Authorization Code Grant (Oauth 2.0) (link)

Step 0: Prerequisites (setup)

Your application should have an Integration Key (/go to admin settings/add integration key/add)

define a redirect URI, Secret Key. Example:

Secret key (auto): b66f1aaf-e4c6-4b90-b460-afb22c6ec482
Integration Key (auto): 85bc2dcc-76b7-4911-9520-ab8ca091339d
Redirect URL: http://localhost:3000/callback

Step 1: Request the Authorization Code (Authorization)

@developer purpose: https://account-d.docusign.com/oauth/auth

https://account-d.docusign.com/oauth/auth?
response_type=code
&scope=YOUR_REQUESTED_SCOPES
&client_id=YOUR_INTEGRATOR_KEY
&redirect_uri=YOUR_REDIRECT_URI

(optional)
&state=YOUR_CUSTOM_STATE (more protection against Cross site forgery)

scope: space separted list of **signature**, **extended** (scope for refresh token), **impersonation** (if using JWT)

Example:
https://account-d.docusign.com/oauth/auth
?response_type=code
&scope=signature%20extended%20impersonation
&client_id=7c2b8d7e-83c3-4940-af5e-cda8a50dd73f
&state=a39fh23hnf23
&redirect_uri=http://example.com/callback/

Application Consent

When an application authenticates to perform actions on behalf of a user, that user will be asked to grant consent Once a user has granted consent to an application, they will no longer be prompted to do so again unless consent is revoked.

After consent has been granted, the Authentication Service returns authentication code to the provided callback URI:

http://example.com/callback/
?code= LONG CODE (valid for 2 minutes)
&state=a39fh23hnf23 (only present if provided initially)

Step 2: Obtain the Access Token (Authentication)

Content-Type: application/x-www-form-urlencoded

curl --header "Authorization: Basic BASE64_COMBINATION_OF_INTEGRATOR_AND_SECRET_KEYS"
--data "grant_type=authorization_code&code=YOUR_AUTHORIZATION_CODE"
--request POST https://account-d.docusign.com/oauth/token

base64 value can be obtained as : btoa(INTEGRATOR_KEY + ':' + SECRET_KEY)

Response:


```
{
  "access_token": "ISSUED_ACCESS_TOKEN",
  "token_type": "Bearer",
  "refresh_token": "ISSUED_REFRESH_TOKEN",
  "expires_in": 28800
}
```


Step 3: Retrieve User Account Data

You will need the account_id and the base_uri claims of the user that your application is acting on behalf on to make calls to the DocuSign API.

curl --request GET https://account-d.docusign.com/oauth/userinfo
--header "Authorization: Bearer YOUR_ACCESS_TOKEN

response


```
{
  "sub": "564f7988-0823-409a-ac8a-781ee556ab7a",
  "name": "Example J Smith",
  "given_name": "Example",
  "family_name": "Smith",
  "created": "2018-04-13T22:03:03.45",
  "email": "Example.Smith@exampledomain.com",
  "accounts": [
    {
      "account_id": "18b4799a-b53a-4475-ba4d-b5b4b8a97604",
      "is_default": true,
      "account_name": "ExampleAccount",
      "base_uri": "https://demo.docusign.net"
    }
  ]
}
```


Step 4: Call the eSignature REST API

To make an eSignature REST API request, you must include:

- The acquired access token in the request's Authorization header
- The base URI for the request, built from the account_id and the base_uri of the user account

Example:

curl -- request GET https://account-d.docusign.com/restapi/v2/accounts/{accountId}/brands
--header "Authorization: Bearer YOUR_ACCESS_TOKEN

accountId (GUID) could be like : 18b4799a-b53a-4475-ba4d-b5b4b8a97604

Additional Details

https://developers.docusign.com/esign-rest-api/guides/authentication/oauth2-code-grant#using-refresh-tokens

JWT (link)

A service integration integrates directly with a DocuSign account and does not authenticate every end user. Instead, the integration obtains permission to impersonate (act as) specific users on a long-term basis

For example, the service integration would likely automate the process of sending the onboarding documents so that the HR alias or manager would not need to manually send the documents each time a new employee was hired.

JWT bearer authentication may impersonate one or more users at will, it can involve a high degree of granted trust. If your application does not need impersonation access or to perform automated operations, use the Authorization Code instead.

The JWT bearer flow does not grant a refresh token. When your access token has expired, you must generate a new one by repeating the JWT Grant flow.

Step 0: Prerequisites (setup)

Your application has an Integration Key (/go to admin settings/add integration key/add)

define a redirect URI, RSA keypair.

Save both the Public and Private keys to a secure place, including the ----BEGIN RSA PRIVATE KEY---- and ----END RSA PRIVATE KEY----- lines as part of the text.

Step 1: Request Application Consent (similar to step 1 of Authorization Code Grant)

If your application will be impersonating a user to make API calls, you will need to get their consent See Obtaining Consent for details on how to obtain consent from individuals and use the Organization Admin panel to obtain admin consent for groups of users.

To request impersonation consent from an individual user, redirect the user's browser to the authorization service URI

```
https://account-d.docusign.com/oauth/auth?
        response_type=code&
        scope=signature%20impersonation
        &client_id=7c2b8d7e-83c3-4940-af5e-cda8a50dd73f
        &redirect_uri=https://client.example.com/callback
```

The user will log in to their DocuSign account and be presented with a request to grant signature and impersonation permissions to your app.

Step 2: Create a JWT Token
Step 3: Obtain the Access Token (authentication)

If successful, an access token will be returned in the response body.

Step 4: Retrieve User Account Data
Step 5: Call the eSignature REST API