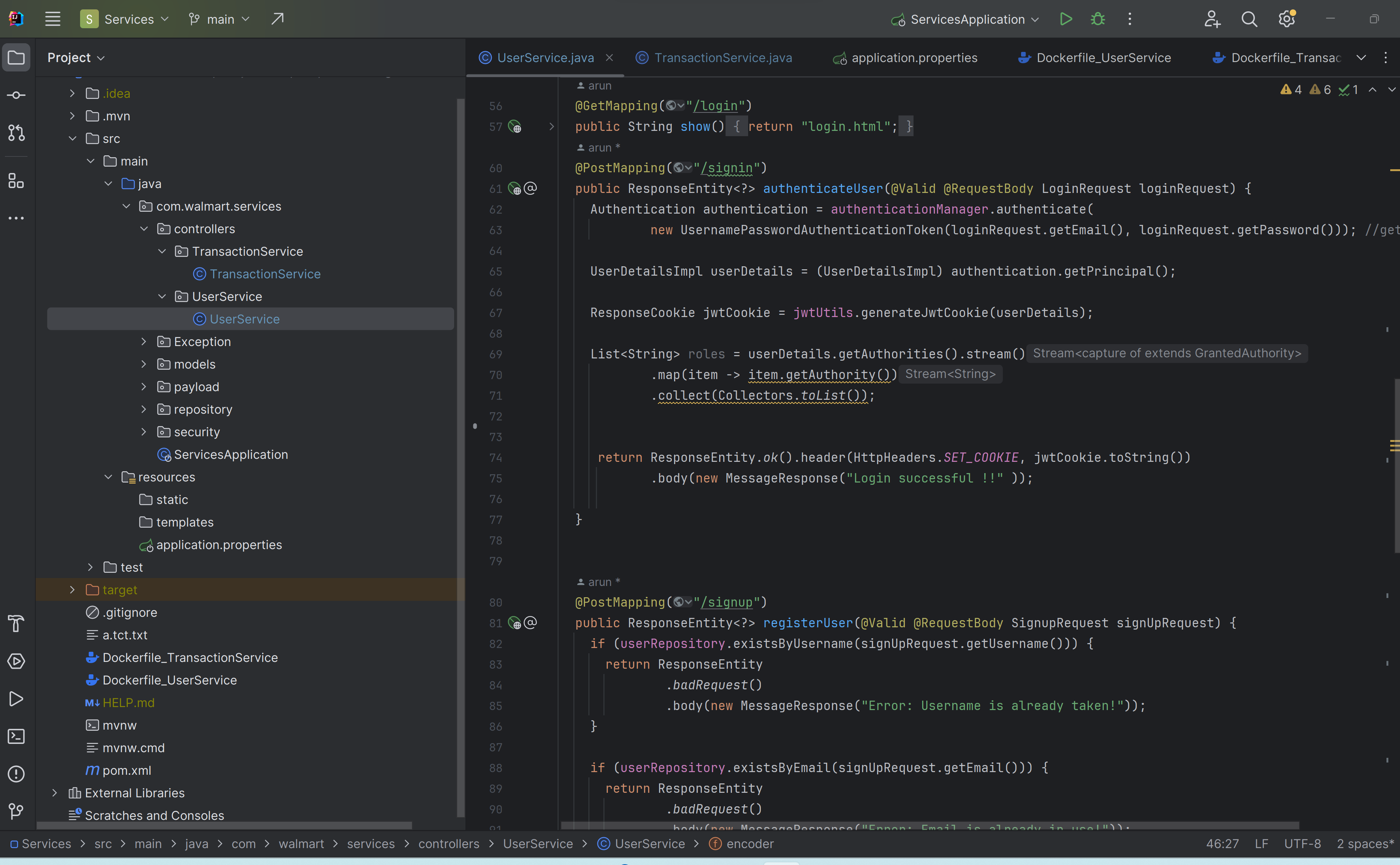
**Backend – Spring Framework**

The complete code is managed in GIT repo with postman project.

https://github.com/arunbalajii/service\_assignment.git

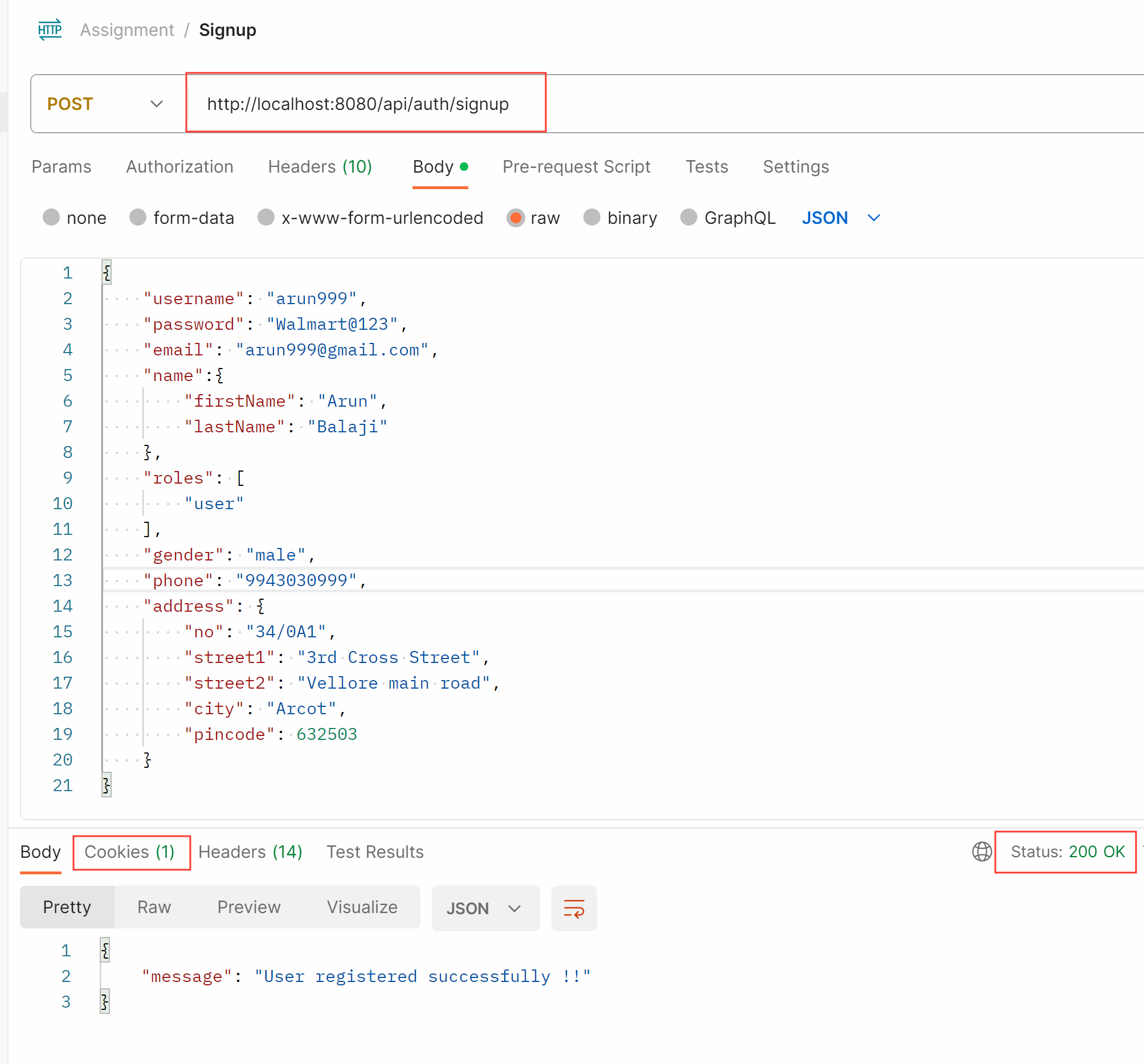
1. **User Service**

User Service controller is designed to let user to register their account and make transactions on it.

****

1. **User Registration:**

Creating new user for this process. It has 3 user checks for already existing username, email and phone num.



1. **User Registration (Existing User):**

If the user is already registered then it is prompted to register with new name likewise for email and phone number

**A screenshot of a computer

Description automatically generated**

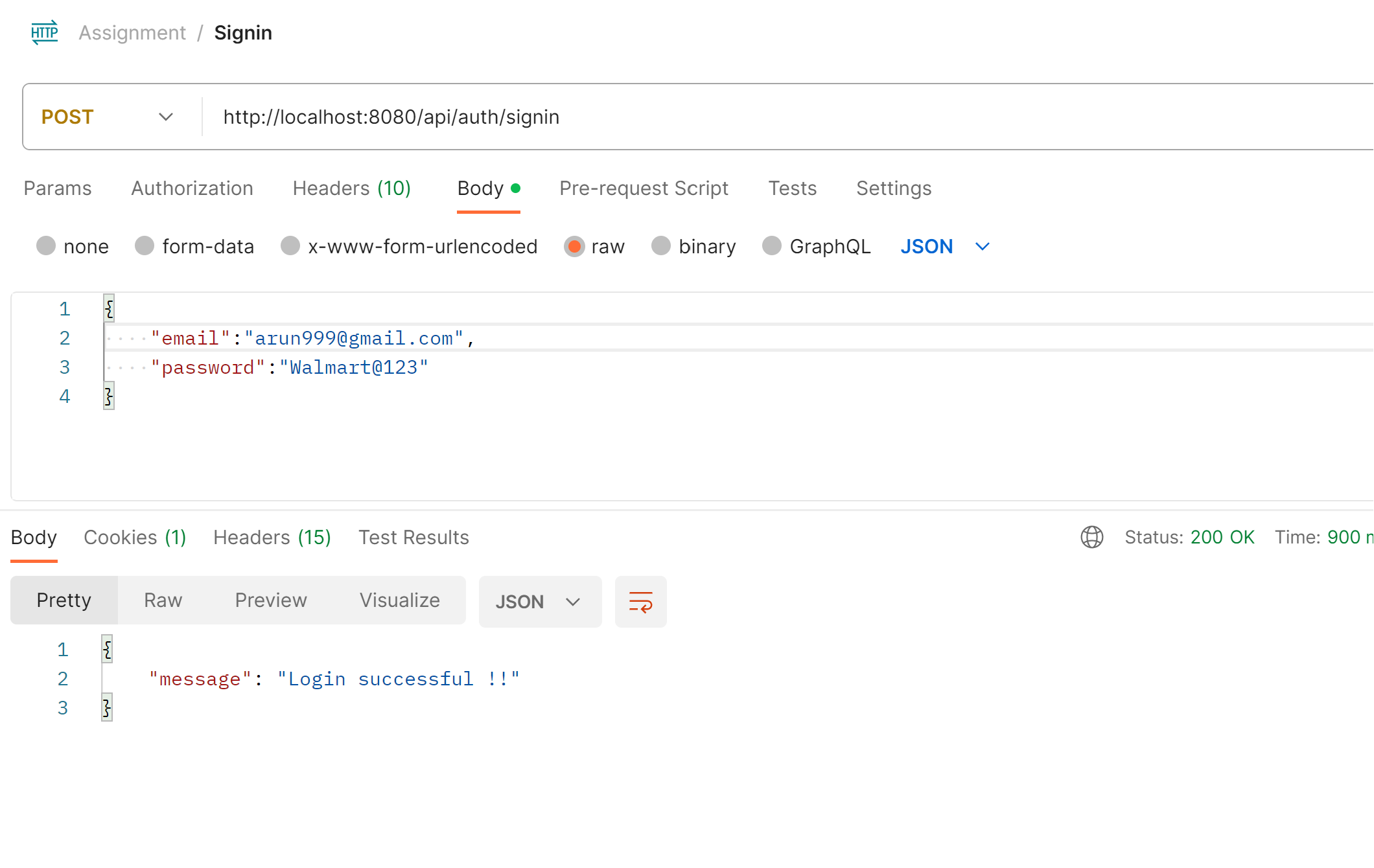
1. **New User entry in DB:**

After registration the user details are captured and pushed to mongoDB. Initially the amount for that account is maintained as 0

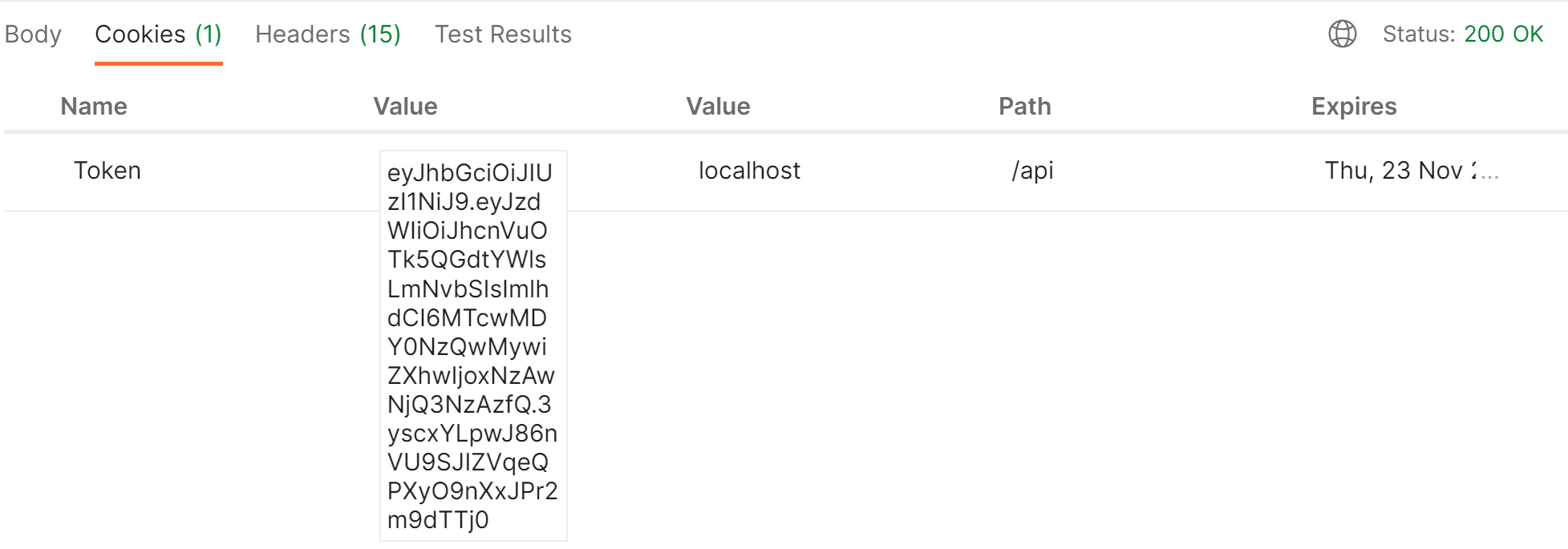


1. **User Login:**

Once user is registered they can able to login with their credentials. This will generate the JWT token saved as cookie will be helpful for further API calls.

****

1. **JWT token generated for future API call validation:**

****

1. **User Login (Bad credentials):**

User with wrong credentials are denied access

**A screenshot of a computer

Description automatically generated**

1. **Transaction Service**

Another controller service which handles the transaction part for the user once after the token is validated

**A screen shot of a computer

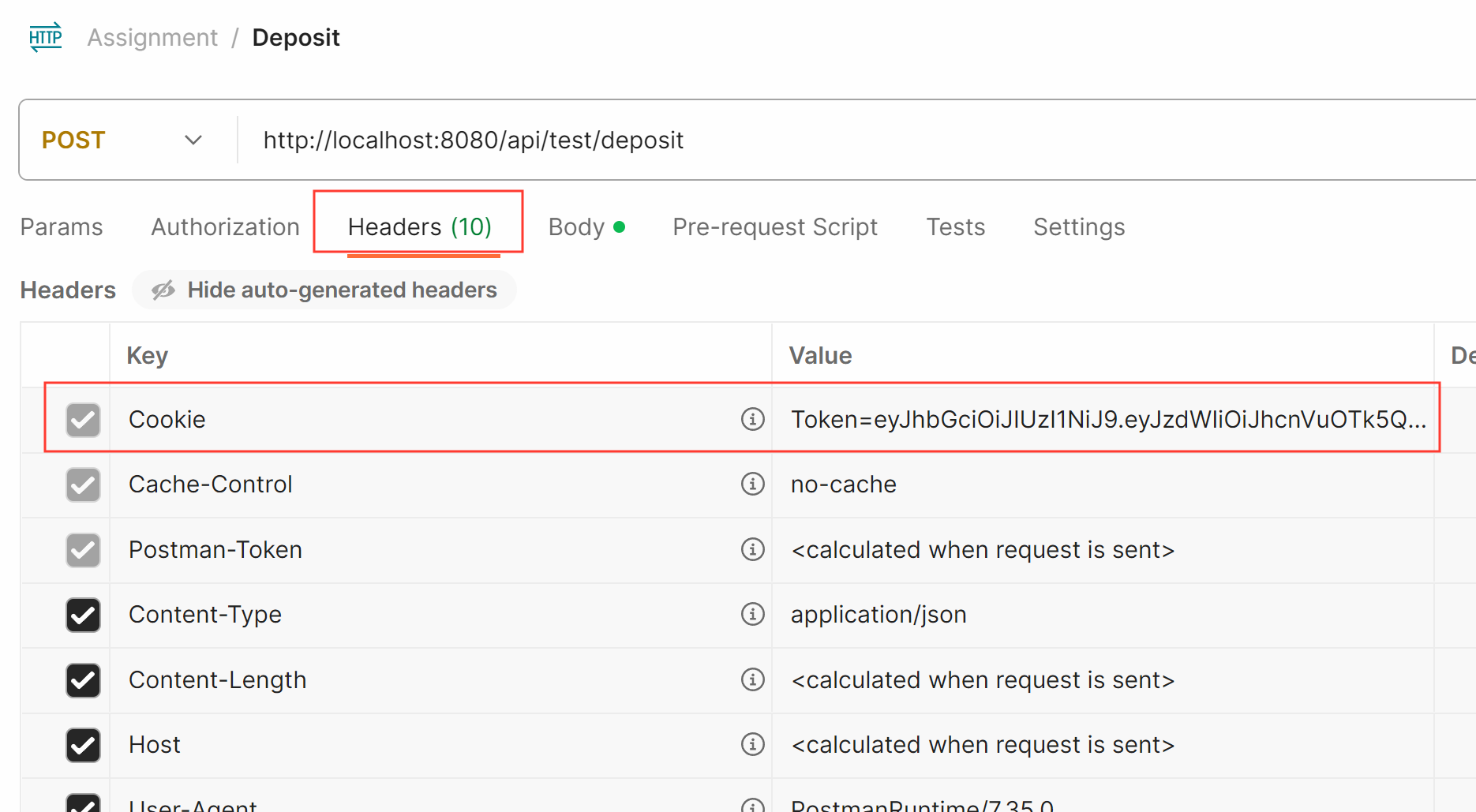
Description automatically generated**

1. **Deposit to User account:**

For the deposit user no need to enter his account details in payload as it is captured and validated from the header token and deposit made to that account

**A screenshot of a computer

Description automatically generated**



1. **Amount credited in DB:**

While the account creation the amount is 0 by default after deposit the DB is updated with his current value.



1. **If token is not present:**

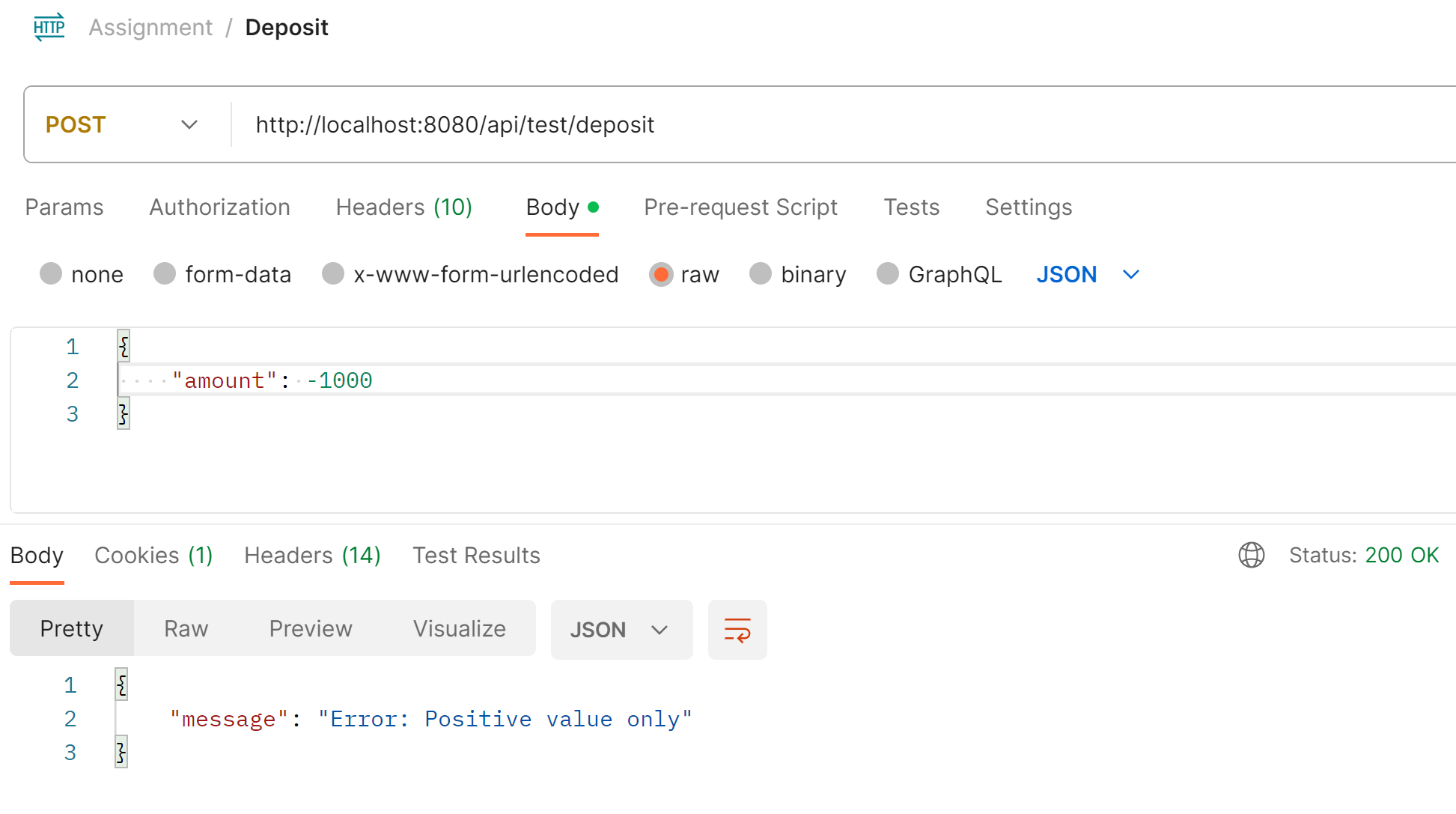
Any user tries to do deposit directly backend will not know their account details hence they are advised to login with their credentials first. So that transaction could be made.

A screenshot of a computer

Description automatically generated

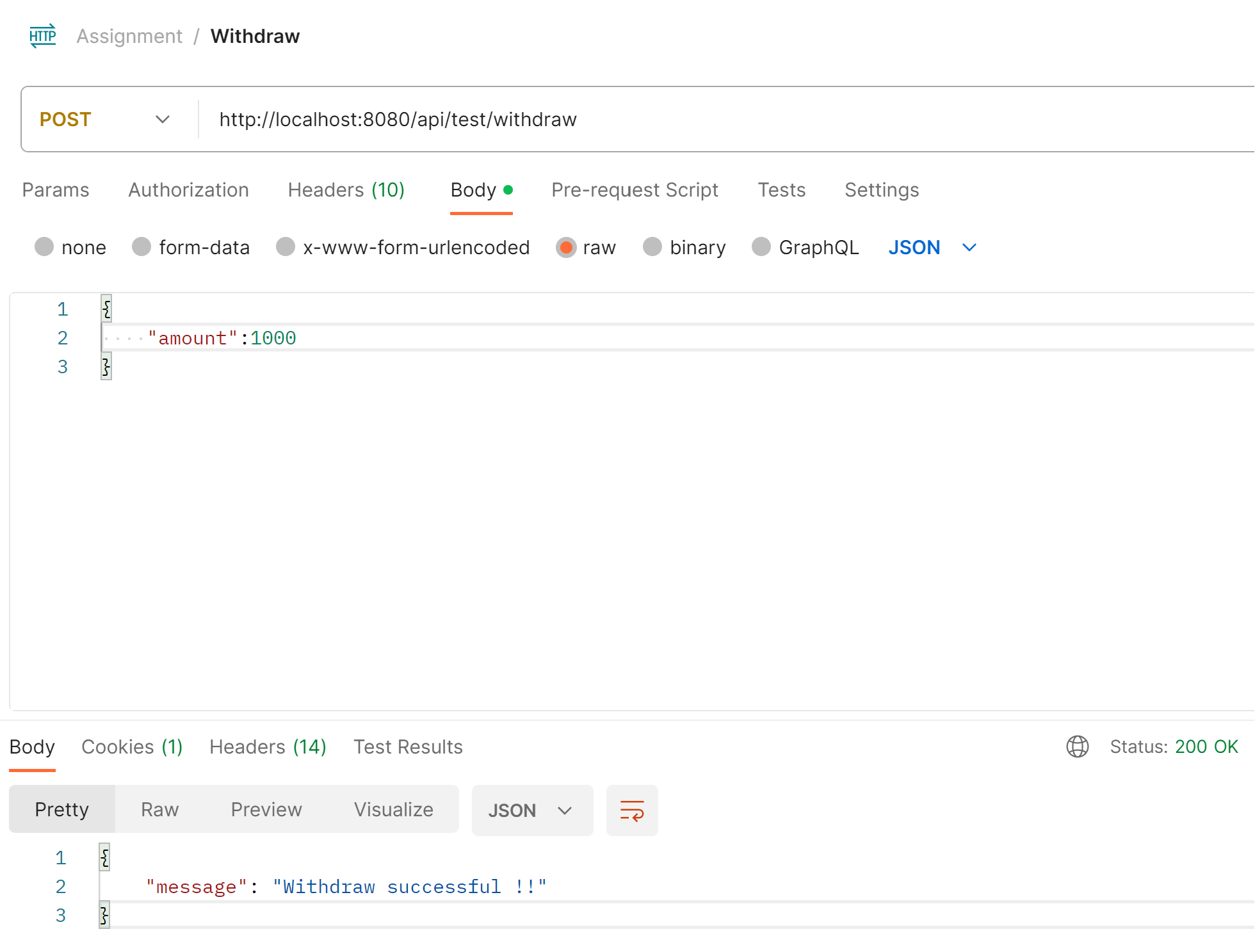
1. **Deposit (negative value):**

Another negative scenario to avoid impact on amount. Conditions are kept to check for the valid amount entered by the user in both deposit and withdraw

****

1. **Withdraw from the user account:**

Like deposit, withdraw is also designed with same set of conditions in validating token and avoiding the negative value. Once all condition met withdraw is allowed

****

1. **Withdraw (negative amount):**

**A screenshot of a computer

Description automatically generated**