WF-04: Admin - Portal Customization

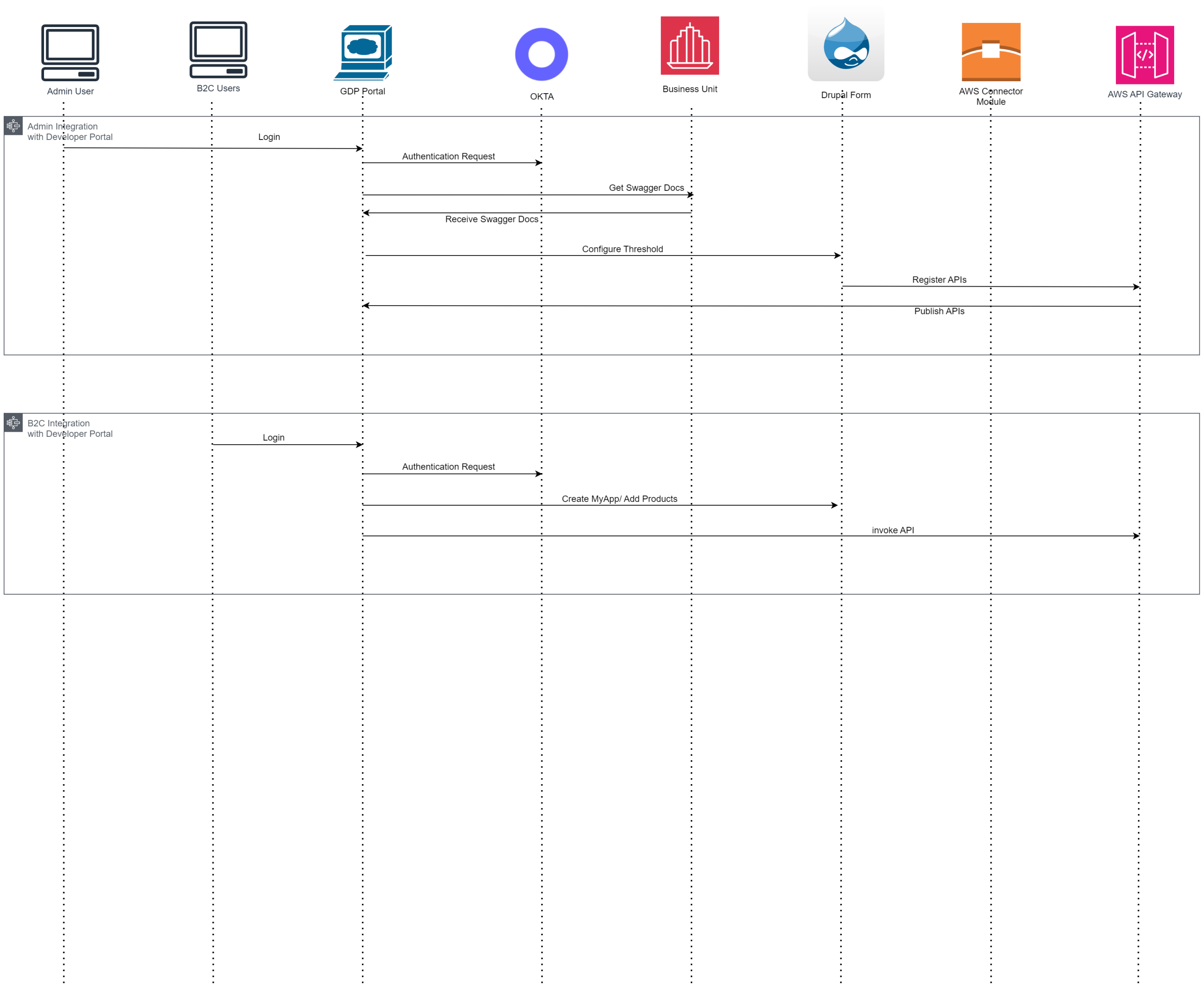
Updates

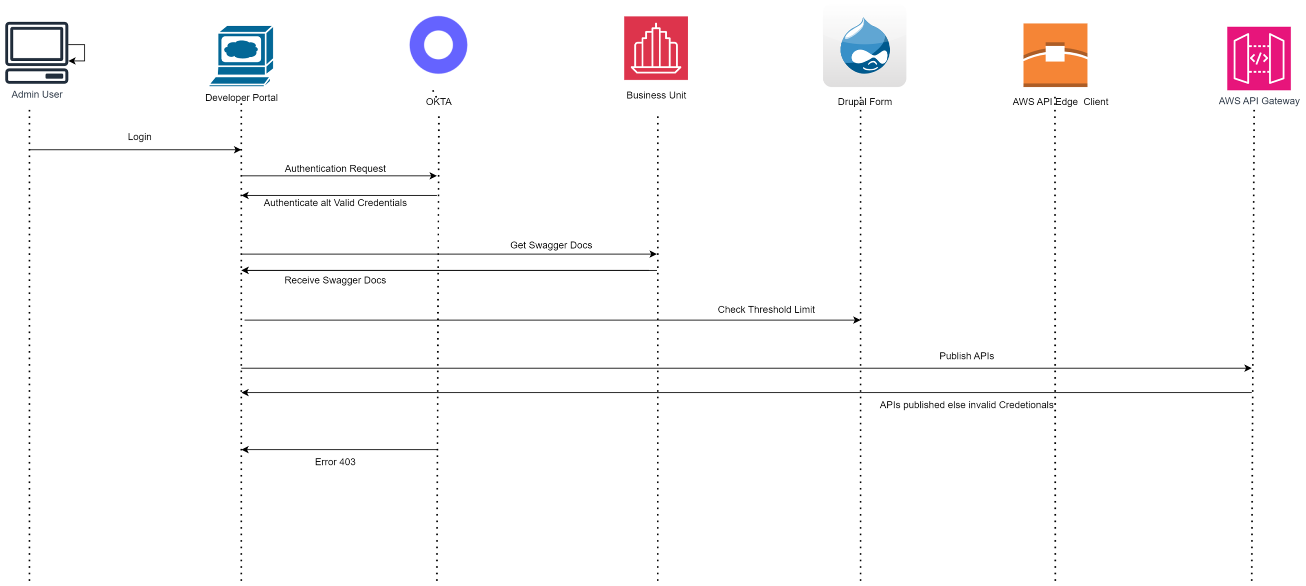
	Last Update	Jira Ticket	Author	Details
1	17 Nov 2023	EUAP-XXXX		
2				

Overview

This is the page to explain how works the Integration between the GDP (Developer Portal) Portal and the AWS Enterprise Gateway. Below you can see the authentication process using the OAuth grant type Authorization Code.

Sequence Diagram

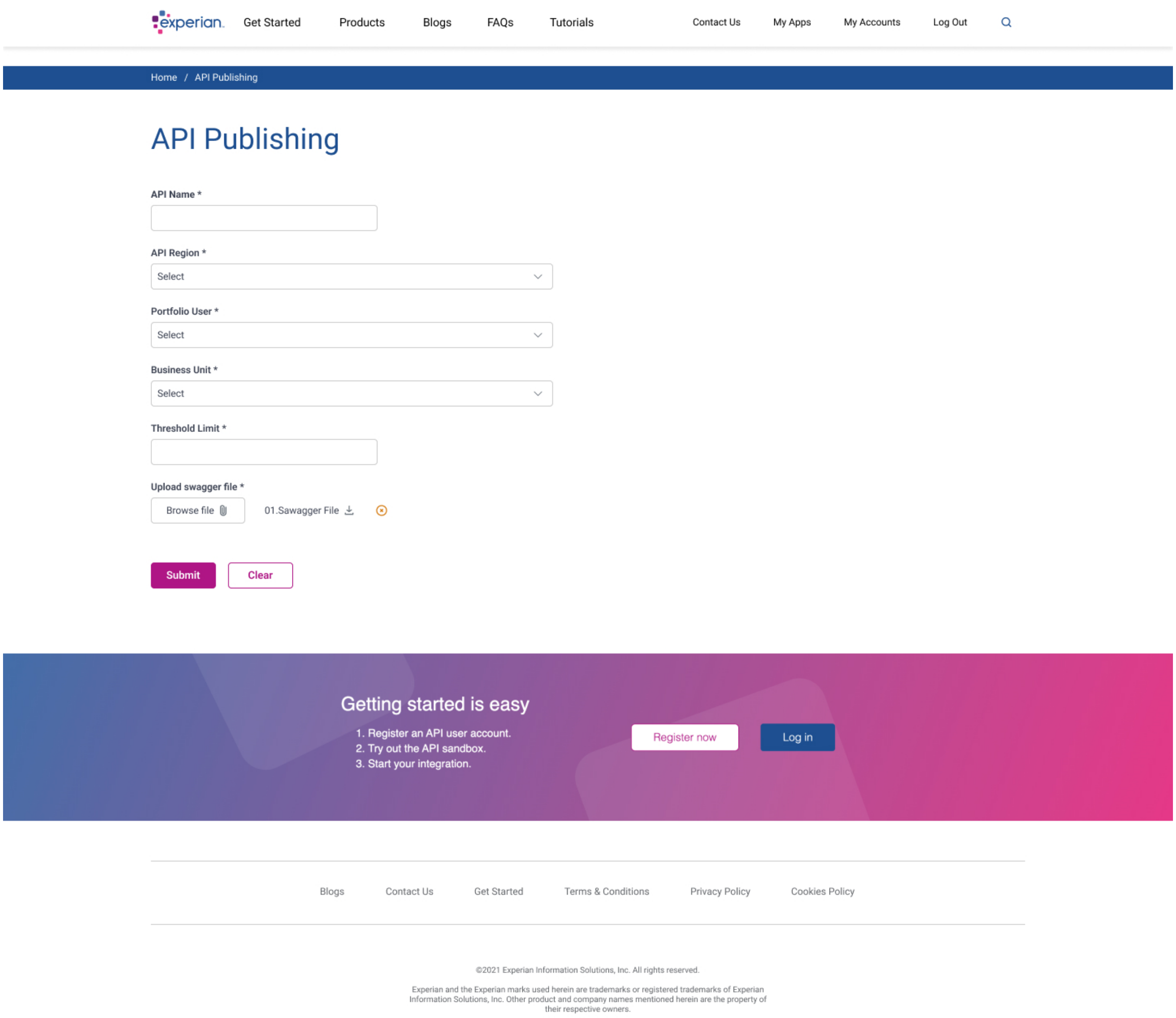




Sequence Diagram Details

1. Admin User
    a. Admin integration with Developer Portal
    b. Login
    c. ...
2. B2C User
    a. ...
3. ...

APICoEAdmin API Input Form.



API Publishing

API Name *

API Region *
Select

Portfolio User *
Select

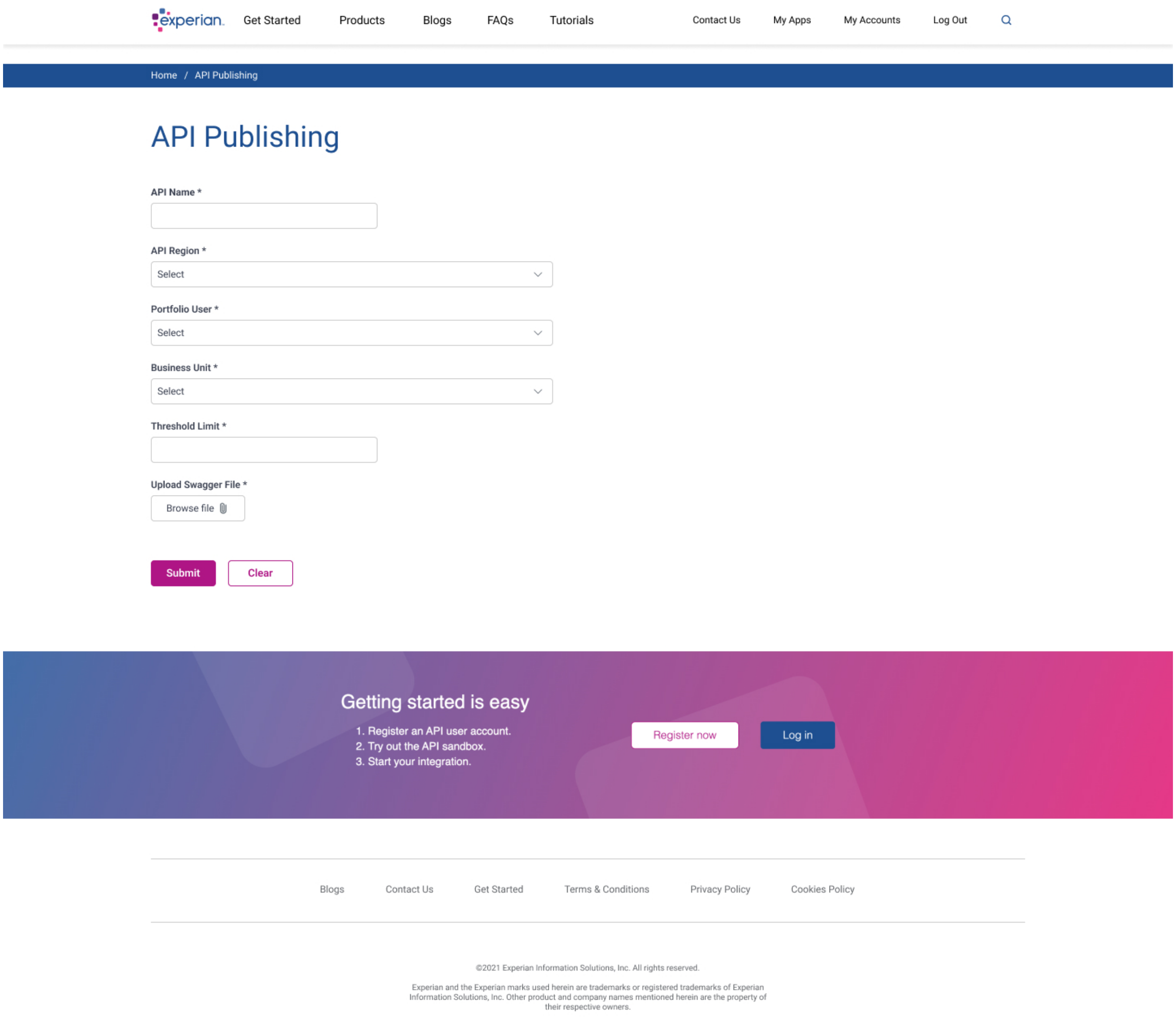
Business Unit *
Select

Threshold Limit *

Upload swagger file *
Browse file 01.Sawagger File

Getting started is easy

1. Register an API user account.
2. Try out the API sandbox.
3. Start your integration.



API Publishing

API Name *

API Region *
Select

Portfolio User *
Select

Business Unit *
Select

Threshold Limit *

Upload Swagger File *

Getting started is easy

1. Register an API user account.
2. Try out the API sandbox.
3. Start your integration.

©2021 Experian Information Solutions, Inc. All rights reserved.

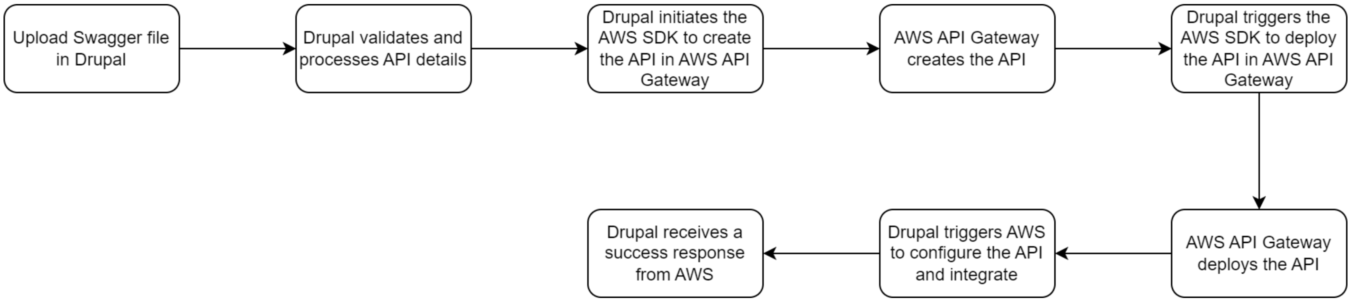
Experian and the Experian marks used herein are trademarks or registered trademarks of Experian Information Solutions, Inc. Other product and company names mentioned herein are the property of their respective owners.

<<ListRegionwiseallAPIs.>>

Another display form should get added that will list all the APIs w.r.t the Region, back to the APICoE Admin

FLOW Diagram

Micro-Level diagram Admin flow end-end(Drupal to AWS)


Upload Swagger file in Drupal
Drupal validates and processes API details
Drupal initiates the AWS SDK to create the API in AWS API Gateway
AWS API Gateway creates the API
Drupal triggers the AWS SDK to deploy the API in AWS API Gateway
AWS API Gateway deploys the API
Drupal triggers AWS to configure the API and integrate
Drupal receives a success response from AWS

Detailed Steps:

- Workflow-1: Publish APIs to AWS Enterprise Gateway using Drupal CMS. Steps are as follows
    - Admin user logs into Developer Portal. Redirect to OKTA or Authentication using JWT authentication type. Returns error '403' if credentials are invalid
    - Receive swagger Docs from Business Unit that needs to be published
    - Use Drupal form to check threshold limits and get API products per region
    - AWS API Edge Client will help publish APIs to API GW

- Log into Developer Portal (https://developer.experian.com) as an admin
- Install a AWS Client Connector app that will integrate Drupal and AWS API GW

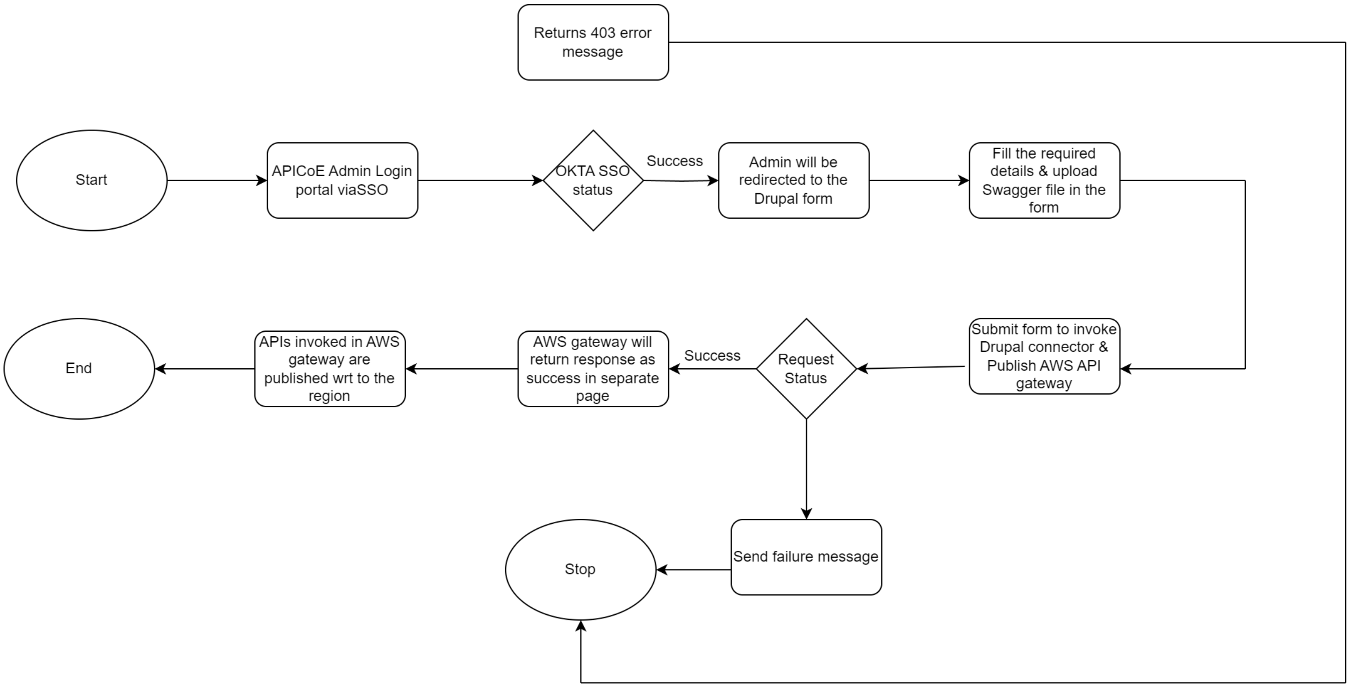
- Receive Swagger Docs from BU
- Open form "Enter API details". Enter information such as 'Region', 'Threshold limits' etc. Customize the APIs before publishing them by way of AWS API GW
  eg. -Restrict the exposure of specific resources, methods, and operations of an API to other applications.
  -Define a custom gateway endpoint by customizing the URL of the gateway endpoint that your users will use to access the API.
- Upload the swagger doc. This will export the APIs to AWS API GW and list them
- API Gateway allows you to publish APIs to Developer Portal from where they are available for consumption by developers and consumers.
  API Gateway also allows you to publish the APIs to the following destinations:
  Service registries. This enables applications to dynamically locate an API Gateway instance that can process that API.
  Integration Server. This is used in API first implementation approach.
- The following sections describe how you can activate an API, customize the gateway endpoint, and publish APIs to different destinations.
    - Activating an API
        - You must first activate the API before publishing it to a portal so that the gateway endpoint is available for developers and consumers to invoke the API.
        - You must have the Activate/Deactivate APIs functional privilege assigned to perform this task. You can activate an API in the Manage APIs page. Alternatively you can also activate the API from the API Details page.
        - The Gateway endpoint is now available, which can be used by the consumers of this API. You can now publish the API to the required destination and expose the API for consumption by the consumers.
    - Once the API is activated, you can define the custom gateway endpoints. For more information about gateway endpoints, see Gateway Endpoints.
    - Once the API is activated, you can enable the tracer. For more information about how to enable the tracer and view the tracing details, see Trace API.
    - Publishing an API to Developer Portal sends the SOAP and REST APIs to Developer Portal on which they are exposed for testing and user consumption. The process of publishing an API to Developer Portal is initiated from API Gateway and is carried out on the Developer Portal server. Doing this involves the following high-level steps:
        - You initiate the publish process by selecting the API to be published, specify the API endpoints to be visible to the consumers, and the Developer Portal communities in which the API is to be published.
        - API Gateway publishes the API to each of the specified Developer Portal communities.
        - During bulk publishing of APIs, the process continues even if API Gateway encounters a failure with Developer Portal.
    - When publishing an API to the Developer Portal destination, keep the following points in mind:
        - The Developer Portal destination must be configured in API Gateway.
        - You must have the Publish to Developer Portal functional privilege.
        - You cannot publish an API if it is in inactive state. You have to activate the API before publishing it.

          Business Exceptions

Exception Type	Exception	What need to be done

Business	APICoE admin SSO Failure	Check with OKTA team for respective roles and access
	Form fields missing in API publishing page	Should not allow user(admin) to click on submit button/submission button should be disabled
	Drupal connector unable to establish connection with AWS API gateway	**Check with Ankit***
	Failed to publish APIs to AWS API gateway	send error message/Email to admin

Flow Diagram:


Returns 403 error message
Start
APICoE Admin Login portal viaSSO
OKTA SSO status
Success
Admin will be redirected to the Drupal form
Fill the required details & upload Swagger file in the form
End
APIs invoked in AWS gateway are published wrt to the region
AWS gateway will return response as success in separate page
Success
Request Status
Submit form to invoke Drupal connector & Publish AWS API gateway
Stop
Send failure message

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

<table>
<tr><th></th><th>Description</th><th>Endpoint</th><th>Verb</th><th>Headers</th><th>Query Params</th><th>Body</th></tr>
<tr><td>1</td><td>Check Access Token</td><td>/checkvalidity</td><td>GET</td><td><table><tr><th></th><th>Name</th><th>Value</th><th>Required?</th></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>2</td><td>JSON WEB Token Request</td><td>/token</td><td>POST</td><td><table><tr><th></th><th>Name</th><th>Value</th><th>Required?</th></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>3</td><td>Revoke Access Token</td><td>/revoketoken</td><td>POST</td><td><table><tr><th></th><th>Name</th><th></th><th>Required?</th></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>
</table>

Response when Successful

	Description	Endpoint	Verb	Status Code	Response Example	Details
1	Check Token	/checkvalidity	GET	200	None	

2	JSON WEB Token	/token	POST	200	{
					"issued_at": "1694809674274",
					"expires_in": "1800",
					"token_type": "Bearer",
					"access_token": "",
					}	

3	Revoke Token	/revoketoken	POST	200	None	

Response when Fail

	Description	Endpoint	Verb	Status Code	Response Example	Example

1	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

2	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

3				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.

4				415	{
					"errors": [
					{
					"errorType": "Unsupported Media Type",
					"message": "Content-Type header is unsupported"
					}
					],
					"success": false
					}
						Error when you are missing the body JSON request

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 body user OR password.

6	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

7				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

8				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

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	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_CONNECT_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_PASSWORD_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_MILLIS	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