

# Auth0 Configuration for Spring Boot Application

## Introduction:

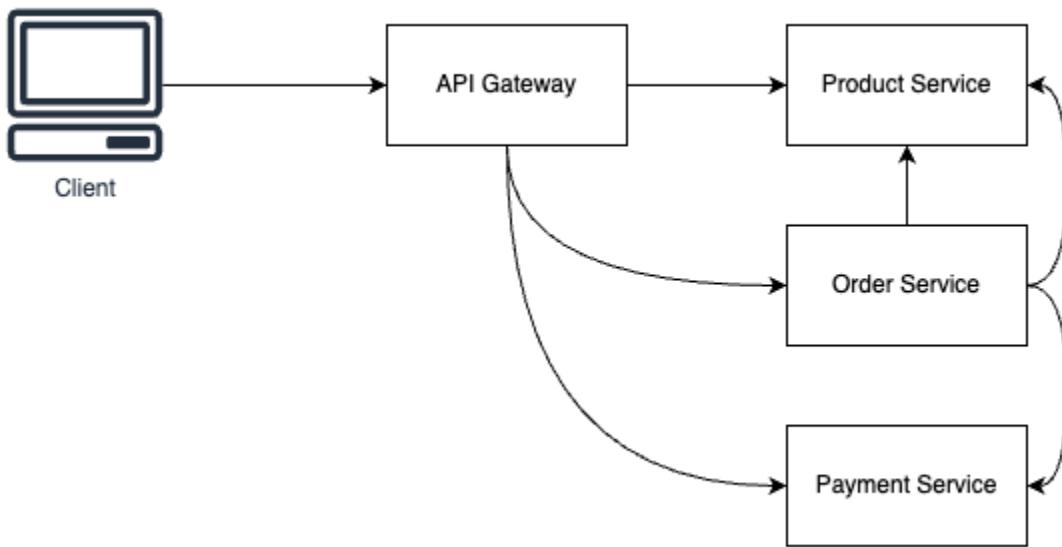
For Auth0 Configuration, you need to create an Auth0/Okta account. Go to the below URL to create your developer account with the free tier.

Got to : <https://auth0.com/api/auth/login?redirectTo=dashboard>

Once the account is created, you must configure the Auth0 dashboard to work with your Spring Boot Microservices.

Below is the architecture of Spring Boot Microservices we are working with.

## Architecture:



In this Architecture, you can see we have multiple services. I have only included the service that needs Security integration.

1. API gateway → This has the API to login to the system and return the access token to be used to invoke other services

2. Product Service → This has the API to manage Products
3. Payment Service → This has the API to manage Payments
4. Order Service → This has the API to manage Orders. Internally, this Service will also invoke the API's of the Product Service and the Payment Service.

Spring Security works with different components:

1. OAuth Client → This is the Client Application, which interacts with Auth0 and authenticates and authorizes the User and generates the token.
2. OAuth Resource → This is the Resource that serves the resources (API). It can validate the token with proper authorization and allow access.

As per above Architecture

Below are the details containing which security component should be added to each service.

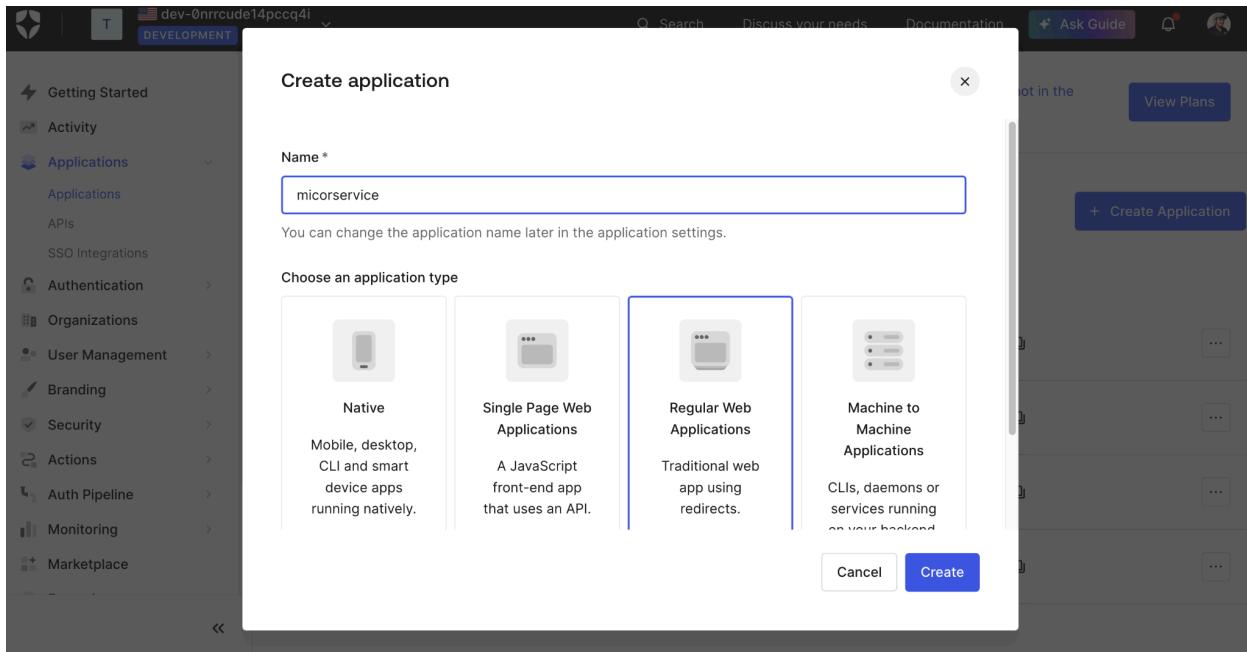
Service	Component
API Gateway	Client, Resource
Product Service	Resource
Order Service	Resource
Payment Service	Resource

## Auth0 dashboard Configuration

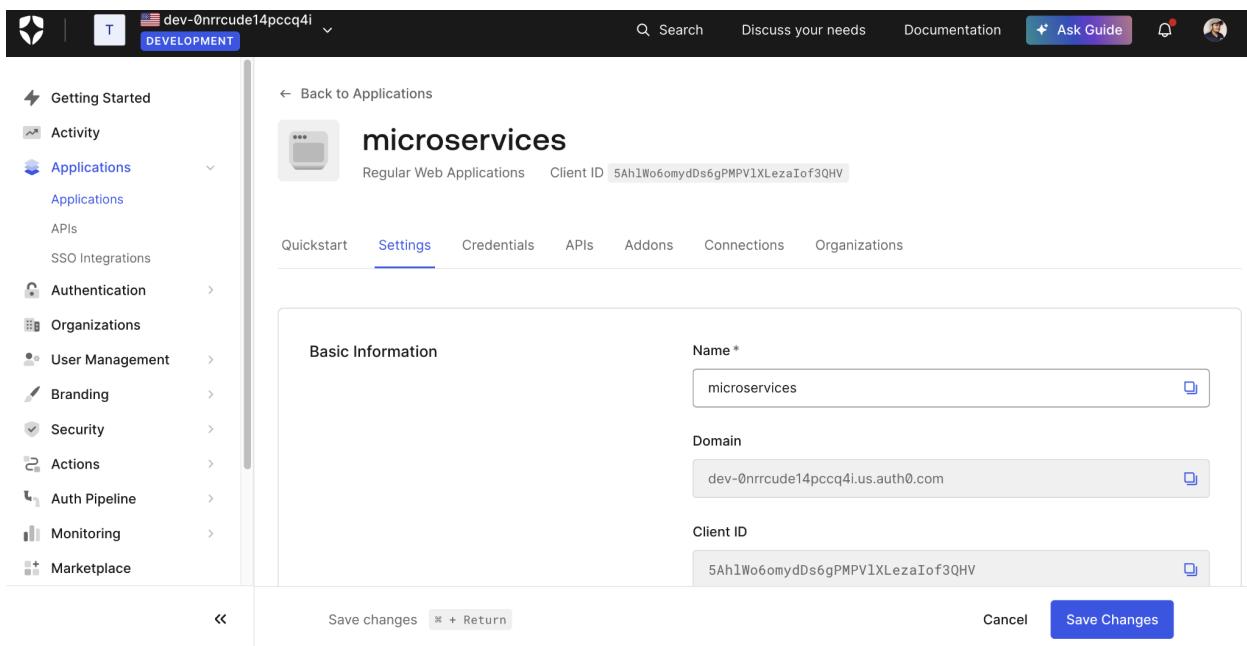
- Login to Auth0 dashboard and go to Applications → Applications
- Click on Create Application

The screenshot shows the Auth0 dashboard interface. At the top, there is a navigation bar with links for 'Getting Started', 'Activity', 'Applications' (which is currently selected), 'APIs', 'SSO Integrations', 'Authentication', 'Organizations', 'User Management', 'Branding', 'Security', 'Actions', 'Auth Pipeline', 'Monitoring', and 'Marketplace'. Below the navigation bar, there is a message: 'Thank you for signing up for Auth0! You have 21 days left in your trial to experiment with features that are not in the Free plan. Like what you're seeing? Please enter your billing information here.' A 'View Plans' button is located next to this message. The main content area is titled 'Applications' and contains a sub-instruction: 'Setup a mobile, web or IoT application to use Auth0 for Authentication. Show more >'. Below this, there is a table listing four applications:

Name	Type	Client ID	Actions
Admin API (Test Application)	Machine to Machine	1j5nYnv8xHdTTRTY5N5oRRtPnkhK2D9	...
Customer APIs (Test Application)	Machine to Machine	4s1w0vYuVW4G0uqag14togi90j4GZNg	...
Default App	Regular Web Applications	5ZW2aivG380F1iy2p9AfQXRX7nQdZbg3	...
microservices	Regular Web Applications	5Ah1W60myD6gPMPY1XLLezaIoT3QHV	...



Once the application is created, go to the application for managing the settings.



- Now once the application is created, you need to configure the application URLs
- Configure Application callback/redirect URL
  - <http://localhost:9090/login/oauth2/code/auth0>
- Configure logout URL
  - <http://localhost:9090/>

The screenshot shows the Auth0 Application Settings interface for a development application named 'dev-0nrcude14pccq4i'. The left sidebar lists various configuration sections: Getting Started, Activity, Applications (selected), APIs, SSO Integrations, Authentication, Organizations, User Management, Branding, Security, Actions, Auth Pipeline, Monitoring, and Marketplace. The main content area is titled 'Application URIs' and contains fields for 'Application Login URI' (set to 'https://myapp.org/login') and 'Allowed Callback URLs' (set to 'http://localhost:9090/login/oauth2/code/auth0'). Below these fields is a note about specifying multiple URLs separated by commas. It also includes sections for 'Allowed Logout URLs' (set to 'http://localhost:9090/') and a note about using custom URI schemes for native clients. At the bottom, there are 'Save changes' and 'Return' buttons, and a 'Cancel' button next to a 'Save Changes' button.

Save the application by clicking Save Changes.

## Create Users and Roles

Once our application is created, we need Users and Roles, which help to log in to the application.

### Create Roles

- Roles allow us to have Role-based Authentication control for our application.
- Customer Role:
  - This Role allows the user to be a Customer and allows the application to: Place an Order, View Order, and View Products.
- Admin Role:
  - This role is for Admin, which helps to maintain the application.
  - To maintain Products.
  - To maintain the Payment reports
  - And other management related APIs, which the Customer should not have access to.
- In Auth0, each Role should be tied to the API (Audience), which helps the application to decide what permissions the user has assigned.
- Then the Role should be tied up to Users.
- Go to User Management → Roles to create new Roles.
- Click on Create Role to create a new Role.

The screenshot shows the Auth0 dashboard interface. On the left, there is a sidebar with various navigation options under 'User Management': Applications, APIs, SSO Integrations, Authentication, Organizations, User Management (which is currently selected), Roles, Branding, Security, Actions, Auth Pipeline, Monitoring, Marketplace, Extensions, and Settings. The main content area has a dark background with a central modal window titled 'New Role'. Inside the modal, there are two input fields: 'Name \*' containing 'Customer' and 'Description \*' containing 'Customer Role'. At the bottom right of the modal are 'Cancel' and 'Create' buttons. Above the modal, a message reads: 'Thank you for signing up for Auth0! You have 21 days left in your trial to experiment with features that are not in the Free plan. Like what you're seeing? Please enter your billing information here.' To the right of the modal, there is a button labeled 'View Plans'.

This screenshot shows the 'Roles' page in the Auth0 dashboard. The sidebar on the left is identical to the previous screenshot, with 'User Management' selected. The main content area is titled 'Roles' and contains the sub-instruction: 'Create and manage Roles for your applications. Roles contain collections of Permissions and can be assigned to Users.' Below this, there is a table listing existing roles:

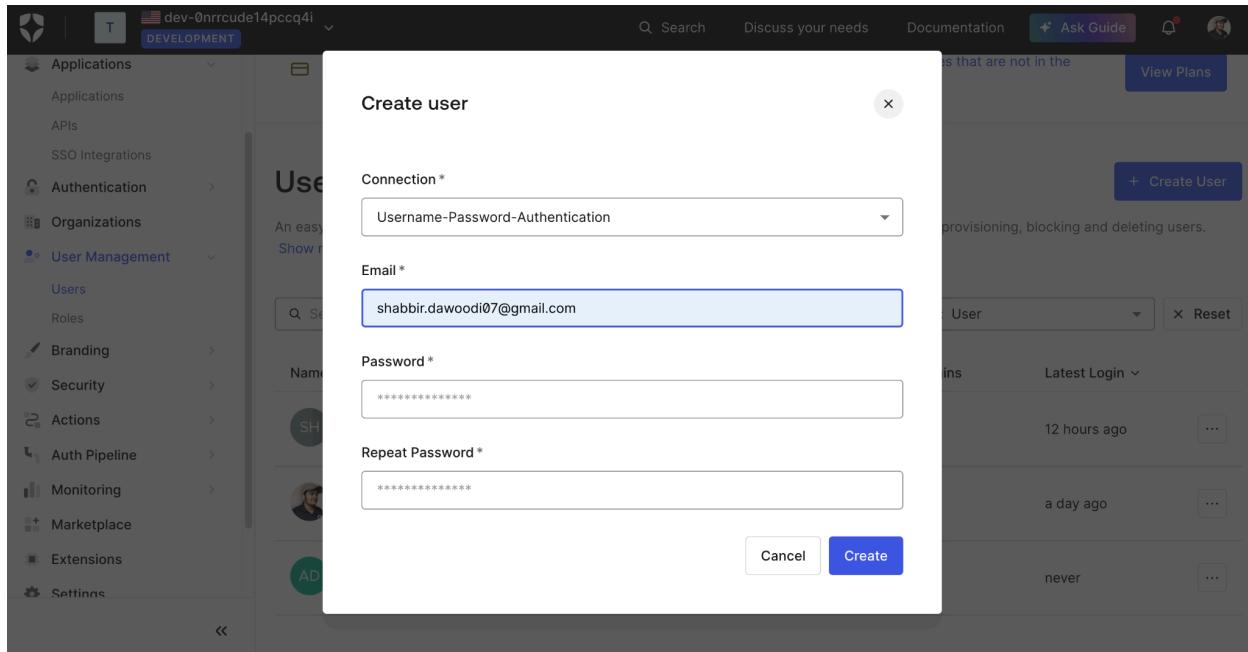
Name	Description	Actions
Admin	Admin	[...]
Customer	Customer	[...]

To the right of the table, there is a blue button labeled '+ Create Role'.

## Create User

- Go to User Management → Users to create Users.

- Click on Create User to create a new User.



Click on the User created to assign Permission and Roles.

The screenshot shows the 'User Management' section of the Auth0 dashboard. On the left is a sidebar with various options like Applications, Authentication, and User Management. In the main area, a user profile for 'shabbir.dawoodi07@gmail.com' is displayed. The 'Details' tab is active, showing the user's name, email (shabbir.dawoodi07@gmail.com, status: UNVERIFIED), and the date 'March 14th 2025, 1:58:5...'. Other tabs include Devices, History, Raw JSON, Authorized Applications, Permissions, and Roles. An 'Actions' dropdown is also visible.

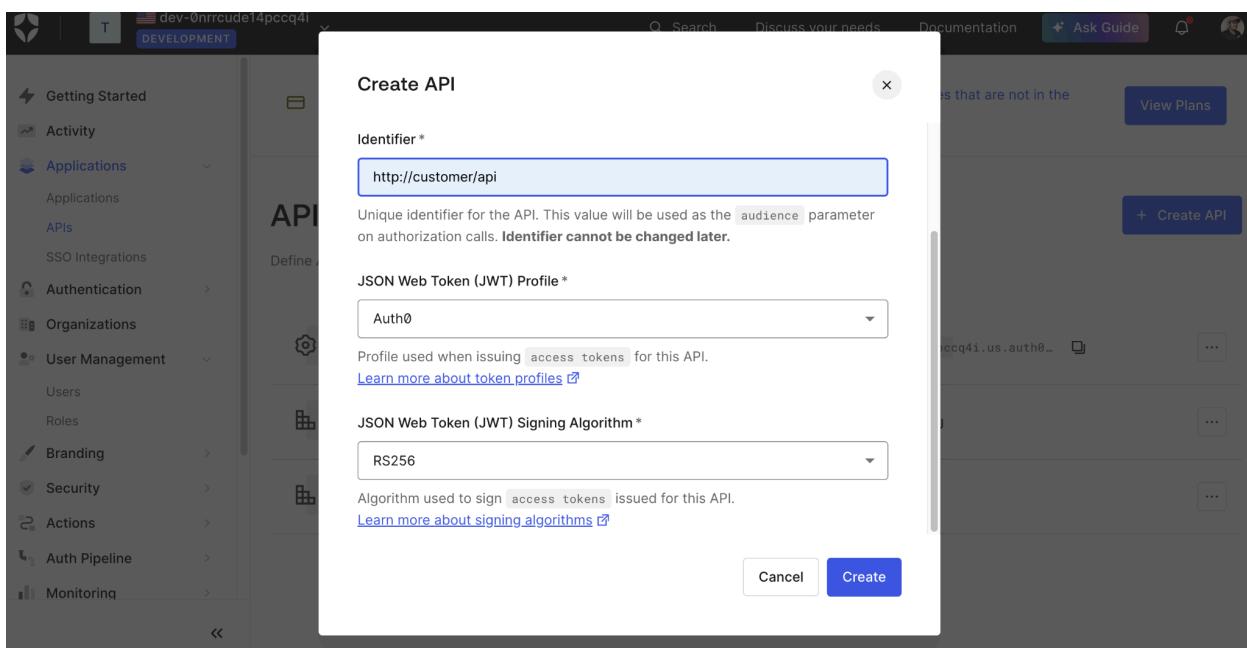
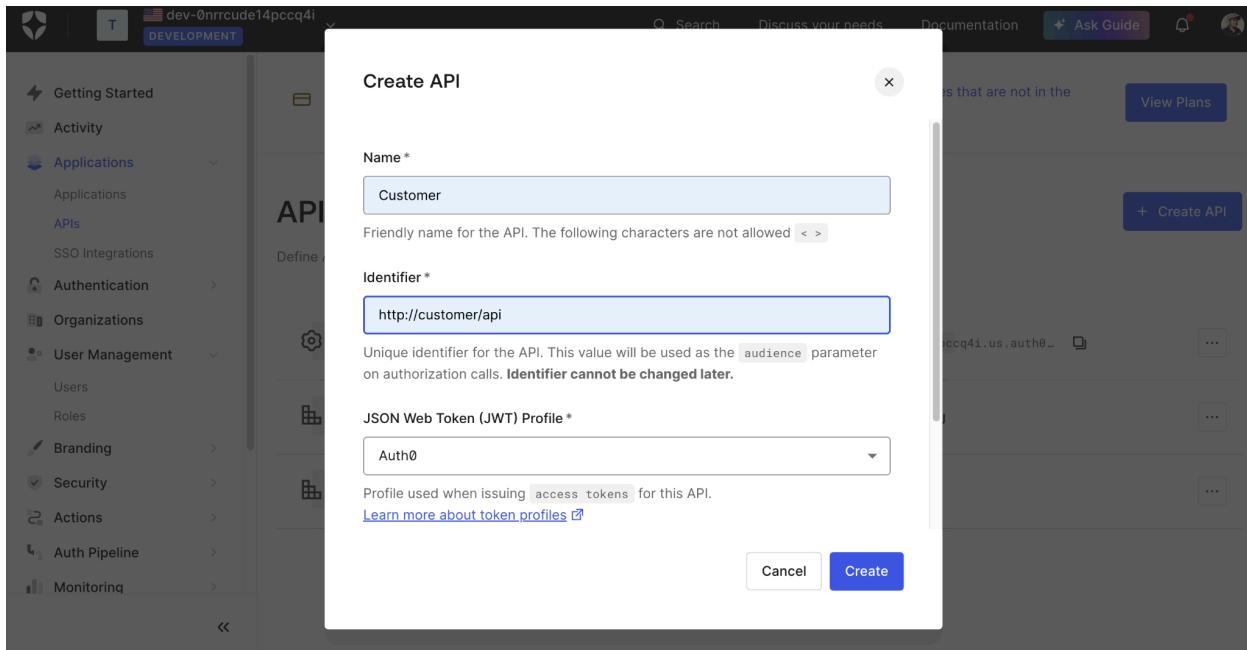
- Go to the Roles Section to assign a Role to the User created.
- For this User, assign the Customer Role as shown in the below Screenshot.

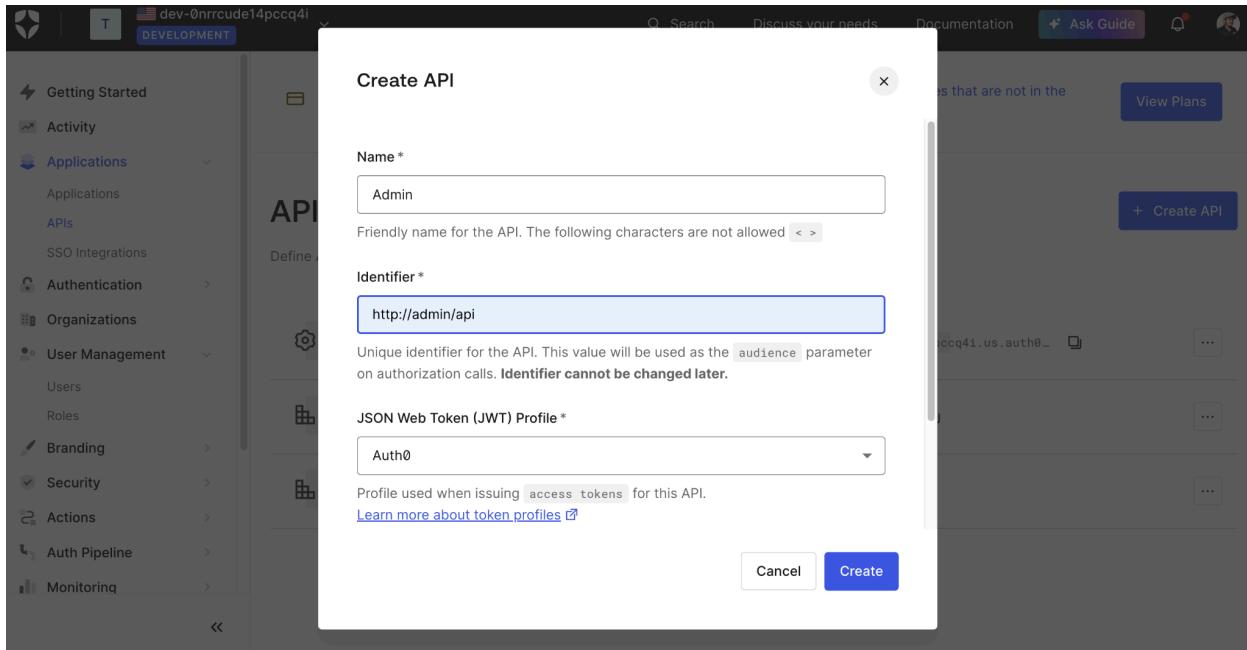
The screenshot shows the Auth0 dashboard interface. On the left, there's a sidebar with various navigation items. Under 'User Management', 'Users' is selected. In the main area, a user profile for 'shabbir.dawoodi07@gmail.com' is displayed. The 'Roles' tab is active, showing a single role assignment: 'Customer' with a description of 'Customer' and an 'Assignment' status of 'Direct'. There's also a 'Raw JSON' tab and a 'View Plans' button.

- Now, we have assigned the Roles to the User
- This User should also have the Permissions assigned.
- Permissions allow the particular User to use the different API's
- Create an API (Audience) that defines the permissions and add those permissions to users.

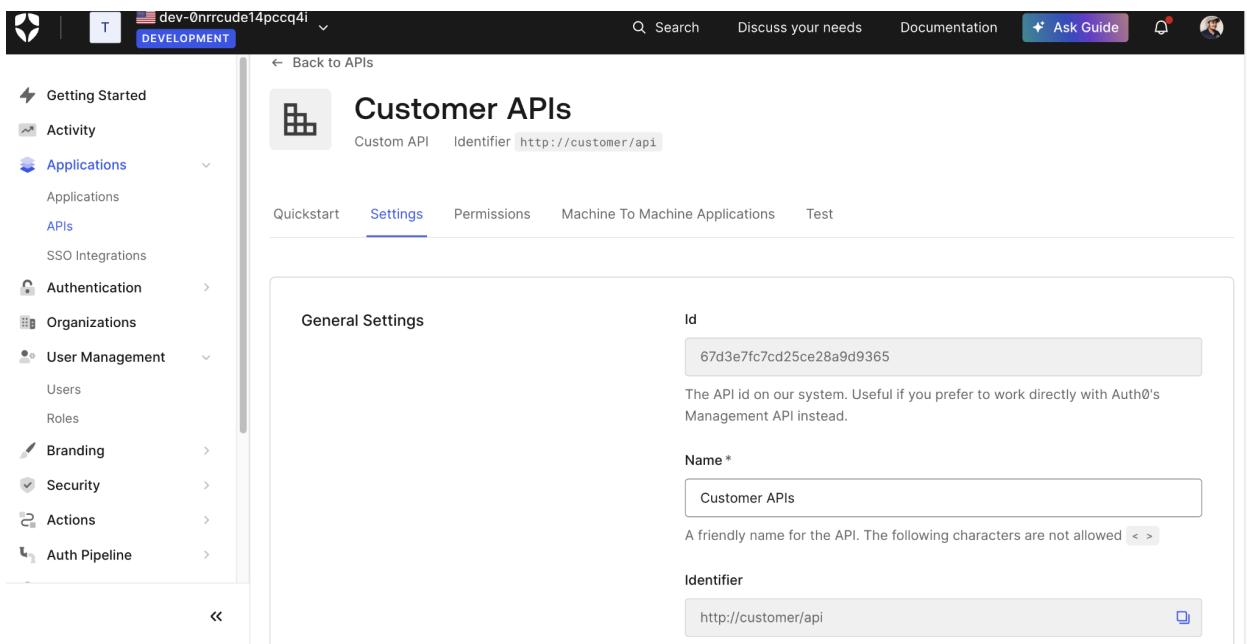
## Create API

- Let's create 2 API's for different purposes of the application
- Customer API to have customer-related permissions
- Admin API to have admin-related permissions
- Go to Applications → APIs to create the API
- Click on Create API to create a new API
- Give a unique name to the API
- Give a unique identifier of the API. Give an Endpoint which is unique, it may not be a valid endpoint.
- Keep JSON Web Token Profile as "Auth0"
- Keep Signing Algorithm as RS256
- Click on Create to create an API
- Repeat the process for Admin API





- For each API, enable Role-based Access Control (RBAC)
- Enable RBAC and add permissions in the Access token
- Now, For Each API created, we need to define permissions.



RBAC Settings

Enable RBAC

If this setting is enabled, RBAC authorization policies will be enforced for this API. [Role](#) and permission assignments will be evaluated during the login transaction.

Add Permissions in the Access Token

If this setting is enabled, the Permissions claim will be added to the access token. Only available if RBAC is enabled for this API.

Access Settings

User Consent Policy

Policy used when asking users for consent to access their information or perform actions on their behalf. [Learn more about user consent policies](#)

Allow Skipping User Consent

Customer APIs

Custom API Identifier <http://customer/api>

Quickstart Settings Permissions Machine To Machine Applications Test

Add a Permission

Define the permissions (scopes) that this API uses.

Permission *	Description *
read:appointments	Read your appointments

+ Add

List of Permissions

These are all the permissions that this API uses.

Permission	Description
Customer	Customer

- Add Permission and Description for the API
- Multiple permissions can be added to the API, which can be used to add for the Roles.
- Go to Machine To Machine Application to add the Application we have created.

The screenshot shows the 'Machine To Machine Applications' section of the platform. On the left, a sidebar lists various application categories like Getting Started, Activity, Applications, APIs, SSO Integrations, Authentication, Organizations, User Management, Branding, Security, Actions, and Auth Pipeline. The 'Applications' category is expanded, showing sub-options: Applications, APIs, and SSO Integrations. The 'APIs' option is selected. The main content area displays a table of applications:

Application	Client Id	Status
Admin API (Test Application)	1j5nYNv8xHdTTRTY5N5oRRrTpNkhK2D9	Unauthorized
Customer APIs (Test Application)	4s14wCvYuV#4Guqagi4togi9Dj4GZNg	Authorized
Default App	5ZW2aiVG300F1iy2p9AfQXRX7nQdZbg3	Unauthorized
microservices	5Ah1Wo6omydDs6gPMPV1XLLezaIof3QHV	Authorized

The screenshot shows the configuration for the 'microservices' application. The sidebar is identical to the previous screenshot. The main content area shows the application details:

- microservices** Client Id: 5Ah1Wo6omydDs6gPMPV1XLLezaIof3QHV
- Select which permissions (scopes) should be granted to this client:
- Grant ID: cgr\_HcBd0aOfy2Dww3SM
- Permissions**: A list box containing 'Customer' with a checked checkbox.
- Organization Support**: Three radio button options:
  - None**: Selected. Description: Machine to machine access cannot be requested.
  - Optional**: Description: Machine to machine access may be requested.
  - Required**: Description: Machine to machine access must be requested.

- Now, go to Users by User Management → Users to add permissions to the User
- Click on the User for which permission needs to be added.
- Go to Permission tab to add the permission
- Click on Assign Permission to assign permission.
- Select the API from which you need to add permission.
- Select all permissions, click on Save to save the permissions.

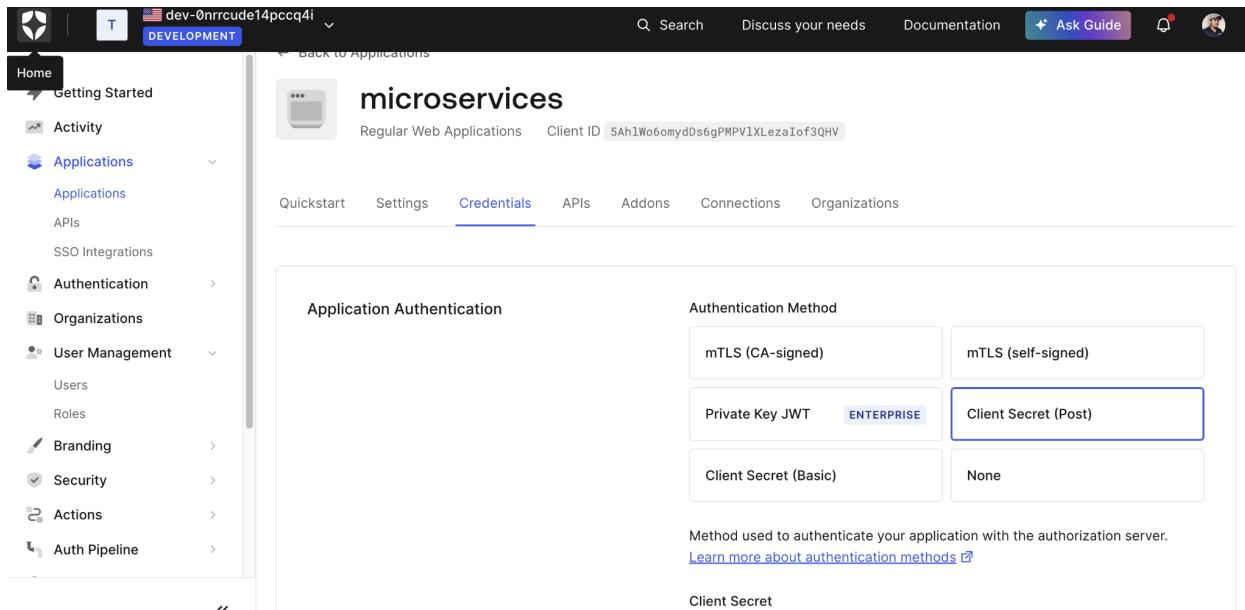
The screenshot shows the 'Add Permissions' dialog box from the Auth0 interface. The dialog has a title 'Add Permissions' and a sub-instruction 'Select permissions from existing APIs'. Below this is a dropdown menu labeled 'Select an API...'. At the bottom right of the dialog is a blue 'Add Permissions' button. The background shows a dark grey dashboard with a sidebar containing various management options like Applications, Authentication, and User Management.

This screenshot shows the 'Add Permissions' dialog with the 'Admin API' selected in the dropdown. Under the 'Permissions' section, there is a list with one item: 'Admin', which has a checked checkbox. To the right of the checkbox is a small blue square icon with a white checkmark. The background of the dialog shows the same 'Add Permissions' interface as the first screenshot, with the 'Customer' row visible in the main table.

Add Customer Permissions. Based on the Users that you have created, add the permissions accordingly.

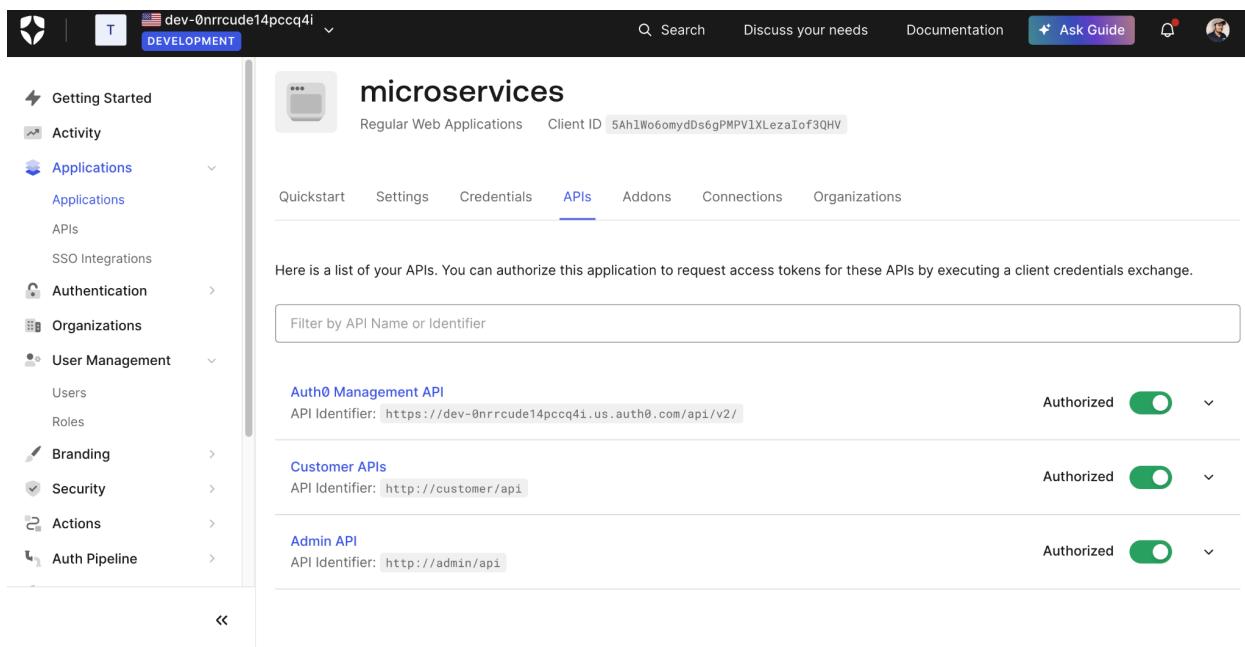
# Application Configuration to add API's and Application Authentication

- Open Application and go to credentials tab to configure credentials.
- Set Application Authentication to “Client Secret (Post)”



The screenshot shows the 'microservices' application configuration page. On the left, there is a sidebar with various tabs like Home, Getting Started, Activity, Applications, Authentication, Organizations, User Management, Branding, Security, Actions, and Auth Pipeline. The 'Applications' tab is currently selected. At the top right, there are links for Search, Discuss your needs, Documentation, Ask Guide, and a user profile. The main content area has a title 'microservices' and a sub-section 'Regular Web Applications'. Below this, there are tabs for Quickstart, Settings, Credentials (which is highlighted in blue), APIs, Addons, Connections, and Organizations. Under the 'Credentials' tab, there is a section titled 'Application Authentication' with a sub-section 'Authentication Method'. It lists several options: mTLS (CA-signed), mTLS (self-signed), Private Key JWT (Enterprise), Client Secret (Post) (which is highlighted with a blue border), Client Secret (Basic), and None. A note below says 'Method used to authenticate your application with the authorization server.' and a link to 'Learn more about authentication methods'. At the bottom, there is a section titled 'Client Secret'.

Go to the API tab and enable all API's



The screenshot shows the same 'microservices' application configuration page, but the 'APIs' tab is now selected in the top navigation bar. The sidebar remains the same. The main content area now displays a list of APIs. It starts with a heading 'Here is a list of your APIs. You can authorize this application to request access tokens for these APIs by executing a client credentials exchange.' Below this is a search bar labeled 'Filter by API Name or Identifier'. The list includes three items: 'Auth0 Management API' (API Identifier: https://dev-0nrrcude14pccq4i.us.auth0.com/api/v2/), 'Customer APIs' (API Identifier: http://customer/api), and 'Admin API' (API Identifier: http://admin/api). Each item has an 'Authorized' toggle switch, which is turned on for all three. There are also dropdown arrows next to each toggle.

For each API added, add the Permissions available from that API as below.

The screenshot shows the configuration for the 'Customer APIs' section. The left sidebar lists various application components like Getting Started, Activity, Applications, Authentication, Organizations, User Management, Branding, Security, Actions, and Auth Pipeline. The main panel displays the 'Customer APIs' details, including its API Identifier: `http://customer/api`. An 'Authorized' toggle switch is turned on. A 'Permissions' section lists the selected scope 'Customer'. Below this, an 'Organization Support' section shows three options: 'None' (selected), 'Optional', and 'Required'. The 'None' option is described as 'Machine to machine access cannot be secured'. The 'Optional' option is described as 'Machine to machine access may be secured'. The 'Required' option is described as 'Machine to machine access must be secured'.

The screenshot shows the configuration for the 'Admin API' section. The left sidebar is identical to the previous screenshot. The main panel displays the 'Admin API' details, including its API Identifier: `http://admin/api`. An 'Authorized' toggle switch is turned on. A 'Permissions' section lists the selected scope 'Admin'. Below this, an 'Organization Support' section shows three options: 'None' (selected), 'Optional', and 'Required'. The 'None' option is described as 'Machine to machine access cannot be secured'. The 'Optional' option is described as 'Machine to machine access may be secured'. The 'Required' option is described as 'Machine to machine access must be secured'.

Now, your Application is configured to use Users, Roles, and Permissions.

## Add Roles and Audience in Access Token

- Auth0 by default does not add Roles and Audience information in the access token.
- When the access token is created, this information is needed to authorize the request.
- To add this information in the Access token, Triggers need to be set up in the Auth0 flow when the user is logged in.
- So, Create the Library which has code to add the details post Login.
- Add Trigger flow to invoke the Custom Library created.
- Go to Actions → Library to create Custom Library
- Go to Custom tab, click on Crate Action, and select Build from Scratch
- Give the appropriate name of the Library.
- Select “Login/post Login” trigger from drop-down.
- Select any Node to run the code.
- Click on Create to create the library.
- This will open the code editor to add the code.
- If the same is not opened, select the library created from the list and it will open the code editor.
- Add the code below and click on deploy to deploy the library.

The screenshot shows the Auth0 Library page. The left sidebar navigation includes: Authentication, Organizations, User Management (with sub-options Users and Roles), Branding, Security, Actions (with sub-options Triggers, Forms, Library), Auth Pipeline, Monitoring, Marketplace, Extensions, and Settings. The main content area is titled 'Library' and contains the message: 'Manage your actions and configuration.' Below this, there are tabs for 'Installed' and 'Custom', with 'Custom' selected. A table lists two actions:

Name	Trigger	Last Update	Last Deploy	Status	More
Add Audience	Login / Post Login	11 hours ago	11 hours ago	DEPLOYED	...
Add Roles to Token	Login / Post Login	11 hours ago	11 hours ago	DEPLOYED	...

At the bottom, it says '2 actions'.

The screenshot shows the Auth0 Library interface. On the left, there's a sidebar with navigation links: Authentication, Organizations, User Management (with sub-links for Users, Roles, and Triggers), Branding, Security, Actions (with sub-links for Triggers, Forms, Library, and Marketplace), Auth Pipeline, Monitoring, Extensions, and Settings. The main area is titled "Library" and contains a message: "Thank you for signing up for Auth0! You have 21 days left in your trial to experiment with features that are not in the Free plan. Like what you're seeing? Please enter your billing information here." Below this is a "Create Action" button, a "View Plans" button, and a dropdown menu with options: "Use a Marketplace Integration", "Choose a template", and "Build from scratch". A "Build from scratch" button is also present. The "Library" section has tabs for "Installed" and "Custom", with "Custom" being selected. In the center, there's a "Create Action" modal window.

The "Create Action" modal is open, showing the following fields:

- Name \***: Add Roles to Token
- Trigger \***: Login / Post Login
- Runtime \***: Node 22 (Recommended)

At the bottom of the modal are "Cancel" and "Create" buttons. The background of the main library interface shows a table with two rows of deployment logs, each with "Last Deploy" at 11 hours ago and "Status" as "DEPLOYED".

The screenshot shows the Auth0 dashboard interface. On the left, there's a sidebar with various navigation items like Authentication, Organizations, User Management, Branding, Security, Actions, and Settings. The 'Actions' section is expanded, showing Triggers, Forms, Library, Auth Pipeline, Monitoring, Marketplace, Extensions, and Settings. The 'Library' section is selected. In the main content area, a modal titled 'Add Roles to Token' is open. The modal has tabs for 'Version History', 'Save Draft', and 'Deploy'. It shows a code editor with the following JavaScript code:

```
4 * @param {Event} event - Details about the user and the context in which they a
5 * @param {PostLoginAPI} api - Interface whose methods can be used to change the
6 */
7 exports.onExecutePostLogin = async (event, api) => {
8   const namespace = "https://dailycodebuffer.com/"; // Custom namespace (must be
9   // Check if the user has assigned roles
10  if (event.authorization && event.authorization.roles) {
11    api.accessToken.setCustomClaim(namespace + "roles", event.authorization.role
12  }
13}
14
15
```

Below the code editor, there's a 'View Samples' button and an 'Add Secret' button.

```
exports.onExecutePostLogin = async (event, api) => {

  const namespace = "https://dailycodebuffer.com/"; // Custom namespace
  (must be unique)

  // Check if the user has assigned roles

  if (event.authorization && event.authorization.roles) {
    api.accessToken.setCustomClaim(namespace + "roles",
    event.authorization.roles);
  }
};
```

Here, the namespace can be any unique namespace for the application. Here I have added my domain as the namespace. This may not be a valid domain or endpoint.

Click on Deploy to deploy the changes.

This Library will add the roles in the access token when the token is generated post login flow.

Create a similar Library for adding the Audience to the token as well as below.

The screenshot shows the Auth0 dashboard with the following details:

- Project Name:** dev-0nrrcude14pccq4i
- Region:** DEVELOPMENT
- Custom Actions:** Custom Actions
- Action Type:** Add Audience
- Description:** Login / Post Login, Runtime: Node 22 (Recommended)
- Code Preview:**

```

4  * @param {Event} event - Details about the user and the context in which they ar
5  * @param {PostLoginAPI} api - Interface whose methods can be used to change the
6  */
7 exports.onExecutePostLogin = async (event, api) => {
8   const namespace = "https://dailycodebuffer.com/"; // Replace with your custom
9   api.accessToken.setCustomClaim(namespace + "audiences", [
10     "http://customer/api",
11     "http://admin/api"
12   ]);
13 }
14
15
16 /**
  
```
- Secrets:** Secrets allow you to securely define secret or privileged values that can be accessed in your running code as properties of the event.secrets object.
- Buttons:** Version History, Save Draft, Deploy, View Samples, Add Secret

```

exports.onExecutePostLogin = async (event, api) => {

  const namespace = "https://dailycodebuffer.com/"; // Replace with your
custom namespace

  api.accessToken.setCustomClaim(namespace + "audiences", [
    "http://customer/api",
    "http://admin/api"
  );
};

}
  
```

Add the above code in the Library and click on deploy to deploy your library.

## Add Triggers

- The two libraries created now to be added as part of the Post Login flow trigger.
- This trigger will be executed when the user logs in to the system and adds the above Roles and Audience information to the token generated.
- From the application, decode the JWT token and check for the Roles and Audience information, and authorize the user to allow the different APIs as per permission.
- Go to Actions → Triggers to add a trigger
- Click on post-login flow to add trigger
- Select both triggers and add to the flow and click on save.

The screenshot shows the Auth0 Triggers page. On the left, there's a sidebar with navigation links for Authentication, Organizations, User Management, Branding, Security, Actions (which is currently selected), Triggers, Forms, Library, Auth Pipeline, Monitoring, Marketplace, Extensions, and Settings. The main content area has a title "Triggers" and a sub-section "Sign Up & Login". It describes triggers for user creation and authentication. Three triggers are listed: "pre-user-registration" (triggers before a user is created), "post-user-registration" (triggers asynchronously after a user is created), and "post-login" (triggers after a user is authenticated but before a token is issued). Below this is a section for "MFA Notifications".

The screenshot shows the Auth0 Post Login configuration page. The sidebar is identical to the previous one. The main area starts with a message about a trial period and a "View Plans" button. Below it, there's a "Choose trigger" link and the title "Post Login". It says "Customize the login trigger for your applications." A flow diagram is shown: "Start User Logged In" leads to "Add Audience", which then leads to "Add Roles to Token". To the right, there's an "Add Action" panel with a search bar and three "Add Audience" and "Add Roles to Token" options. A note says "All changes are live".

- Go to Custom Action and add both actions in the flow.
- Drag and drop both the actions to the Post Login Flow
- Click on Apply to apply the settings