B2C Okta/AWS Cognito - Authentication & Authorization Flow

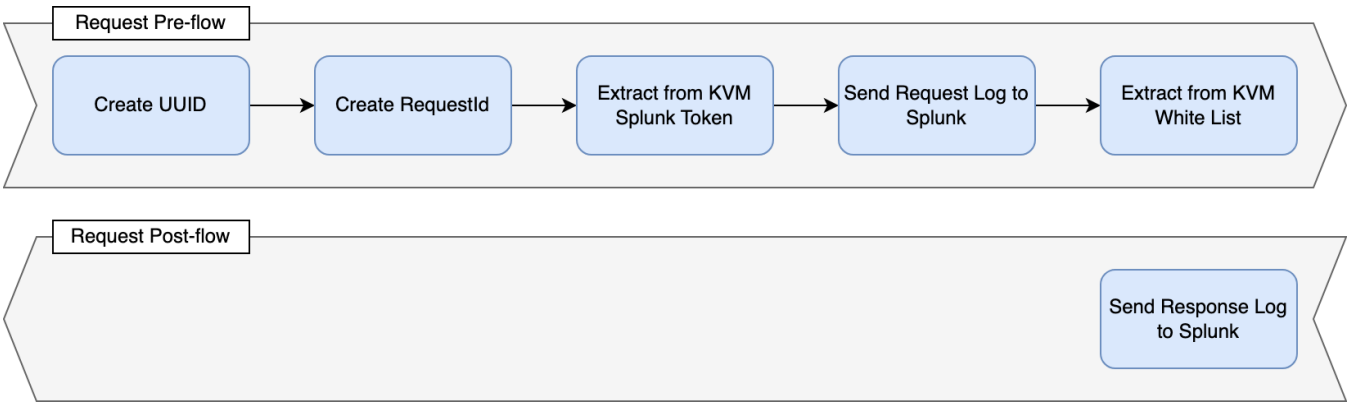
- Overview
- High Level Design
- Authentication Process
- Specifications
    - Version
    - Path
    - Endpoints, HTTP Verb, Headers, Body and Query Params table
    - Response when Successful
    - Response when Fail
- Proxy Dependencies
    - Target Server
    - KVMs
    - TLS Key Store
    - Virtual Host
    - Cache
- Bitbucket Repository
- Tests Knowledge
- Apigee Pipeline
    - Differences Between the V1 and V2 versions of the proxy EITS-Authorization-Oauth2
- Pendencies

Overview

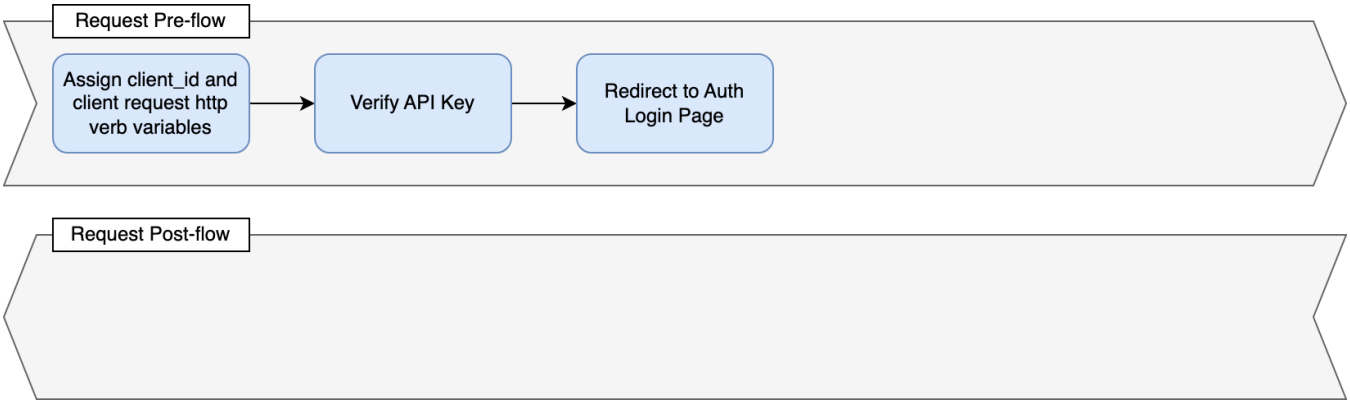
This documentation is to help APICoE team learn how works the Apigee Proxy **EITS-Authorization-Oauth2-V2**.

High Level Design

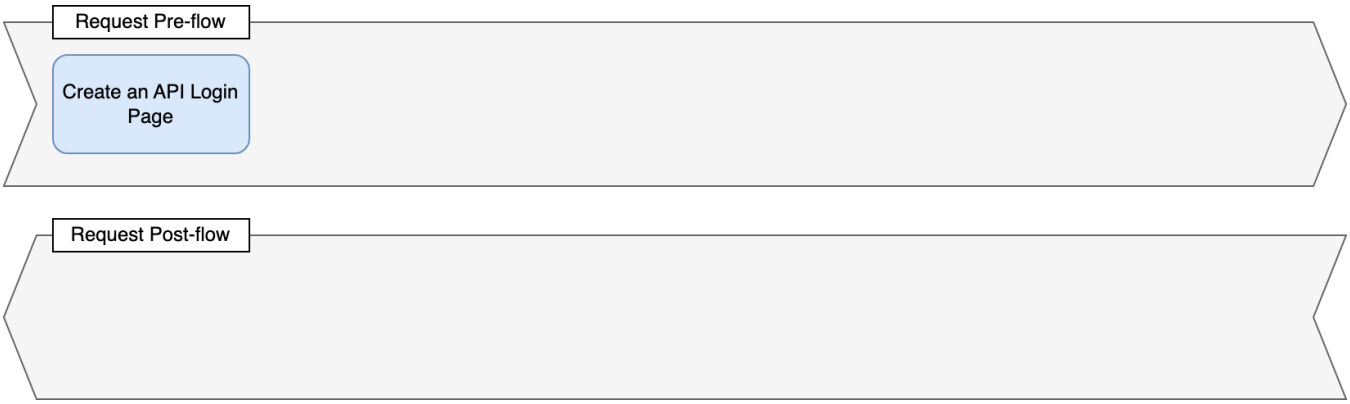
The diagram below has the components/policies that happens for any API endpoint requested.



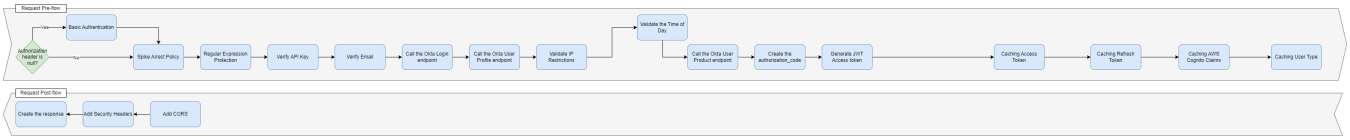
The diagram below is to the endpoint GET /authorize (Authorize)



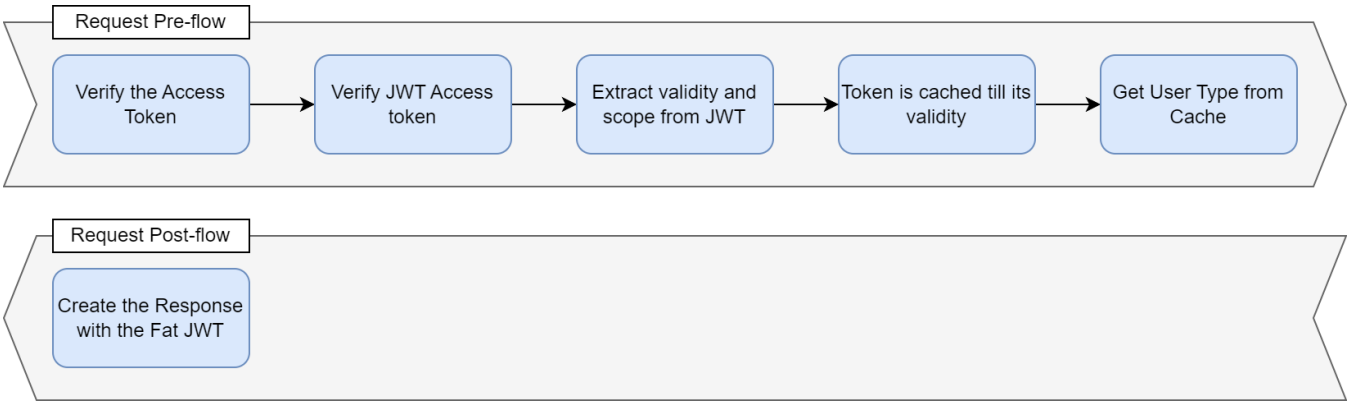
The diagram below is to the endpoint GET /login (Auth Login Page)



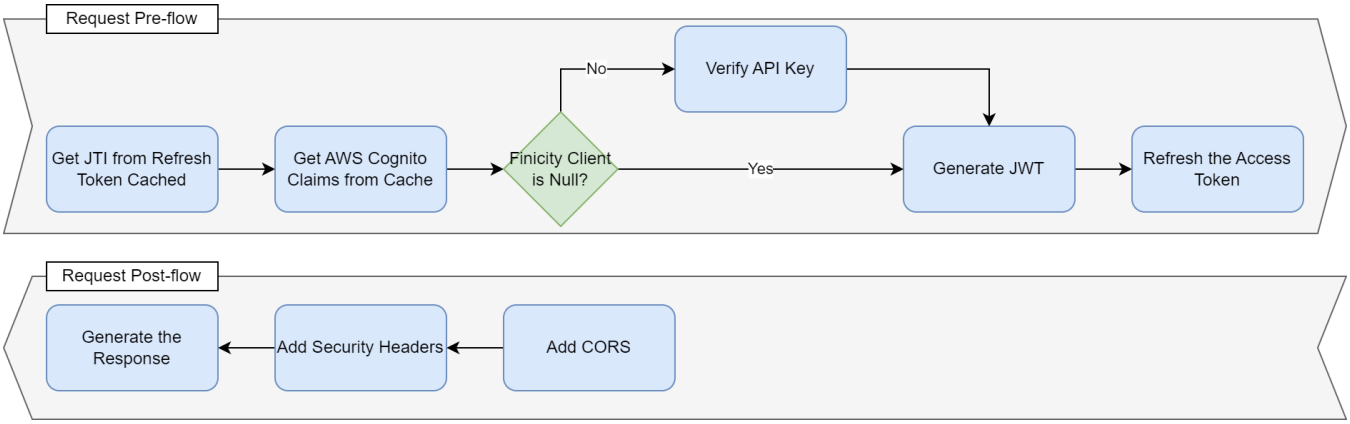
The diagram below is to the endpoint POST /token (JSON Web Token)



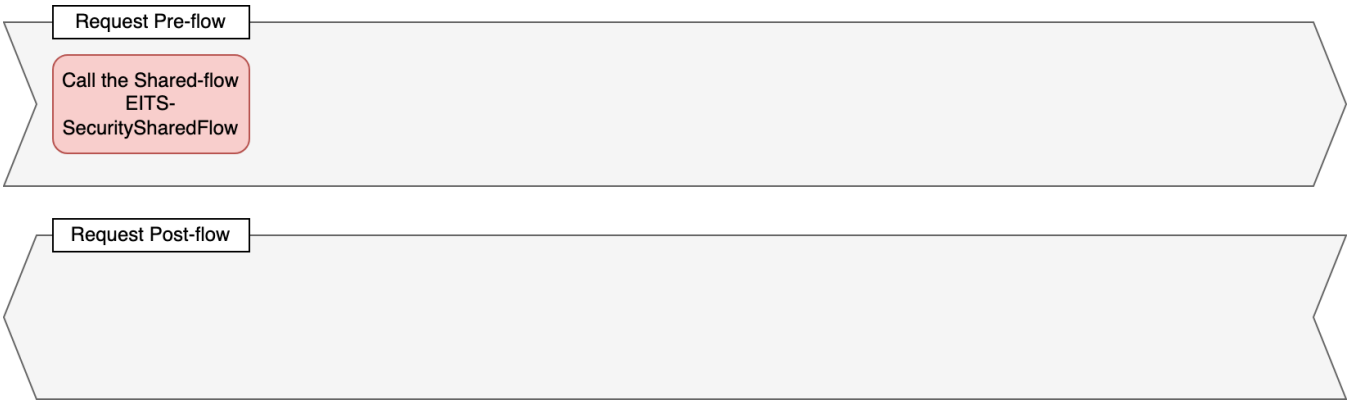
The diagram below is to the endpoint GET /showjwt (Show the Cached Fat JWT)



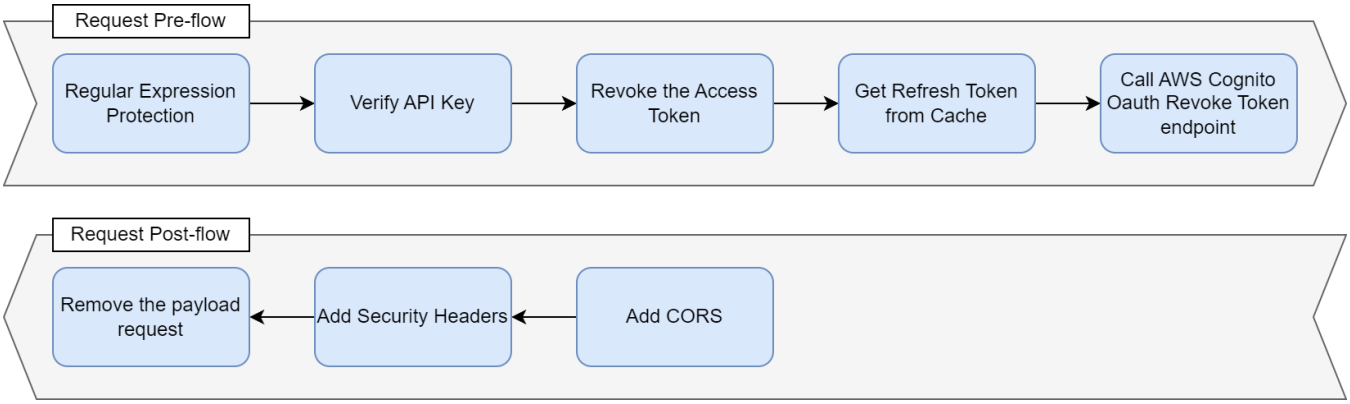
The diagram below is to the endpoint POST /token (Refresh Token)



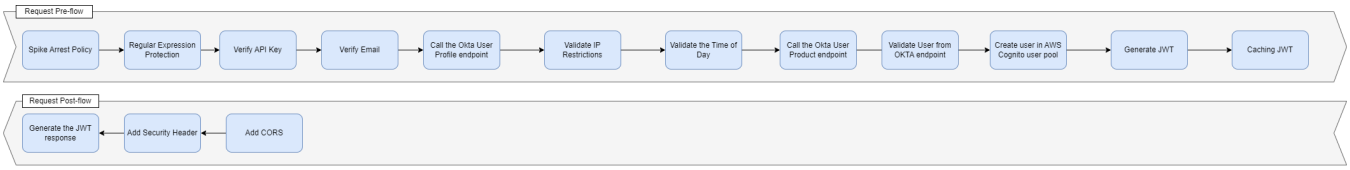
The diagram below is to the endpoint GET /checkValidity (Check Thin JWT)



The diagram below is to the endpoint POST /revokeToken (Revoke Thin JWT)



The diagram below is to the endpoint POST /token (Client Credentials Authentication)



Authentication Process

The authentication process to generate the access token depends of which endpoint you will call, the most common endpoint called by the customers is the '**JSON WEB Token Request**' endpoint. This endpoint requires the **client_id** and **client_secret** from AWS Cognito user pool App Client, authorization code and redirect uri for generating access and refresh token . To understand more about how to create a request to consume this endpoint, check below the '**Endpoints, HTTP Verb, Headers, Body and Query Params table**' to learn how to do that.

Specifications

Version

- Version: /v2

Path

- Path: /oauth2
- Full path with URL and Version: https://{ENVIRONMENT-ORGANIZATION}-api.experian.com/oauth2/v2

Endpoints, HTTP Verb, Headers, Body and Query Params table

<table>
<tr><td>1</td><td>Authorize</td><td>/authorize</td><td>GET</td><td><table><tr><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td>client_id</td><td>{{client_id}}</td><td>Yes</td></tr><tr><td>2</td><td>client_secret</td><td>{{client_secret}}</td><td>Yes</td></tr><tr><td>3</td><td>response_type</td><td>token</td><td>Yes</td></tr><tr><td>4</td><td>state</td><td>{{state}}</td><td>Yes</td></tr><tr><td>5</td><td>code</td><td>{{code}}</td><td>Yes</td></tr><tr><td>6</td><td>redirect_uri</td><td>{{redirect_uri}}</td><td>Yes</td></tr></table></td><td></td><td></td></tr>
<tr><td>2</td><td>Auth Login Page</td><td>/login</td><td>GET</td><td></td><td><table><tr><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td>client_id</td><td>{{client_id}}</td><td>Yes</td></tr><tr><td>2</td><td>response_type</td><td>token</td><td>Yes</td></tr><tr><td>3</td><td>client_secret</td><td>{{client_secret}}</td><td>Yes</td></tr><tr><td>4</td><td>code</td><td>{{code}}</td><td>Yes</td></tr><tr><td>5</td><td>state</td><td>{{state}}</td><td>Yes</td></tr><tr><td>6</td><td>redirect_uri</td><td>{{redirect_uri}}</td><td>Yes</td></tr></table></td><td></td></tr>
<tr><td>3</td><td>Show Cached JWT</td><td>/showjwt</td><td>GET</td><td><table><tr><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td>authorization</td><td>Bearer {{access_token}}</td><td>Yes</td></tr><tr><td>2</td><td>client_id</td><td>{{client_id}}</td><td>Yes</td></tr><tr><td>3</td><td>client_secret</td><td>{{client_secret}}</td><td>Yes</td></tr></table></td><td><table><tr><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td>product</td><td>{{product}}</td><td>Yes</td></tr></table></td><td></td></tr>
<tr><td>4</td><td>Check Access Token</td><td>/checkvalidity</td><td>GET</td><td><table><tr><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td>authorization</td><td>Bearer {{access_token}}</td><td>Yes</td></tr></table></td><td></td><td></td></tr>
<tr><td>5</td><td>JSON WEB Token Request</td><td>/token</td><td>POST</td><td><table><tr><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td>client_id</td><td>{{client_id}}</td><td>Yes</td></tr><tr><td>2</td><td>client_secret</td><td>{{client_secret}}</td><td>Yes</td></tr><tr><td>3</td><td>exp-system-info</td><td>{{exp-system-info}}</td><td>No</td></tr></table></td><td></td><td>JSON:

{
"username":"{{user_name}}",
"password":"
{{user_password}}"
}</td></tr>
<tr><td>6</td><td>Access Token</td><td>/token</td><td>POST</td><td><table><tr><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td>client_id</td><td>{{client_id}}</td><td>Yes</td></tr><tr><td>2</td><td>client_secret</td><td>{{client_secret}}</td><td>Yes</td></tr><tr><td>3</td><td>content-type</td><td>application/x-www-form-urlencoded</td><td>Yes</td></tr></table></td><td></td><td>Form-encoded:<table><tr><td></td><td></td><td></td></tr><tr><td>1</td><td>grant_type</td><td>password</td></tr><tr><td>2</td><td>access_token</td><td>{{access_token}}</td></tr></table></td></tr>
<tr><td>7</td><td>Revoke Access Token</td><td>/revoketoken</td><td>POST</td><td><table><tr><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td>client_id</td><td>{{client_id}}</td><td>Yes</td></tr><tr><td>2</td><td>client_secret</td><td>{{client_secret}}</td><td>Yes</td></tr><tr><td>3</td><td>token</td><td>{{token}}</td><td>Yes</td></tr></table></td><td></td><td>Empty JSON

{}</td></tr>
<tr><td>8</td><td>Client Credentials</td><td>/token</td><td>POST</td><td><table><tr><td></td><td></td><td></td><td></td></tr><tr><td>1</td><td>grant_type</td><td>client_credentials</td><td>Yes</td></tr><tr><td>2</td><td>client_id</td><td>{{client_id}}</td><td>Yes</td></tr><tr><td>3</td><td>client_secret</td><td>{{client_secret}}</td><td>Yes</td></tr></table></td><td></td><td>Empty JSON

{}</td></tr>
</table>

Response when Successful

	Description	Endpoint	Verb	Status Code	Response Example	Details
1	Authorize	/authorize	GET	302	None	This endpoint will redirect to 'Auth Login Page'
2	Auth Login Page	/login	GET	200	HTML Login Page	
3	Show Cached Fat JWT	/showjwt	GET	200	{ "access_token": "", "user_type": "" }	
4	Check Token	/checkvalidity	GET	200	None	
5	JSON WEB Token	/token	POST	200	{ "issued_at": "1694809674274", "expires_in": "1800", "token_type": "Bearer", "access_token": "", "refresh_token": "" }	
6	Refresh Token	/token	POST	200	{ "issued_at": "1694809674274", "expires_in": "1800", "token_type": "Bearer", "access_token": "", "refresh_token": "" }	
7	Revoke Token	/revoketoken	POST	200	None	
8	Implicit Token Auth	/token	POST	302	None	This endpoint will redirect to the URL defined on query param 'redirect_uri' with the access_token
9	Client Credentials Auth	/token	POST	200	{ "issued_at": "1694810104743", "expires_in": "1800", "token_type": "Bearer", "access_token": "" }	

Response when Fail

	Description	Endpoint	Verb	Status Code	Response Example	Example
1	Authorize	/authorize	GET	401	{ "errors": [{ "errorType": "Unauthorized", "message": "Access is denied due to invalid 'username' or 'password'. For further assistance, please contact Experian Helpdesk at 800-854-7201 or TSCAPISupport@experian.com" }], "success": false }	Error when you are missing the header client_id OR header response_type
2	Auth Login Page	/login	GET	401	{ "errors": [{ "errorType": "Unauthorized", "message": "Access is denied due to invalid 'username' or 'password'. For further assistance, please contact Experian Helpdesk at 800-854-7201 or TSCAPISupport@experian.com" }], "success": false }	Error when you are missing the Query param client_id

3				404	{
					"errors": [
					{
					"errorType": "Not Found",
					"message": "resource does not exist"
					}
					],
					"success": false
					}
						Error when you are missing the Query param respone_type

4	Show Cached JWT	/showjwt	GET	401	{
					"errors": [
					{
					"errorType": "Unauthorized",
					"message": "Access is denied due to invalid 'username' or 'password'. For further assistance, please contact Experian Helpdesk at 800-854-7201 or TSCAPISupport@experian.com"
					}
					],
					"success": false
					}
						Error when you are missing the header Authorization

5				500	{
					"errors": [
					{
					"errorType": "Internal Server Error",
					"message": "Internal Server Error. If problems persist, please contact apisupport@experian.com"
					}
					],
					"success": false
					}
						Error when you are missing the query param product

6	Check Token	/checkvalidity	GET	401	{
					"errors": [
					{
					"errorType": "Unauthorized",
					"message": "Access is denied due to invalid access token"
					}
					],
					"success": false
					}
						Error when you are missing the header Authorization

7	JSON Web Token Request	/token	POST	400	{
					"errors": [
					{
					"errorType": "Bad Request",
					"message": "The 'client_id' and 'client_secret' attributes are required"
					}
					],
					"success": false
					}
						Error when you are missing the header client_id OR client_secret

8				401	{
					"errors": [
					{
					"errorType": "Unauthorized",
					"message": "Access is denied due to invalid 'username' or 'password'. For further assistance, please contact Experian Helpdesk at 800-854-7201 or TSCAPISupport@experian.com"
					}
					],
					"success": false
					}
						Error when the body user OR password are wrong.

9				415	{
					"errors": [
					{
					"errorType": "Unsupported Media Type",
					"message": "Content-Type header is unsupported"
					}
					],
					"success": false
					}
						Error when you are missing the body JSON request

10				500	{
					"errors": [
					{
					"errorType": "Internal Server Error",
					"message": "Internal Server Error. If problems persist, please contact apisupport@experian.com"
					}
					],
					"success": false
					}
						Error when you are missing the body user OR password.

11	Refresh Token	/token	POST	400	{
					"errors": [
					{
					"errorType": "Bad Request",
					"message": "The 'client_id' and 'client_secret' attributes are required"
					}
					],
					"success": false
					}
						Error when you are missing the header client_id OR client_secret

12				400	{
					"errors": [
					{
					"errorType": "Bad Request",
					"message": "The 'username' and 'password' attributes are required"
					}
					],
					"success": false
					}
						Error when you are missing the form-encoded refresh_token OR grant_type attribute

13	Revoke Access Token	/revoketoken	POST	401	{
					"errors": [
					{
					"errorType": "Unauthorized",
					"message": "Access is denied due to invalid 'username' or 'password'. For further assistance, please contact Experian Helpdesk at 800-854-7201 or TSCAPISupport@experian.com"
					}
					],
					"success": false
					}
						Error when you are missing the header client_secret

14				401	{
					"errors": [
					{
					"errorType": "Unauthorized",
					"message": "Failed to resolve API Key variable request.header.client_id"
					}
					],
					"success": false
					}
						Error when you are missing the header client_id

15				500	{
					"errors": [
					{
					"errorType": "Internal Server Error",
					"message": "Internal Server Error. If problems persist, please contact apisupport@experian.com"
					}
					],
					"success": false
					}
						Error when you are missing the header token

16	Client Credentials	/token	POST	400	{
					"errors": [
					{
					"errorType": "Bad Request",
					"message": "The 'client_id' and 'client_secret' attributes are required"
					}
					],
					"success": false
					}
						Error when you are missing the header client_id OR client_secret

17				415	{
					"errors": [
					{
					"errorType": "Unsupported Media Type",
					"message": "Content-Type header is unsupported"
					}
					],
					"success": false
					}
						Error when you are missing the body JSON request

18				401	{
					"errors": [
					{
					"errorType": "Unauthorized",
					"message": "Access is denied due to invalid client id or client secret"
					}
					],
					"success": false
					}
						Error when you send a wrong header client_id OR client_secret

Proxy Dependencies

Target Server

No target server

KVMs

	Map Identifier	Name	Encrypted	Details
1	FINICITY_TOKEN_CUSTOMIZATION	ACCESS_TOKEN_EXPIRY_MILLIS	Yes	If the request header exp-System-info is equals Yes, the values from this KVM will be used
2		ACCESS_TOKEN_EXPIRY_SEC	Yes	
3		REFRESH_TOKEN_EXPIRY_MILLIS	Yes	
4				
5	AWS_Cognito_JWT_Keys	private	Yes	The values from this KVM are to generate the JWT or JWE
6				
7		public	Yes	
8	OKTA_API_KEY_ENCR	OktaAPIKey	Yes	The value from this KVM is necessary to Apigee-Okta integration. This is the authorization bearer token.
9	OKTA_OPEN_ID_CONNECT_ATTR	clientId	Yes	This KVM looks like it is not finished by the developer.
10		clientSecret	Yes	
11		AUTH_SERVER_ID_DEFAULT	Yes	
12	SPIKE_ARREST_RATE	AUTHORIZATION_CODE_GRANT	Yes	The values from this KVM are the reference rate to the policy Spike Arrest, for each endpoint is a different value
13		CLIENT_CREDENTIALS_GRANT	Yes	
14		REVOKE_GRANT	Yes	
15		REFRESH_GRANT	Yes	
16		PASSWORD_CHANGE	Yes	
17				
18	SPLUNK_TOKEN	AUTHORIZATION_TOKEN	Yes	The values from this KVM are to the integration between Apigee and Splunk
19		SPLUNK_URL	Yes	
20	IP_WHITELIST_CONFIG	HeaderToCheck	Yes	The values from this KVM are necessary to validate the IP range and to the validate the IP restrictions.
21		ExtraIPsToCheck	Yes	
22	TOKEN_EXPIRY_TIME	ACCESS_TOKEN_EXPIRY_MILLIS	Yes	The values from this KVM are necessary to the expiration time for the access token and refresh token.
23		ACCESS_TOKEN_EXPIRY_SEC	Yes	
24		REFRESH_TOKEN_EXPIRY_MILLIS	Yes	
25				

TLS Key Store

No applicable/No target Server

Virtual Host

- secure

Cache

	Resource	Prefix	Key Fragment	Details
1	JWT_Cache	Oauth	JWT	This cache keep the Fat JWT token, when the endpoint '**Show Cached JWT**' is called we retrieve the FAT Token using the JTI from the access token and the API Product from the query param.
2			JTI	
3			AWS API Product	
4	REFRESH_TOKEN_VS_JTI_CACHE	RTJTI	Refresh Token	This cache keep the refresh token
5	AWS_Cognito_CLAIMS_CACHE	Oauth	COGNITOCLAIMS	This cache keep the Cognito claims
6			JTI	
7	JWT_Cache	UserType	JTI	This cache keep the user type from Okta profile
8	JWE_Cache	Oauth	JWE	This cache keep the JWE, IF the custom attribute '**Cache_Encrypted_Payload**' from API Product, is equals YES.
9			Okta User	
10	JWT_Cache	OktaOauth	Okta Access Token	This cache keep the Okta access token
11	JWT_Cache	OktaOauth	Okta Refresh Token	This cache keep the Okta refresh token

Bitbucket Repository

Link: https://code.experian.local/projects/APDE/repos/eits-authorization-oauth2-v2/browse

Tests Knowledge

- Coverage Test
- Integration/Components Test
- Static Code Analysis
- Unit Test
- Tests Libraries Dependencies

Apigee Pipeline

- Apigee CI/CD Pipeline

Differences Between the V1 and V2 versions of the proxy **EITS-Authorization-Oauth2**

- The version 2 fixed the issues with Apigee name definitions good practice.
- The version 2 fixed the issue with the ServiceCallout when authenticating on Okta, this policy was overridden the value of the response flow variable.
- The version 2 removed the hard coded value EXPERIAN to the JWT claim ISS/ISSUER. Now it is dynamic, and the value is the host server like 'dev-us-api.experian.com/oauth2/v2'

Pendencies

- This proxy is missing the Integration tests