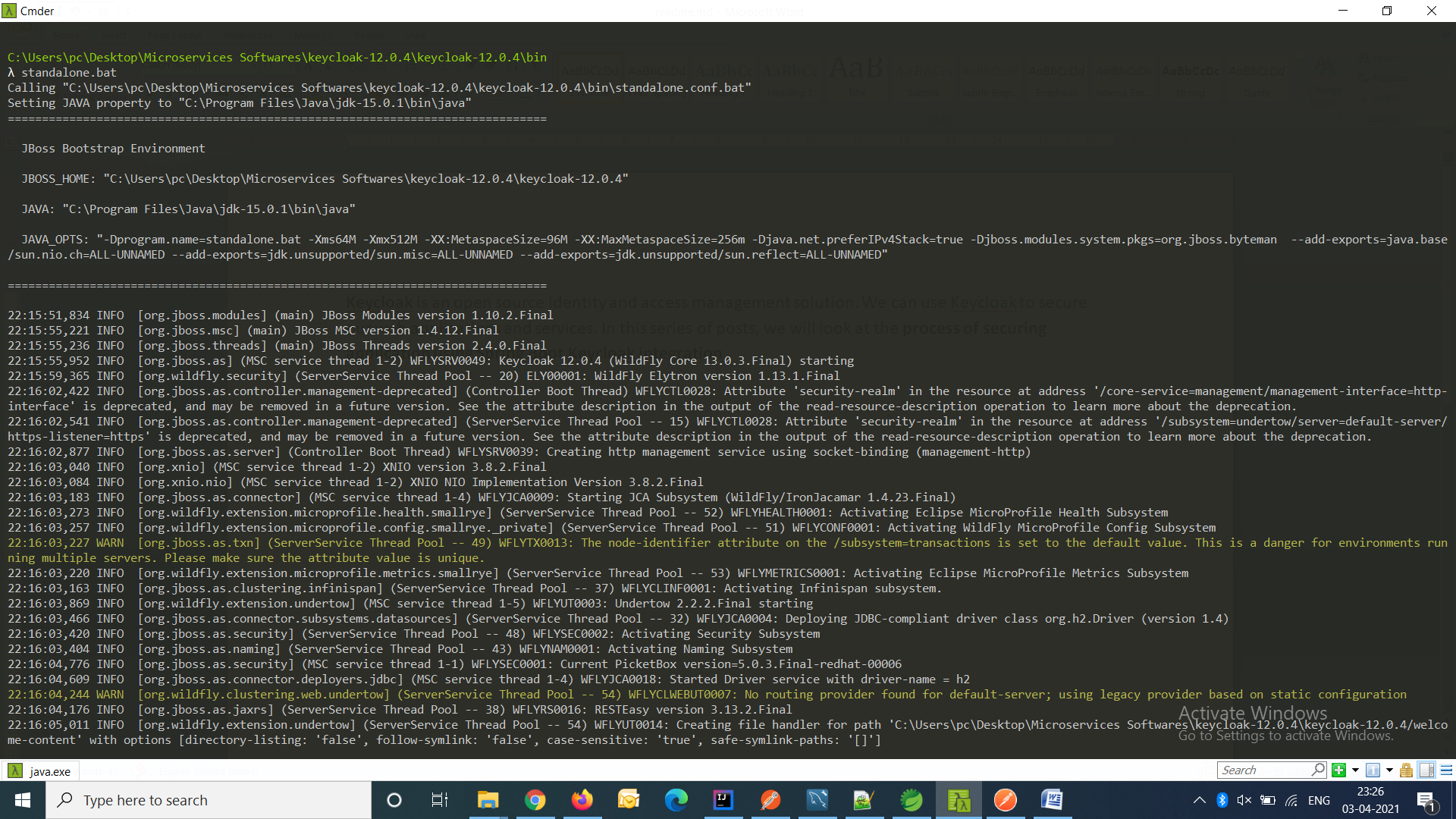
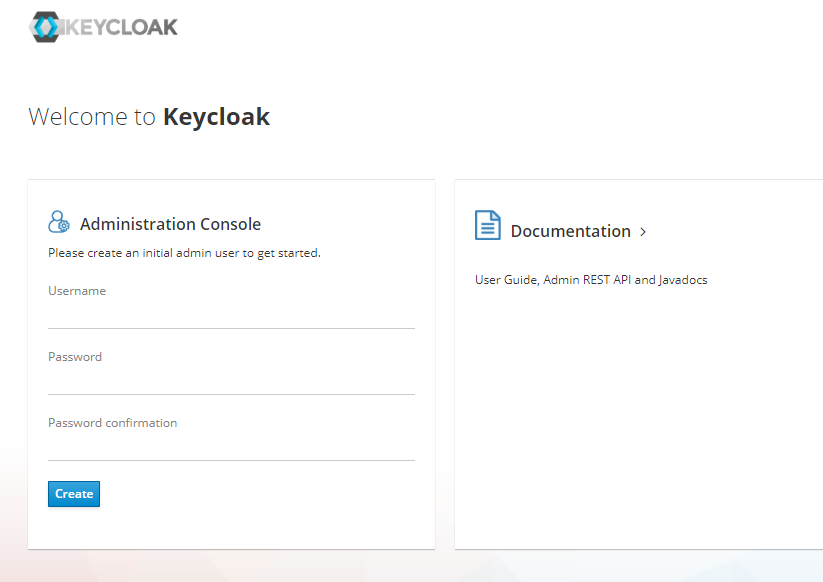
**Keycloak** is an open source identity and access management solution. We can use Keycloak to secure our web applications and services. In this series of posts, we will look at the **process of securing application using Spring Boot Keycloak integration**.

You can download keycloak from here - <https://www.keycloak.org/downloads>



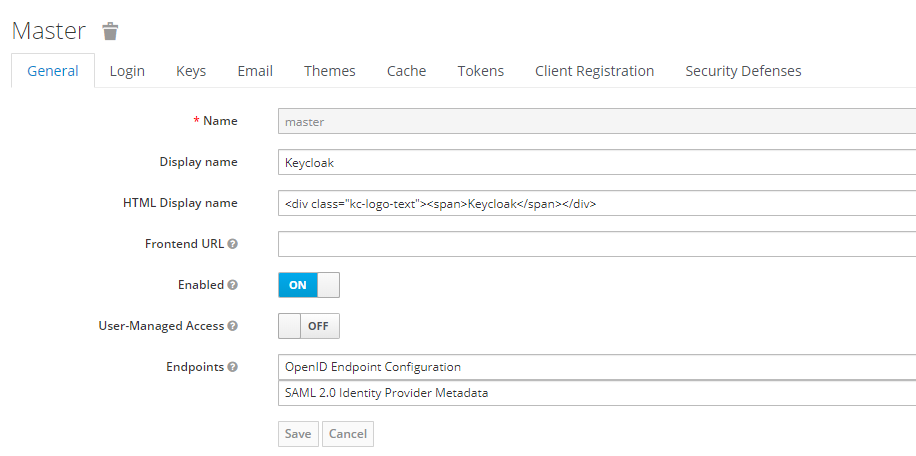
Once the server starts up, we have to setup an admin user for the same. By default Keycloak does not come with any user. To setup the admin user, we have to navigate to <http://localhost:8080/auth>. Here, we should see a sign up form where we can setup the admin user.

Initially setup admin user using admin/admin



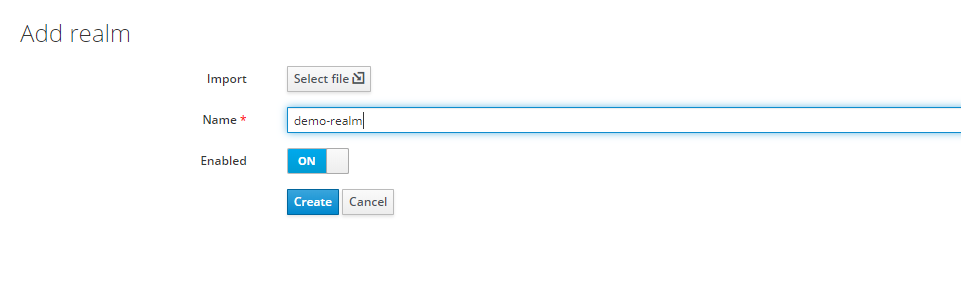
Keycloak works on a **multi-tenant model**. Basically, this means that one keycloak instance can support multiple applications or tenants. You can think of tenant as an isolated group of users and application.

By default, there is a **master**realm in Keycloak. However, that is reserved for admin purposes. The role of this realm is to manage the keycloak instance. Therefore, we should not use it for our application.



Instead, we should create a new realm. To do so, hover over the dropdown in the top left corner where it says Master and then click the button ‘**Add Realm**‘.

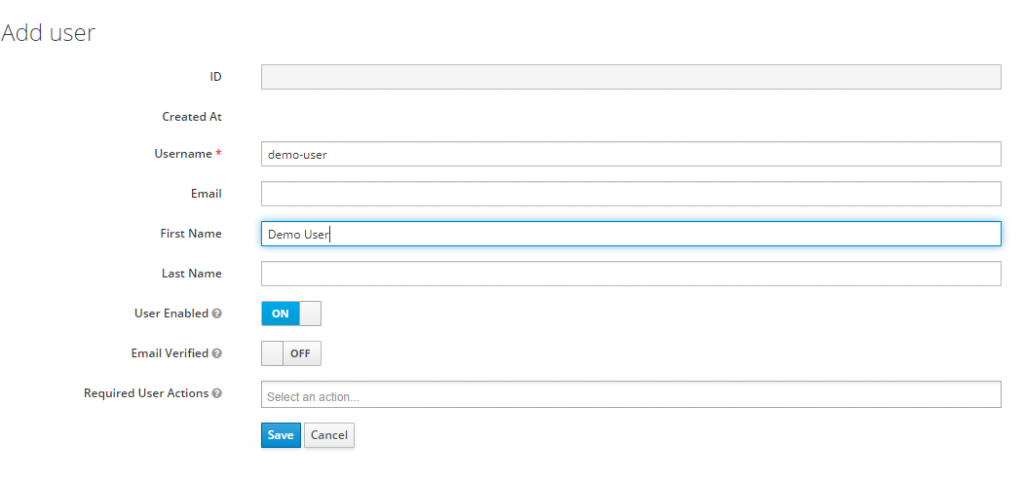
This will open up a dialog box where you can fill the name of the realm.



In this case, we have given the name as **demo-realm**. After that, we can simply click Create and the realm will be created in our keycloak instance.

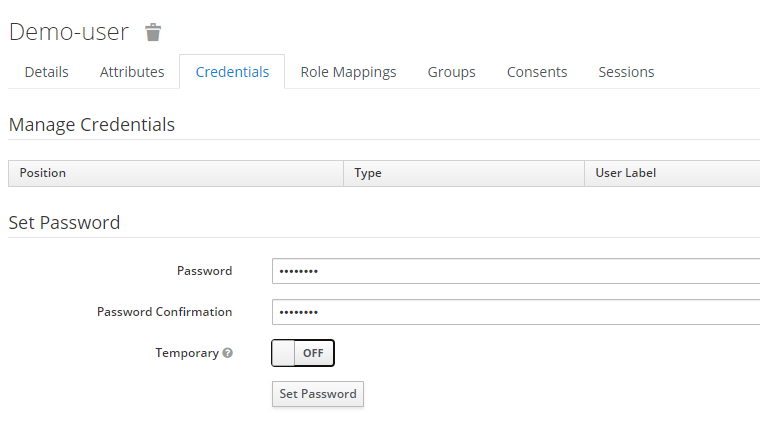
4 – Add Users to Realm

Now that the realm is ready, we need to add users to this realm. To add users, we can click on the Users menu item in the left hand menu bar. This will open another screen where click **Add User** button.



For the time being, I have entered username as demo-user and that is basically the only mandatory field over here. Upon saving, the user will be created and also an unique database ID generated for the user. Apart from it, a bunch of tabs will also appear on the screen to setup other details for the user such as Credentials, Roles, Groups and so on.

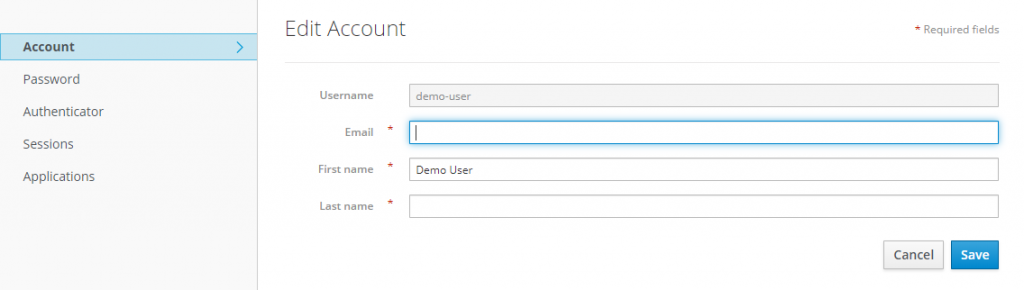
For the user to be able to login, we need to set an initial password for the user. That can be done by clicking the Credentials tab.



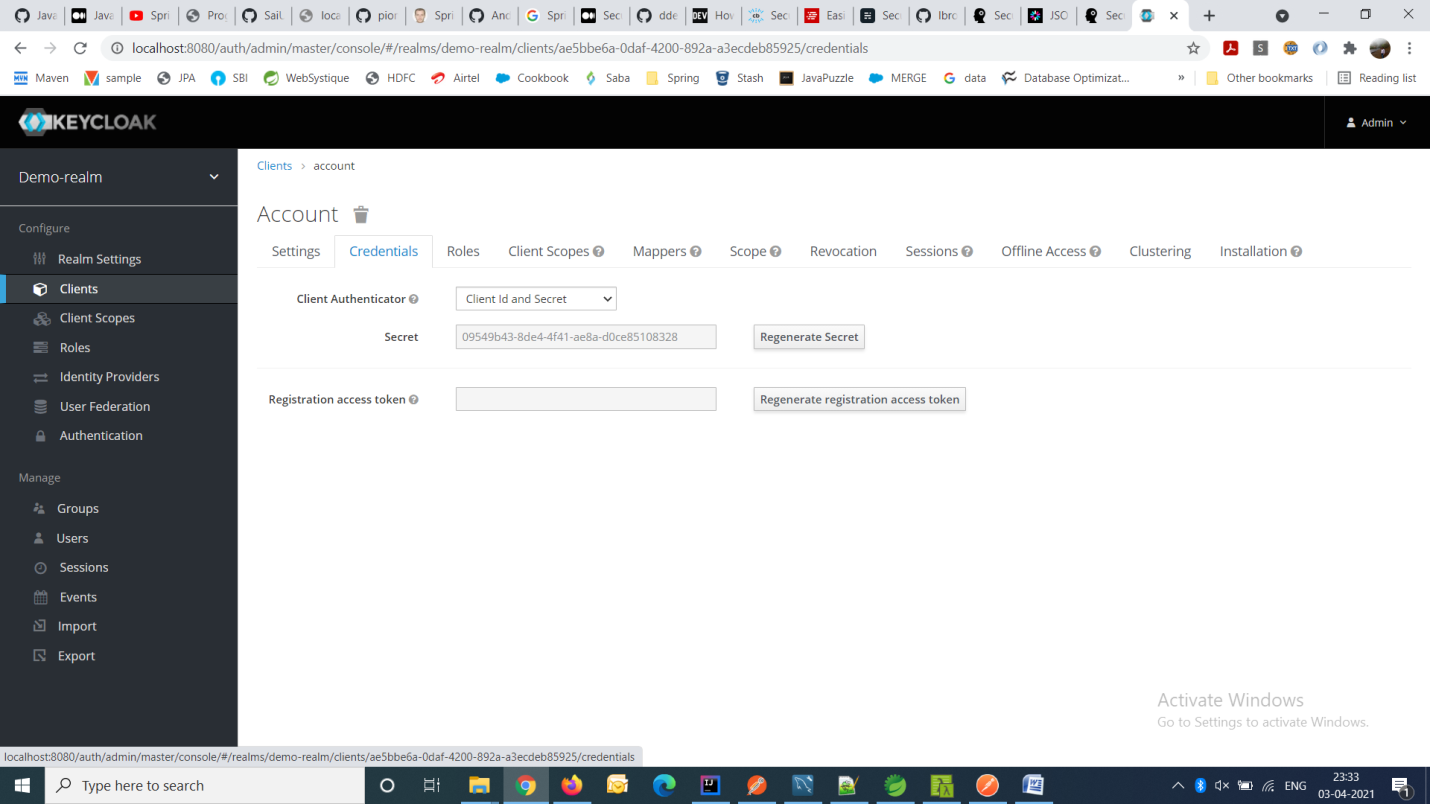
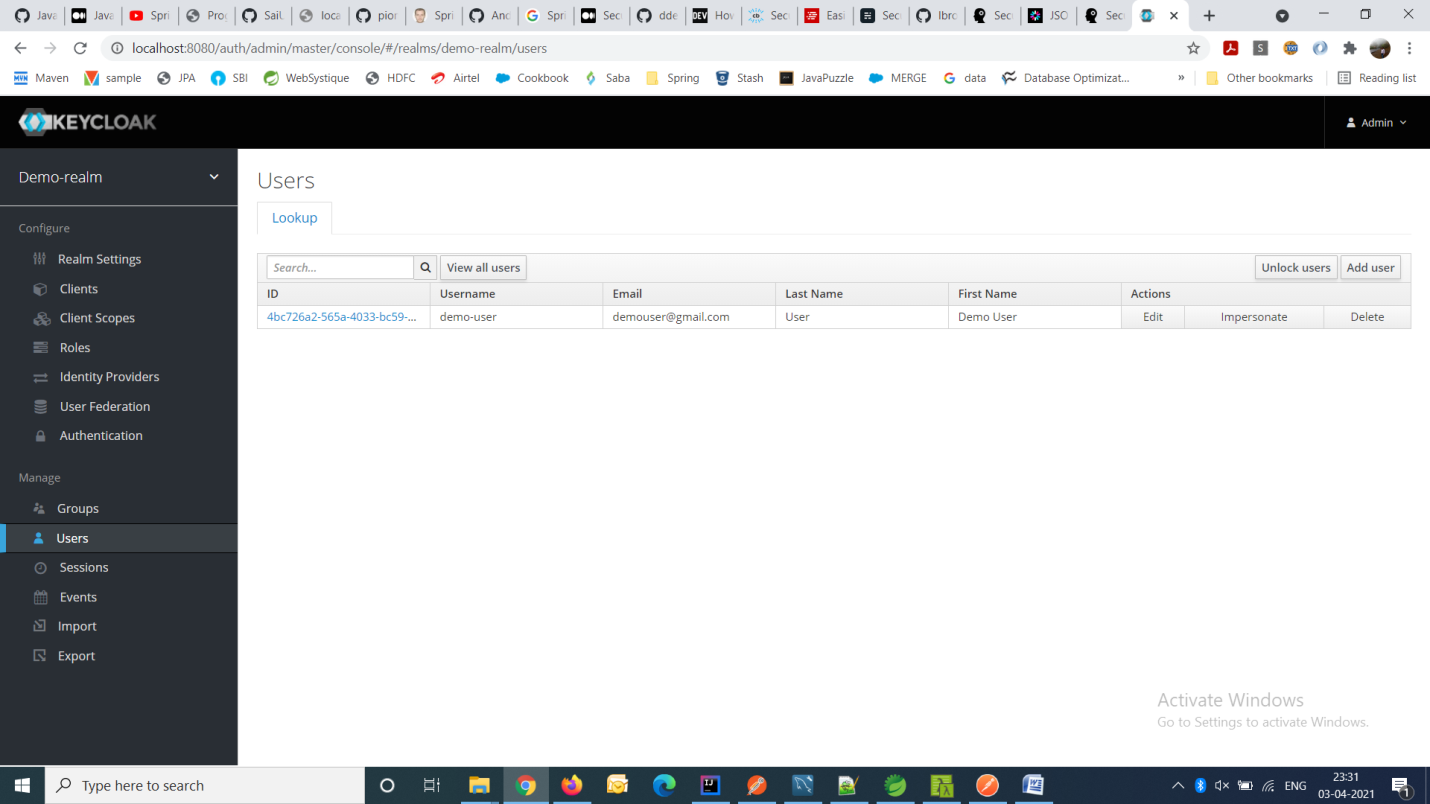
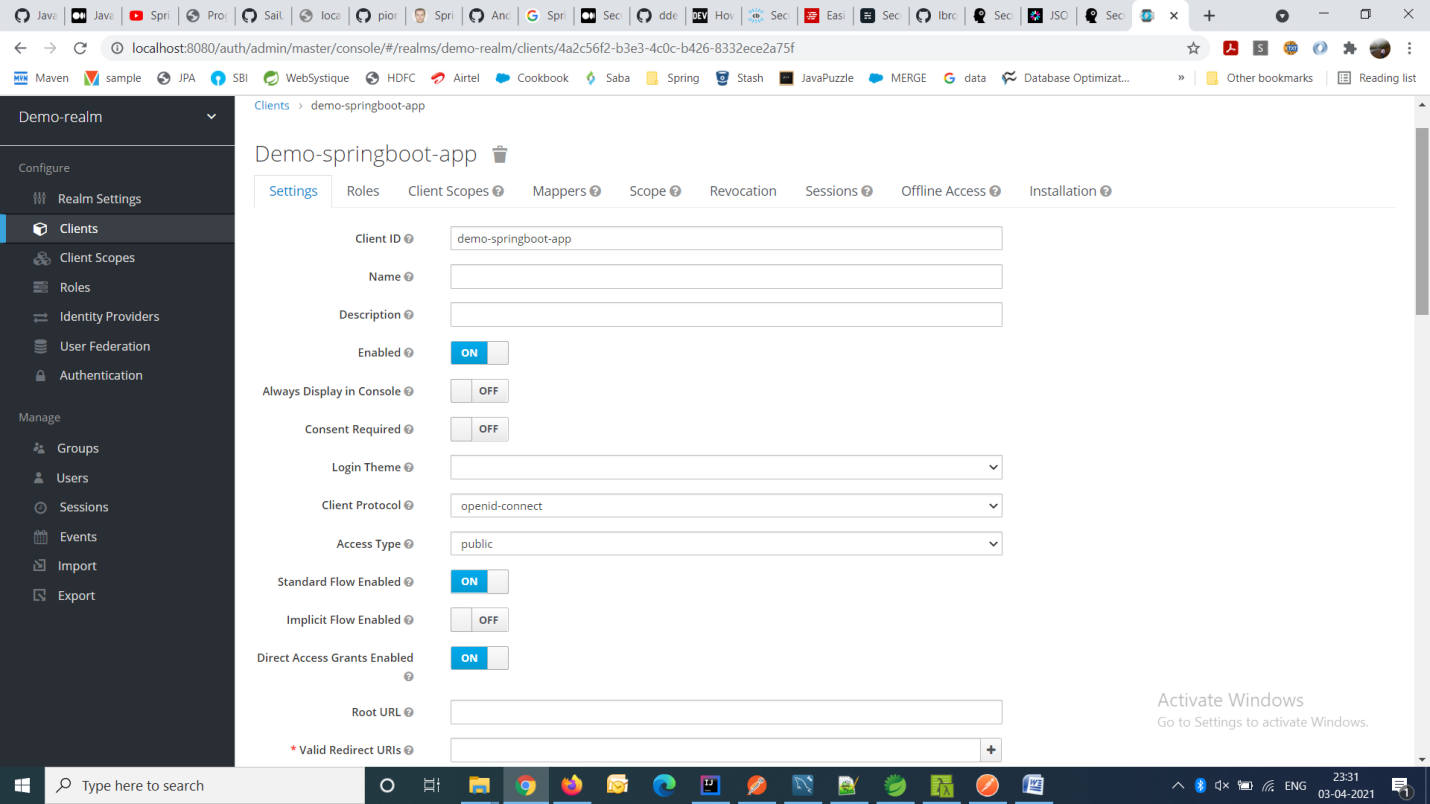
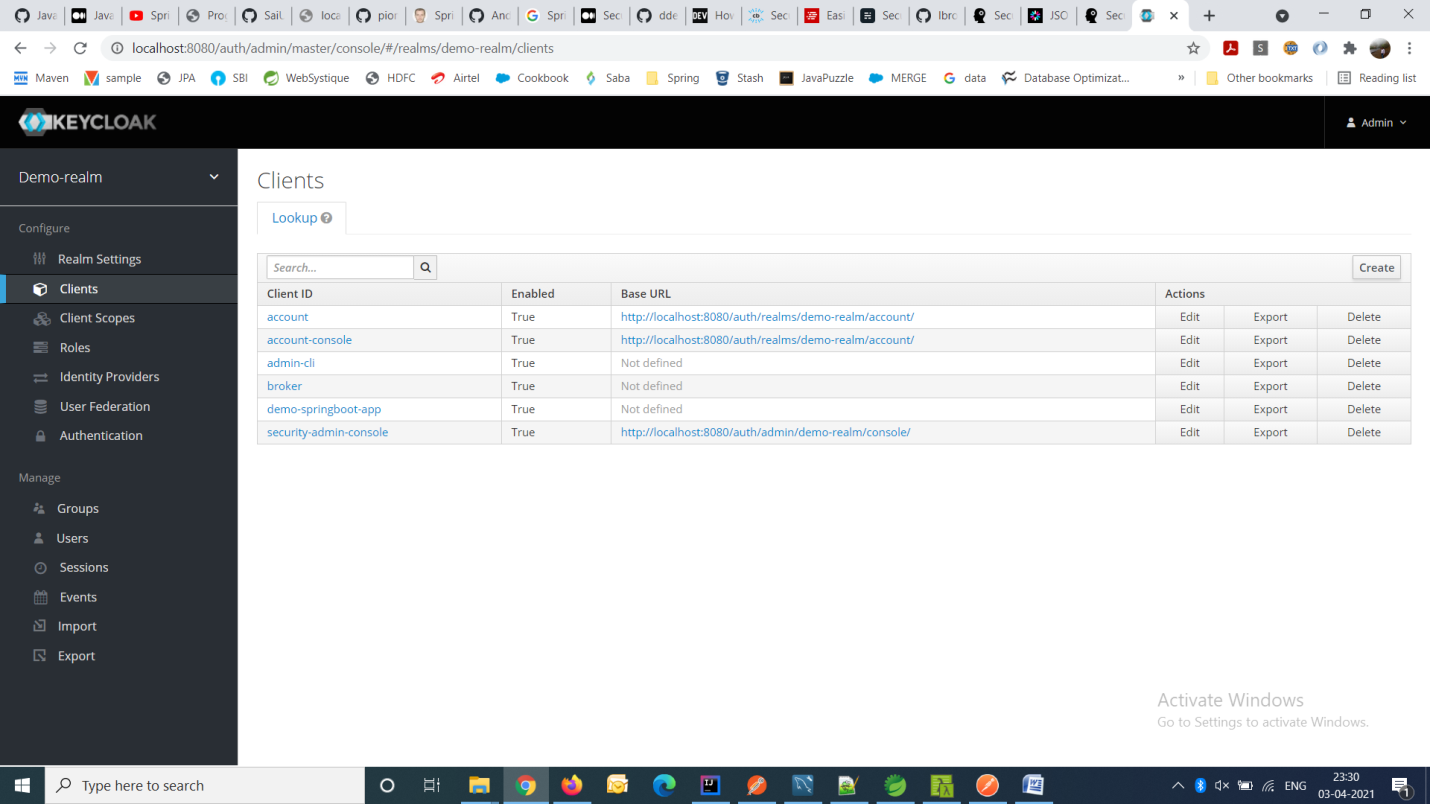
If we set the Temporary flag to ON, user will have to reset the password first time they log in.

The user was created in the **demo-realm** and hence, he/she can only login to that realm. To login to our **demo-realm**, we need to go to Account Console view for the particular realm. In our case, the path is <http://localhost:8080/auth/realms/demo-realm/account>.

Once the user logs in, they can manage their information as well.



With this, we have successfully installed Keycloak in our local machine and also setup a new realm as well as added users to the realm.

server.port=8000

keycloak.realm = demo-realm

keycloak.auth-server-url = http://localhost:8080/auth

keycloak.ssl-required = external

keycloak.resource = demo-springboot-app

keycloak.credentials.secret = 09549b43-8de4-4f41-ae8a-d0ce85108328

keycloak.use-resource-role-mappings = true

keycloak.bearer-only = true

curl --location --request POST 'http://localhost:8080/auth/realms/demo-realm/protocol/openid-connect/token' \

--header 'cache-control: no-cache' \

--header 'content-type: application/x-www-form-urlencoded' \

--header 'postman-token: ebd9baa6-f58e-47bc-72e1-52de2d1830cd' \

--header 'Cookie: KEYCLOAK\_LOCALE=en; JSESSIONID=4DDD6553BC67A20DDA0943CD8872DABD.desktop-nq639du; JSESSIONID=4DDD6553BC67A20DDA0943CD8872DABD' \

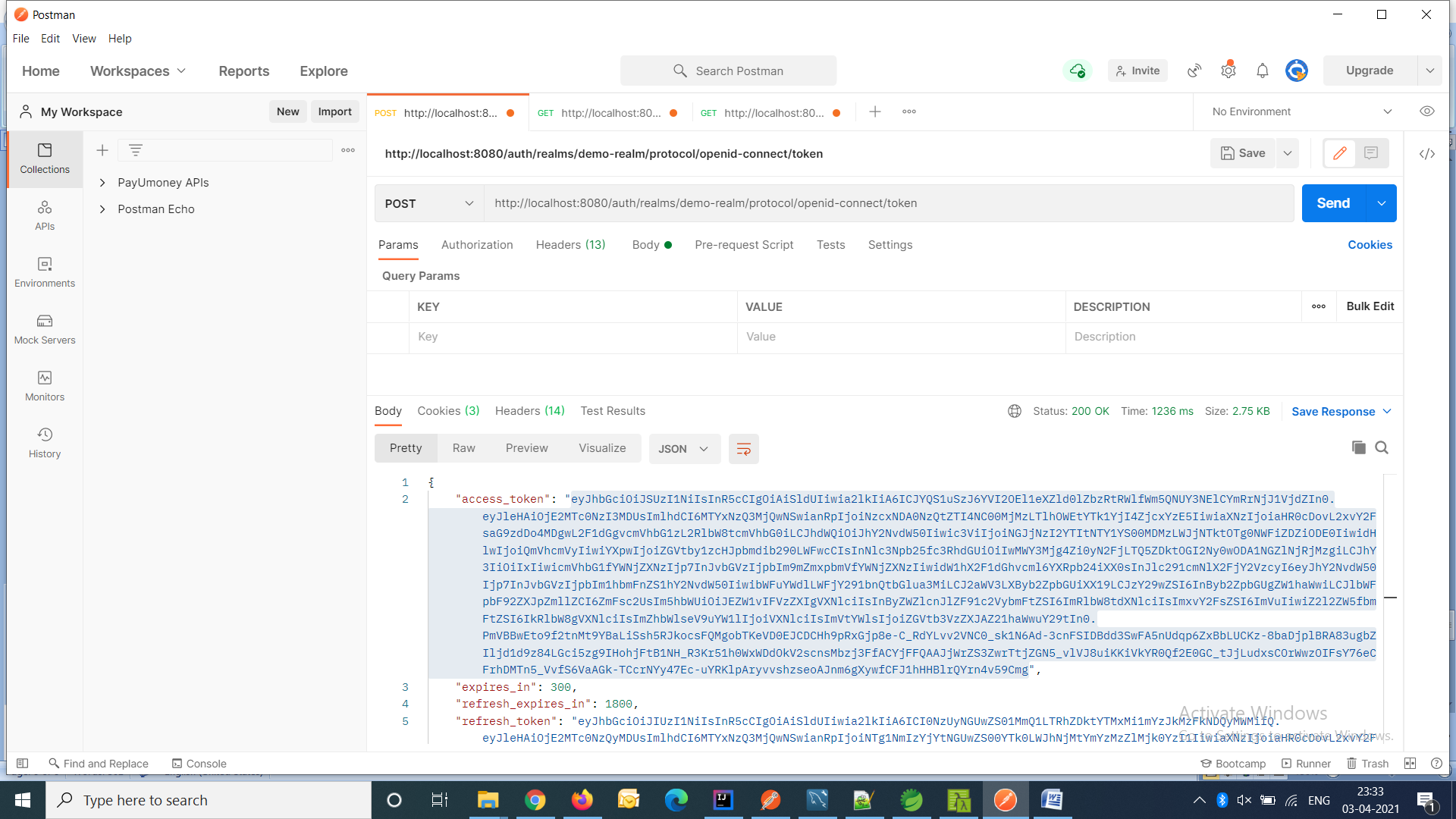
--data-urlencode 'grant\_type=password' \

--data-urlencode 'client\_id=demo-springboot-app' \

--data-urlencode 'client\_secret=09549b43-8de4-4f41-ae8a-d0ce85108328' \

--data-urlencode 'username=demo-user' \

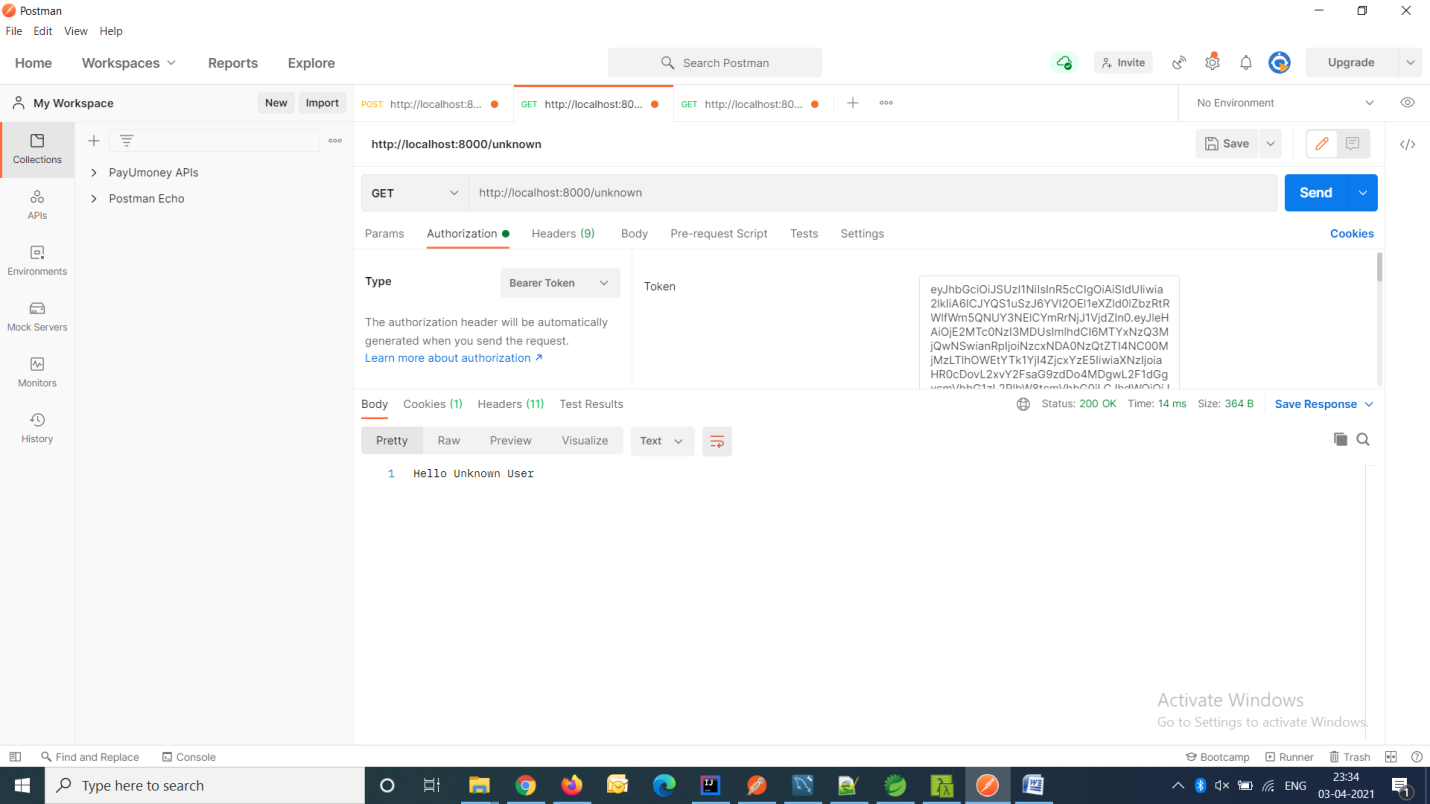
--data-urlencode 'password=test'



curl --location --request GET 'http://localhost:8000/unknown' \

--header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJYQS1uSzJ6YVI2OEl1eXZld0lZbzRtRWlfWm5QNUY3NElCYmRrNjJ1VjdZIn0..PmVBBwEto9f2tnMt9YBaLiSsh5RJkocsFQMgobTKeVD0EJCDCHh9pRxGjp8e-C\_RdYLvv2VNC0\_sk1N6Ad-3cnFSIDBdd3SwFA5nUdqp6ZxBbLUCKz-8baDjplBRA83ugbZIljd1d9z84LGci5zg9IHohjFtB1NH\_R3Kr51h0WxWDdOkV2scnsMbzj3FfACYjFFQAAJjWrZS3ZwrTtjZGN5\_vlVJ8uiKKiVkYR0Qf2E0GC\_tJjLudxsCOrWwzOIFsY76eCFrhDMTn5\_VvfS6VaAGk-TCcrNYy47Ec-uYRKlpAryvvshzseoAJnm6gXywfCFJ1hHHBlrQYrn4v59Cmg' \

--header 'Cookie: JSESSIONID=4DDD6553BC67A20DDA0943CD8872DABD'



curl --location --request GET 'http://localhost:8000/admin-user' \

--header 'Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUIiwia2lkIiA6ICJYQS1uSzJ6YVI2OEl1eXZld0lZbzRtRWlfWm5QNUY3NElCYmRrNjJ1VjdZIn0..PmVBBwEto9f2tnMt9YBaLiSsh5RJkocsFQMgobTKeVD0EJCDCHh9pRxGjp8e-C\_RdYLvv2VNC0\_sk1N6Ad-3cnFSIDBdd3SwFA5nUdqp6ZxBbLUCKz-8baDjplBRA83ugbZIljd1d9z84LGci5zg9IHohjFtB1NH\_R3Kr51h0WxWDdOkV2scnsMbzj3FfACYjFFQAAJjWrZS3ZwrTtjZGN5\_vlVJ8uiKKiVkYR0Qf2E0GC\_tJjLudxsCOrWwzOIFsY76eCFrhDMTn5\_VvfS6VaAGk-TCcrNYy47Ec-uYRKlpAryvvshzseoAJnm6gXywfCFJ1hHHBlrQYrn4v59Cmg' \

--header 'Cookie: JSESSIONID=4DDD6553BC67A20DDA0943CD8872DABD'

