



TGH

Making Integrations Simpler



How to Create JWT Token Authentication from Okta and Implementation

Author

Mohd Nizamuddin



How to create an authentic JWT token from OKTA and Implementation

Introduction

What is meant by the JWT token?

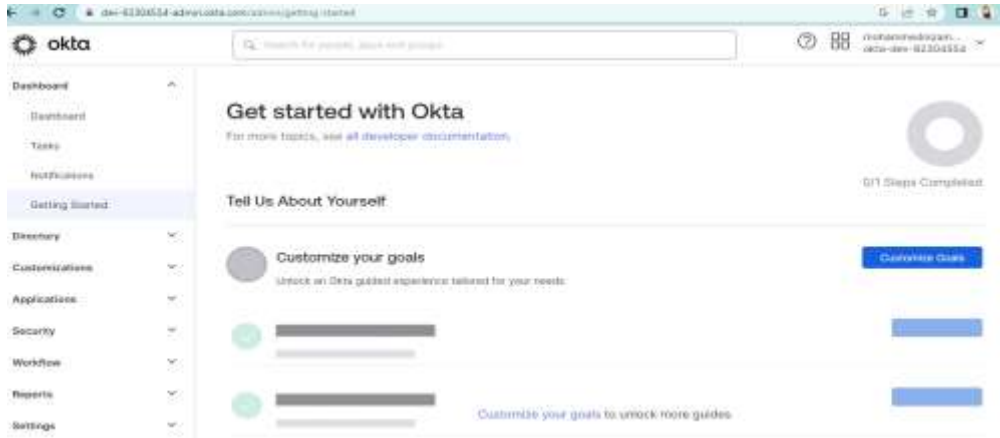
JWT stands for JSON Web token is a proposed internet standard for creating data with optional signature and optional encryption whose payload holds JSON that asserts somenumber of claims.

Creation of OKTA Account

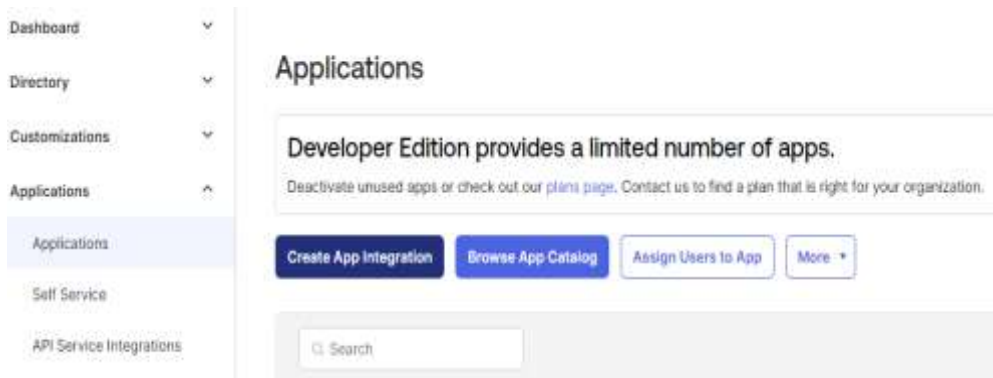
Step 01: In this use case will create an OKTA ID and implement a token by using this link <https://developer.okta.com/login/>



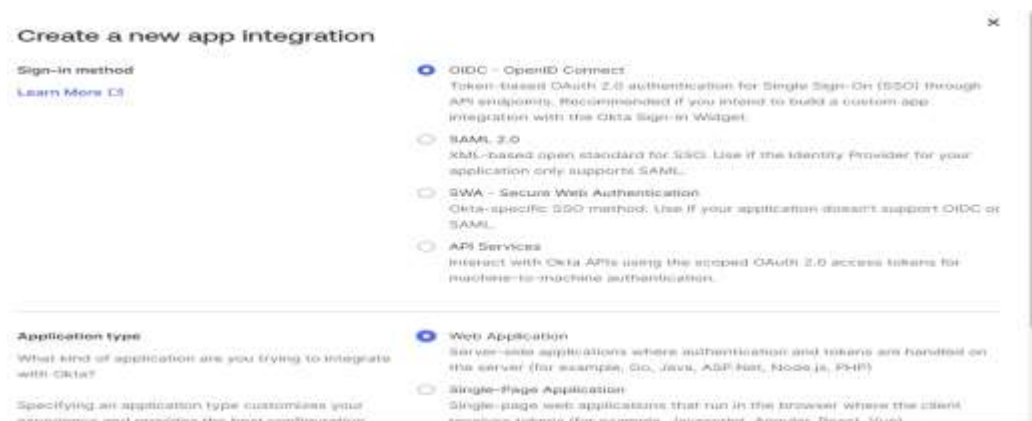
Step 02: Followed by continue with Google Account by adding username and password, then present to the OKTA Home page.



Step 03: Now in the Dashboard we have an option called APPLICATIONS drop-down button then we have an option called APPLICATIONS and then go with CREATE APP INTEGRATION



Step 04: Then select OIDC - OPENID CONNECT in sign-in method and select WEB APPLICATION in Application type then select NEXT



Step 05: New Web Application Integration tab opens. Here, we need to give the Application Integration Name: JWT_PRACTICE, for Grant type select the Client Credentials check box.



New Web App Integration

General Settings

App integration name:

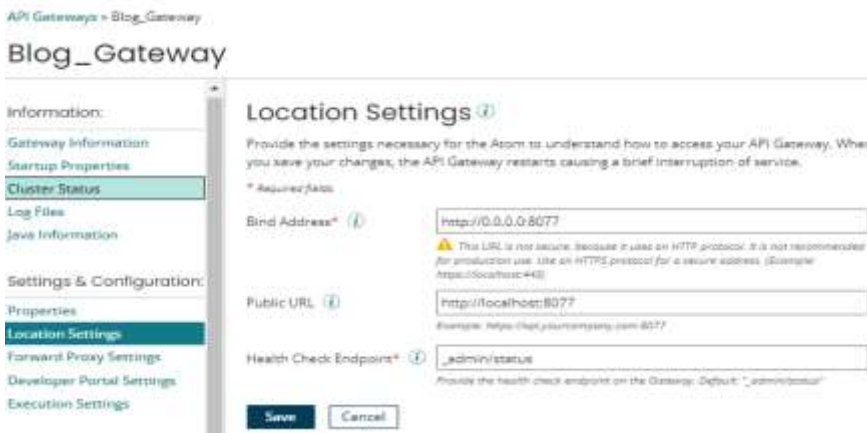
Logo (Optional): 

Grant type:

- Client acting on behalf of itself
 - Client Credentials
 - Authorization Code
 - Refresh Token
 - Implicit (hybrid)
- Client acting on behalf of a user
 - Authorization Code
 - Refresh Token
 - Implicit (hybrid)

[Learn More](#)

Step 06: Then next is Sign-in redirect URI and Add URI from Gateway => Location Settings => Copy the Public URL



API Gateways > Blog_Gateway

Blog_Gateway

Information:

- Gateway Information
- Startup Properties
- Cluster Status
- Log Files
- Java Information

Settings & Configuration:

- Properties
- Location Settings
- Forward Proxy Settings
- Developer Portal Settings
- Execution Settings

Location Settings

Provide the settings necessary for the Apom to understand how to access your API Gateway. When you save your changes, the API Gateway restarts causing a brief interruption of service.

* Required fields

Bind Address*
⚠ This URL is not secure, because it uses an HTTP protocol. It is not recommended for production use. Use an HTTPS protocol for a secure address. (Example: https://localhost:443)

Public URL
Example: https://api.yourcompany.com:8077

Health Check Endpoints*
Provide the health check endpoint on the Gateway. Default: "_admin/status"

Step 07: and paste it in the Add URI box.



Sign-in redirect URIs Allow wildcard * in sign-in URI redirect.

Okta sends the authentication response and ID token for the user's sign-in request to these URIs

[Learn More](#)

Step 08: Then at the bottom of the page we have option called ASSIGNMENTS, Select the check boxes Allow everyone in your organization to access

Assignments

Controlled access

Select whether to assign the app integration to everyone in your org, only selected group(s), or skip assignment until after app creation.

- Allow everyone in your organization to access
- Limit access to selected groups
- Skip group assignment for now

Enable immediate access (Recommended)

Recommended if you want to grant access to everyone without pre-assigning your app to users and use Okta only for authentication.

- Enable immediate access with **Federation Broker Mode**

i To ensure optimal app performance at scale, Okta End User Dashboard and provisioning features are disabled. Learn more about [Federation Broker Mode](#).

Save **Cancel**

And then SAVE.

Step 09: By selecting on SAVE we get a pop-up as Application Successfully Completed and then Client ID and Client Secret will be generated.

Client Credentials [Edit](#)

Client ID [Copy](#)
 Ooaa81fdj0otpb7w75d7
 Public identifier for the client that is required for all OAuth flows.

Client authentication
 Client secret
 Public key / Private key

Proof Key for Code Exchange (PKCE) Require PKCE as additional verification

CLIENT SECRETS

[Generate new secret](#)

Creation date	Secret	Status
Jul 4, 2023	Active Copy

Now,

In boomi we are creating a new process.

Step 10: Create a new process with start shape as Connector type Web Service Server and Action as Listen

Start Shape ⓘ

The Start shape is the main shape that begins the process flow. It is automatically added to each new process and it cannot be removed.

Process Mode: General

Type: Connector Trading Partner Data Passthrough No Data

General Parameters

Display Name:

Connector: ⓘ

Connection: The Atom Web Server will manage connection settings.

Action:

Operation: ⓘ

Step 11: and in operation select operation type as GET and object as jwt which will be added to the URL

=> As we are selecting as GET we don't need to give a request profile.

=> and our output is in XML format then we will need to give the XML profile and then SAVE.

Blog | Web Services Server Connector Oper...

Options Archiving Tracking

Connector Action:

Simple URL Path: ⓘ

SOAP URL Path: ⓘ

Operation Type: ⓘ

Object: ⓘ

Expected Input Type: ⓘ

Response Output Type: ⓘ

Response Profile: ⓘ

Result Content Type: ⓘ

Attachment Cache: ⓘ

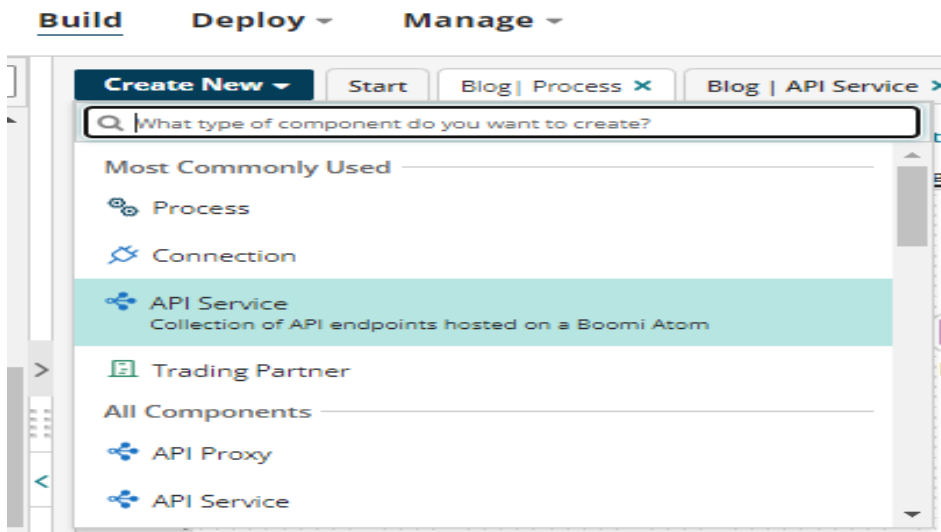
Previous Save on 05 Jul 2023 at 07:03:4

Step 12: and in message shape we are passing the data followed by Return document shape.

Step 13: Then we will Create package Component and Deploy the process.

Step 14: Lets create a API Service Component.

In the process canvas page From the Create New option, we can select APIService Component.



Step 15: Now, it shows a API Service Component home page where we need to add Published APITitle => Published version Number => Base API Path

Blog | API Service - API Service ⓘ Folder Add Description

General REST SOAP OData Profiles Documentation

Published API Title*
55 characters remaining.

Published Version Number* ⓘ
17 characters remaining.
Valid characters are a-z, A-Z, 0-9, _ and .

Published Description
4000 characters remaining.

Service Configuration

Set the base portion of the URL for requests to the API defined by the API object. The full URL is generated when the API component is deployed.

Base API Path
The Base URL Path must be unique for each environment that the API component.

[Save](#) [Save and Close](#) [Close](#) Previous Save on 05 Jul 2023 at 07:07:16 PM UTC+5:30 [Revert](#)

Step 16: Then go to REST tab and select import an Endpoint by using existing process

Import an Endpoint

Choose the method for importing one or more API endpoints into

- Import method
- Create a new process
 - Import from an external service file
 - Import from an external service URL
 - Use an existing process
 - Import processes from an environment

Step 17: and add the REST API and select the Existing process.

Use an Existing Process i

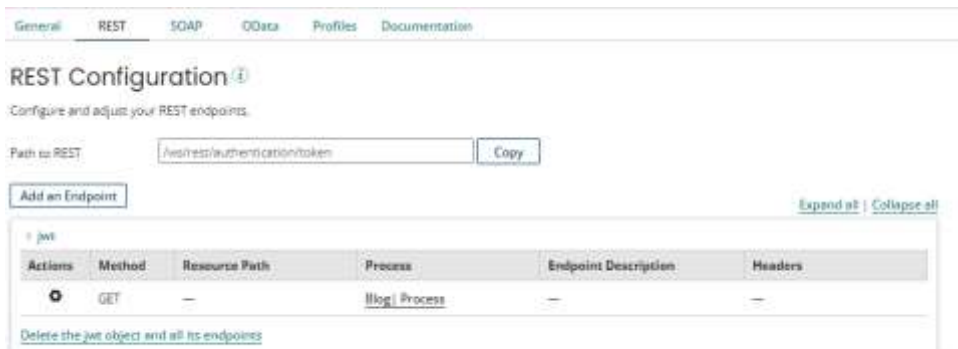
Choose a Web Service listener process.

* Required fields.

Process*

Add to* REST SOAP OData

Then it will look like by adding the endpoint



General REST SOAP OData Profiles Documentation

REST Configuration ⓘ

Configure and adjust your REST endpoints.

Path to REST:

[Expand all](#) | [Collapse all](#)

Actions	Method	Resource Path	Process	Endpoint Description	Headers
	GET	-	Blog Process	-	-

[Delete the jwt object and all its endpoints](#)

Step 18: Now, create a package component and Deploy it to the Environment which is attached to the Gateway.

Step 19: Boomi will offer multiple services, in that we have one service known as API Management



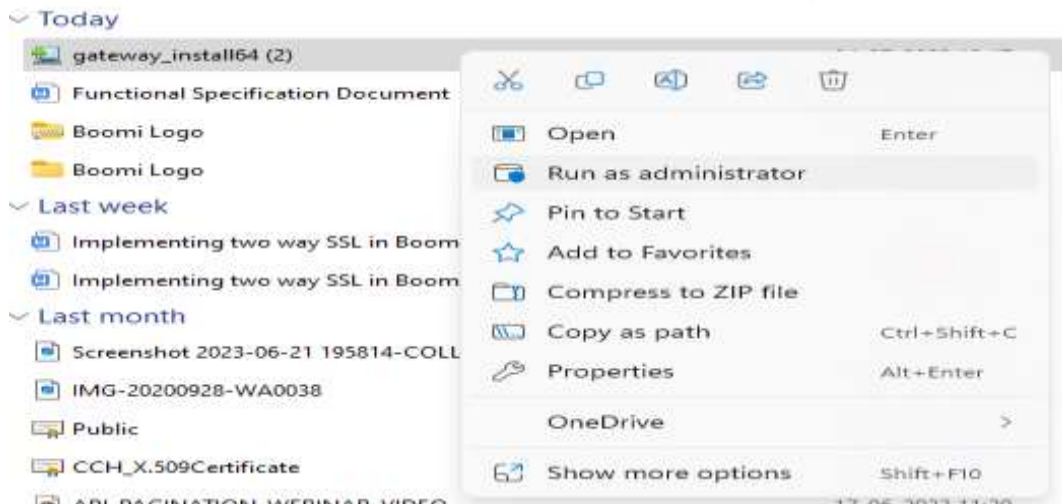
Step 20: Then it goes to the API Management home page there we have option called CONFIGURESERVER in that we have option called GATEWAYS



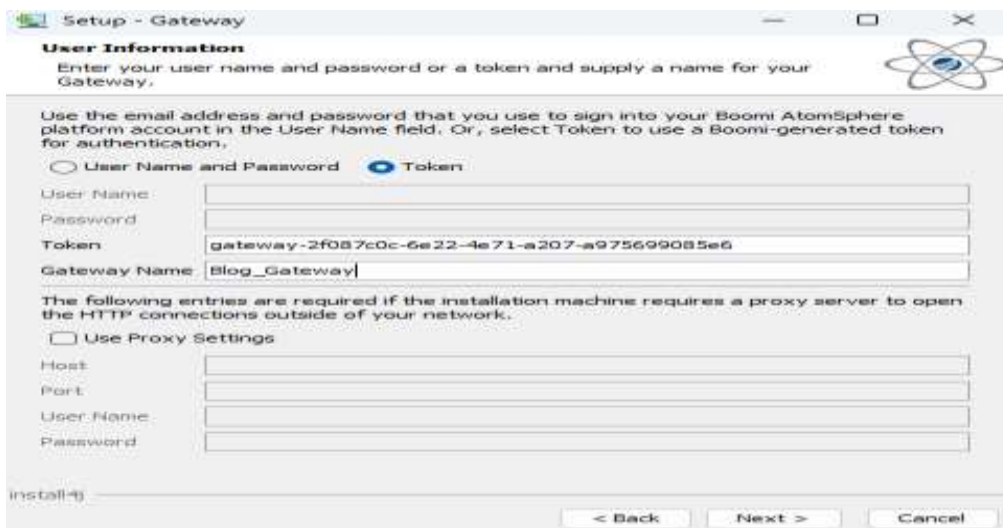
Step 21: Here we need to Add a Gateway and then setup a Gateway by selecting an Operating System and in Security options we need to copy a Gateway Installer Token and then select Download Installer.



Step 22: Now, go to Download list where the Gateway is downloaded and then right-click the Installer and select RUN as Administrator

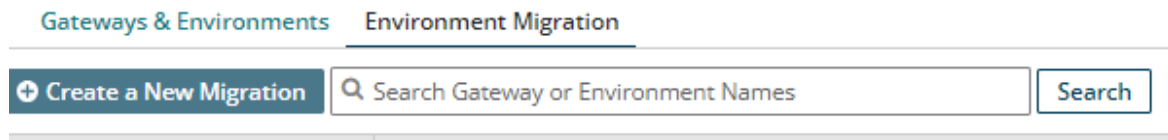


Step 23: Then select Run Anyway => then select as Next => Then in the user information select as Token and give the Gateway Name and the token which we have copied earlier for installation.



Step 24: Then select Next and Next to download the JRE files and wait for Finishing to install the Gateway on our local machine.

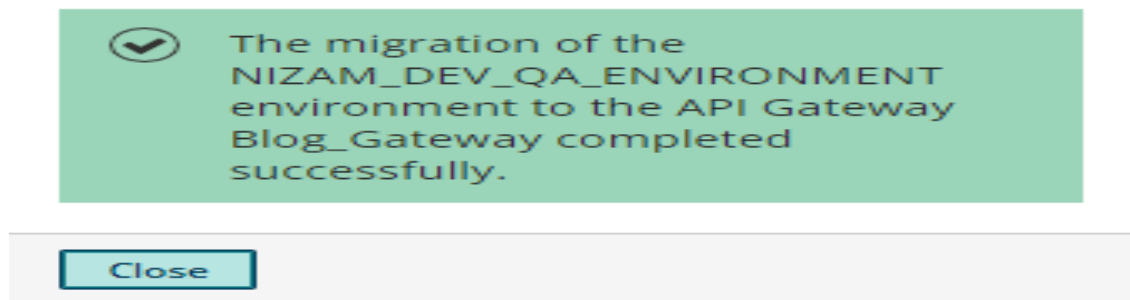
Step 25: Gateway has been installed, now we need to migrate this to Environment migration by selecting the tab.



Step 26: Now, we need to attach an Environment to the Gateway and then Create migration => now, select the Gateway which we have installed



Step 27: Then select Save and Continue and check the endpoints and select Save and Continue, it will be confirming the status and we need to check again and select Save and Continue. Then at the end we will finish the process and get a pop-up as Gateway Installed Successfully.



Step 28: After creating a Gateway we need to create an Application which is under the Configure APIs and Applications tab.



Step 29: Now Create an Application by selecting the Gateway Name, Application Name, Application Owner Name, Application owner Email and then SAVE.

Applications > Create an Application

Create an Application (?)

Specify the application settings.
** Required fields.*

General Settings

Gateway: Blog_Gateway

Status: Enabled Disabled by Publisher

Application Name*: Blog
255 characters remaining

Description:
4000 characters remaining

Application Owner Information (?)

Application Owner Name*:
255 characters remaining

Application Owner User Name (?):
255 characters remaining

Application Owner Email*:
255 characters remaining

Step 30: Now, Create Plan for the Application in Configure Server tab



Step 31: After selecting a Plan now configure a Plan by giving the proper name of the Plan and also the messagesize, Rate limit, Quota limit to the Plan.



API Management | **Configure Server** | Configure APIs and Applications

Create a Plan

Choose from the options provided to create a new Plan. Once created, a plan can be associated with any API Deployment.

* Required fields.

Plan Name

Description

Message Size Unlimited
 Maximum of KB

Rate Limit Unlimited
 Maximum of calls per minutes

Quota Limit Unlimited
 Maximum of calls per months

Step32: After creating a Plan go to Authentication from Configure Server tab



API Management | **Configure Server**

- Gateways
- Plans
- Authentication**

Step 33: and configure a New Authentication Source by giving Authentication Source Name, Description and Identity Provider Type.

In Identity Provider Type we have two options i.e, Basic Authentication (Gateway) and JWT Authentication. Here we go with JWT Authentication.



Add an Authentication Source

* Required fields.

Authentication Source Name

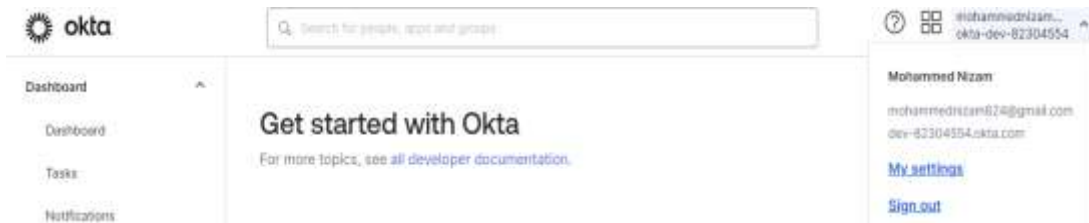
Description

Identity Provider Type

- Select One
- Basic Authentication (Gateway)
- JWT Authentication**

OK Cancel

Step 34: After Selecting an Identity Provider Type as JWT Authentication we need to add an IdentityProvider URL which is required. This Identity Provider URL need to take below from OKTA user name i.e, dev-82304554.okta.com



Step 35: Prepend this OKTA ID with https:// and Append with /oauth2/default.

At the end, it will be in the form of <https://dev-82304554.okta.com/oauth2/default>



Step 36: After Authentication we need to Configure APIs and Applications with the Deployed APIs and check with the Authentication method. Here we can check that the process which we have configured and created API Service Component will be reflecting in this Deployed APIs

APIs by Name | APIs by Gateway | APIs by URL

Detail	API Name	Environment	API Version	Authentication	Plans	Custom URL?	Execution Settings	Status
View	APIServiceComponent-jwt(practice)-APIService	Gourav-prod	1.0	jwt_practice	1 Plan (Keyless)	No	Gateway	✔ Sent to the Gateway

Step 37: then need to add previously created Plan for the Gateway.

Add a plan Make One Plan Keyless

Keyless	Name	Maximum Message Size	Rate Limit	Quota Limit	Usage	Description	Remove
<input type="checkbox"/>	blog_plan	Unlimited	Unlimited	Unlimited	0	—	

and then SAVE it.

Step 38: After selecting a Plan for the Deployed process. Then view the process

APIs by Name | APIs by Gateway | APIs by URL

Detail	API Name	Environment	API Version	Authentication	Plans	Custom URL?	Execution Settings	Status
View	APIServiceComponent-jwt(practice)-APIService	Gourav-prod	1.0	jwt_practice	1 Plan	No	Gateway	✔ Sent to the Gateway

Step 39: then a new tab appears where we go to Rest tab

General | **REST**

REST information about this API.
 You do not have the ability to view these REST endpoints through the Swagger Visualization Portal because the Developer Portal is disabled and all of the Acorns within the environment are using Gateway as the Authentication type.

Gateway information: Blog_Gateway (1 atom)

Base URL Path: <http://localhost:8077/ws/rest/PRACTISE>
 Swagger Reference: <http://localhost:8077/ws/rest/PRACTISE/swagger.json>
 OpenAPI Reference: <http://localhost:8077/ws/rest/PRACTISE/openapi.json>

REST Endpoints for Blog_Gateway with Gateway Listener Authentication

Expand All | Collapse All

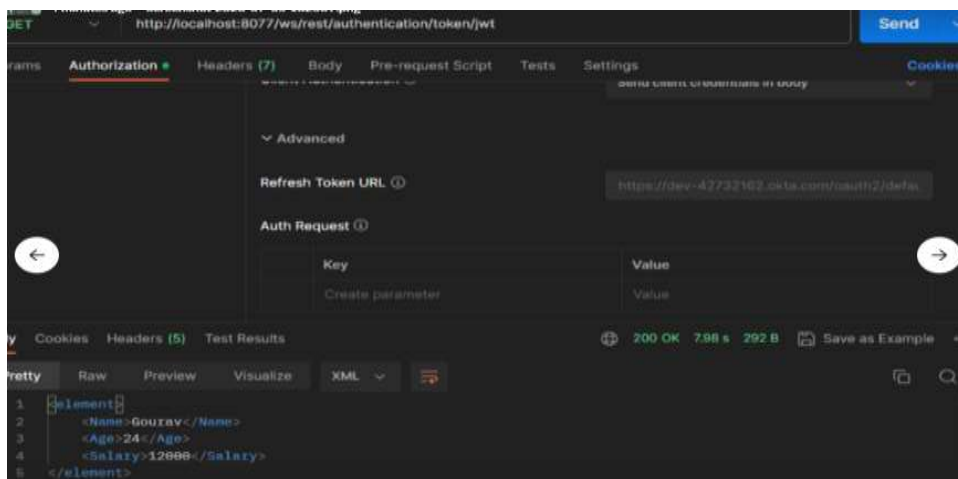
Method	Endpoint Name	Endpoint Path
GET	(apiName)	http://localhost:8077/ws/rest/PRACTISE/EMRCA... <input type="button" value="Copy"/>

Step 40: In the new tab copy the Endpoint path. Now go to Postman for checking the API.

Step 41: Add a new request from the Postman and paste the same URL which we have copied from the Deployed API

Step 42: In Authorization select type as Oauth 2.0 and Add authorization data to the Request Header

In the new tab copy the Endpoint path. Now go to Postman to check the API.



Step 43: Now we need to Configure New Token by selecting Grant type as Authorization Code =>Call back URL as <http://localhost:8077>

Step 44: Auth URL as same as Identity Provider URL followed by /v1/authorize i.e, <https://dev-35000642.okta.com/oauth2/default/v1/authorize>

Step 45: Access URL token as same as Identity Provider URL followed by /v1/token i.e, <https://dev-35000642.okta.com/oauth2/default/v1/token>

Step 46: Copy Client ID and Client Secret from OKTA and paste in postman

Client Credentials [Edit](#)

Client ID: 00aa8bfdj0a1pb7w75d7 [Copy](#)
Public Identifier for the client that is required for all OAuth flows.

Client authentication: Client secret Public key / Private key

Proof Key for Code Exchange (PKCE): Require PKCE as additional verification

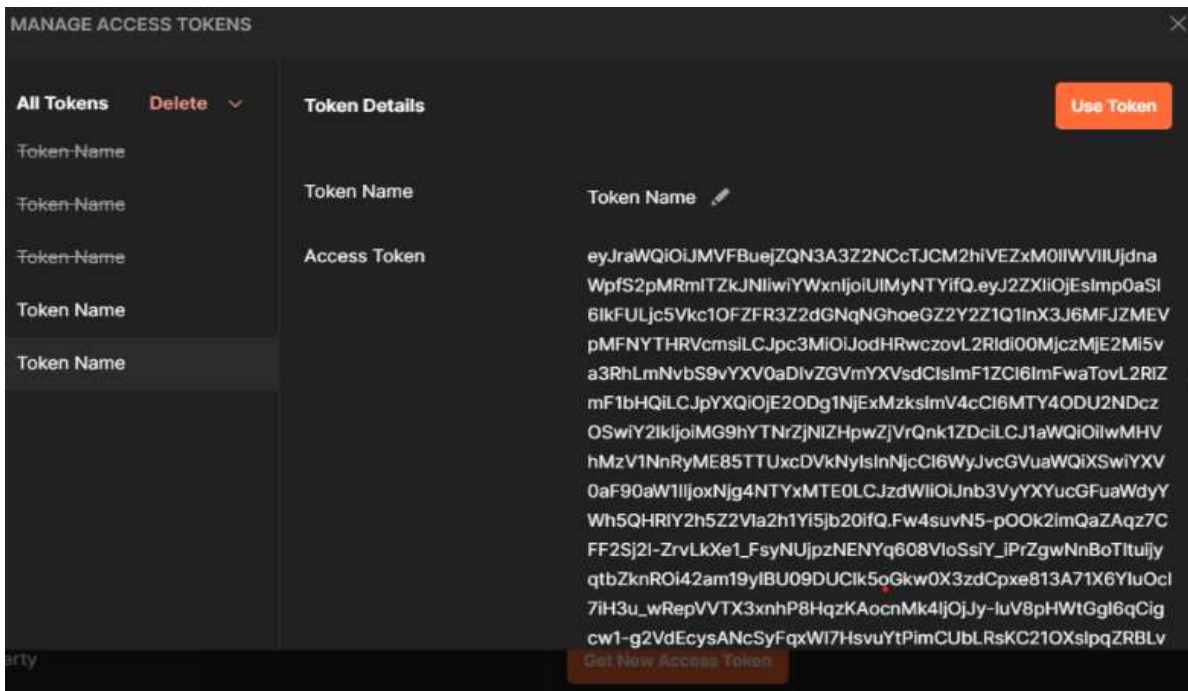
CLIENT SECRETS

[Generate new secret](#)

Creation date	Secret	Status
Jul 4, 2023 Copy Share	Active ▼

Step 47: Scope as OpenID and State as ‘a’

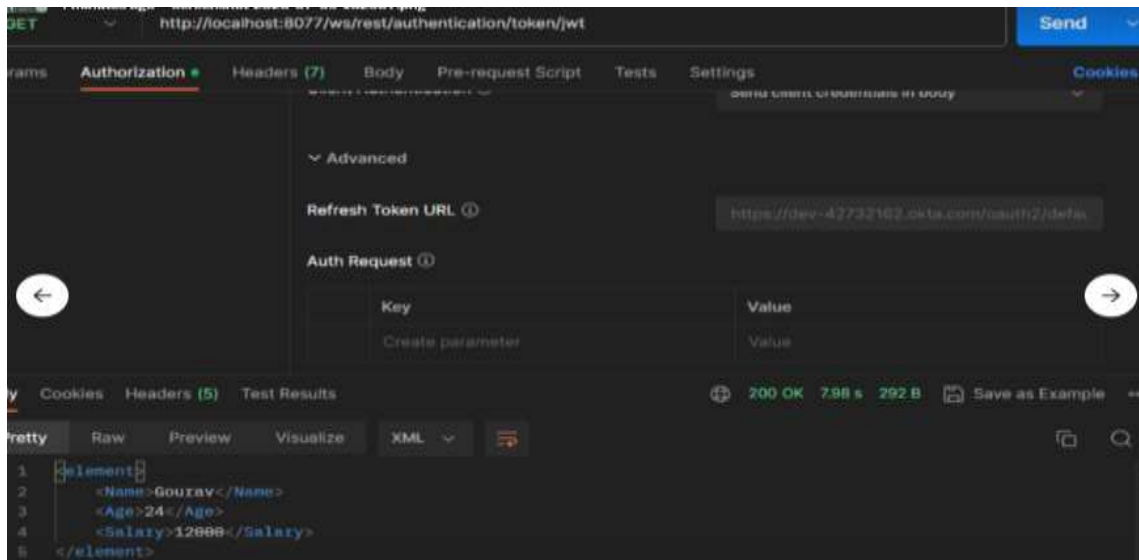
Step 48: Then we need to Hit the button as Get New Access token



The screenshot shows the 'MANAGE ACCESS TOKENS' interface. On the left, there is a list of tokens, all labeled 'Token Name', with a 'Delete' dropdown menu. On the right, the 'Token Details' panel is open for an 'Access Token'. The token details include the token name and a long alphanumeric string representing the token value. At the bottom of the panel, there is a 'Get New Access Token' button.

Step 49: Then use Token

Step 50: and at last, Hit the URL and will get the response.





TGH

Making Integrations Simpler



TGH Software Solutions Pvt. Ltd.

www.techygeekhub.com

At TGH, we specialize in driving digital transformation through seamless Integration Technologies.

Operating as an INTEGRATION FACTORY, we serve as a one-stop shop for all your integration needs. Our expert team is well-versed in enterprise software and legacy system integration, along with leading iPaaS technologies like Boomi, MuleSoft, Workato, OIC, and more.

We're committed to enhancing business processes and solving problems through our integration expertise.



Email address

connect@techygeekhub.com



Phone number

+ 011-40071137
+ 91-8810610395



Our offices

Noida Office

iThum
Plot No -40, Tower A,
Office No: 712,
Sector-62, Noida,
Uttar Pradesh, 201301

Hyderabad Office

Plot no: 6/3, 5th Floor,
Techno Pearl Building,
HUDA Techno Enclave,
HITEC City, Hyderabad,
Telangana 500081

