B2B Okta/AWS Cognito - Authentication & Authorization Flow

Overview

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

Flow Diagram

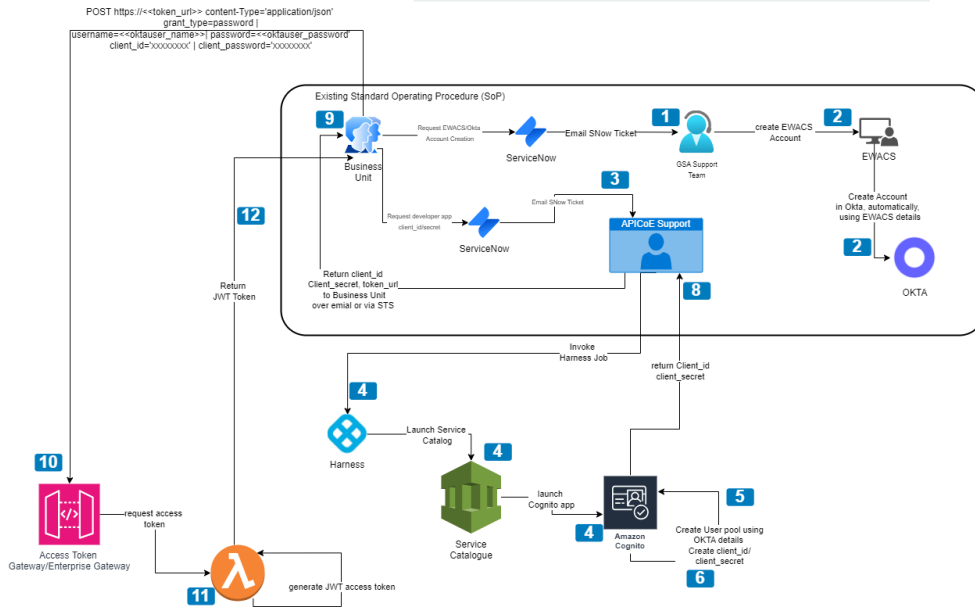### Onboarding a new B2B user

Pre-requisites

- The Business User is already vetted.

The numbered steps legend:

1. Business Unit will raise a ServiceNow request to onboard a new B2B user
2. The GSA Support team will then go ahead and create an user profile on OKTA
3. Business Unit will request client_id/client_secret, token_url for the developer app from APICoE team
4. APICoE team will Launch Cognito Client App via Service catalog
5. Create Client App in B2B-Cognito user pool based on B2B user profile details defined using drupal forms.
6. Client_id/client_secret is generated for the Cognito Client app
7. APICoE team maps Cognito Client application with respective API GW
8. AWS Cognito returns 'Client_id/client_secret' to APICoE team
9. APICoE team shares client_id/client_secret, token_url with Business Unit
10. Business user uses token_url to invoke Access Token Gateway using 'grant_type=password', Okta user/password
11. Access Token Gateway sends the 'access token request' to Enterprise Gateway
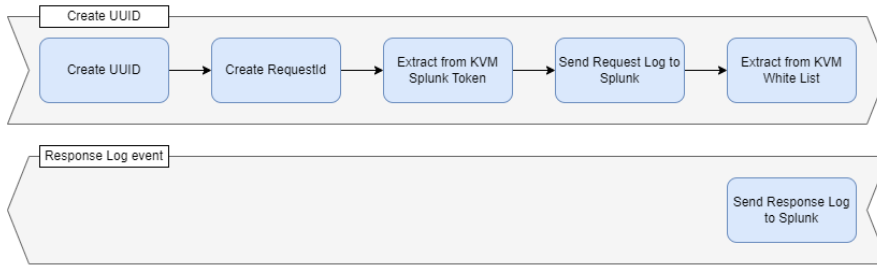12. Enterprise Gateway returns the 'JWT access token' back to the Business user

POST https://<<token_url>> content-Type='application/json'
grant_type=password |
username=<<oktauser_name>>| password=<<oktauser_password'
client_id='xxxxxxxx' | client_password='xxxxxxxx'

Existing Standard Operating Procedure (SoP)

Business Unit — Request EWACS/Okta Account Creation — ServiceNow — Email SNow Ticket — GSA Support Team — create EWACS Account — EWACS

Request developer app client_id/secret — ServiceNow — Email SNow Ticket — APICoE Support

Create Account in Okta, automatically, using EWACS details — OKTA

Return client_id Client_secret, token_url to Business Unit over emial or via STS

Return JWT Token

Invoke Harness Job

Launch Service Catalog — Harness

launch Cognito app — Service Catalogue — Amazon Cognito

return Client_id client_secret

Create User pool using OKTA details
Create client_id/ client_secret

request access token — Access Token Gateway/Enterprise Gateway

generate JWT access token

EWACS & OKTA Details (for step 2) for Service Now ticket	
Section	Select/Include
Region	<<Region Name>>
Description	Request EWACS account created in EWACS Production
Details	First Name: Last Name: Email: DL or team name Telephone User ID User Role: End User EWACS Product Name (to be assigned to the account) List of IPs (as per IP restriction requirement)
Assignment Group	Select "EITS IAM SSO Requests" for US account Select "EITS IAM EWACS Requests" for UK account

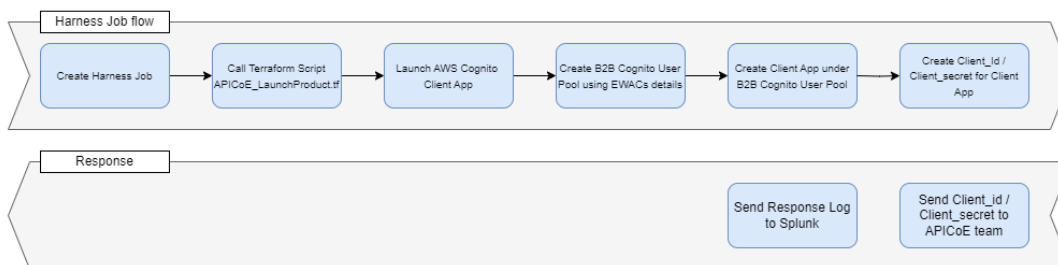| Developer App Creation (For step 3) for Service Now ticket	
Section	Select/Inclue
Region	Select Appropriate Region
Request Type	Select "Credentials"
Organization	Create appropriate organization
Environment	Select "Production"
Details	Apigee Developer App Creation: API Product Name (please be sure to provide the AWS Cognito Product name and not the EWACS product name. Even though they sometimes are the same, please be sure to validate it): Username (please provide EWACS created username): Email Address (please provide the email address linked to the EWACS username): Account Owner First and Last Name (Please provide as stated in EWACS username): *** If the client needs these credentials sent to an email address other than the one linked to the user, please be sure to state it in

Micro level flow diagram

Splunk Integration for Logging

This will invoke a UUID for Splunk to log events

Create UUID

| Create UUID | → | Create RequestId | → | Extract from KVM Splunk Token | → | Send Request Log to Splunk | → | Extract from KVM White List |

Response Log event
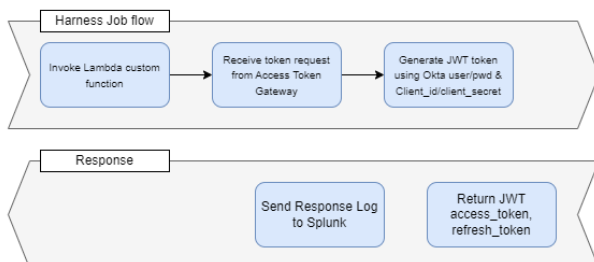
| Send Response Log to Splunk |

Harness Job flow:

This will trigger a 'APICoE_launchCognitoproduct.tf' that will launch the AWS Cognito Client App, create user pool, client_id/client_secret and send the client_credentials to the APICoE team

Harness Job flow

| Create Harness Job | → | Call Terraform Script APICoE_LaunchProduct.tf | → | Launch AWS Cognito Client App | → | Create B2B Cognito User Pool using EWACs details | → | Create Client App under B2B Cognito User Pool | → | Create Client_Id / Client_secret for Client App |

Response

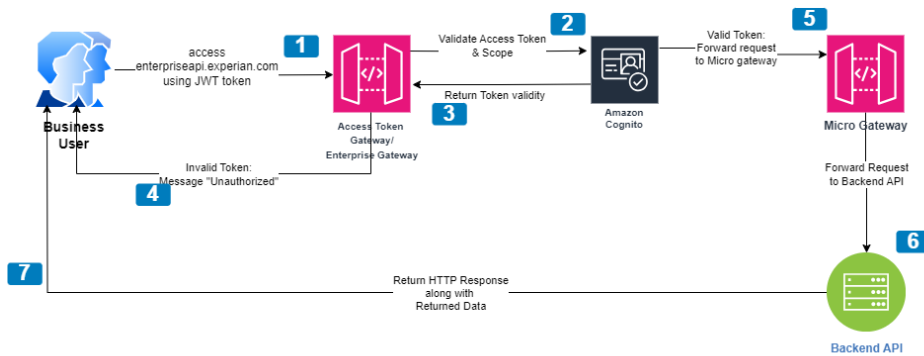| Send Response Log to Splunk | Send Client_id / Client_secret to APICoE team |

Lambda Custom Token generator:

This will triggered by the a 'APICoE_launchCognitoproduct.tf' that will launch the AWS Cognito Client App, create user pool, client_id/client_secret and send the client_credentials to the APICoE team

Harness Job flow

| Invoke Lambda custom function | → | Receive token request from Access Token Gateway | → | Generate JWT token using Okta user/pwd & Client_id/client_secret |

Response

| Send Response Log to Splunk | Return JWT access_token, refresh_token |

B2B User API Invocation Flow

Section	Select/Include	
EWACS & OKTA Details		
Section	Select/Include	
Region	<<Region Name>>	
Description	Request EWACS account created in EWACS Production	
Details	First Name: Last Name: Email: DL or team name Telephone User ID User Role: End User EWACS Product Name (to be assigned to the account) List of IPs (as per IP restriction requirement)	
Assignment Group	Select "EITS IAM SSO Requests" for US account Select "EITS IAM EWACS Requests" for UK account	

1 Access API using "enterpriseapi.experian.com" with JWT access_token passed as a request header

2 Validate Access token that API Gateway had generated and shared it with AWS Cognito, during B2B onboarding

3 AWS Cognito returns token_validity to Enterprisegateway

4 API Gateway receives token_validity response. If token is invalid, message "Unauthorized" is sent back to the Business User

5 If the API token is valid, forward API request to relevant micro gateway

6 Micro gateway forwards the API rquest to invoke the Backend API

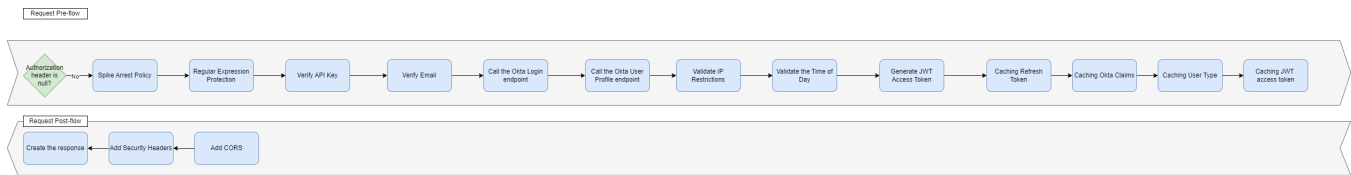**7** Backend Gateway return HTTP Response along with Returned Data

The diagram below enlists the components/policies applicable for any API endpoint requested.
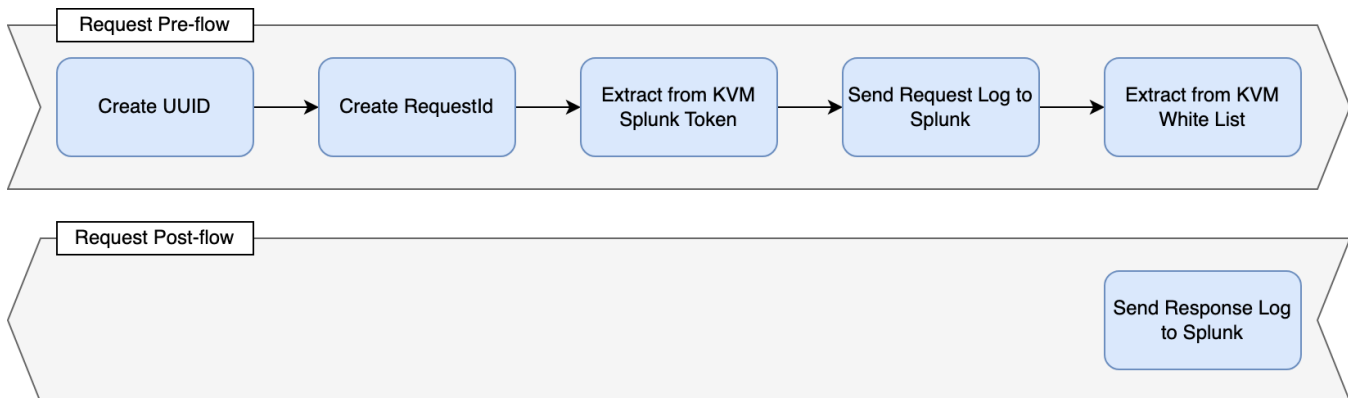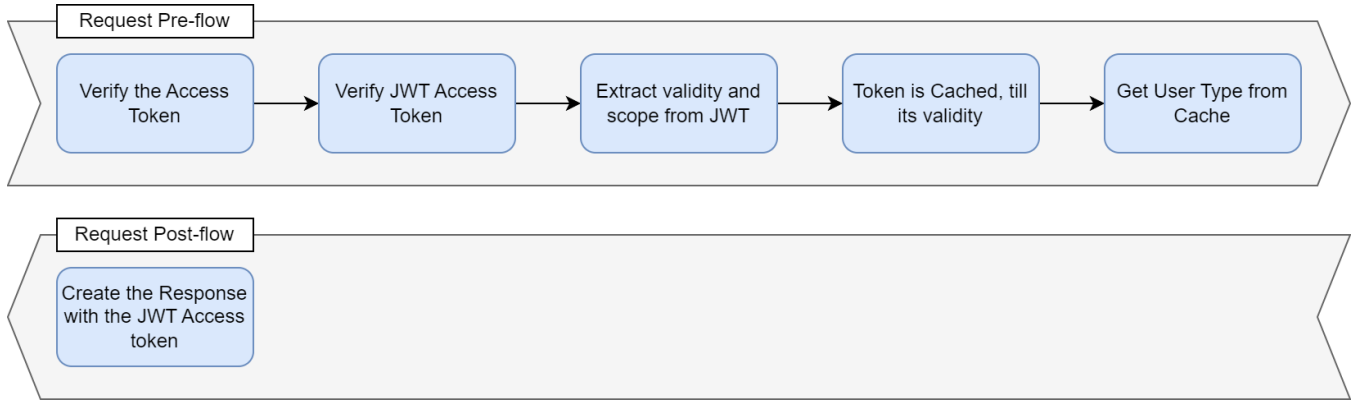

This diagram represents the endpoint GET /authorize (Authorize)
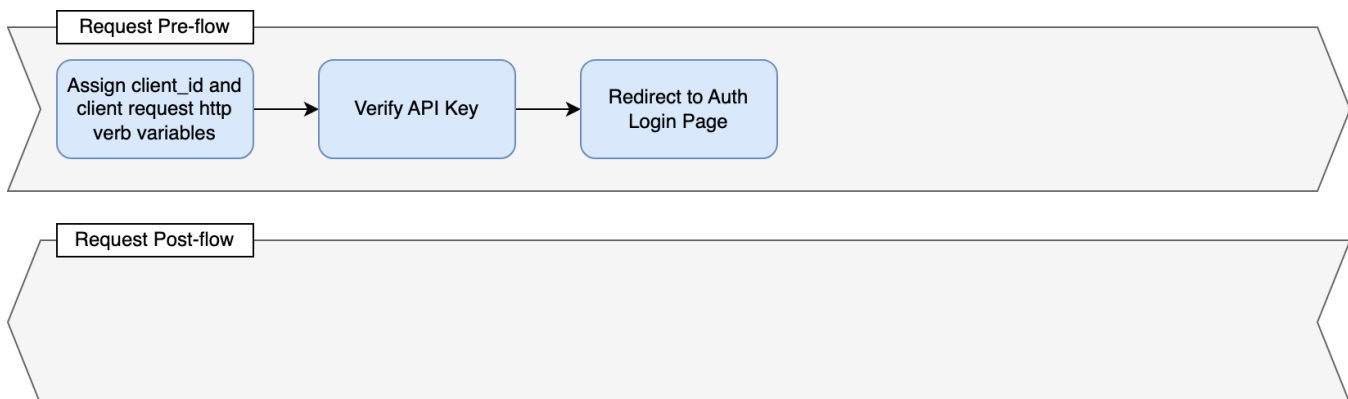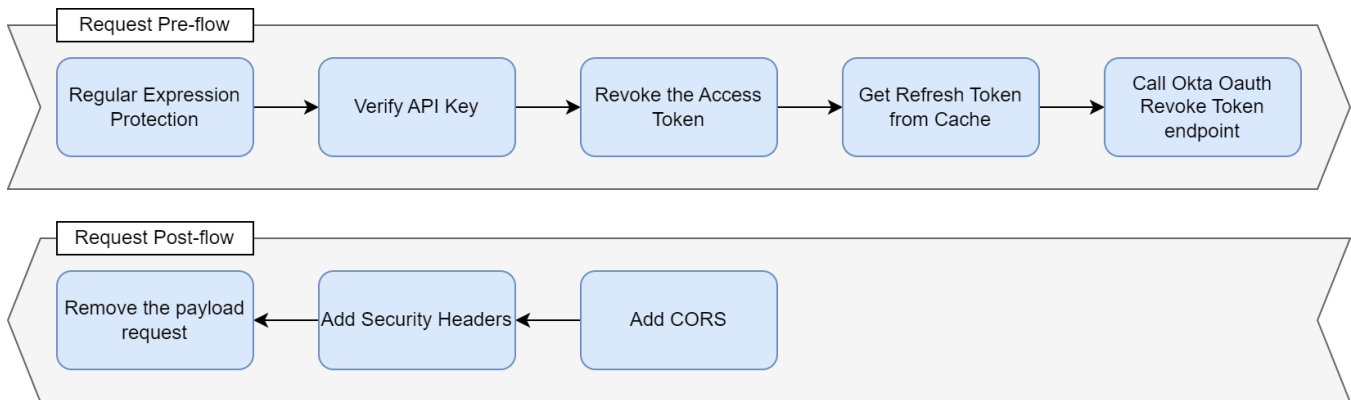

The diagram below represents endpoint POST /token (JSON Web Token) using grant_type=password
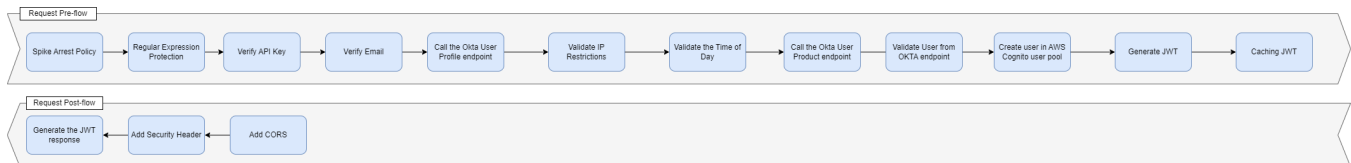
Authorization header is null? → Spike Arrest Policy → Regular Expression Protection → Verify API Key → Verify Email → Call the Okta Login endpoint → Call the Okta User Profile endpoint → Validate IP Restrictions → Validate the Time of Day → Generate JWT Access Token → Caching Refresh Token → Caching Okta Claims → Caching User Type → Caching JWT access token

Request Post-flow

Create the response ← Add Security Headers ← Add CORS

The diagram below represents the endpoint GET /showjwt

Request Pre-flow

Verify the Access Token → Verify JWT Access Token → Extract validity and scope from JWT → Token is Cached, till its validity → Get User Type from Cache

Request Post-flow

Create the Response with the JWT Access token

The diagram below represents the endpoint POST /revokeToken (Revoke JWT)

Request Pre-flow

Regular Expression Protection → Verify API Key → Revoke the Access Token → Get Refresh Token from Cache → Call Okta Oauth Revoke Token endpoint

Request Post-flow

Remove the payload request ← Add Security Headers ← Add CORS

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

Request Pre-flow

Spike Arrest Policy → Regular Expression Protection → Verify API Key → Verify Email → Call the Okta User Profile endpoint → Validate IP Restrictions → Validate the Time of Day → Call the Okta User Product endpoint → Validate User from OKTA endpoint → Create user in AWS Cognito user pool → Generate JWT → Caching JWT

Request Post-flow

Generate the JWT response ← Add Security Header ← Add CORS

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 App Client, and the user and password from Okta. 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

	Description	Endpoint	Verb	Headers	Query Params	Body
1	Authorize	/authorize	GET			
2	Auth Login Page	/login	GET			
3	Show Cached JWT	/showjwt	GET			
4	Check Access Token	/checkvalidity	GET			
5	JSON WEB Token Request	/token	POST			

1. Authorize — Headers

	Name	Value	Required?
1	client_id	{{client_id}}	Yes
2	client_secret	{{client_secret}}	Yes
3	response_type	token	Yes
4	state	{{state}}	Yes
5	code	{{code}}	Yes
6	redirect_uri	{{redirect_uri}}	Yes

2. Auth Login Page — Query Params

	Name	Value	Required?
1	client_id	{{client_id}}	Yes
2	response_type	token	Yes
3	client_secret	{{client_secret}}	Yes
4	code	{{code}}	Yes
5	state	{{state}}	Yes
6	redirect_uri	{{redirect_uri}}	Yes

3. Show Cached JWT — Headers

	Name	Value	Required?
1	authorization	Bearer {{access_token}}	Yes
2	client_id	{{client_id}}	Yes
3	client_secret	{{client_secret}}	Yes

3. Show Cached JWT — Query Params

	Name	Value	Required?
1	product	{{product}}	Yes

4. Check Access Token — Headers

	Name	Value	Required?
1	authorization	Bearer {{access_token}}	Yes

5. JSON WEB Token Request — Headers

	Name	Value	Required?
1	client_id	{{client_id}}	Yes
2	client_secret	{{client_secret}}	Yes
3	exp-system-info	{{exp-system-info}}	No

5. JSON WEB Token Request — Body

JSON:

```
{
    "username":"{{user_name}}",
    "password":"{{user_password}}"
}
```

6	Access Token	/token	POST							Form-encoded:		
					Name	**Value**	**Required?**				**Name**	**Value**
				1	client_id	{{client_id}}	Yes			1	grant_type	password
				2	client_secret	{{client_secret}}	Yes			2	access_token	{{access_token}}
				3	content-type	application/x-www-form-urlencoded	Yes					
7	Revoke Access Token	/revoketoken	POST							Empty JSON {}		
					Name		**Required?**					
				1	client_id	{{client_id}}	Yes					
				2	client_secret	{{client_secret}}	Yes					
				3	token	{{token}}	Yes					
8	Client Credentials	/token	POST							Empty JSON {}		
					Name	**Value**	**Required?**					
				1	grant_type	client_credentials	Yes					
				2	client_id	{{client_id}}	Yes					
				3	client_secret	{{client_secret}}	Yes					

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 JWT	/showjwt	GET	200	{ "access_token": "", }	
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": "", }	
6	Revoke Token	/revoketoken	POST	200	None	
7	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

#	Name	Endpoint	Method	Status	Response	Description

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_CUSTO MIZATION	ACCESS_TOKEN_EXPIRY_MIL LIS	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	Apigee_JWT_Keys	private	Yes	The values from this KVM are to generate the JWT or JWE
4		public	Yes	
5	OKTA_API_KEY_ENCR	OktaAPIKey	Yes	The value from this KVM is necessary to Apigee-Okta integration. This is the authorization bearer token.
6	OKTA_OPEN_ID_CONNE CT_ATTR	clientId	Yes	This KVM looks like it is not finished by the developer.
7		clientSecret	Yes	
8		AUTH_SERVER_ID_DEFAULT	Yes	
9	SPIKE_ARREST_RATE	RESOURCE_OWNER_PASSWO RD_GRANT	Yes	The values from this KVM are the reference rate to the policy Spike Arrest, for each endpoint is a different value
10		CLIENT_CREDENTIALS_GRANT	Yes	
11		REVOKE_GRANT	Yes	
12		REFRESH_GRANT	Yes	
13		PASSWORD_CHANGE	Yes	
14	SPLUNK_TOKEN	AUTHORIZATION_TOKEN	Yes	The values from this KVM are to the integration between Apigee and Splunk
15		SPLUNK_URL	Yes	
16	IP_WHITELIST_CONFIG	HeaderToCheck	Yes	The values from this KVM are necessary to validate the IP range and to the validate the IP restrictions.
17		ExtraIPsToCheck	Yes	
18	TOKEN_EXPIRY_TIME	ACCESS_TOKEN_EXPIRY_MIL LIS	Yes	The values from this KVM are necessary to the expiration time for the access token and refresh token.
19		ACCESS_TOKEN_EXPIRY_SEC	Yes	

20		REFRESH_TOKEN_EXPIRY_MILLIS	Yes	

TLS Key Store

Not 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			Apigee API Product	
3	OKTA_CLAIMS_CACHE	Oauth	OKTACLAIMS	This cache keep the Okta claims
4	JWT_Cache	UserType	JTI	This cache keep the user type from Okta profile
5	JWE_Cache	Oauth	JWE	This cache keep the JWE, IF the custom attribute '**Cache_Encrypted_Payload**' from API Product, is equals YES.
6			Okta User	
7	JWT_Cache	OktaOauth	Okta Access Token	This cache keep the Okta **access** token
8	JWT_Cache	OktaOauth	Okta Refresh Token	This cache keep the Okta **refresh** token