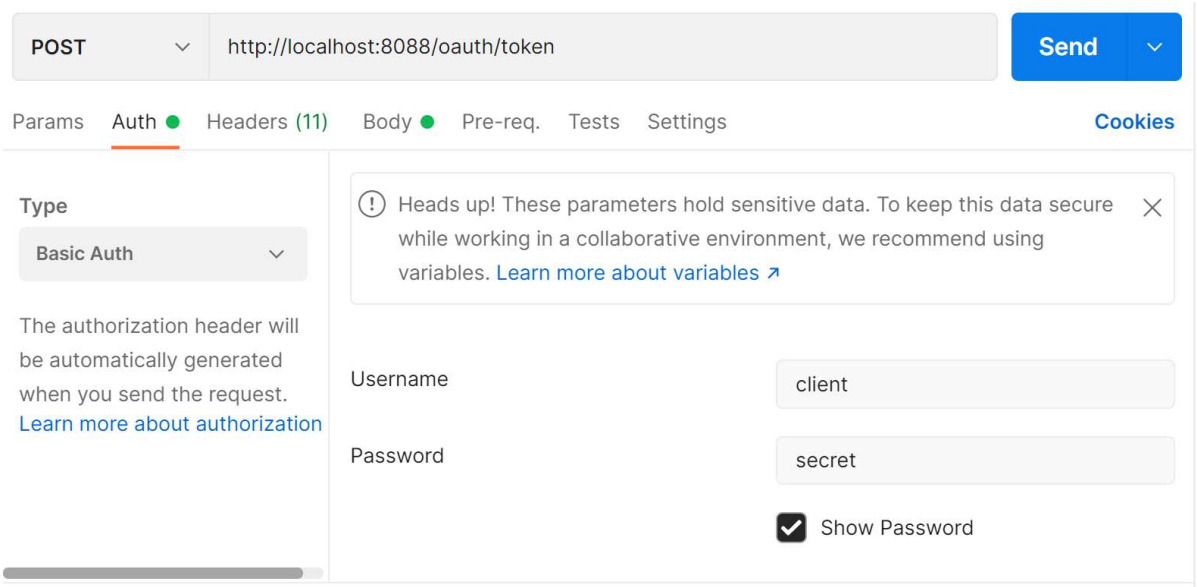
Lab 11

Part 1

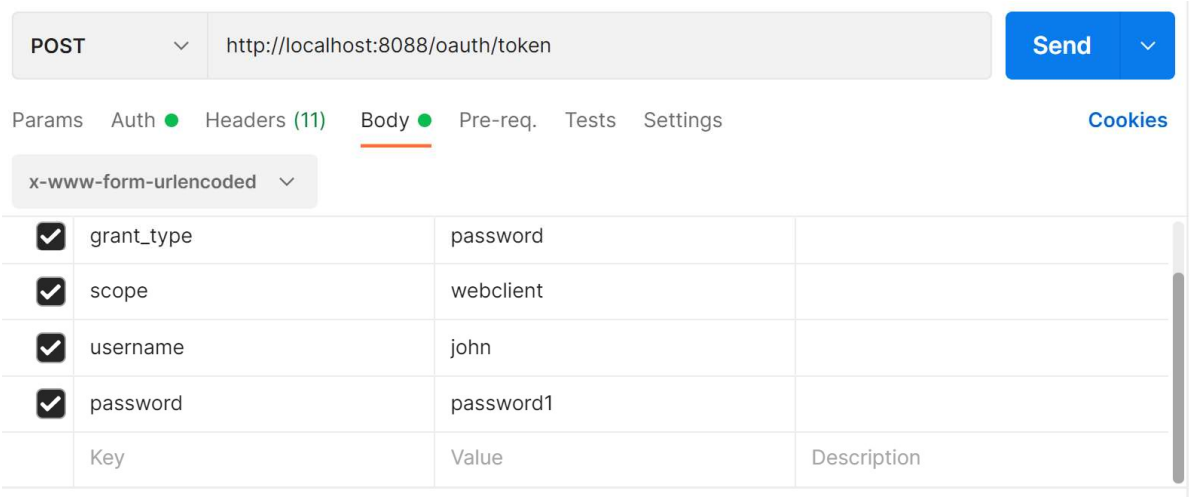
Given is the project **AuthorizationSever**. Run the file **AuthenticationServerApplication.java**.

Start postman and select the **POST** request and enter the URL **http://localhost:8088/oauth/token**

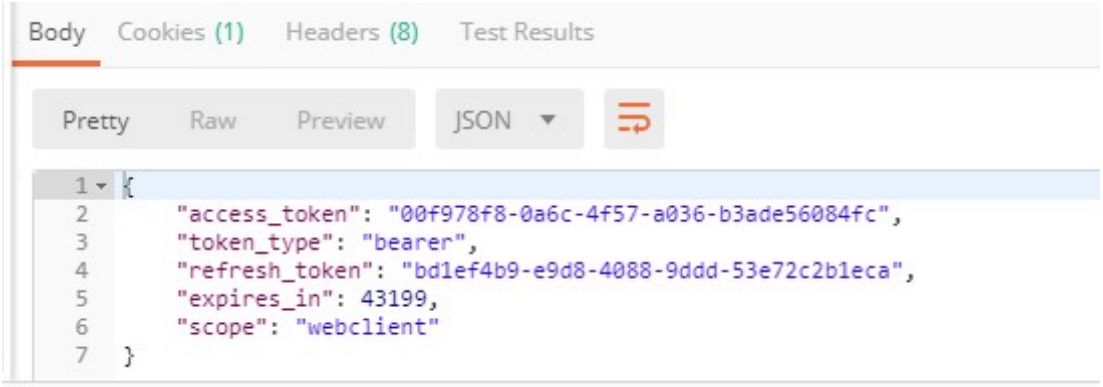
Enter the following information in the Authorization tab:



Enter the following information in the Body tab:

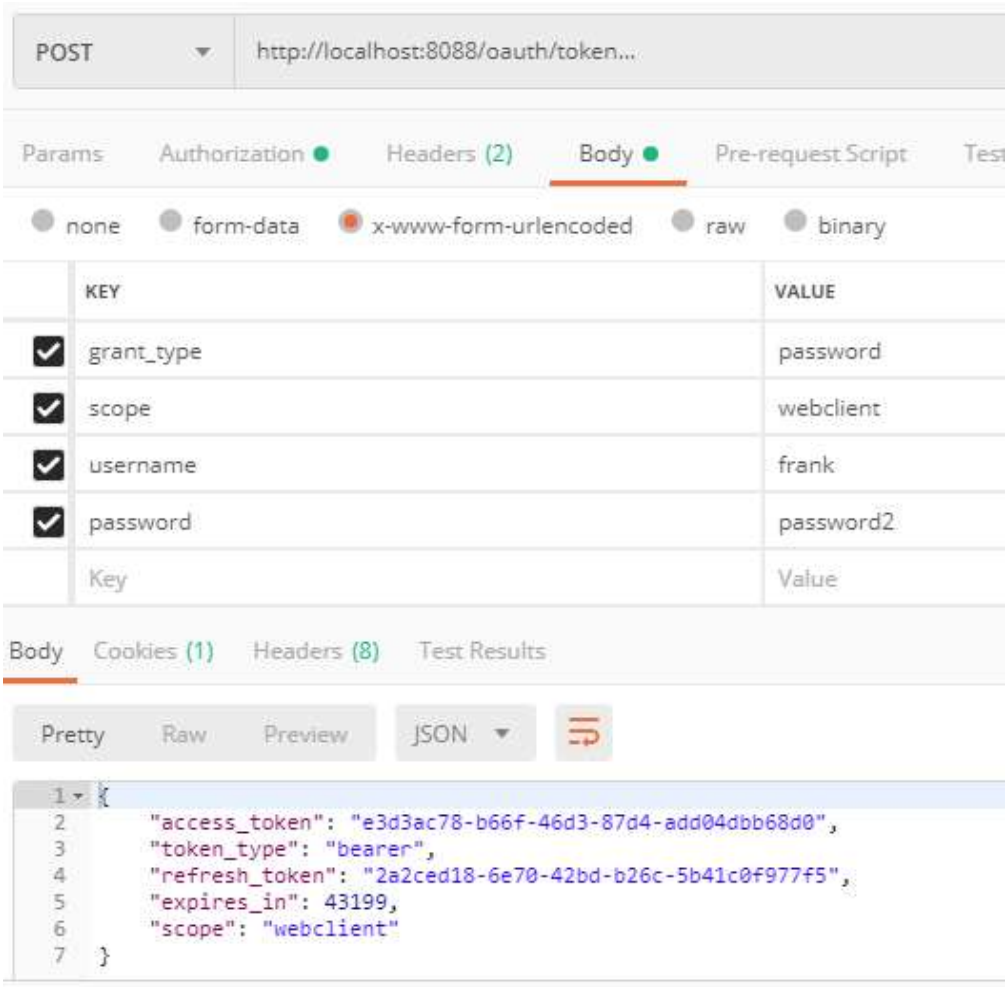


If you click the **send** button, you should get back the token for John:



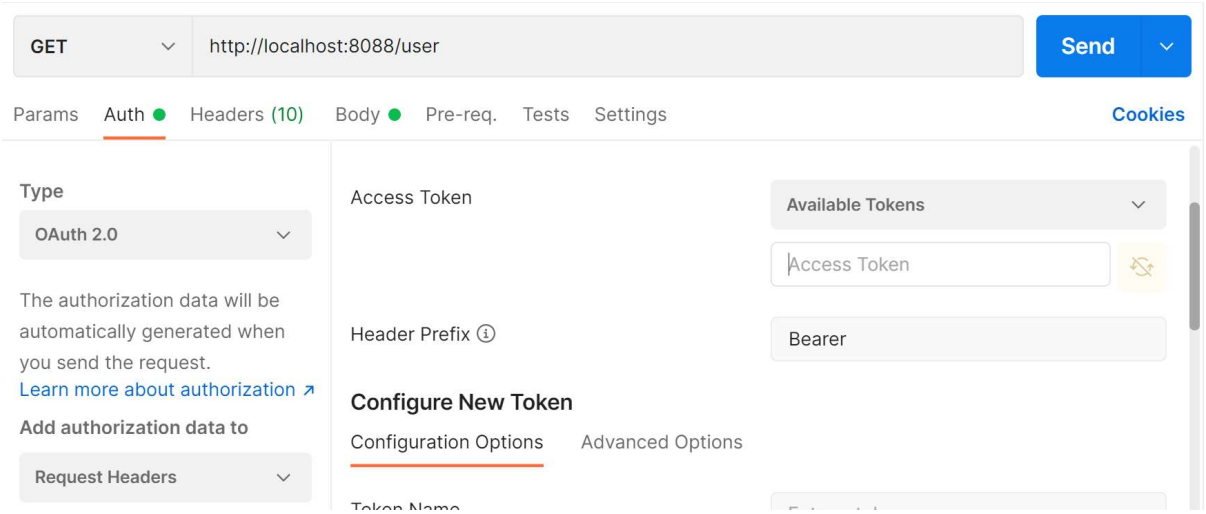
Copy the access token somewhere so we can use it later for user john

Now modify the username and password to retrieve the token for frank:



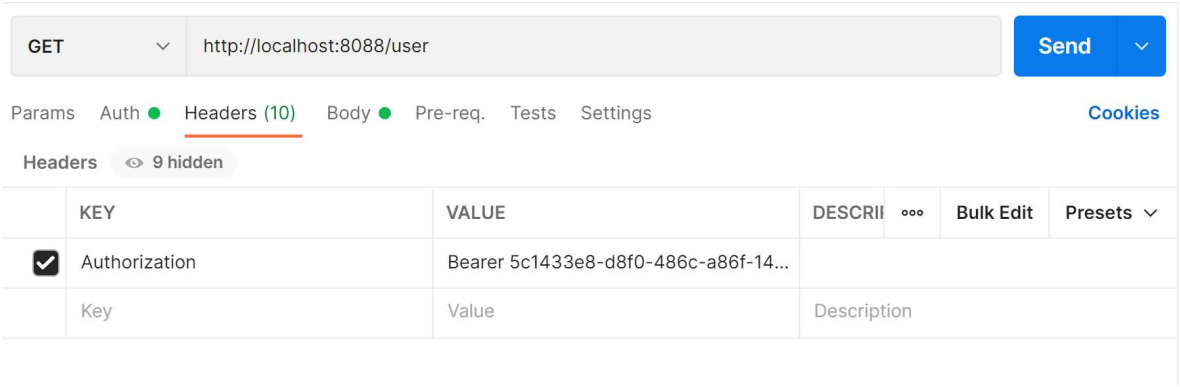
Copy the access token somewhere so we can use it later for user frank

Now change the request typoe to **GET**, and enter the url **http://localhost:8088/user**

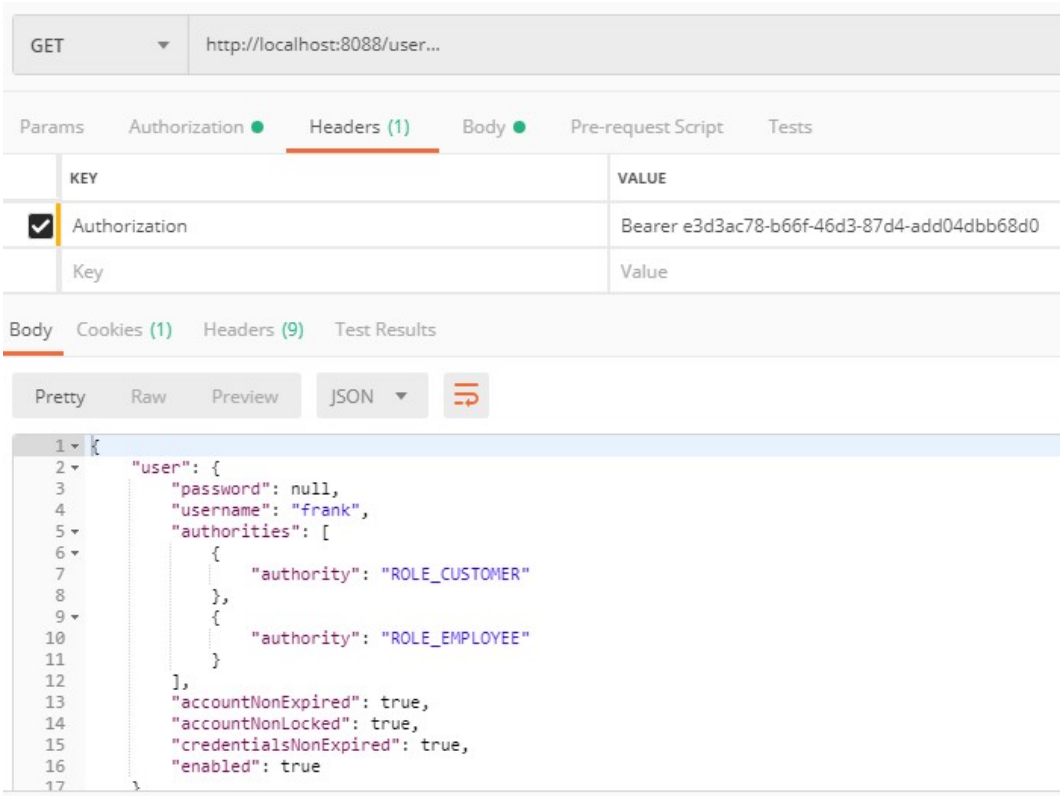


In the **Authorization** tab, select **OAuth 2.0**.

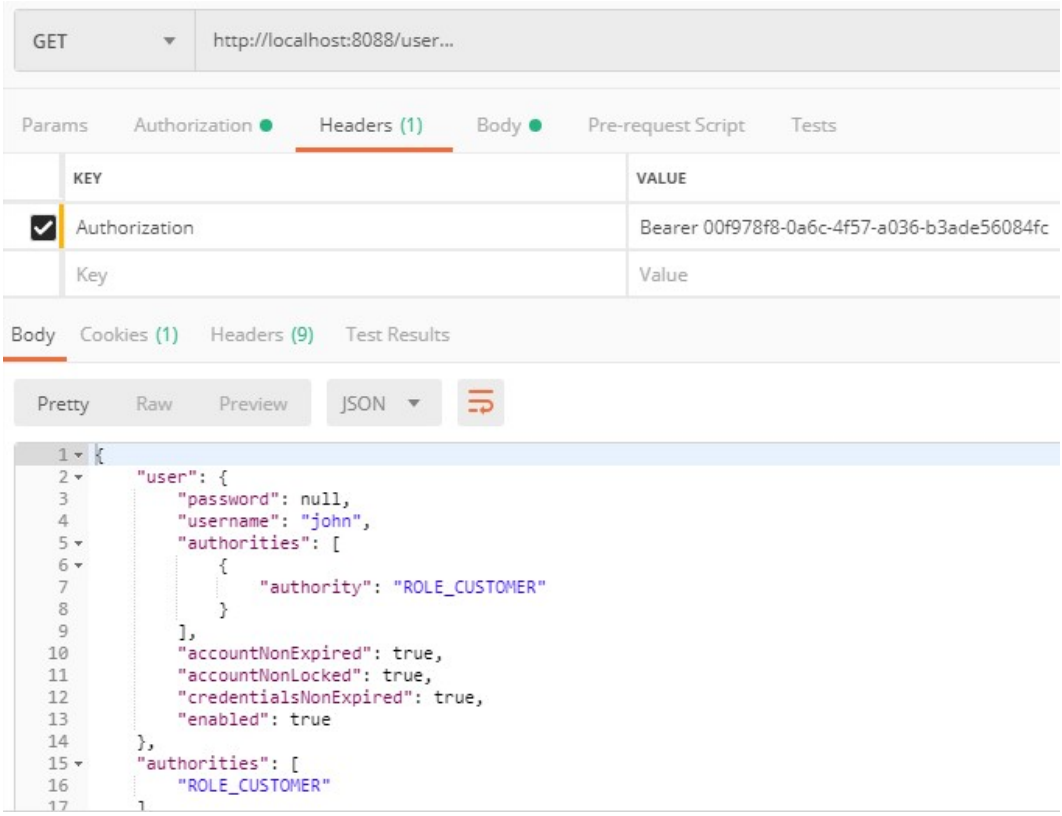
In the **Headers** tab, enter the following header with the token for frank:



When you click the **Send** button, you get the user information for frank:



Now change the token, to get the information for john:



Now change the in-memory users so that we have 3 users:
One with the role customer, one with the roles customer and employee, and one with the roles customer, employee and manager.

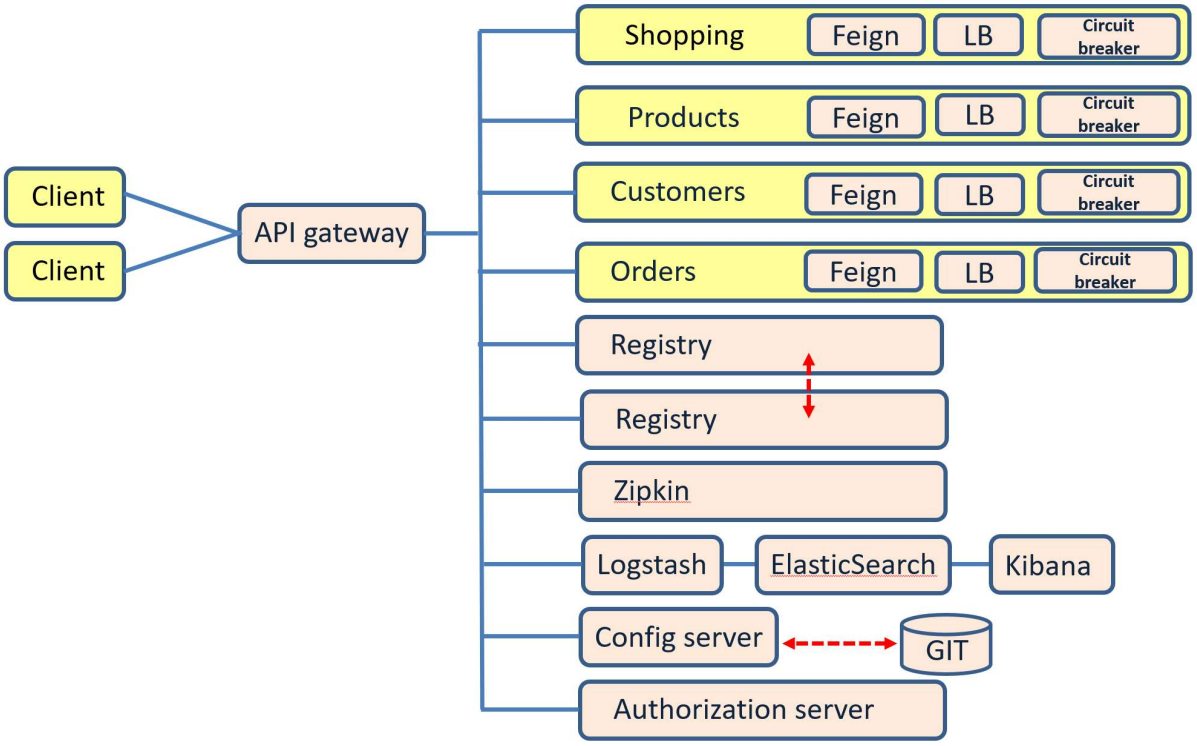
Then create 3 microservices A, B and C.
C contains salary data
B contains employee contact data (phone)
A is our actual company service that is used by customers, employees and managers

In A you can call productdata that is accessible by all customers, employees and managers
In A you can call employee contact data that is accessible only by employees and managers. A will call B to get the actual employee contact data.
In A you can call salary data that is accessible only by managers. A will call C to get the actual salary data.

Check that only customers can call product data
Check that only employees can call employee data
Check that only managers can call salary data

Part 2

Suppose we have a complete microservice architecture like this:



a. Draw a sequence diagram showing all calls between the different services for the following scenario where we start up the different microservices.

b. Draw a sequence diagram showing all calls between the different services for the following scenario:
   - The client application call the method placeOrder() on the Orders service.
   - The Orders service calls the Products service to update the available products on stock amount.

Part 3

Write 2 separate microservices: SavingAccountService and CheckingAccountService. Implement the logic that you can create accounts and that you can deposit and withdraw to and from each account.

Implement a client application that transfers money from her checking account to her saving account. Make sure that this transfer action runs within one transaction. Test if the transaction works by simulating an error in your code.

What to hand in?

1. A zip file containing all services for part 1
2. A PDF for part 2
3. A zip file containing all services for part 2
4. Write a readme.txt file with the following statement and sign with your name:
   a) Status of the lab. Describe here if you finished all parts of the lab or not. If you did not finish the lab, describe which parts are finished, and which parts not. Describe clearly why some parts are not finished.
   b) Write the following statement and sign with your name:

   I hereby declare that this submission is my own original work and to the best of my knowledge it contains no materials previously published or written by another person. I am aware that submitting solutions that are not my own work will result in an NC of the course.

   I am aware that I am not allowed to share solutions with other students.

   I am aware that if I submit only parts of this lab that points will be subtracted.

   I am aware that if my lab submission does not contain this readme.txt file that I do not get points for this lab.

   [your name as signature]