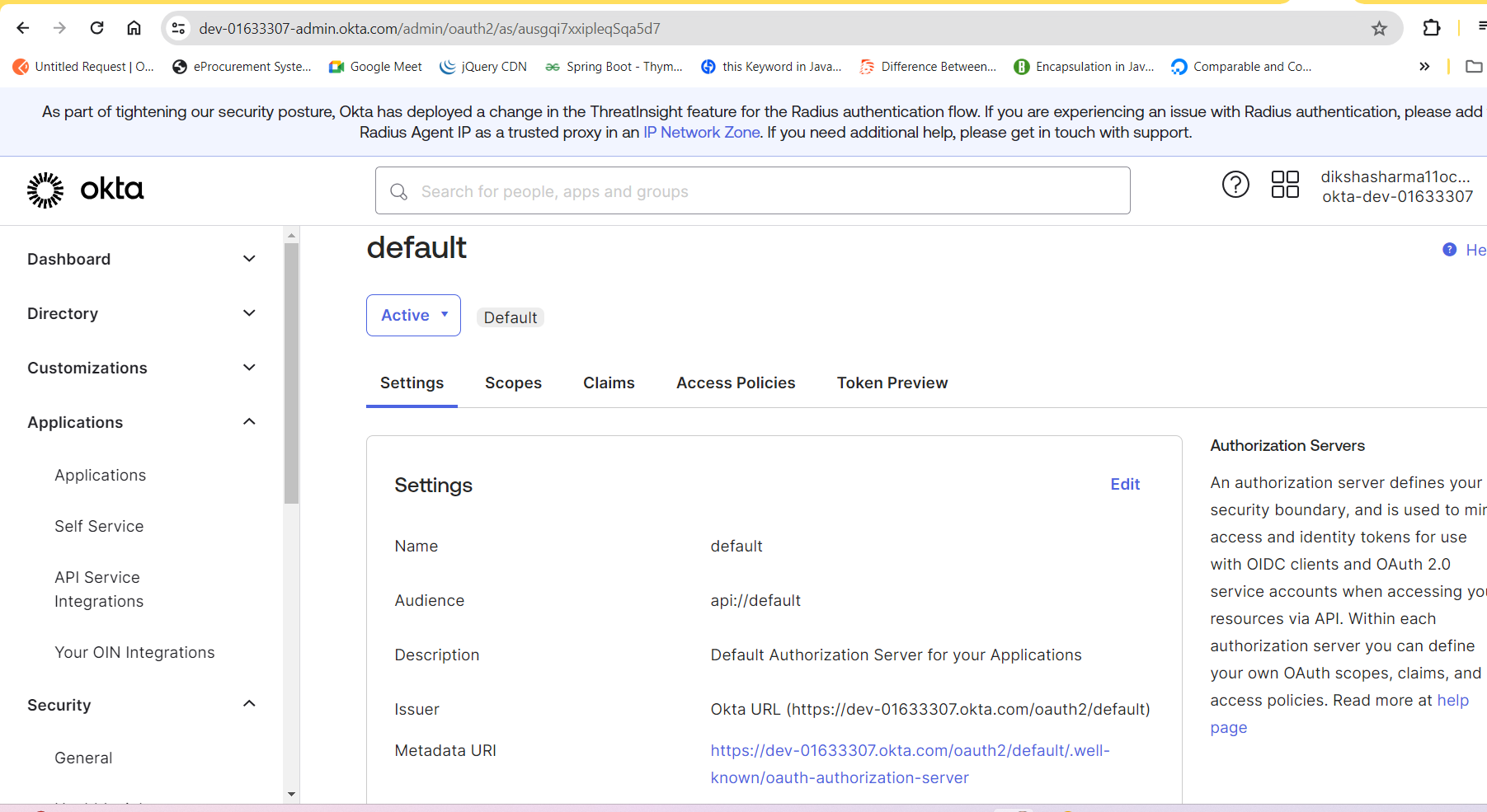
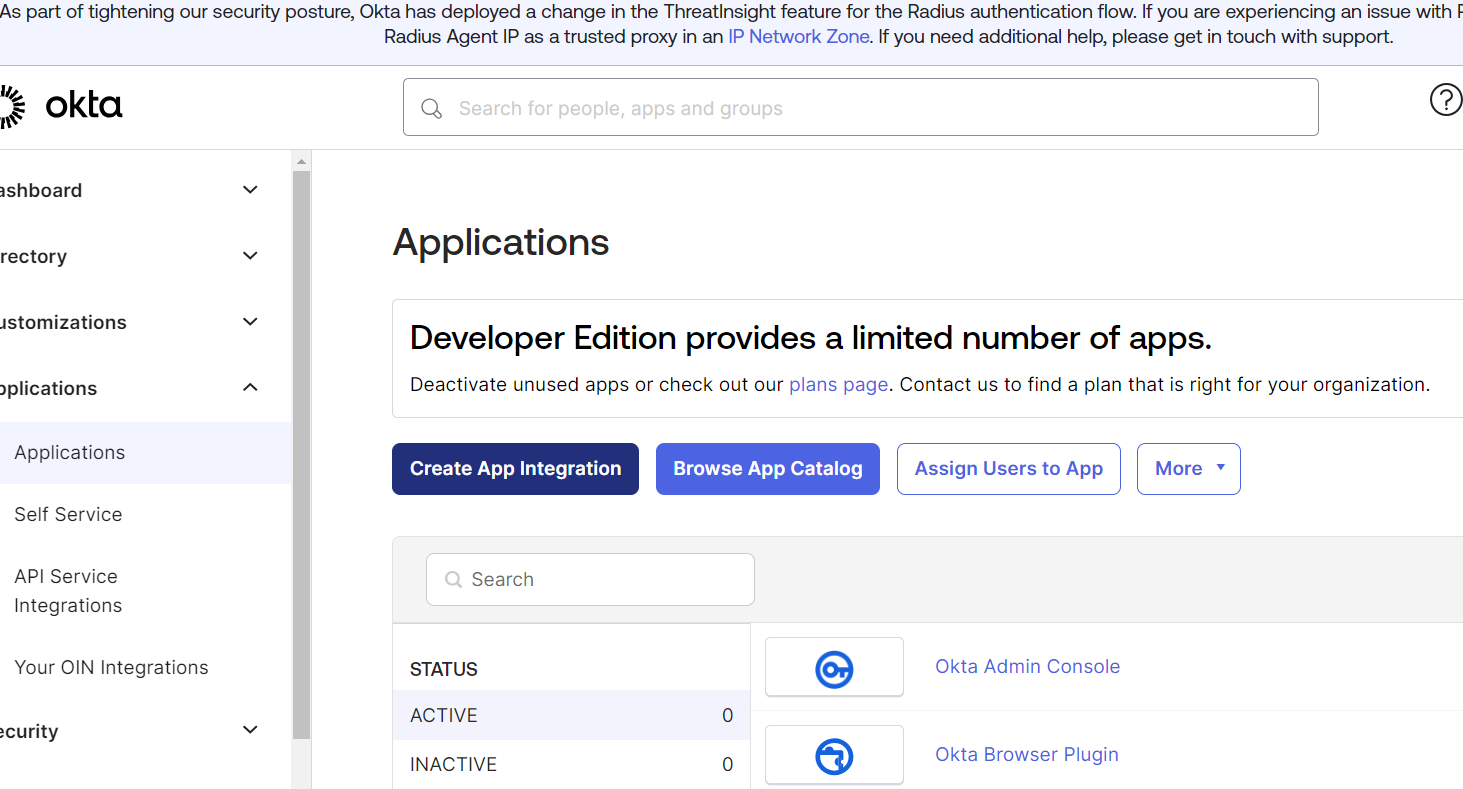
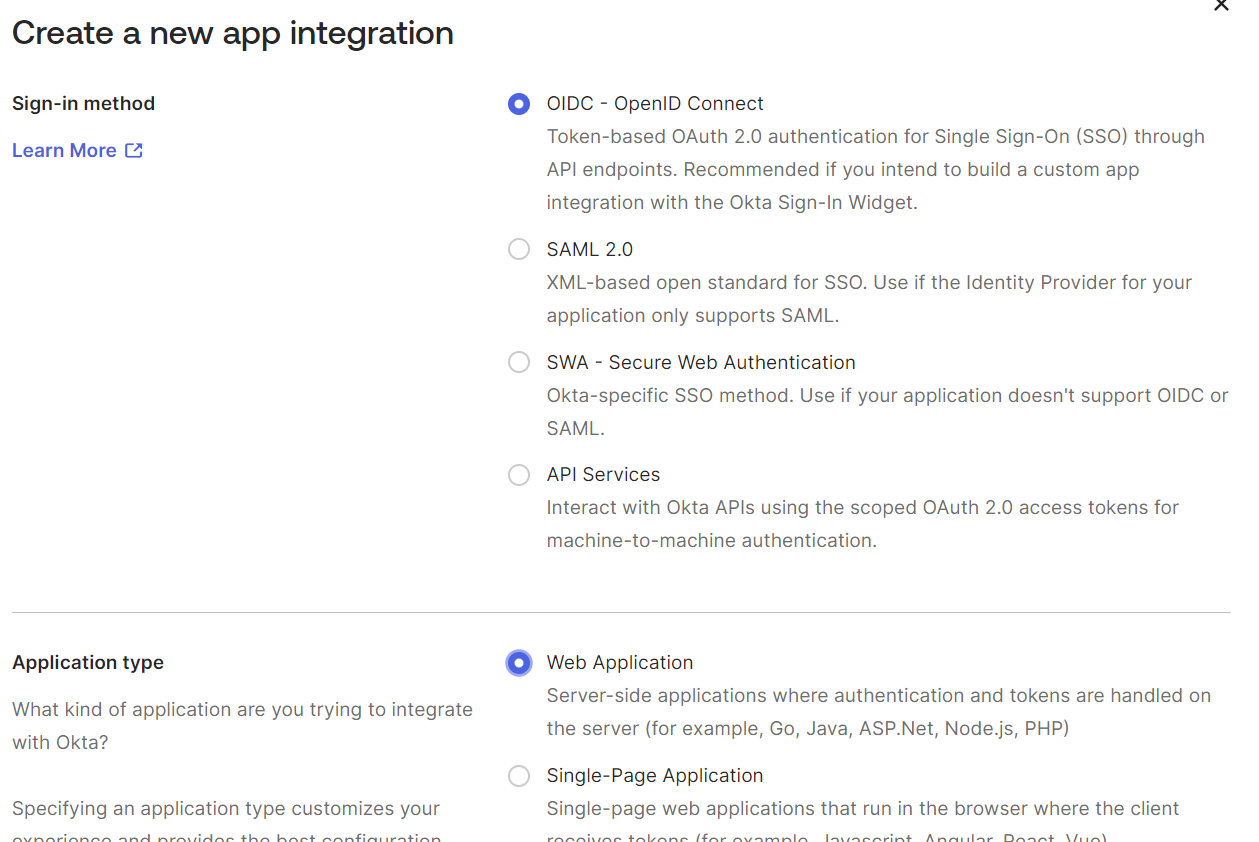
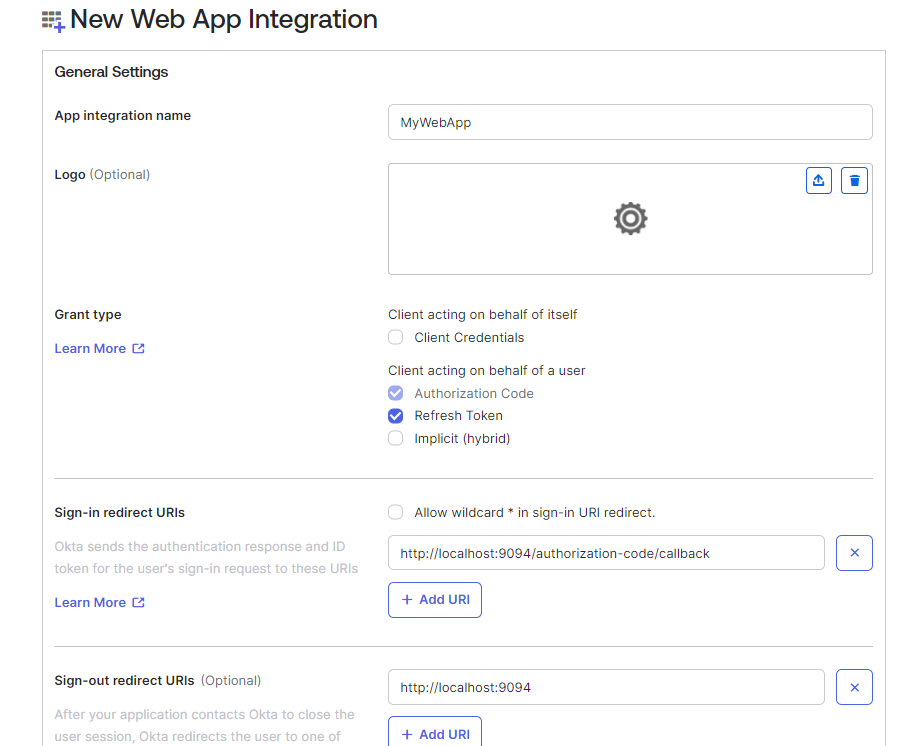
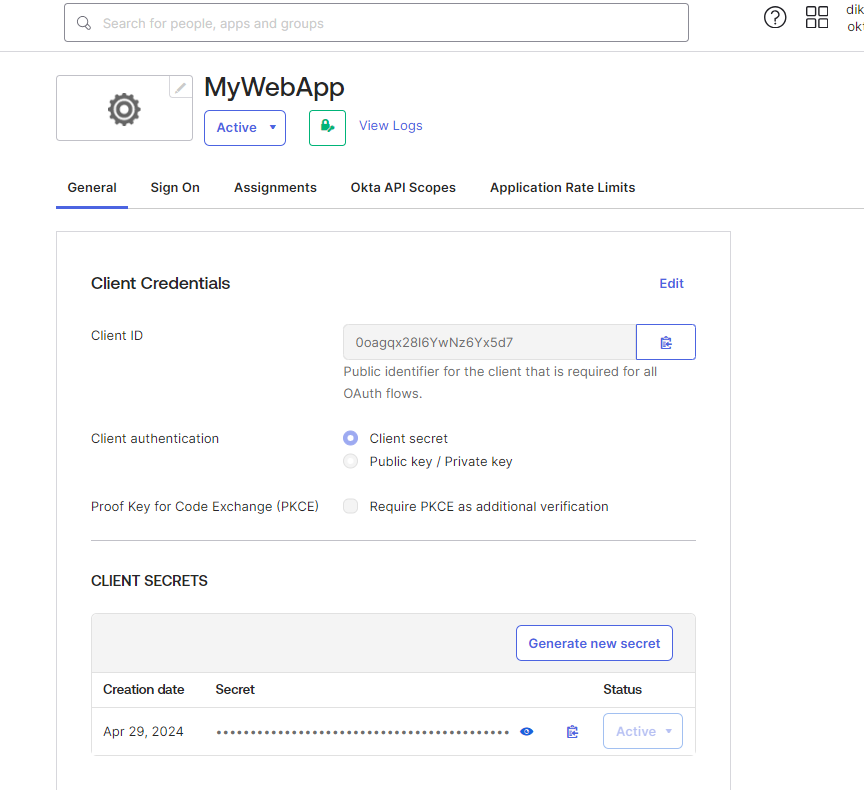
Click on applications in Okta developer website  


Click on Create ap application and select oidc as first option

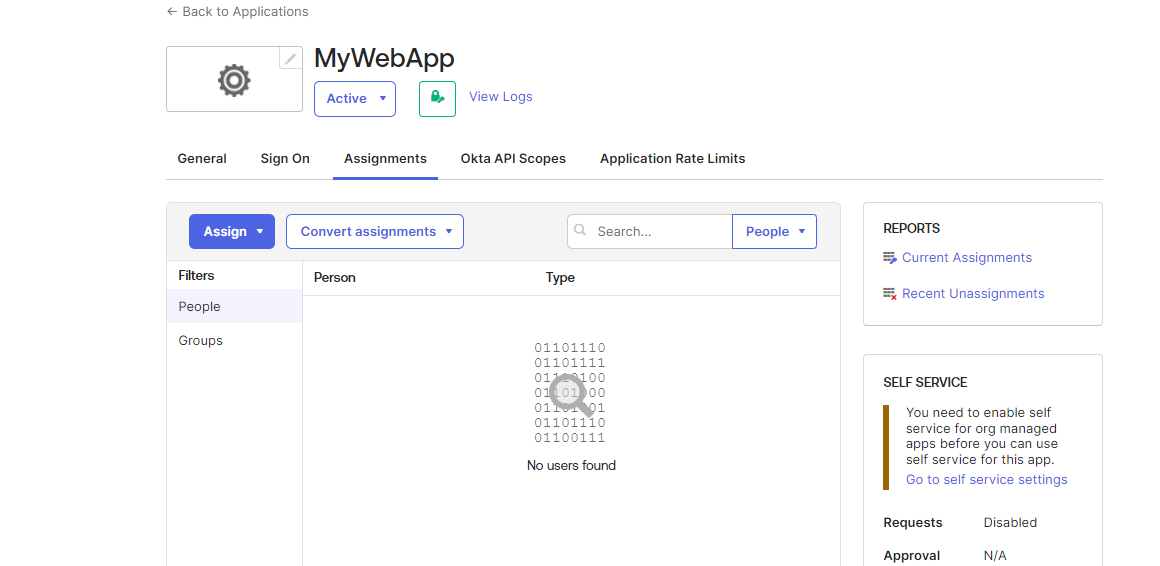


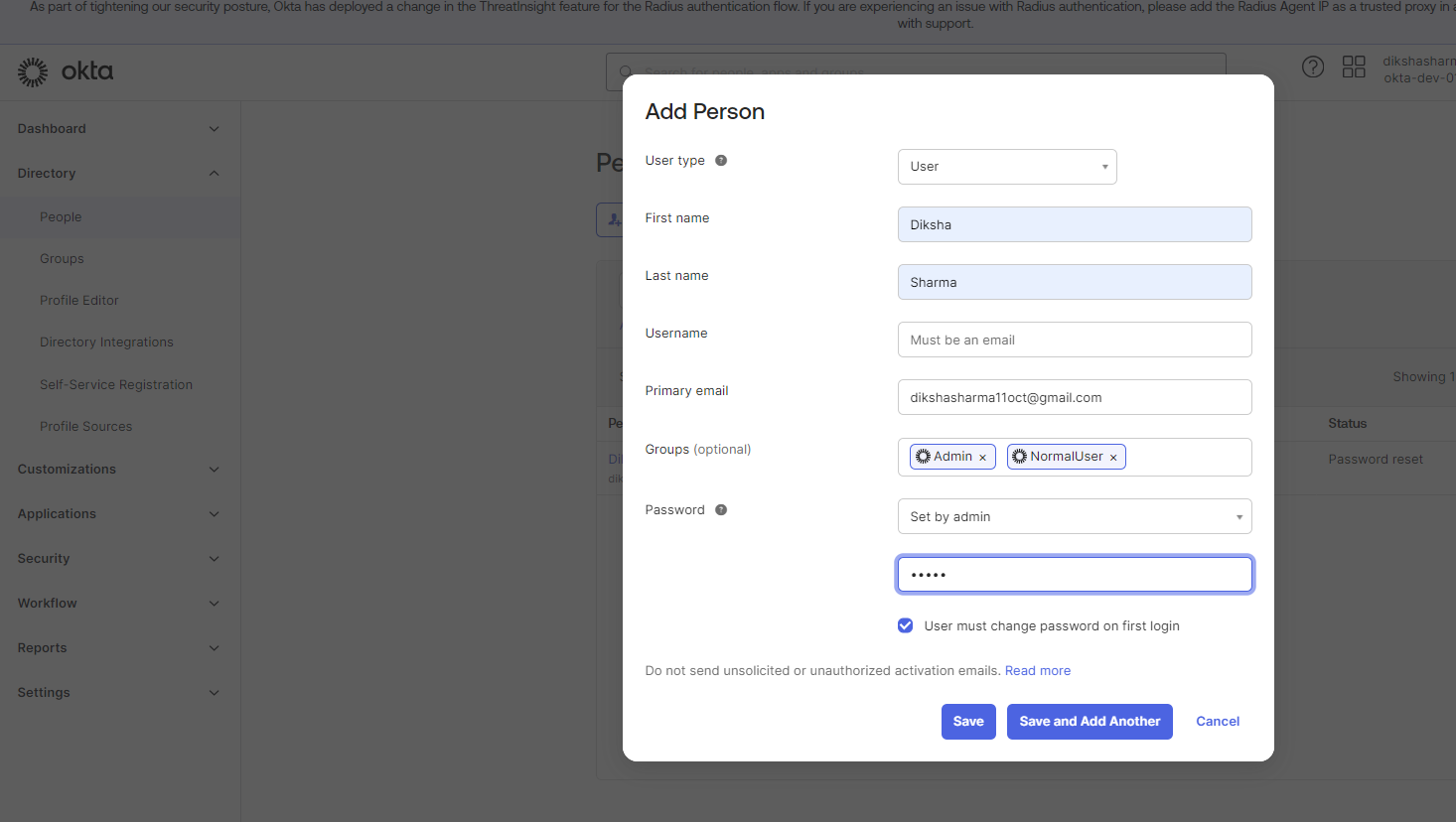


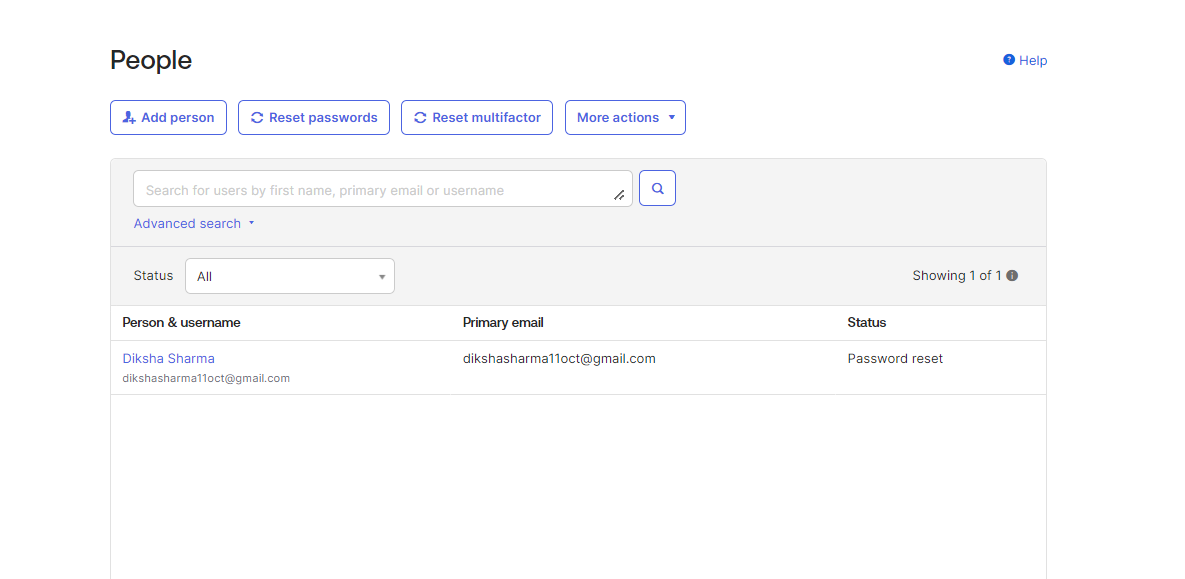


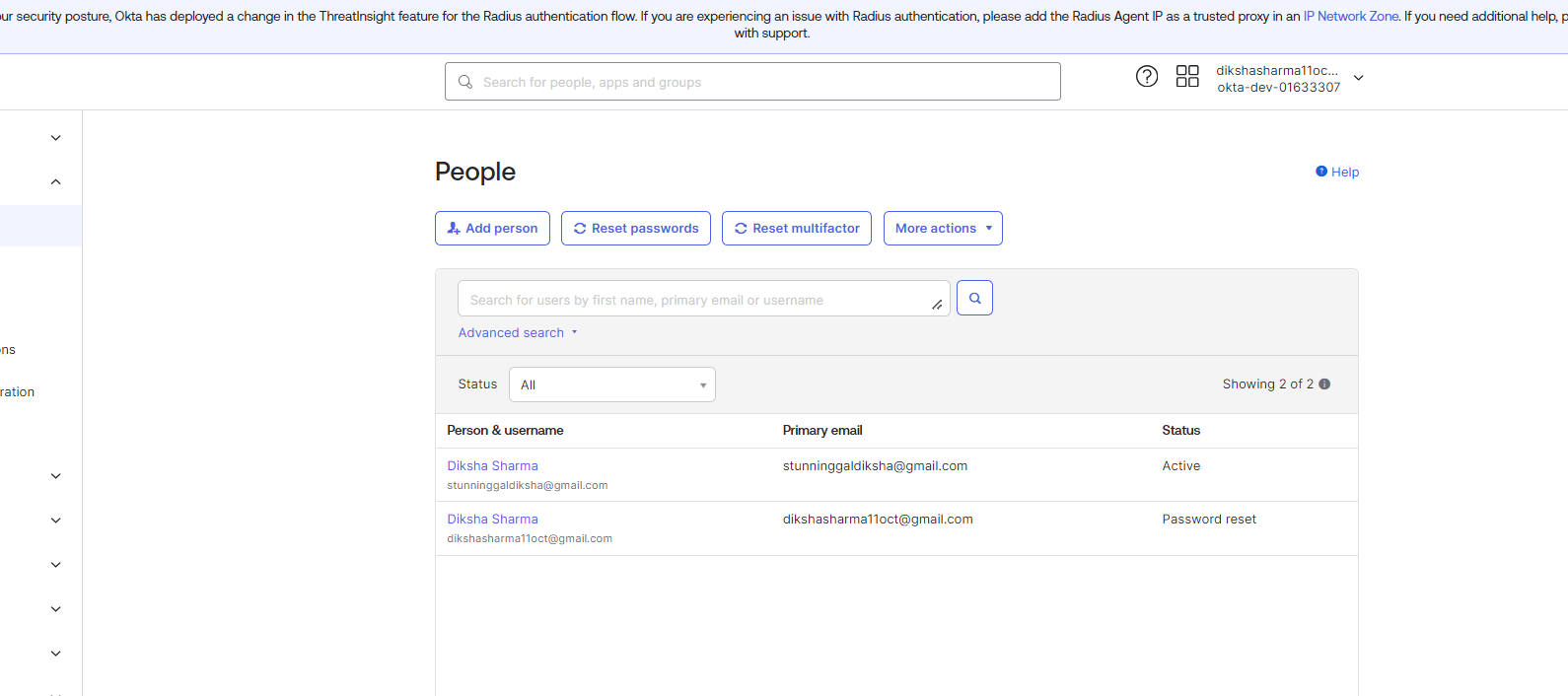
Add save groups can be skip so you add it later and now you will get client Id and Secret Key.  


To add people and groups here in assignment =>Go to Directory

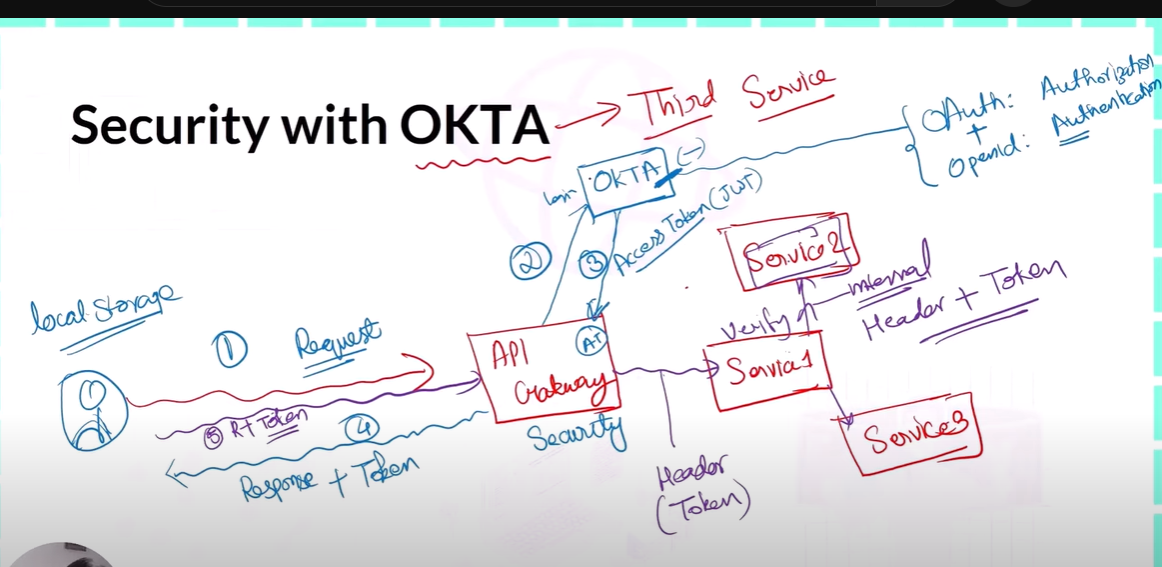


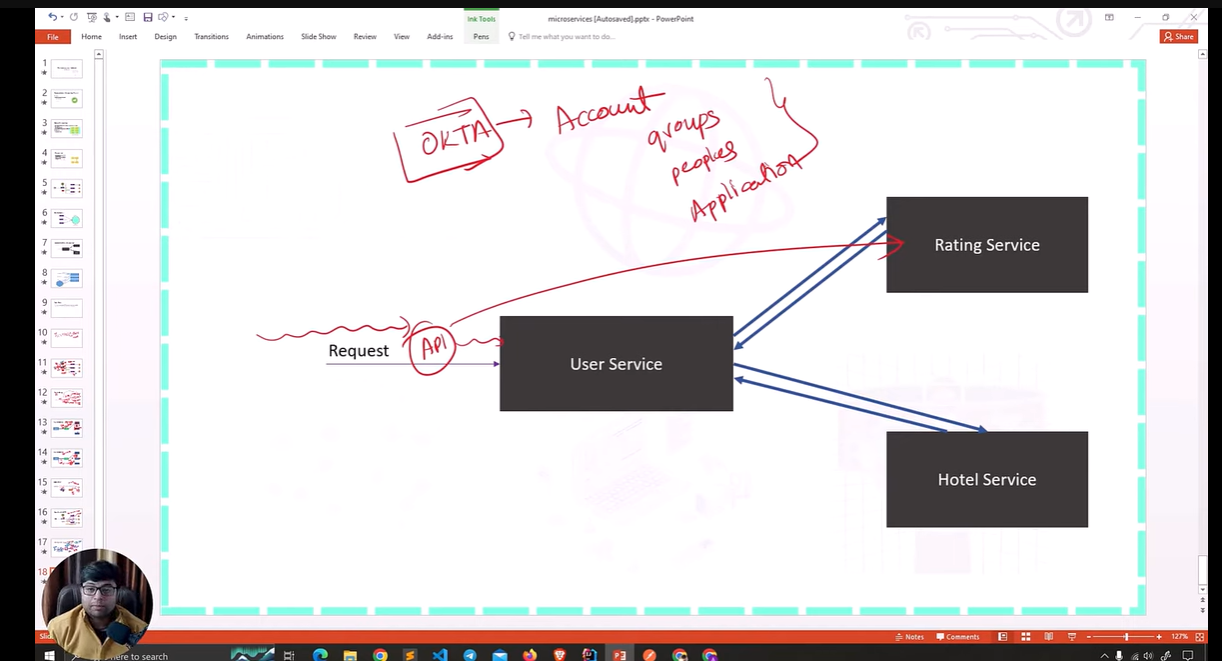


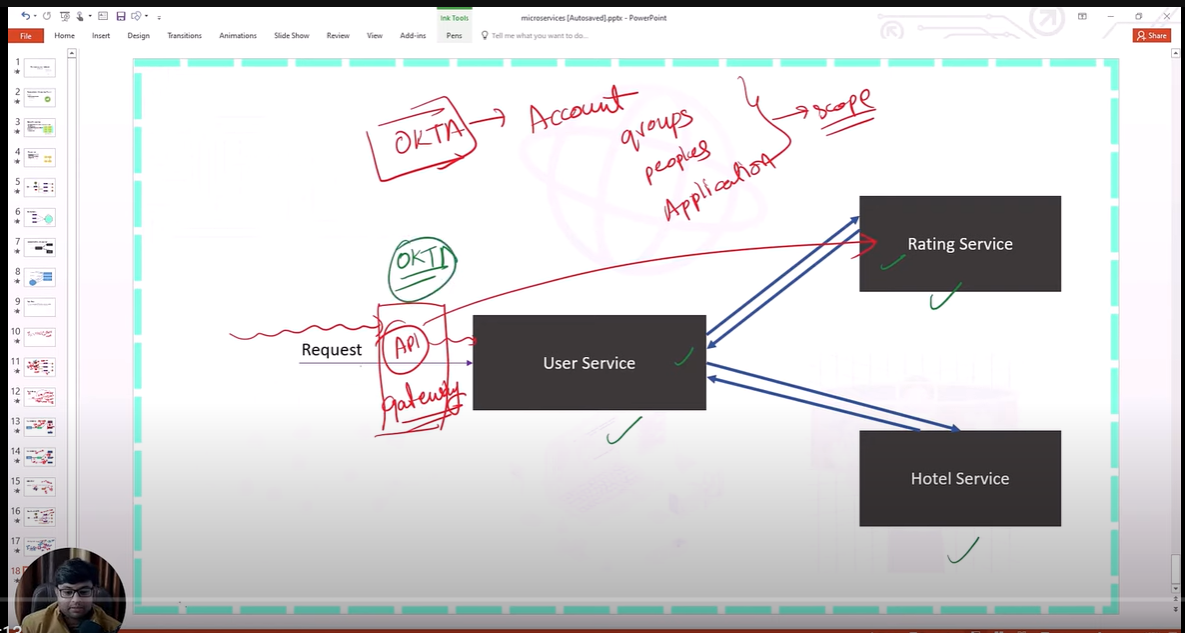
Pswd is dik@12345  




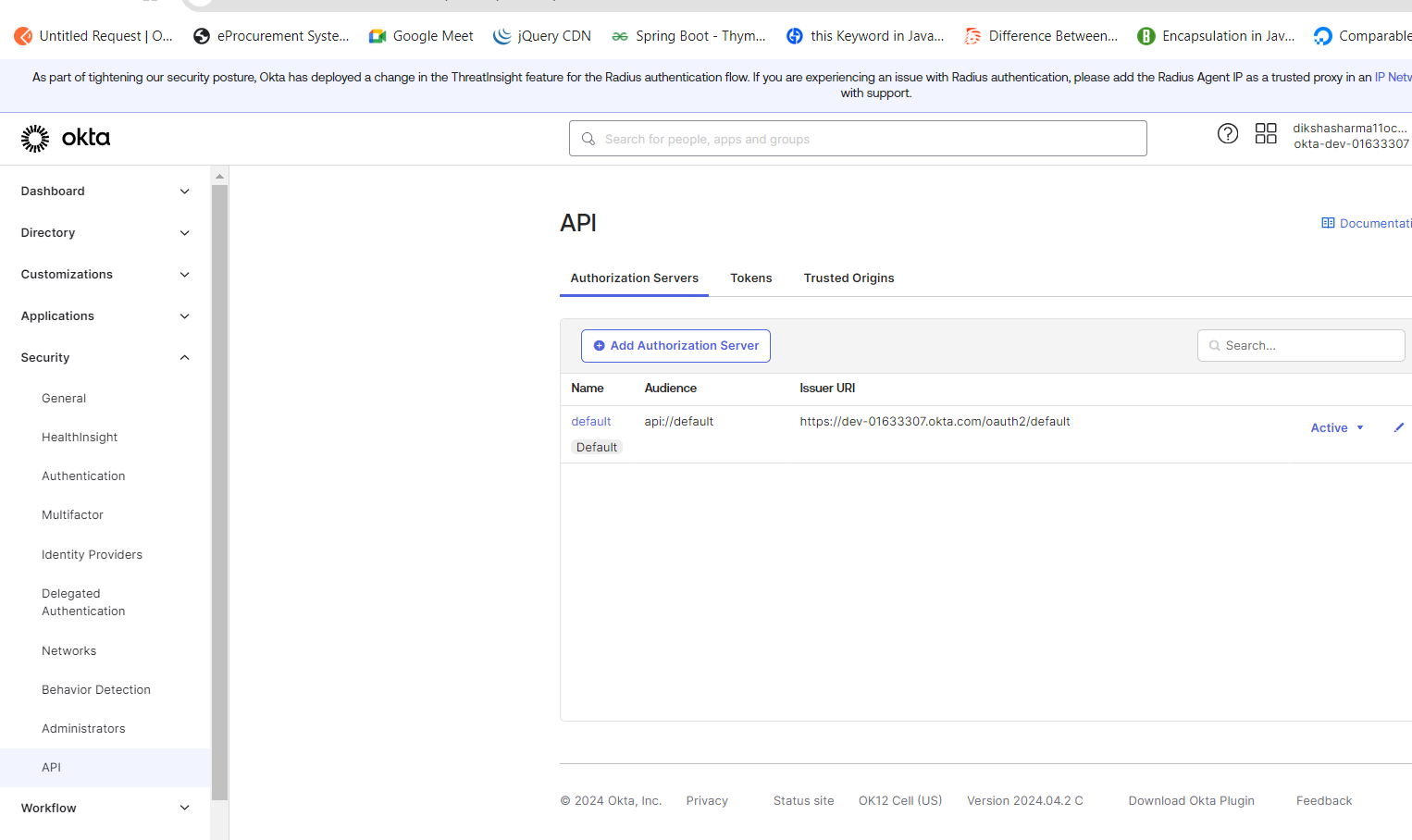
Click on first Diksha Sharma user and it will take you to the application

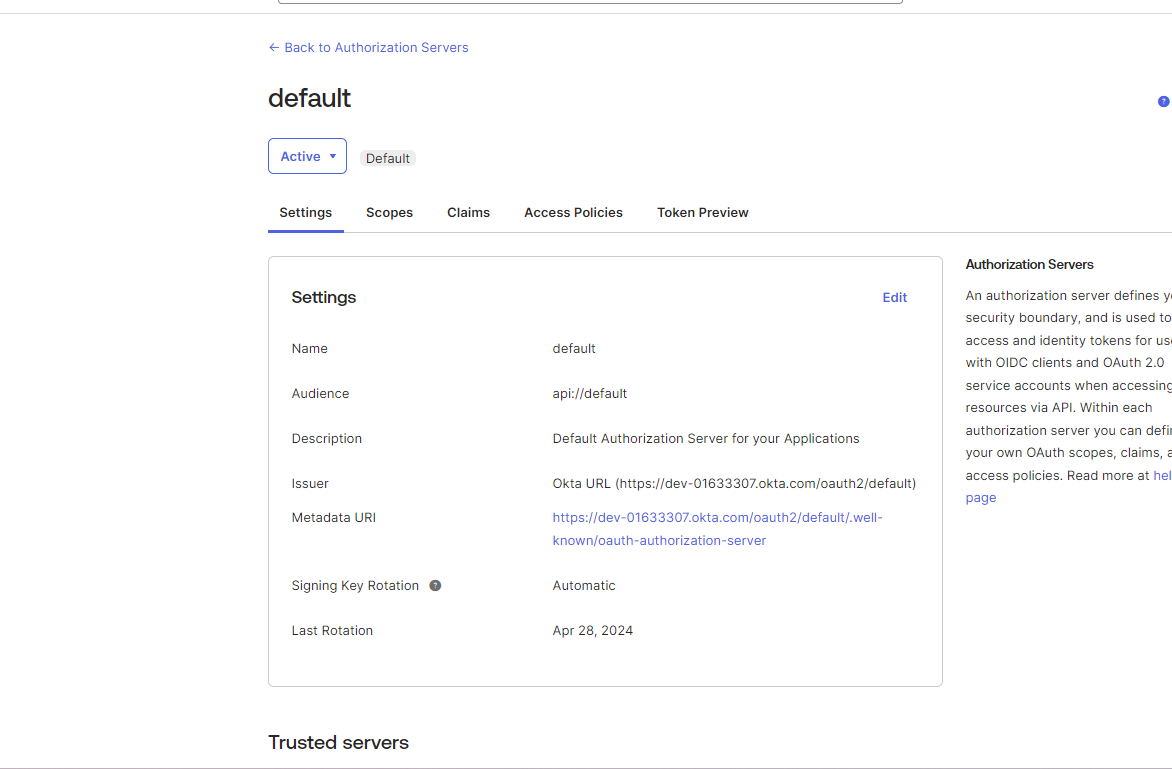




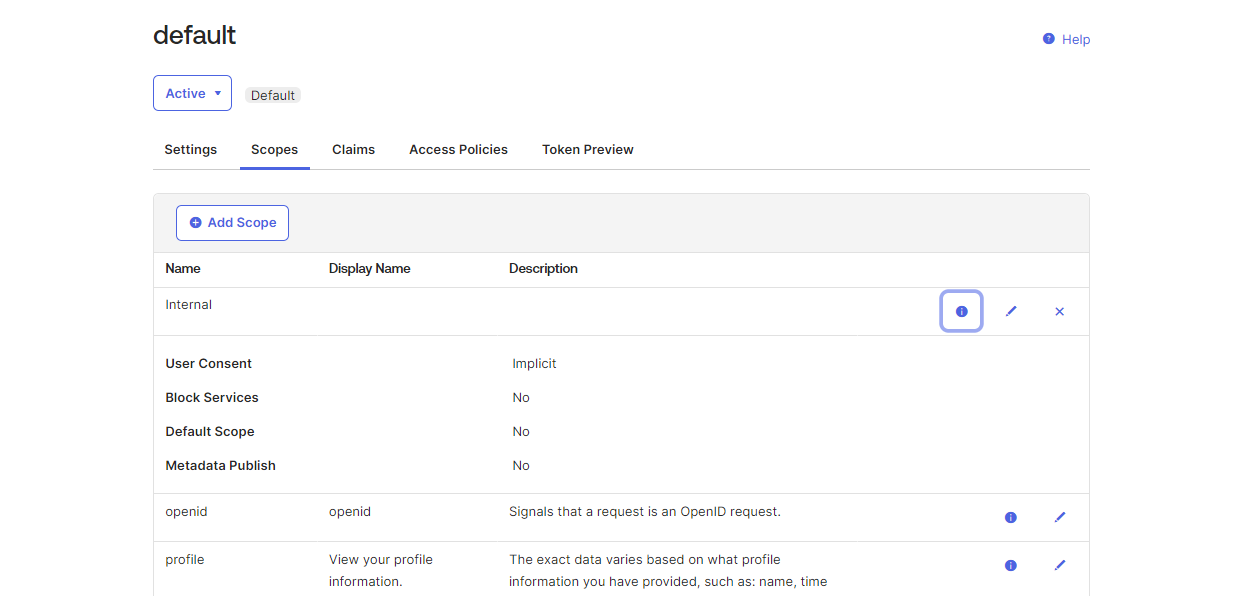


Click on security->api->default

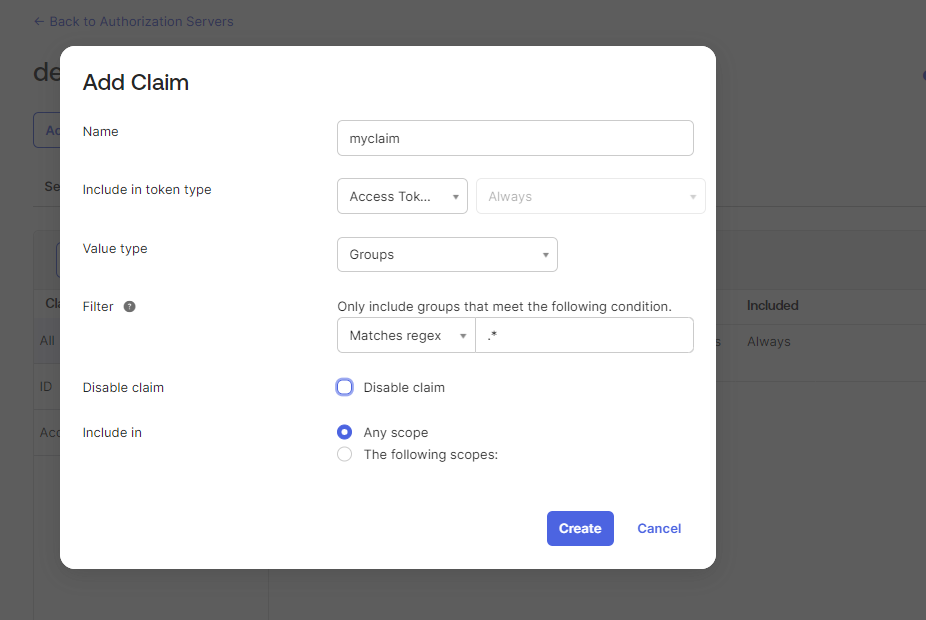




Add scope here only name of scope

  
then add claims

Click on save



Use these configuration here

#Okta config

okta.oauth2.issuer=https://dev-01633307.okta.com/oauth2/default

okta.oauth2.audience=api://default

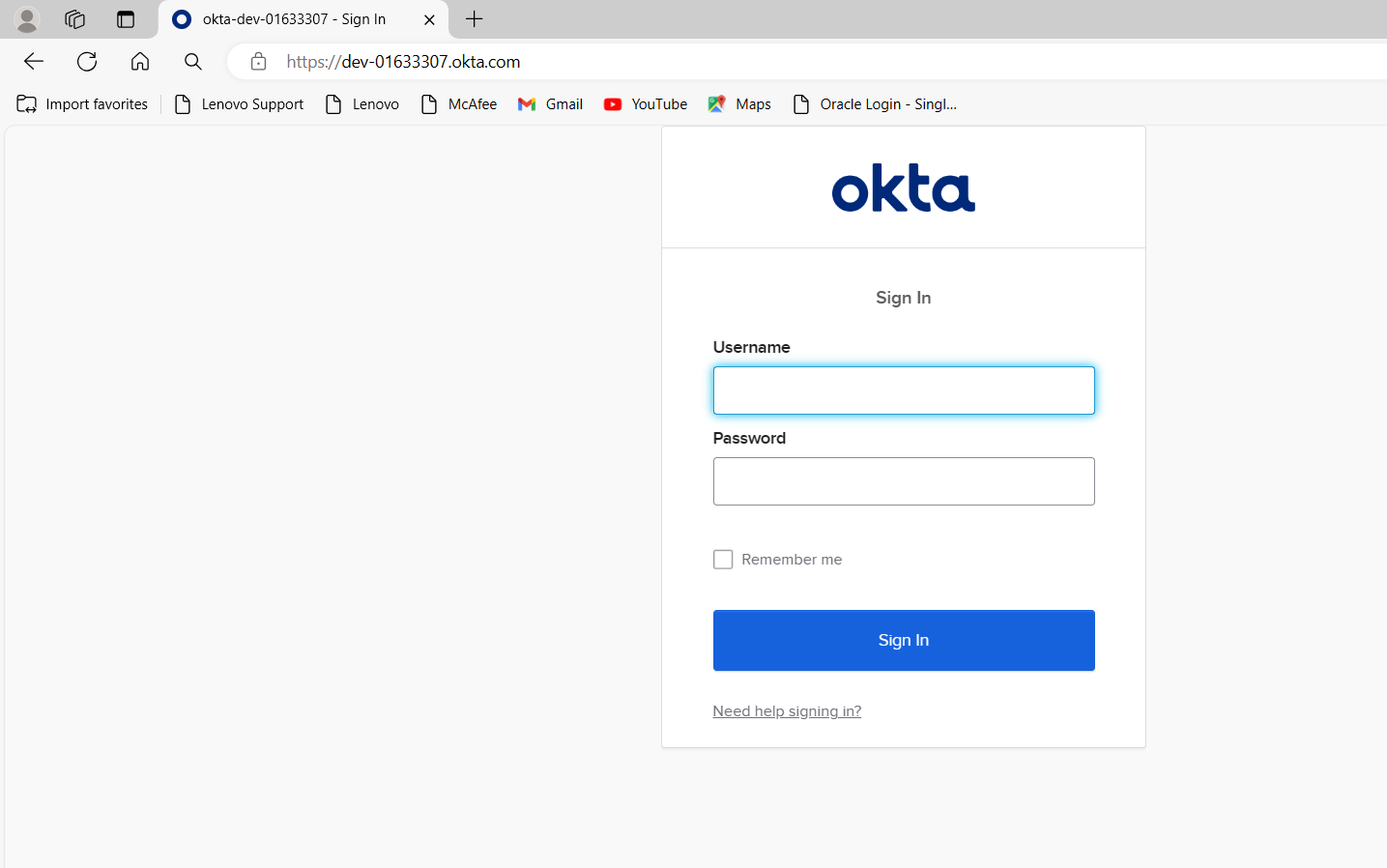
okta.oauth2.client-id=0oagqx28l6YwNz6Yx5d7

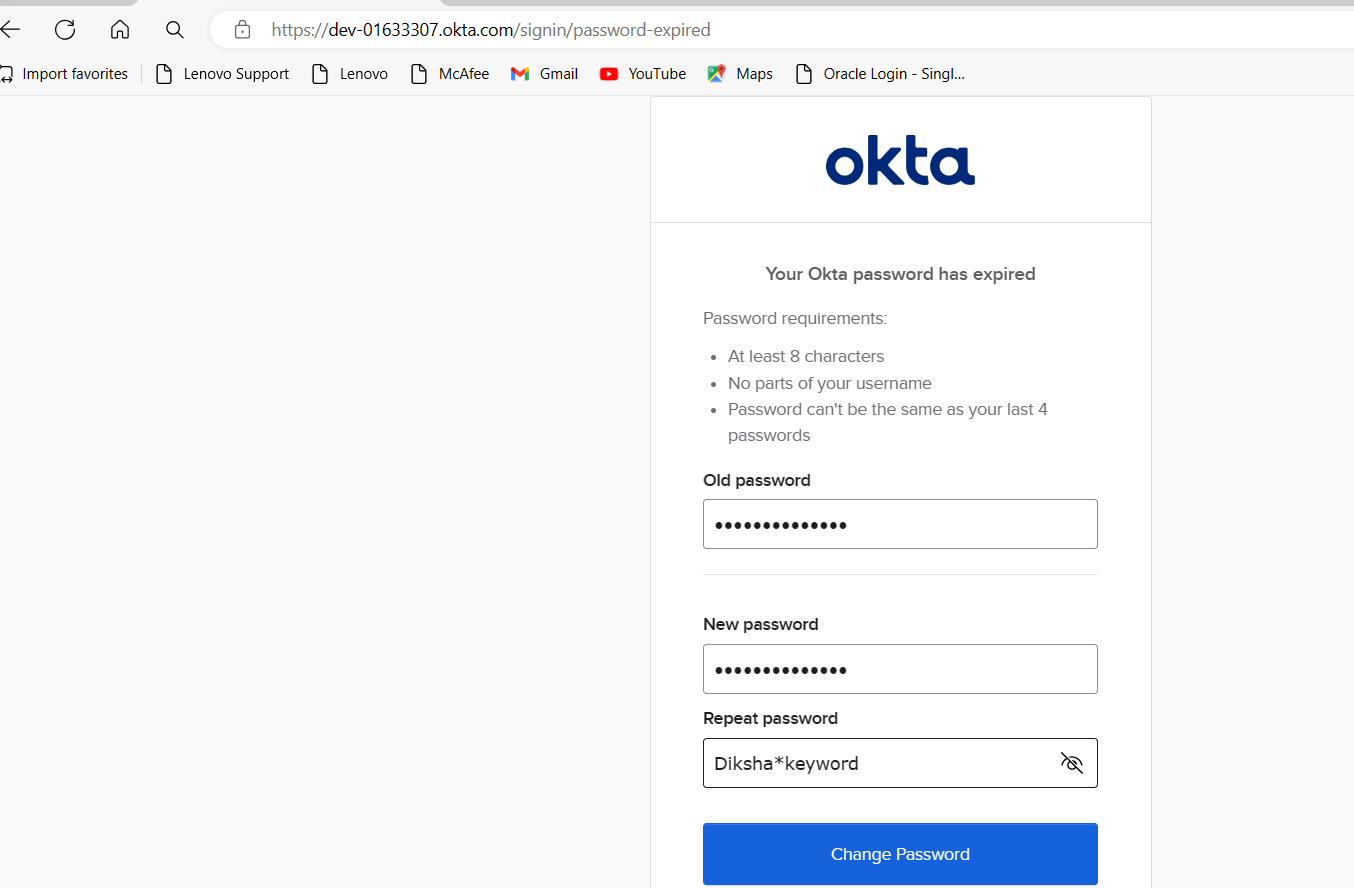
okta.oauth2.client-secret=KuktCATcQ-iaQmDjpAkMyozipTtUpMLHx7Broj3mCgwFdsZdNjkNLuM6lNOMjWyY

okta.oauth2.scopes=openid, profile, email, offline\_access

eureka.client.service-url.defaultZone=http://localhost:8761/eureka

Click on reset password to reset the passcode



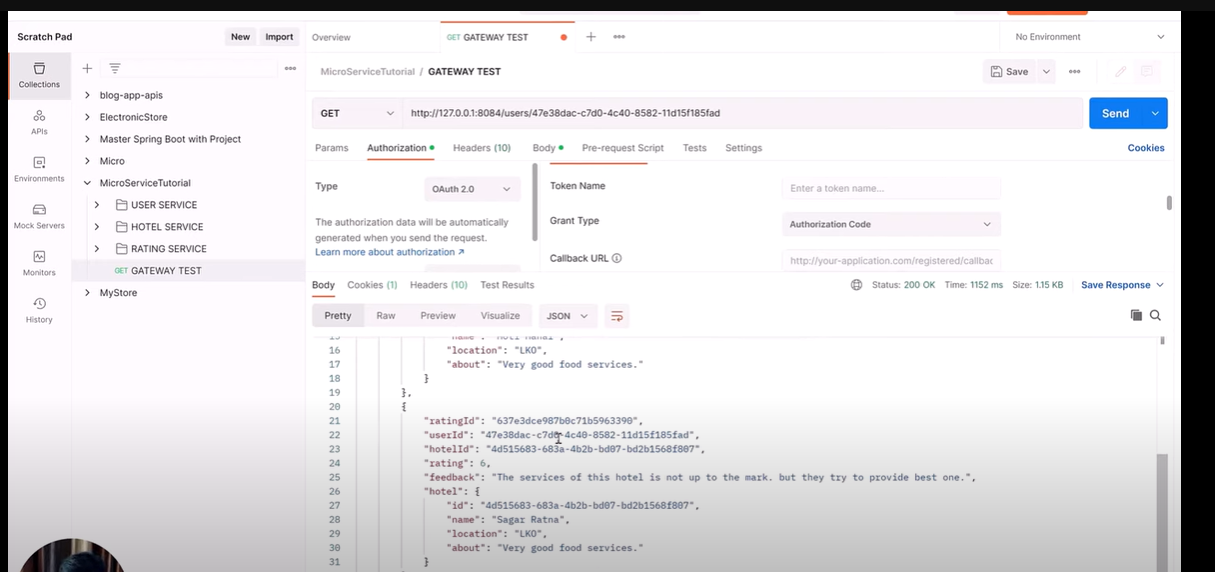


diksha#12345

diksha@847-> stunninggaldiksha@gmail.com



Use above token to access api those url has been registered with apiGateway



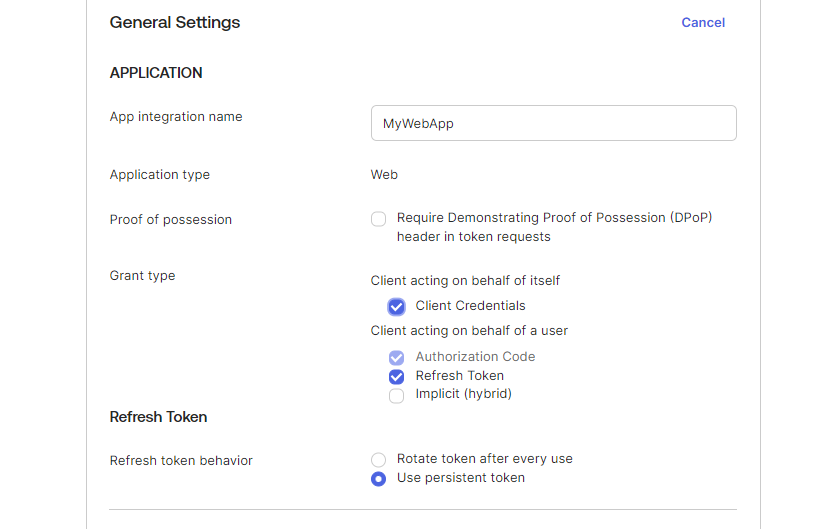
Eureka server: http://localhost:8761/

Hit this url for sign in localhost:9094/auth/login

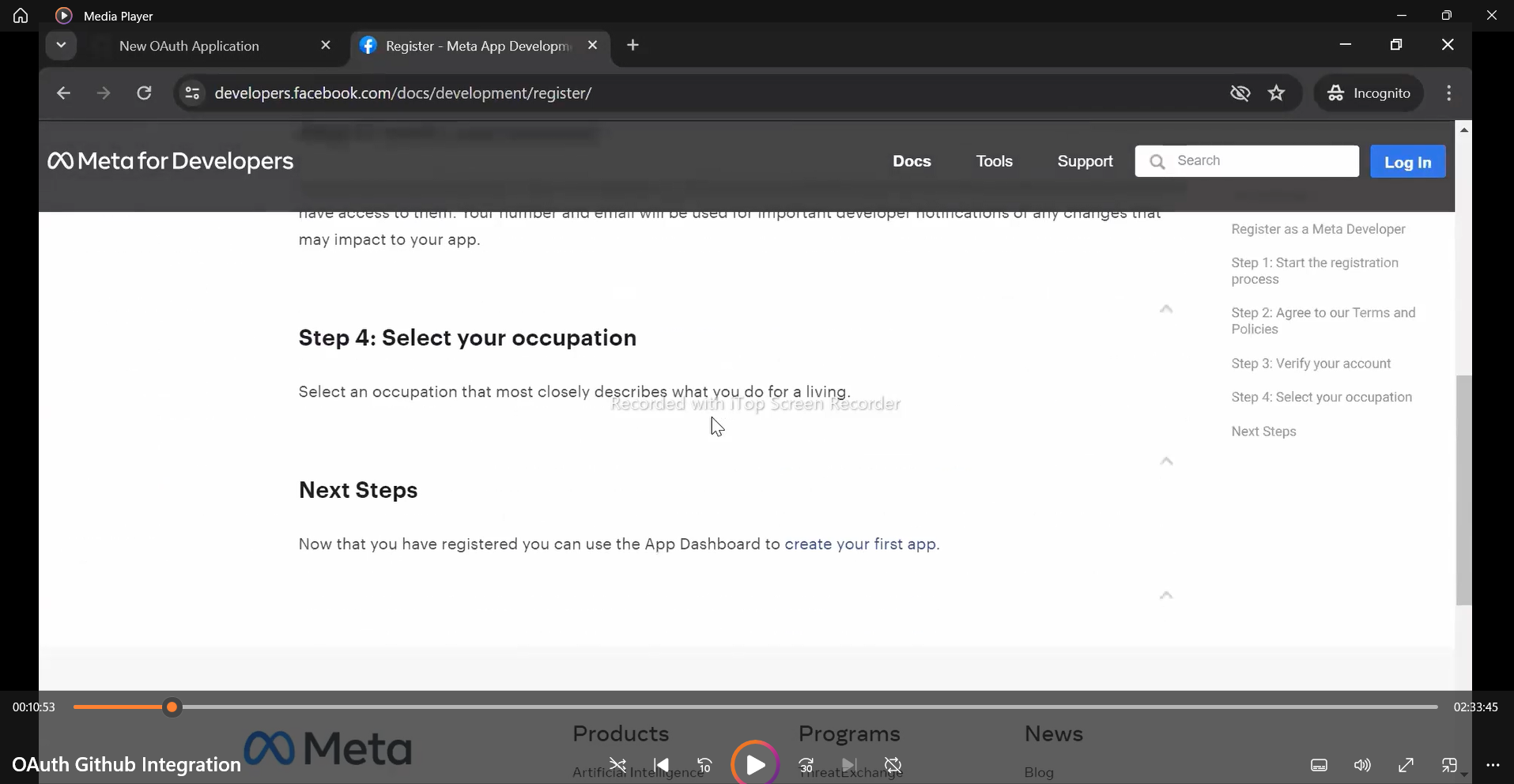
Need to make interceptor to call other service that is calling either by restClient or feign client

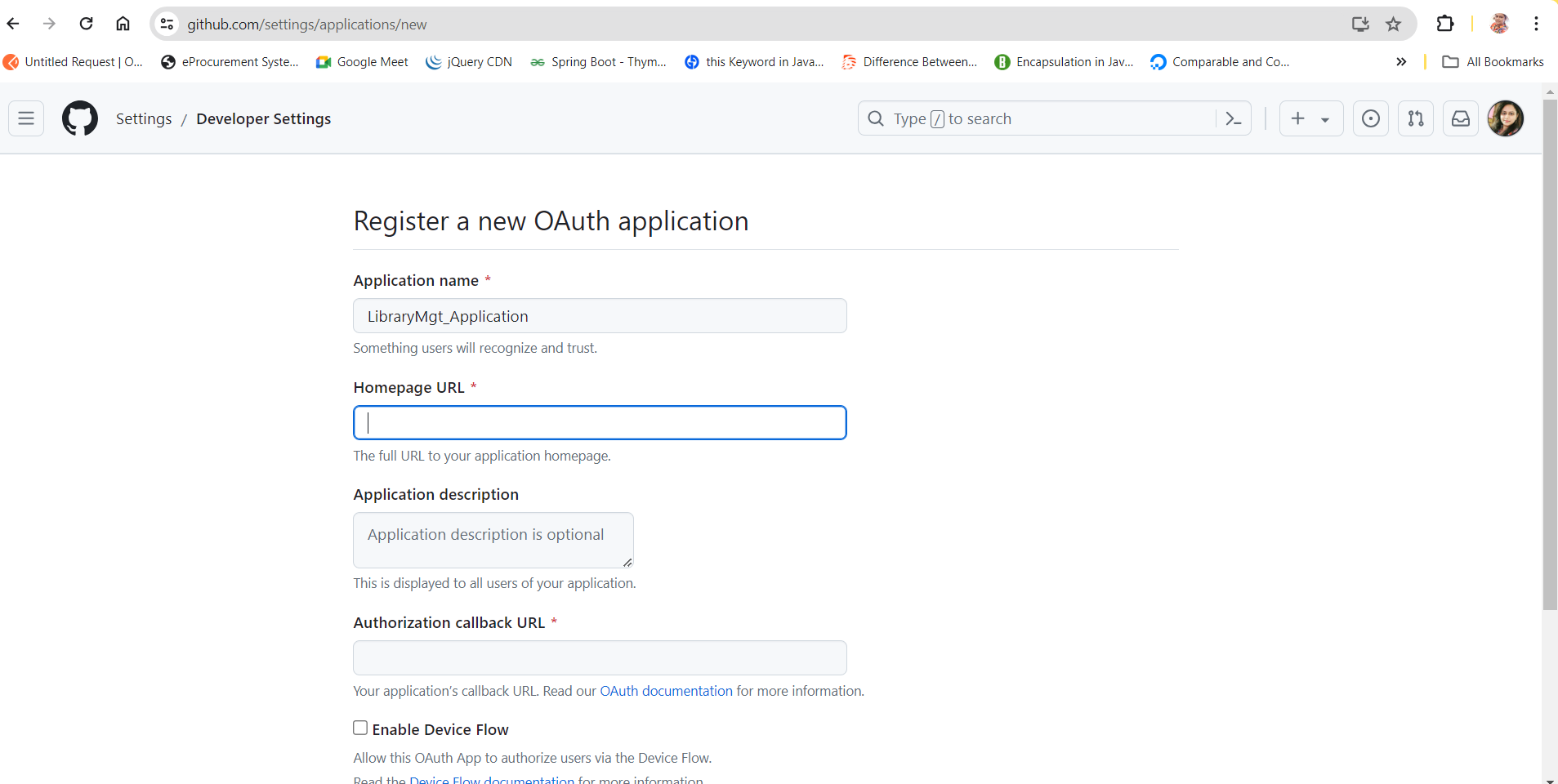
This interceptor will interrupt the calling of these feign clients and will provide a token

//declare the bean of OAuth2AuthorizedClient manager to get the token for above both interceptor



For Facebook Meta open this page for O-auth





**Operation->Login,**

**App-name->LinkedIn/GitHub/Google**

**Where you want to github to pass this token that is fallback url for authorization purpose**

**Uri when we login from gfg by using github**

https://github.com/login/oauth/authorize?client\_id=dd1ea782edfc6b22e890

&redirect\_uri=https%3A%2F%2Fauth.geeksforgeeks.org%2Fgithub-callback.php&

scope=user%3Aemail&

state=a4cdc2fb2df5cd81e1c94ea6a9d5ffb5a8ecd5b1b4080f6e5c5fafb7ec251bcc

and

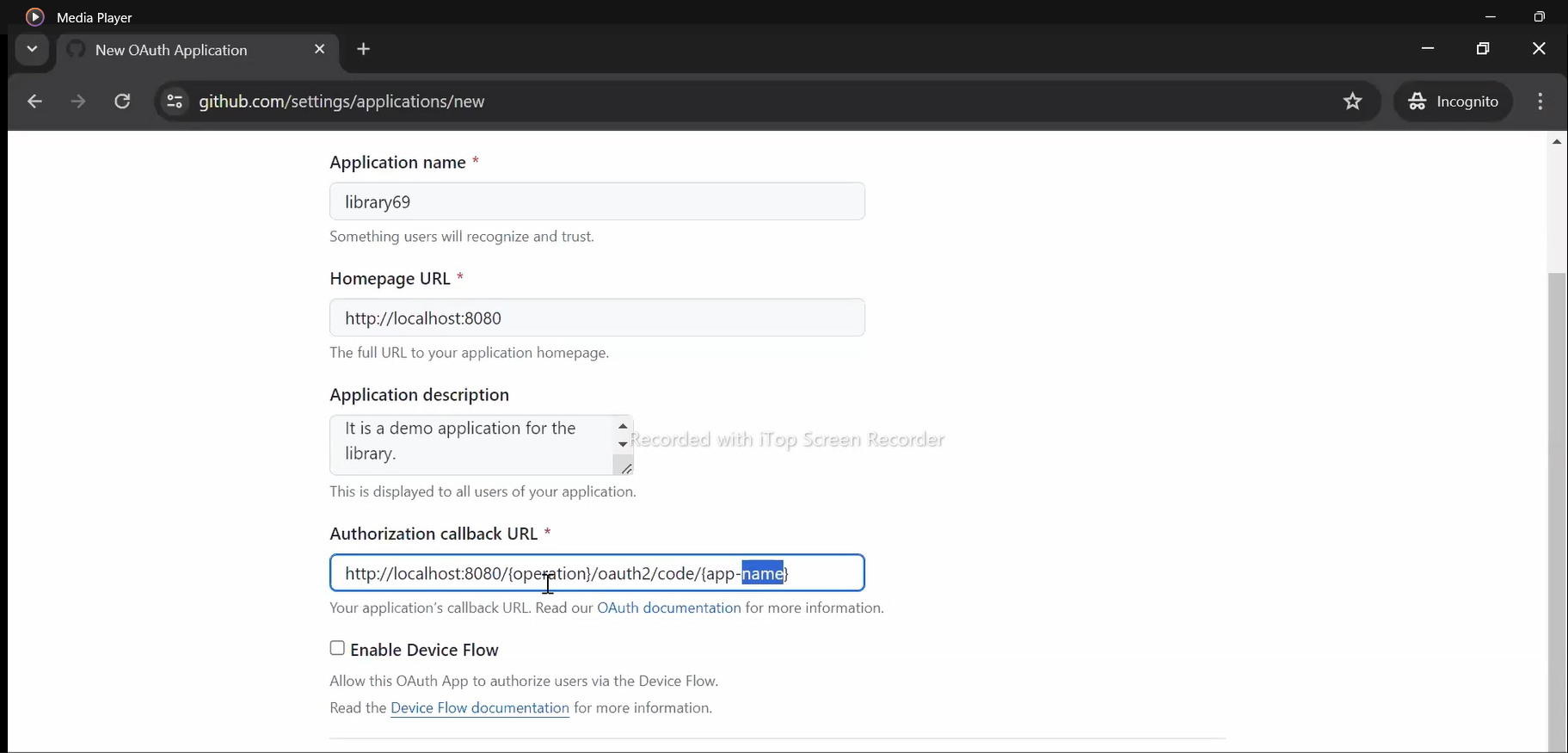
it’s decoded url ->use this link for decode->https://meyerweb.com/eric/tools/dencoder/  
https://github.com/login/oauth/authorize?client\_id=dd1ea782edfc6b22e890

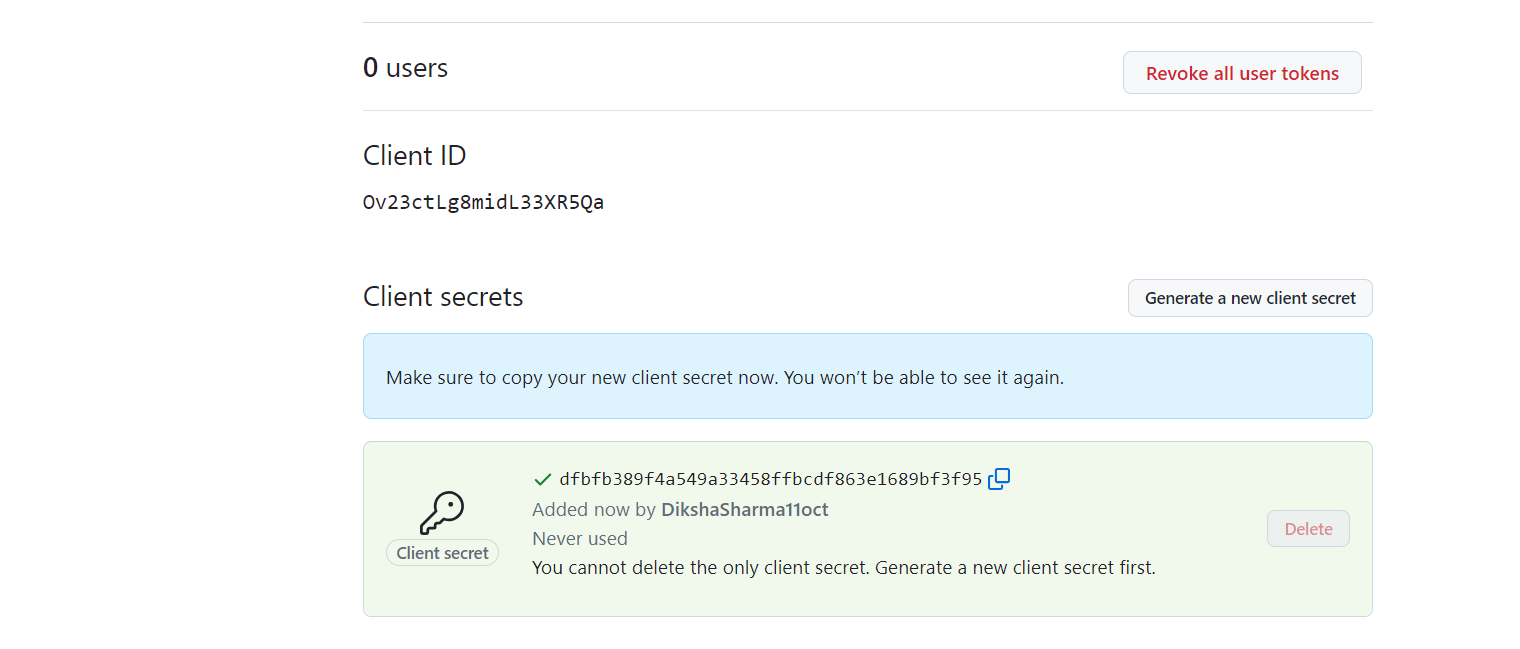
&redirect\_uri=https://auth.geeksforgeeks.org/github-callback.php&

scope=user: email&

state=a4cdc2fb2df5cd81e1c94ea6a9d5ffb5a8ecd5b1b4080f6e5c5fafb7ec251bcci can change the url, gfg can change the url but github will provide the details on basis of fallback method that is where scopes comes.  
using secret authentication comes in to picture  
what is authenticationv/s authrozation here  
ehcache where it stores the data.

We keep a seprate application->Application name and key name





**To Find Out the GitHub Scopes:**

<https://docs.github.com/en/apps/oauth-apps/building-oauth-apps/scopes-for-oauth-apps>  
state decides from where the request is coming. it’s a random generated string.  
it’s like csrf

Authenticate Object is responsible for authenticating

