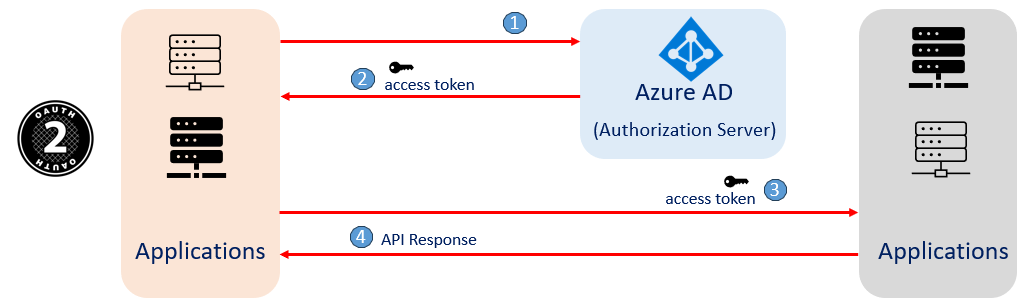
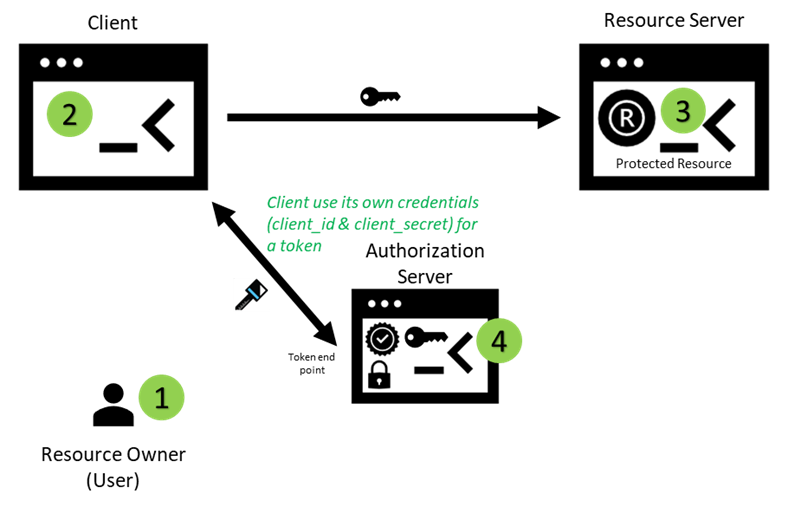
**Part 3: OAuth 2.0 Client Credentials Grant with Azure AD**

<https://medium.com/@shoaib.alam/part-3-oauth-2-0-client-credentials-grant-with-azure-ad-36ac0f9ab2b7>



In client credentials grant flow, the client is identical to the resource owner and request an access token to access their own resources, not on behalf of a user. This is a common situation, in which there are back-end systems that need to communicate directly with each other and not necessarily on behalf of any particular user.

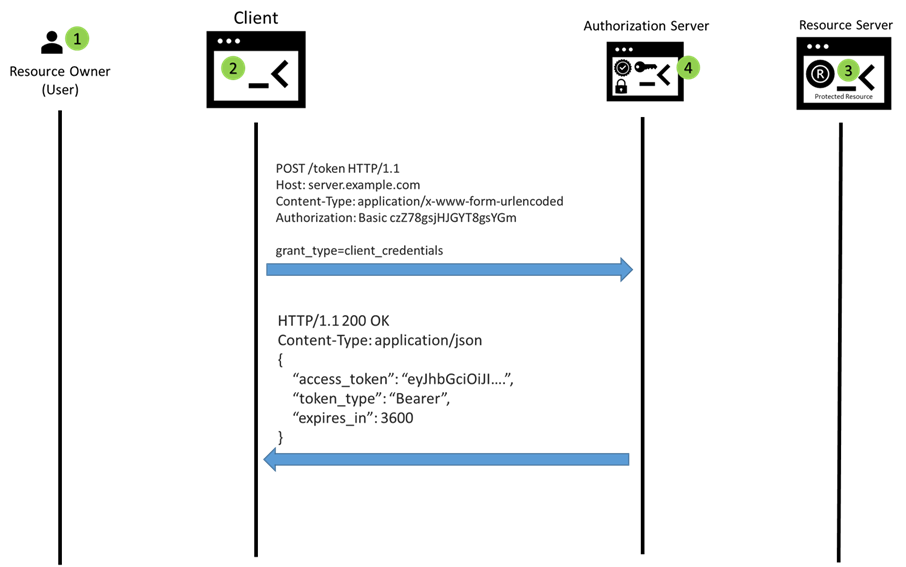
The client requests a token from the token endpoint and uses a grant\_type parameter value as client\_credentials. It doesn’t have an authorization code or other temporary credential to trade for the token. Instead, the client authenticates itself directly by using client\_id and client\_secret in the HTTP Basic auth header. The client can also request specific scopes inside this call using the scope parameter.



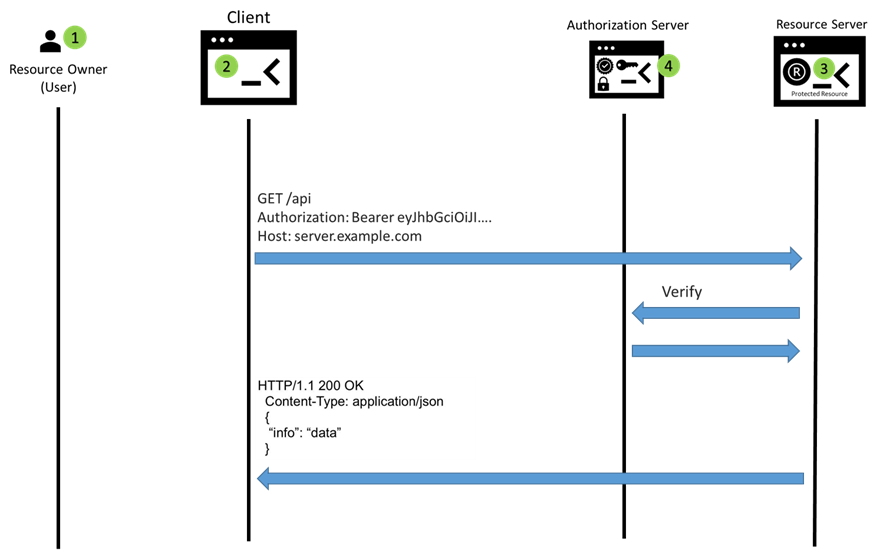
The response from the authorization server is an access token. The client credentials flow does not issue a refresh token because the client is assumed to be in the position of being able to request a new token for itself at any time without involving a separate resource owner and it makes refresh token unnecessary in this context.

**Sequence Diagram**

**Token End Point**



**Resource Access**



**Register an application and get access token**

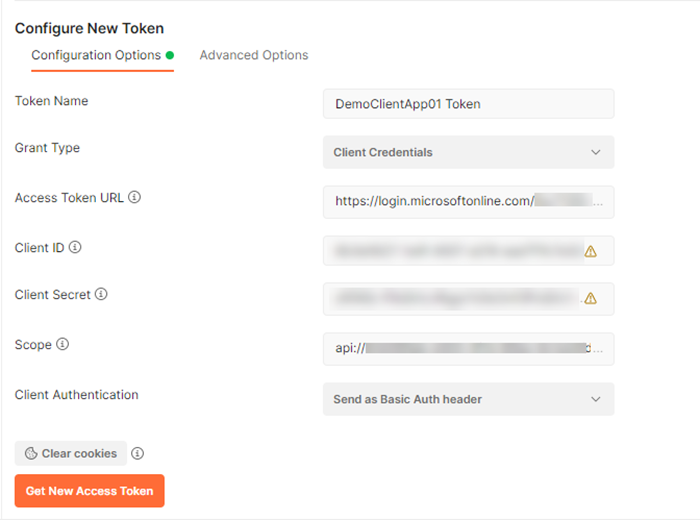
Before this flow can be used, the client must register with the OAuth server same as for authorization code grant flow. After successful registration the client receives a client\_id and a client\_secret

In [Part 2B](https://medium.com/@shoaib.alam/part-2b-oauth-2-0-authorization-code-grant-with-azure-ad-d97b213ecc9) I already registered two applications DemoClientApp01 and DemoWebApp in Azure AD. I am going to use the same applications to get access token for Client Credential flow.

I also assume that Postman is also downloaded and installed. In this part we are going to use client credential as a grant type. Rest of the parameters are as follows:

* Token Name: Any Suitable Name for a Token
* Grant Type: Client Credentials
* Access Token URL: https://login.microsoftonline.com/<<your tenant ID>>/oauth2/v2.0/token
* Client ID: 8\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
* Client Secret: z\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
* Scope: api://{application-id}/.default

*Note: application-id in scope parameter is the application id of DemoWebApp.*



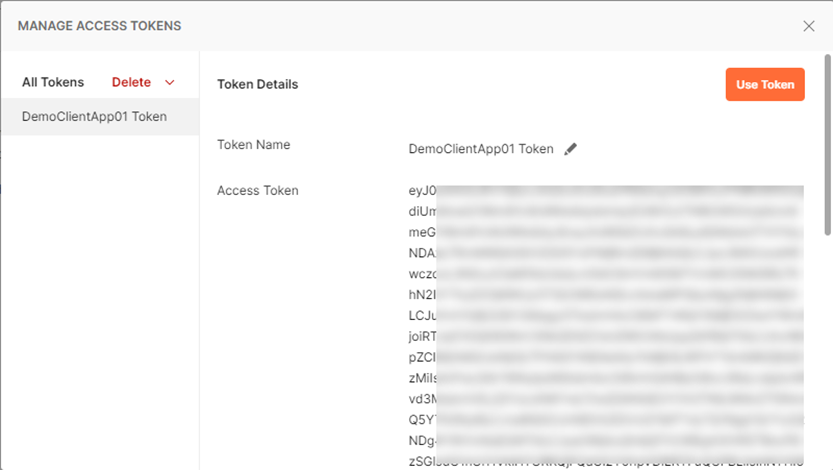
The Client Credentials flow is a server-to-server flow. There is no user authentication involved in the process. It is not going to prompt for any user to log-in.

Get Shoaib Alam’s stories in your inbox

Join Medium for free to get updates from this writer.

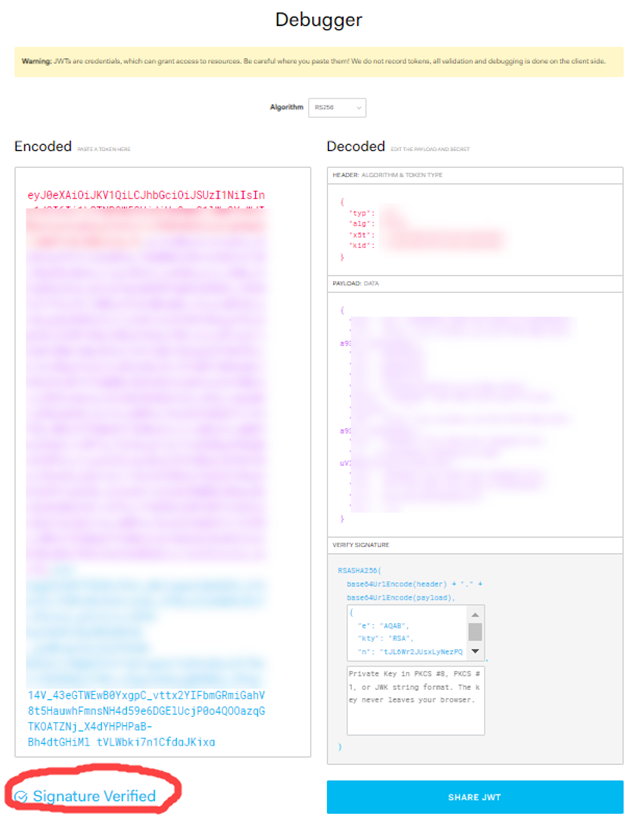
Subscribe

When you press “Get New Access Token”, a new Access Token for a client credential flow is issued by Azure AD.



**Decode JWT Access Token**

Let’s open the website jwt.io and copy & paste the token acquire in the previous step.

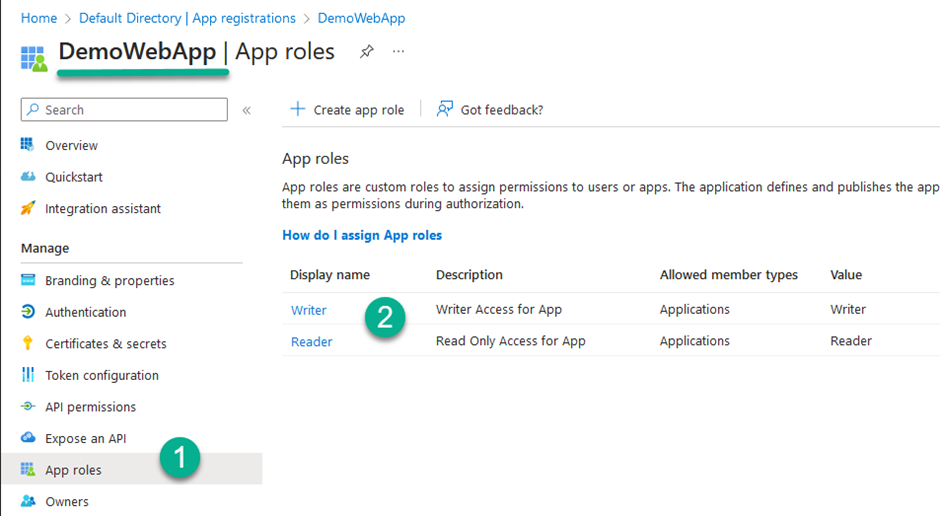


**Application Permission**

In client credential flow the application acts on its own with no user signed in. In the scenario where there is no signed-in user present, App-only access uses app roles instead of delegated scopes. App roles are also referred to as application-only permissions or application permissions.

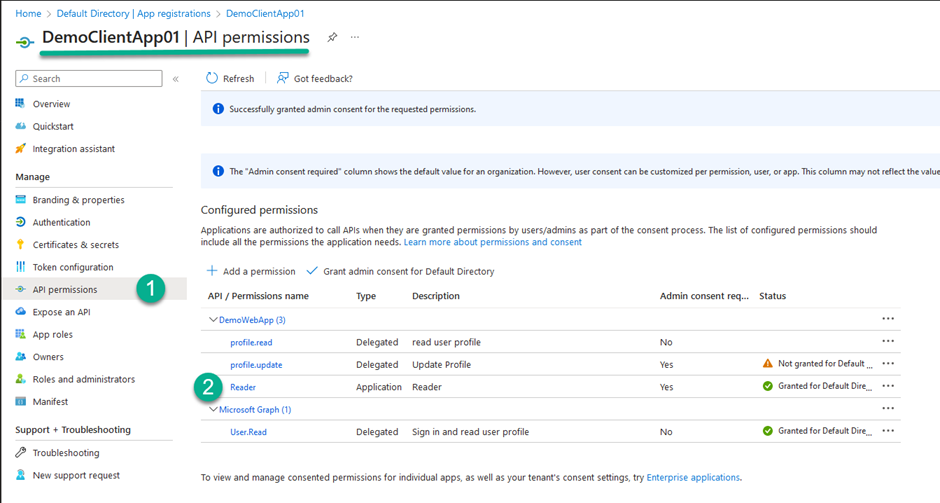
**Add application roles**

From Azure Active directory navigate to DemoWebApp and add an app role from **App Roles > Create app role**and create Reader and Writer Applications roles.



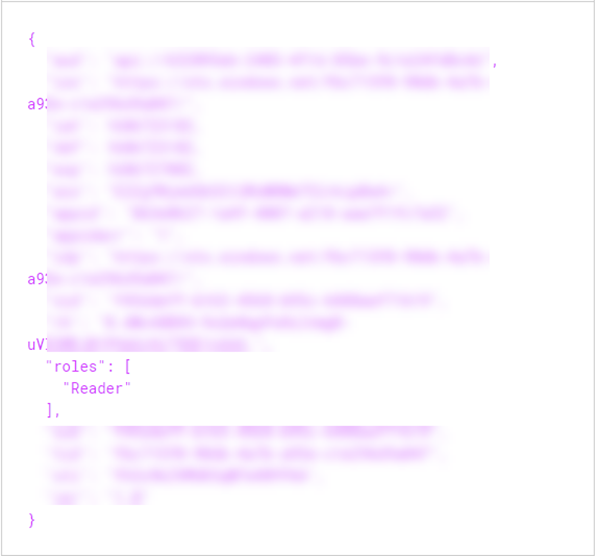
**Add application permissions**

From Azure Active directory navigate to DemoClientApp01 and add permission from **API Permissions > Add a permission > My APIs**and select the DemoWebApp. Select **Application Permission** from type of permissions. Choose **Reader** application permission and click Add Permissions. Grant admin consent for **Reader** role.



**Access token using postman**

Use the OAuth parameters as mentioned in the above section and use client credential as a grant type to acquire a new access token. When you decode the access token a new role **Reader** is added in the token.



Note: You cannot receive both Application and Delegated permissions ([Part 2B](https://medium.com/@shoaib.alam/part-2b-oauth-2-0-authorization-code-grant-with-azure-ad-d97b213ecc9)) in the same token, it is an either/or scenario. Which type you receive depends entirely on which OAuth Grant you used to request the token:

*Authorization Code: Delegated Permissions in the token with a scp claim*

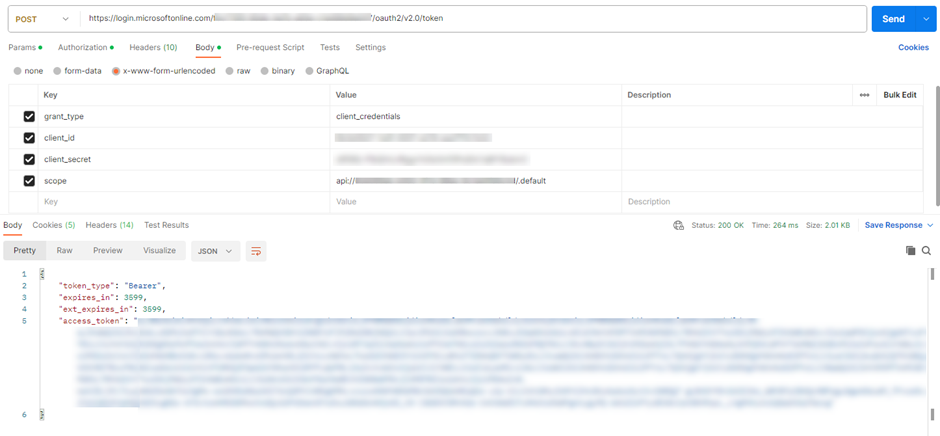
*Client Credentials: Application Permissions in the tokens with roles claim*

**Get access token using POST Request**

Open Postman and create the HTTP POST request which requires the followings for Body.

* grant\_type
* client\_id
* client\_secret
* Scope

Send the POST request to the token end point and it will generate an access token.



For postman – Redirect URL

[https://www.getpostman.com/oauth2/callback](https://www.getpostman.com/oauth2/callback.)