**How to execute the project**

**Step 1: Creating a Database in MongoDB, Adding a MongoDB Collection and Data**

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

***use rest\_framework***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

***db.createCollection('users');***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

***db.users.insert([{\_id:1, name:"John", account:"SEK,GBP"},{\_id:2, name:"Sam", account:"EURO,AUD"},{\_id:3, Name:"Sarah", account:"USD,INR,SEK"}]);***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

***db.users.find().pretty();***

***\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\****

**Step 2:**

**Using Spring Initializer to Create a Project**

## Adding Model to Spring Boot Project:

Create a folder called Model in src/main/java/model

Model is the description of data in database (table name is used as class, fields, then generating setters and getters) data structure stored in the database to Spring Boot.

New Java class named: “Users”

## Adding Repository to Spring Boot Project

## We can create the connector between the model and MongoDB. This is done through a Repository interface.

## Create a new folder called "repositories" in src/main/java/repositories

New Java class named: “UsersRepositoty”

*The name of this repository is extremely important because it tells MongoDB the collection that it will be querying.*

## Adding the MongoDB Connection Info

## To tell spring the connection information of our MongoDB, we will need to add connection details to the application.properties file, located in the "src/main/resources" folder.

## Creating REST Controller

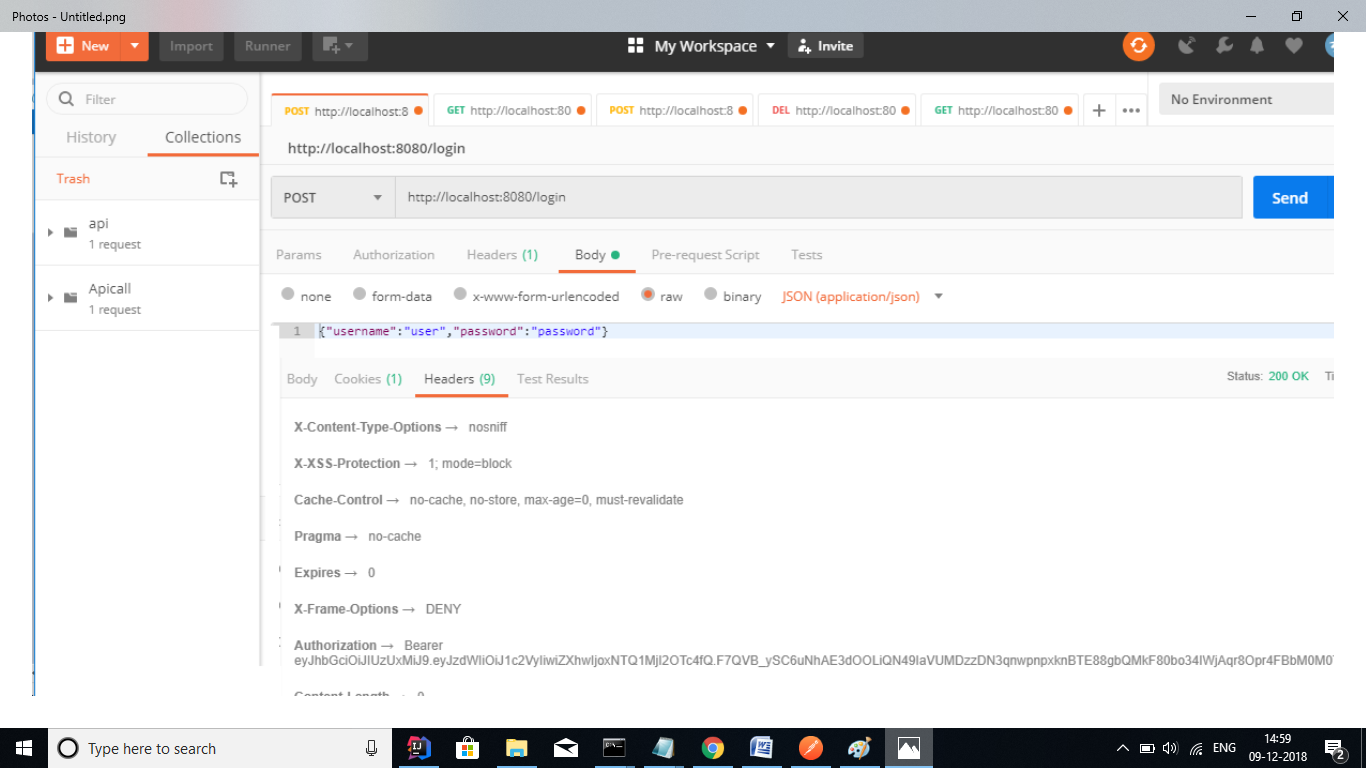
## We can establish the endpoints that we can contact to interact with the database. This will be done in a Spring Rest Controller, which uses Request Mappings to map requests with functions. We can create a Rest Controller by adding a file called UsersController.java to the src/main/java/controller folder.

## The *@RestController* annotation tells Spring that this class will requested by URL and will return data to the requester. The *@RequestMapping* annotation specifies the base URL that the controller will be handling, so any request to the host starting with "/pets" will be directed to this controller. The *@Autowired* annotation creates an instance of the UsersRepository object that will allow us to access and modify the Users database.

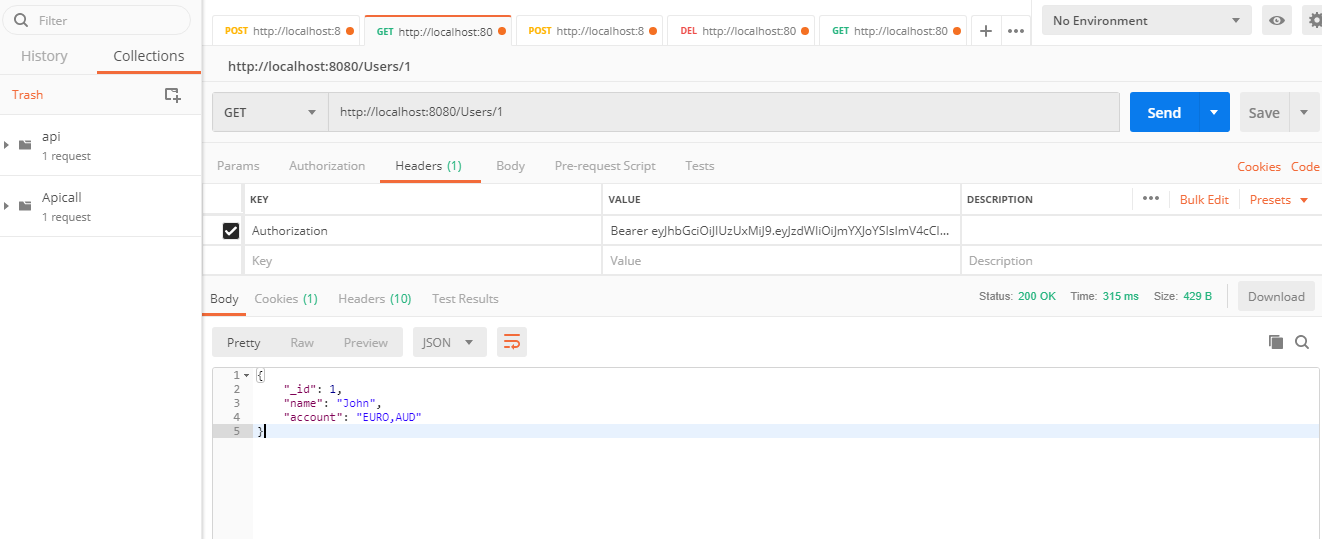
### Running the Spring Boot Application

When our server is up and running, we can test this out by querying *http://localhost:8080/users*, and we should get a message saying "Access Denied." To authenticate ourselves, we will send a POST request to */login* with our user's credentials in the body: {"username":"user", "password":"password"}

Use Postman as client to access the URL..



The restricted the URL’s can be accessed by the bearer token generated in the earlier step:



## Other URLs are listed below:

## GET- <http://localhost:8090/Users>

## GET- <http://localhost:8080/Users/1>

## POST - <http://localhost:8080/Users>

## {

## "\_id": 5,

## "name": "Sarah",

## "account": "EURO,AUD"

## }

## Delete - <http://localhost:8090/Users/5>

**Swagger UI**

