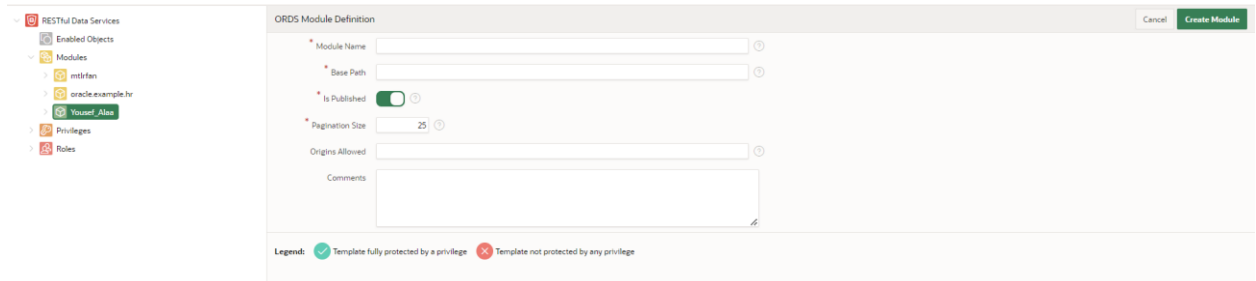
# How to create APIs using apex

Firstly, we go to SQL Workshop and chose RESTful Services,

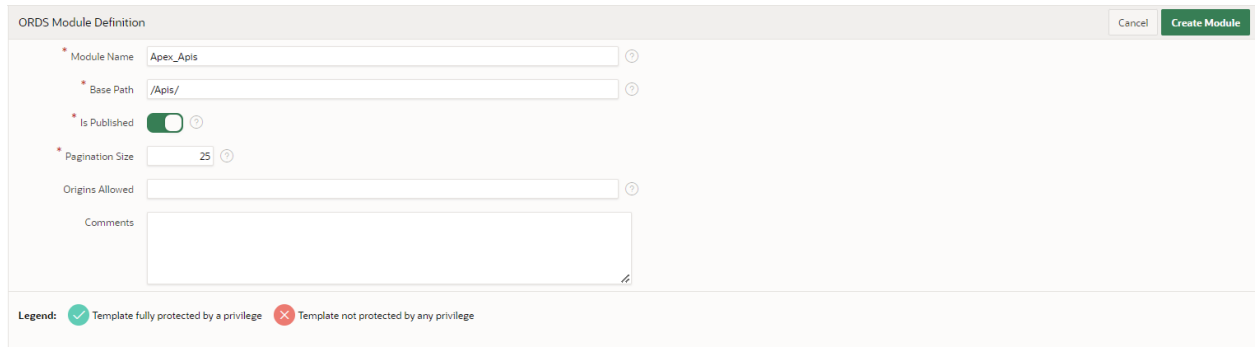
Now we need to create new module:



- Press create Module:

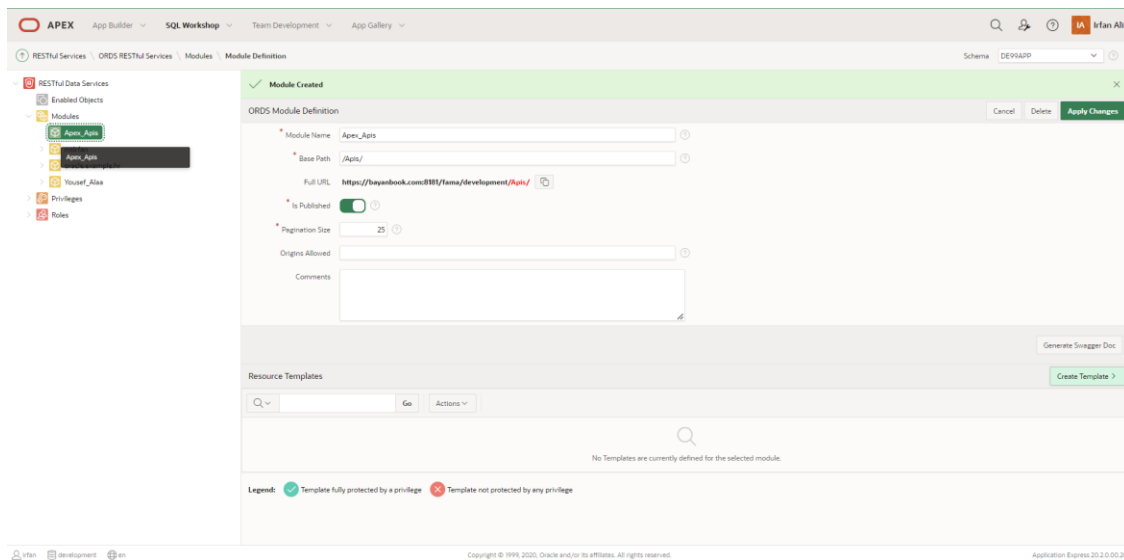


- Enter **Module name** and **Base path**:

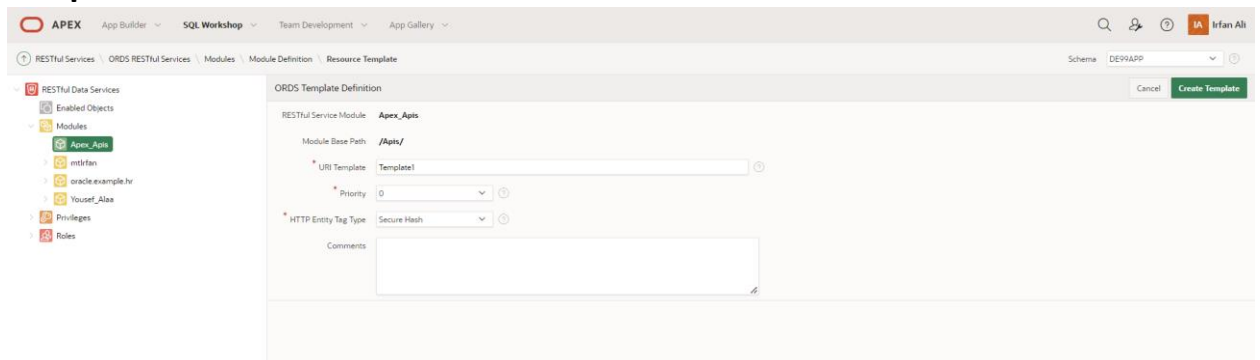


- Then **press create Module**

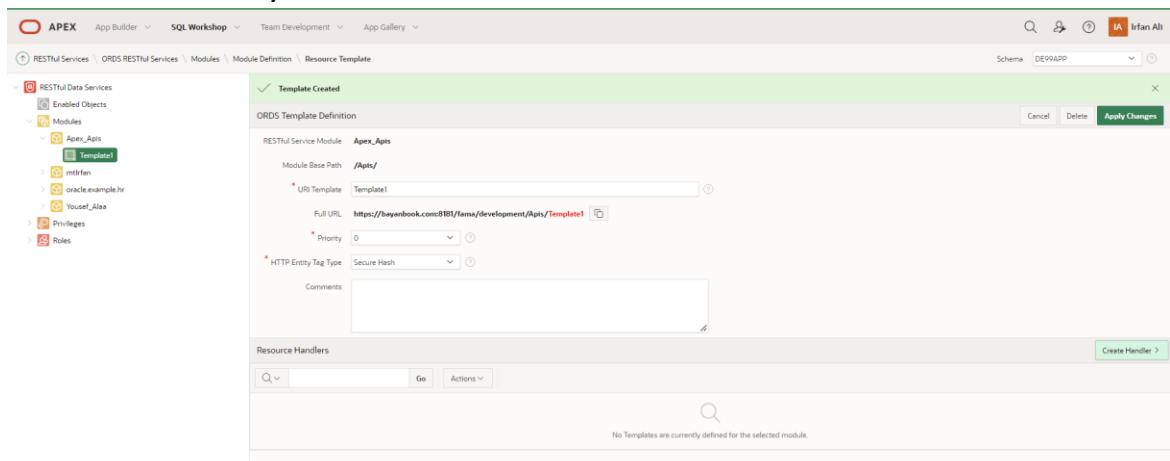
Every module has Templates and each template has APIs so now we need to create template:



- Press create template and give it **URL Template** then press create template:



- Now we are ready to create our APIs:



- Press **Create Handler**:

ORDS Handler Definition

RESTful Service Module: **Apex\_Apis**

Module Base Path: **/Apex/**

URI Template: **Template1**

Method: **GET**

Source Type: **Collection Query**

Format: **JSON**

Pagination Size:

Comments:

Source:

- Specify the type of the API (GET, POST, PUT, DELETE).
- In the source we type SQL query, for example if we want to make GET query that get all the data from one table called ADDRESS\_TYPES we use this query:

```
select * from ADDRESS_TYPES
```

- last step we press Create Handler.
- Now we have our API ready for running

✓ Handler Created

ORDS Handler Definition

RESTful Service Module: **Apex\_Apis**

Module Base Path: **/Apex/**

URI Template: **Template1**

Full URL: <https://bayanbook.com/8181/tama/development/Apis/Template1>

Method: **GET**

Source Type: **Collection Query**

Format: **JSON**

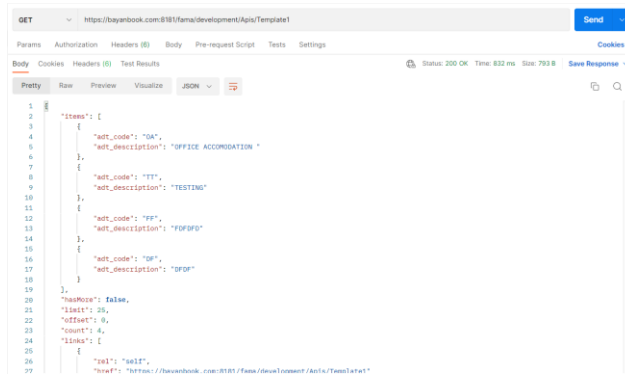
Pagination Size:

Comments:

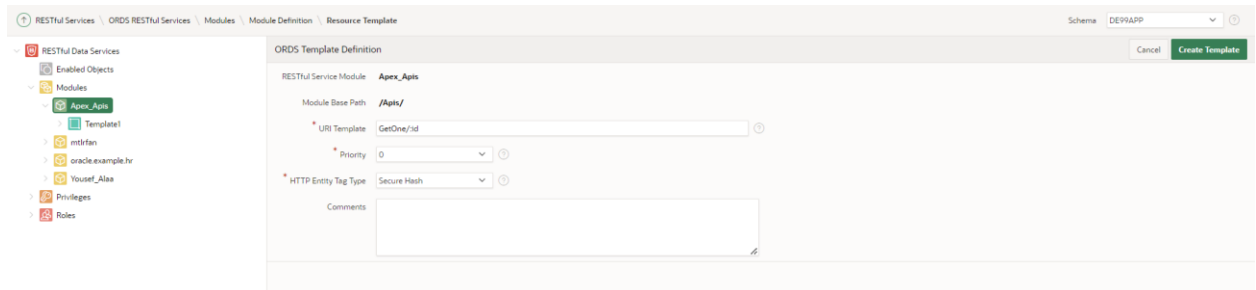
Source:

```
1 select * from ADDRESS_TYPES
```

- We use Full URL provided to run our api. Let's test it using postman:



- Now we have an API that response with all the rows of the table, what if we want an API that responses with only one row depending on something that API user type.
- We can't have two APIs that have the same Method in one handler so we create new handler.
- In the URL Template we type our route/: dependence for example we will create template called GetOne that has GET API that get one row from table with specific id, then we type GetOne/:id then create the template. ':' means this will be an input



- Now create handler **with method GET**:
- Now we have id as variable and we can use it in our query:

```

SELECT * FROM YOUSEF_CATEGORY
WHERE ID=:id

```

- Now we can consume the API using postman:

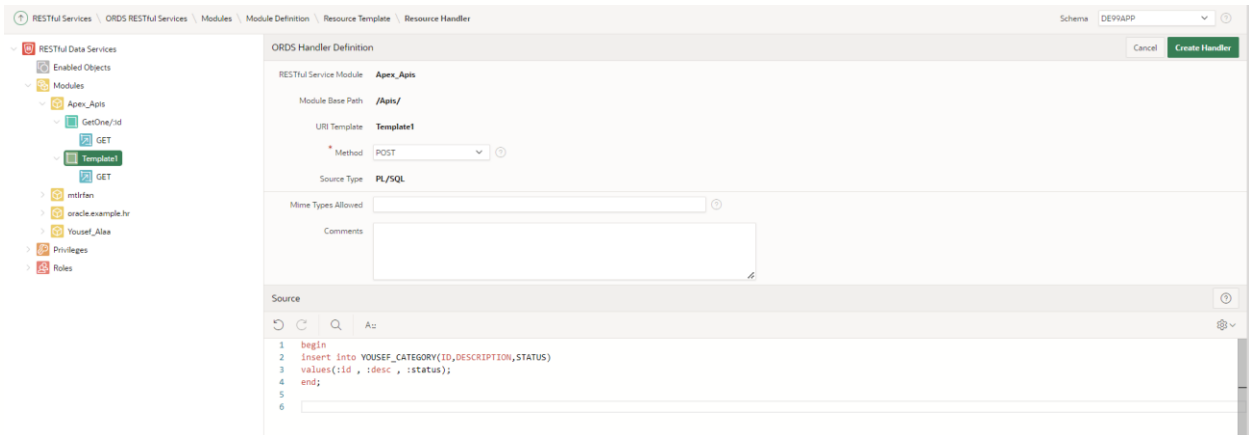
The screenshot shows a Postman interface with a GET request to `https://bayanbook.com:8181/fama/development/Apis/GetOne/1`. The response status is 200 OK, with a time of 176 ms and a size of 662 B. The response body is displayed in JSON format, showing a list of items and pagination information.

```
1  {
2    "items": [
3      {
4        "id": 1,
5        "description": "development",
6        "status": "inactive"
7      }
8    ],
9    "hasMore": false,
10   "limit": 25,
11   "offset": 0,
12   "count": 1,
13   "links": [
14     {
15       "rel": "self",
16       "href": "https://bayanbook.com:8181/fama/development/Apis/GetOne/1"
17     },
18     {
19       "rel": "describedby",
20       "href": "https://bayanbook.com:8181/fama/development/metadata-catalog/Apis/GetOne/item"
21     },
22     {
23       "rel": "first",
24       "href": "https://bayanbook.com:8181/fama/development/Apis/GetOne/1"
25     }
26   ]
27 }
```

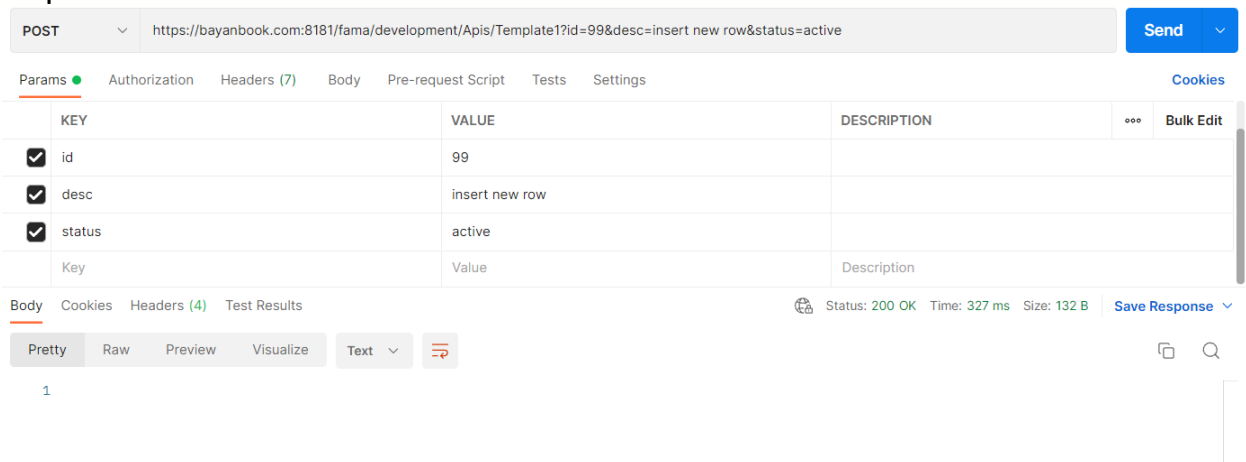
## Post request:

- We can use template1 as it doesn't any post request or create new template.
- We will use template1 then create new Handler then specify the method as post and insert out PL/SQL code in the source for example if we want to insert new row in a table that has columns of (ID, DESCRIPTION, STATUS) we will use this code:

```
begin
insert into YOUSEF_CATEGORY(ID, DESCRIPTION,STATUS)
values(:id , :desc , :status);
end;
```



- New create handler and the api is ready to be consumed:  
If we used postman to consume the api we will just add the parameter and click send to insert in the table and get status of 200 means successful request:



- If we try to add the same row again in the table we will get an error message as we can't have two rows this the same primary key:

POST <https://bayanbook.com:8181/fama/development/Api/Template?id=99&desc=insert new row&status=active> Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> id	99	
<input checked="" type="checkbox"/> desc	insert new row	
<input checked="" type="checkbox"/> status	active	
Key	Value	Description

Body Cookies Headers (6) Test Results Status: 555 Time: 365 ms Size: 16.83 KB Save Response

Pretty Raw Preview Visualize HTML

```

1 <!DOCTYPE html>
2 <html>
3 <style type="text/css" media="screen">
4   footer,
5   header {
6     display: block;
7   }
8
9   html {
10    font-family: sans-serif;
11    -ms-text-size-adjust: 100%;
12    -webkit-text-size-adjust: 100%;
13  }
14
15  body {
16    margin: 0;
17  }

```

## Security of the APIs:

What if we want to add some privilege so that only authorized people can access out APIs.

1- Add new role:

From roles press Create Role:

APEX App Builder SQL Workshop Team Development App Gallery

RESTful Services ORDS RESTful Services Roles Schema DE99APP Create Role

Name	Owner	Created On	Updated On
oracle.ords.roles.autorest.DE99APP	DE99APP	29-AUG-2022 12:50:15	29-AUG-2022 12:50:15
OAuth2 Client Developer	ORDS	04-OCT-2021 22:15:17	04-OCT-2021 22:15:17
SQL Developer	ORDS	04-OCT-2021 22:15:17	04-OCT-2021 22:15:17
RESTful Services	ORDS	04-OCT-2021 22:15:17	04-OCT-2021 22:15:17
Schema Administrator	ORDS	04-OCT-2021 22:15:17	04-OCT-2021 22:15:17
oracle.ords.roles.autorest.any.schema	ORDS	04-OCT-2021 22:15:17	04-OCT-2021 22:15:17
SODA Developer	ORDS	04-OCT-2021 22:15:17	04-OCT-2021 22:15:17
Items Role	DE99APP	31-AUG-2022 10:57:33	31-AUG-2022 10:57:33
GenAI_Role	DE99APP	30-AUG-2022 10:55:22	30-AUG-2022 10:55:22

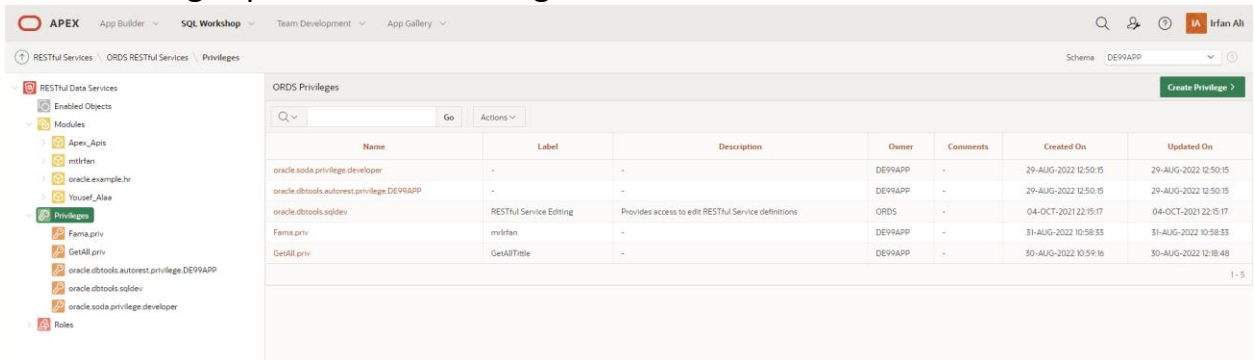
Then give it a name and press Create Role:

ORDS Role Definition Cancel Create Role

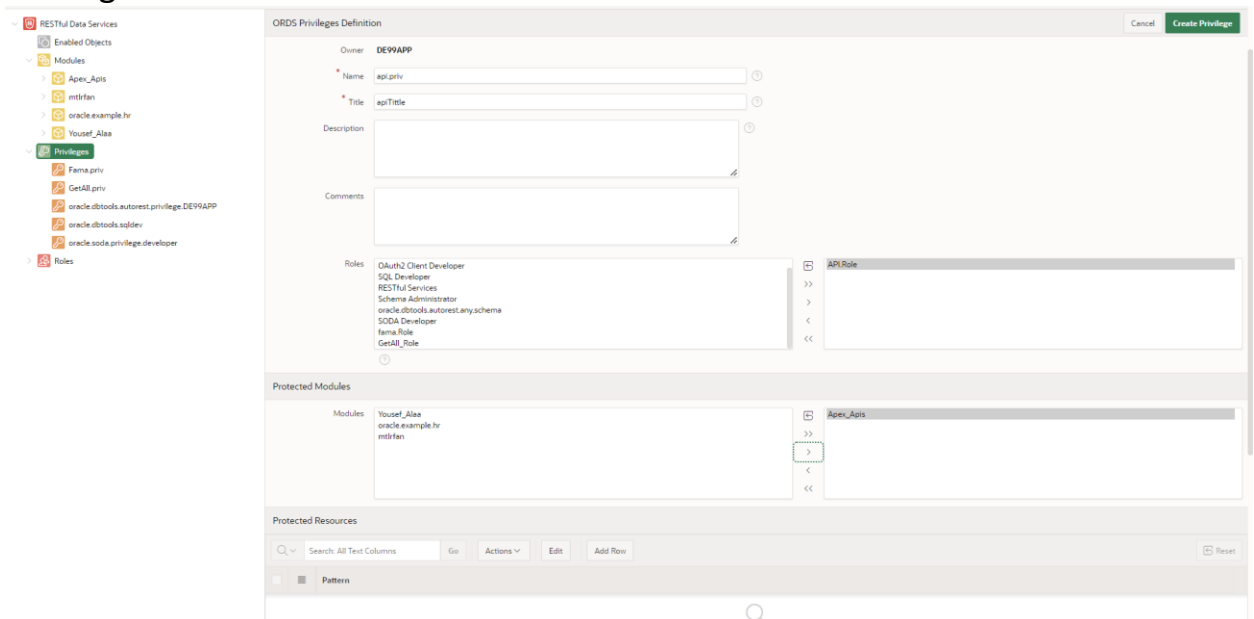
Owner DE99APP

\* Role Name API.Role

## 2- Create new Privilege: From Privileges press create Privilege:



- Then give it name, title roles, and modules or specific route then press create Privilege:



- Now our APIs are secured and anyone want to access it should be authorized.
- If we want to give any user access to our APIs:

1- Add the user as client as user to our system. Example query:

```
BEGIN
  oauth.create_client (p_name      => 'user1',
                     P_grant_type => 'client_credentials',
                     P_description => 'Client with access to Employee
Resources',
                     P_support_email => 'hr.client.one@ordsdemo.com',
                     P_privilege_names => NULL);
COMMIT;
END;
```

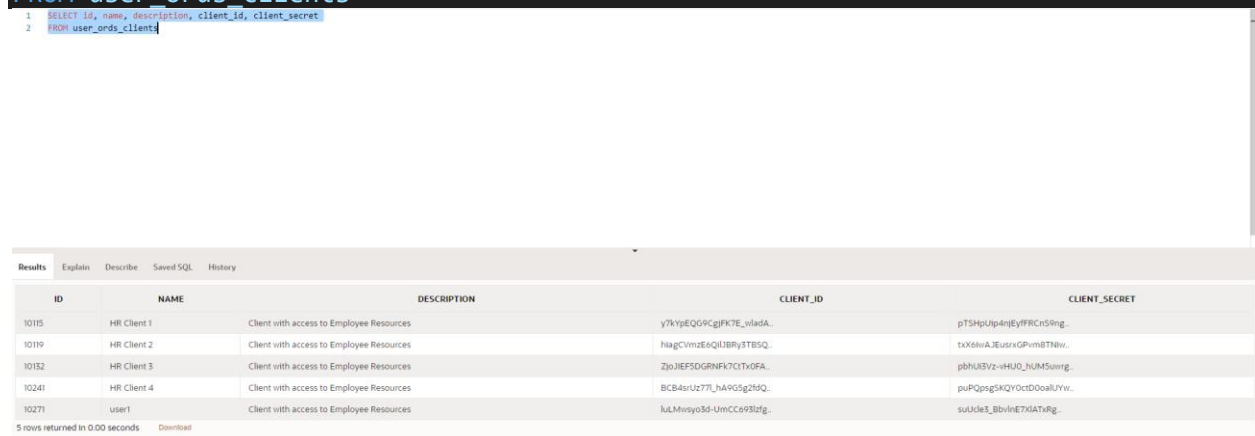


- 2- Give privilege to this user to the role we used to make the Privilege.  
Example query:

```
BEGIN
  oauth.create_client (p_name          => 'user1',
                      P_grant_type     => 'client_credentials',
                      P_description     => 'Client with access to Employee
Resources',
                      P_support_email  => 'hr.client.one@ordsdemo.com',
                      P_privilege_names => NULL);
COMMIT;
END;
```

- 3- Now the user has access to the APIs and can generate access token with his client id and client secret. To get the client secret and id we use this query:

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



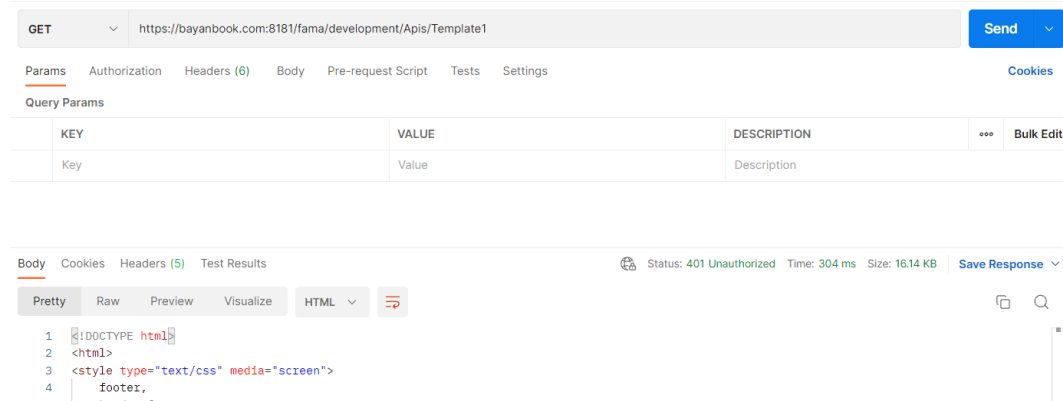
The screenshot shows a SQL query result in a web interface. The query is `SELECT id, name, description, client_id, client_secret FROM user_ords_clients`. The result is displayed as a table with 5 rows. The columns are ID, NAME, DESCRIPTION, CLIENT\_ID, and CLIENT\_SECRET. The data is as follows:

ID	NAME	DESCRIPTION	CLIENT_ID	CLIENT_SECRET
10115	HR Client 1	Client with access to Employee Resources	y7kYpEQQ9CgJfK7E_vlaaA..	pT5HpUlp4njEYfFRcn5ng..
10119	HR Client 2	Client with access to Employee Resources	hiagCvnmZsqUJBPj3TB5Q..	txx8bwAJEusxGPvm8TNbw..
10132	HR Client 3	Client with access to Employee Resources	ZjoJIEF5DGRNFk7CtXoFA..	pbbHUBVz-vHUO_hUM5uerrg..
10241	HR Client 4	Client with access to Employee Resources	BCB4srUz77_hAPG5g2Hq..	puPQpsg5KQVOctD0aalUvw..
10271	user1	Client with access to Employee Resources	IuLMwryso3d-UmCCe93Izg..	suJck3_BbVlnE7XATvRg..

5 rows returned in 0.00 seconds

Now let's test the secured APIs using postman:

- 1- If anyone tried to consume the APIs will get status of 401 unauthorized:



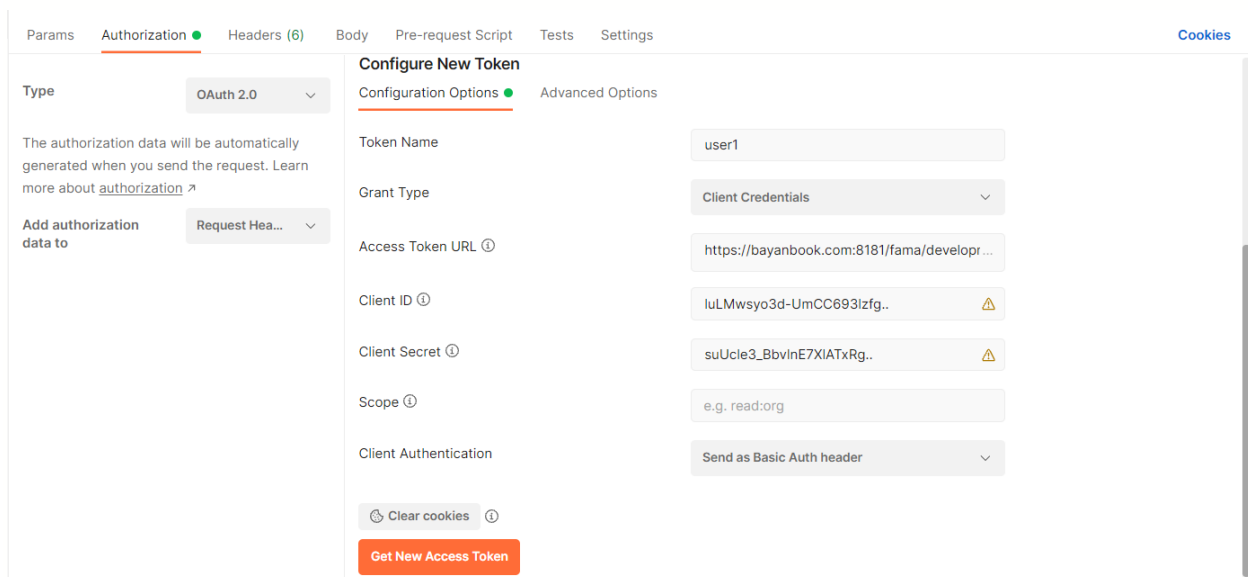
The screenshot shows a Postman API client interface. The request is a GET request to `https://bayanbook.com:8181/fama/development/Api/Template1`. The response is a 401 Unauthorized status. The response body is HTML code, which is partially visible in the 'Pretty' view.

```
1 <!DOCTYPE html>
2 <html>
3 <style type="text/css" media="screen">
4   footer,
5   header {
```

To make an authorization in postman

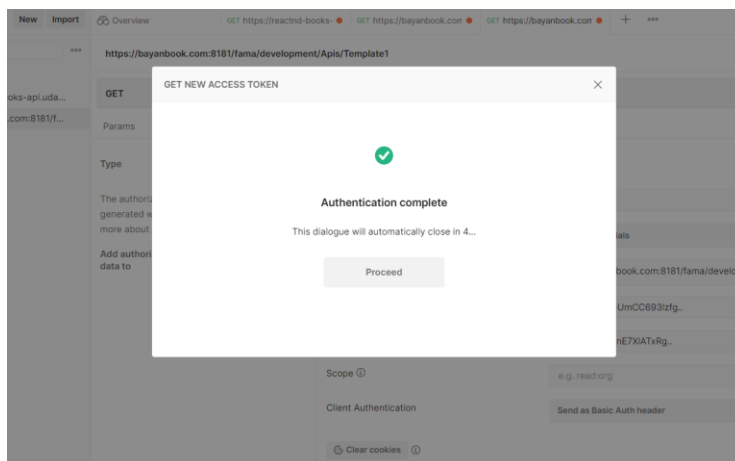
- 1- we go to authorization.
- 2- Chose the type of OAuth 2.0.
- 3- Chose add authorization data to: Request Headers.
- 4- Go to configuration New Token.
- 5- Give it Token Name
- 6- Chose Grant Type: Client Credentials.
- 7- Access token URL to be: <https://bayanbook.com:8181/fama/development/oauth/token>.
- 8- Specify the Client ID.
- 9- Specify the Client Secret.

Then press Get New Access Token:



The image shows the 'Configure New Token' dialog in Postman. The 'Type' is set to 'OAuth 2.0'. The 'Add authorization data to' dropdown is set to 'Request Headers'. The 'Configuration Options' tab is active. The 'Token Name' is 'user1'. The 'Grant Type' is 'Client Credentials'. The 'Access Token URL' is 'https://bayanbook.com:8181/fama/development/oauth/token'. The 'Client ID' is 'luLMwsyo3d-UmCC693lzfg..'. The 'Client Secret' is 'suUcle3\_BbvlnE7XIATxRg..'. The 'Scope' is 'e.g. read:org'. The 'Client Authentication' is set to 'Send as Basic Auth header'. There is a 'Clear cookies' button and a 'Get New Access Token' button at the bottom.

Then you get successful message. Used the generated token to access the APIs.



Now we get status of 200 means successful access:

GET <https://bayanbook.com:8181/fama/development/Api/Template1> Send

Params **Authorization** Headers (7) Body Pre-request Script Tests Settings Cookies

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

Header Prefix ⓘ Bearer

**Configure New Token**

Configuration Options ● Advanced Options

Token Name user1

Body Cookies Headers (6) Test Results

Status: 200 OK Time: 481 ms Size: 876 B Save Response

Pretty Raw Preview Visualize JSON ⌵ ≡

```
1  {
2    "items": [
3      {
4        "adt_code": "OA",
5        "adt_description": "OFFICE ACCOMODATION "
6      },
7      {
8        "adt_code": "TT",
9        "adt_description": "TESTING"
10     },
11     {
12       "adt_code": "FF",
13       "adt_description": "FDFDFD"
14     },
15     {
16       "adt_code": "DF",
17       "adt_description": "DFDF"
```