Google Cloud

Edge Security

OAuth - Client Credentials

OAuth grants

An OAuth Grant is a credential representing the resource owner's authorization. More often than not, we tend to think of grants in terms of the process used to obtain an access token.

Grant Type	Typical Use Case	Complex?
No specific resource owner is involved		
Client Credentials	Business system interactions, where resources being operated on are owned by the partner, not a particular user	No
A specific resource owner is involved		
Resource Owner Password Credentials	Resources are owned by a particular user and the requesting application is trusted	A bit
Authorization Code	Resources are owned by a particular user and the requesting application is untrusted	Very
Implicit	Resources are owned by a particular user, and the requesting application is an untrusted browser-based app written in a scripting language such as JavaScript	Very, and potentially insecure as well

Client credentials grant type

Resources participating in client credentials grant type

- Business Partner System Consuming API
- Apigee Generating and Validating Token
- Backend API Resource

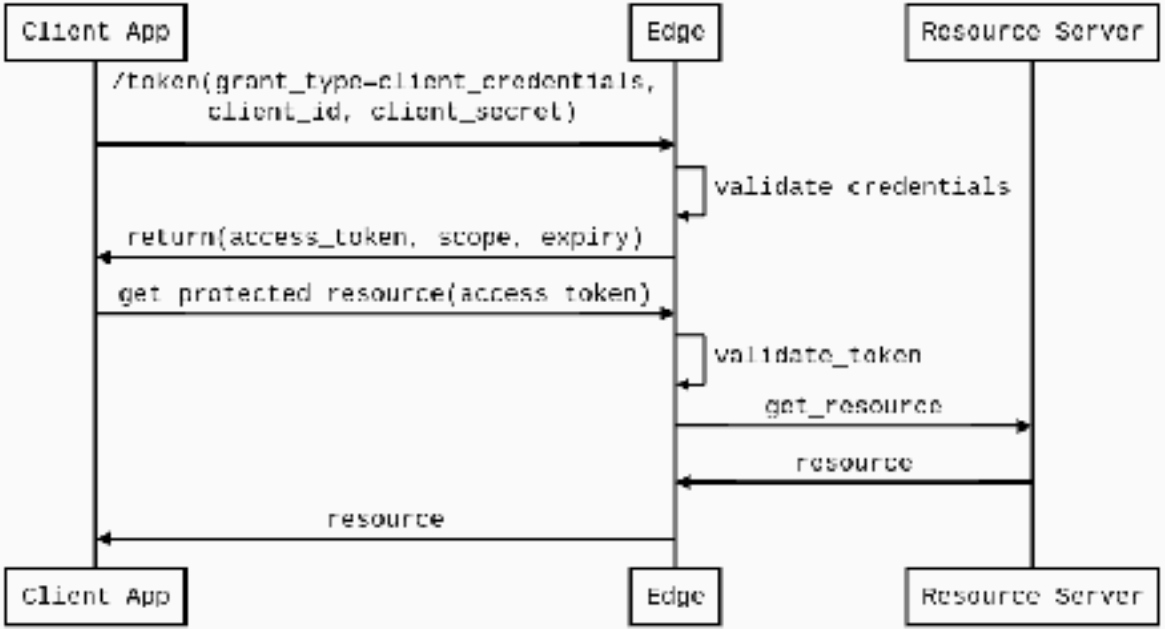
No specific resource owner is involved. Typically server to server.

Least complex to implement

Least secure among all grant types – Useful for *trusted apps* in some situations

No Refresh Token Generated

Client credentials - Sequence diagram



Generate access token policy

Generate tokens for the client credentials grant using the OAuthV2 policy.
Be sure to set the token endpoint to use SSL.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OAuthV2 async="false" continueOnError="false" enabled="true" name="oauth-generate-token">
    <DisplayName>OAuth Generate  Token</DisplayName>
    <Operation>GenerateAccessToken</Operation>
    <ExpiresIn>86400000</ExpiresIn>
    <SupportedGrantTypes>
      <GrantType>client_credentials</GrantType>
    </SupportedGrantTypes>
    <GrantType>request.queryparam.grant_type</GrantType>
  <GenerateResponse/>
</OAuthV2>
```

Using client credentials

Developer passes Base64 encoded Basic Auth request (client_id:client_secret) to generate access_token

Request:

```
http://api.yourcompany.com/oauth/token?
grant_type=client_credentials
```

Header:

```
Authorization: Basic
d293IHlvdSByZWFsbHkgdG9vayB0aGUgdGltZSB0b
yBkZWNvZGUgdGhpcz8g
```

Response:

```
{
    "issued_at": "1393889380896",
    "application_name": "030fdcea-
cf97-12084aea513c",
    "scope": "",
    "status": "approved",
    "api_product_list": "[weather]",
    "expires_in": ”86400",
    "developer.email": "tesla@weathersample.com",
    "organization_id": "0",
    "token_type": "BearerToken",
    "client_id": "RqBca4HGxdyaDM6AAPIHfQ53kLLIGFMf",
    "access_token": "4WCAchNNtVyK8JsACl1HP7mlWW1X",
    "organization_name": "jokeindex",
    "refresh_token_expires_in": "0",
    "refresh_count": "0"
}
```

Verify OAuth token policy

The OAuthV2 policy's "VerifyAccessToken" operation will validate the access token for subsequent requests for all grant types.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<OAuthV2 async="false" continueOnError="false" enabled="true" name="VerifyOAuthToken">
    <DisplayName>OAuth Verify Token</DisplayName>
    <Operation>VerifyOAuthToken</Operation>
</OAuthV2>
```

Set the access token as the Bearer token in the Authorization header of the http request.

```
curl -H "Authorization: Bearer {access_token}"
http://myorg-test.apigee.net/v1/cc/oauth_cc_weather/forecastrss?w=12797282
```

THANK YOU