Contenido

[Running the application Item Rest API locally 2](#_Toc54521797)

[Installation Requirements: 2](#_Toc54521798)

[Running the backend API 2](#_Toc54521799)

[Configuring Postman 5](#_Toc54521800)

[Testing the backend application with Postman 7](#_Toc54521801)

[Getting the token for the adminUser 7](#_Toc54521802)

[Getting the token for the customerUser 8](#_Toc54521803)

[Example of adding token to the saveItem request: 10](#_Toc54521804)

# Running the application Item Rest API locally

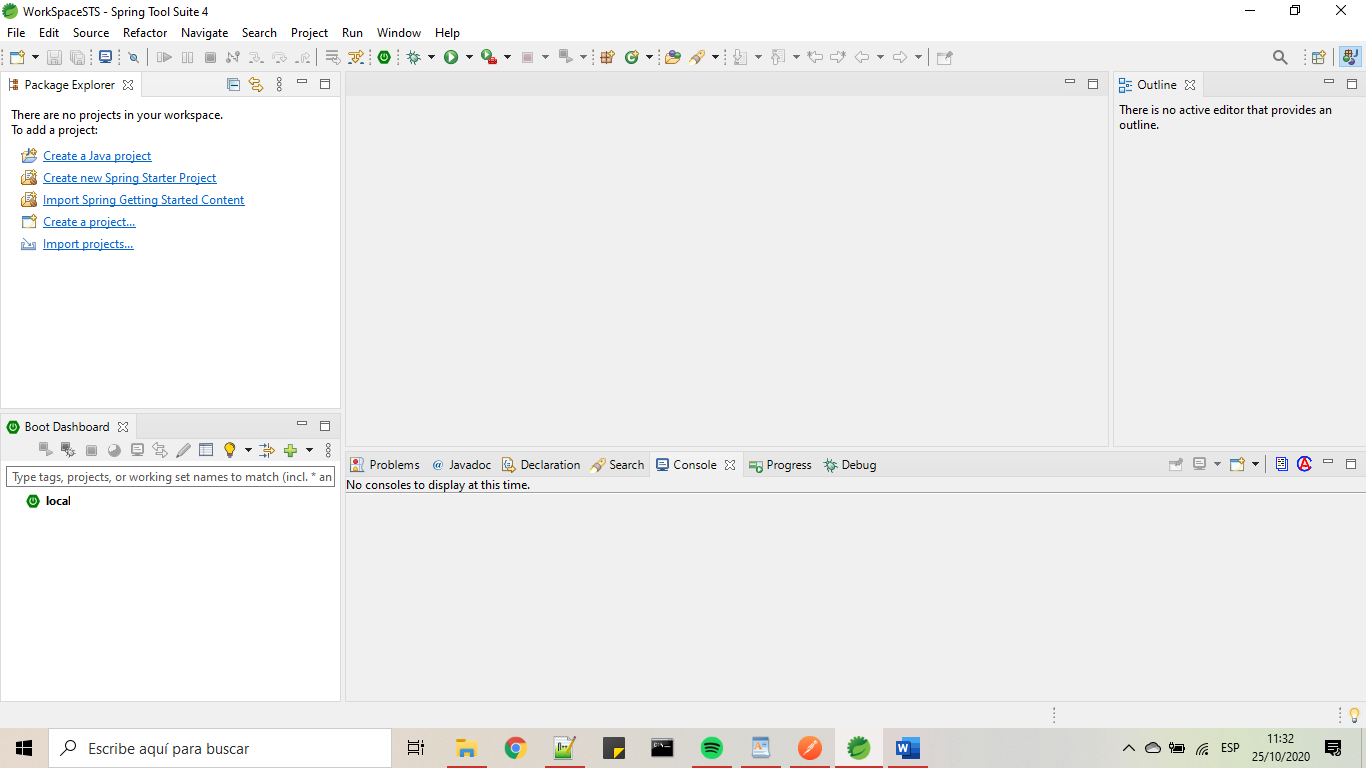
## Installation Requirements:

* JDK 1.8
* Spring Tool Suite
* Postman

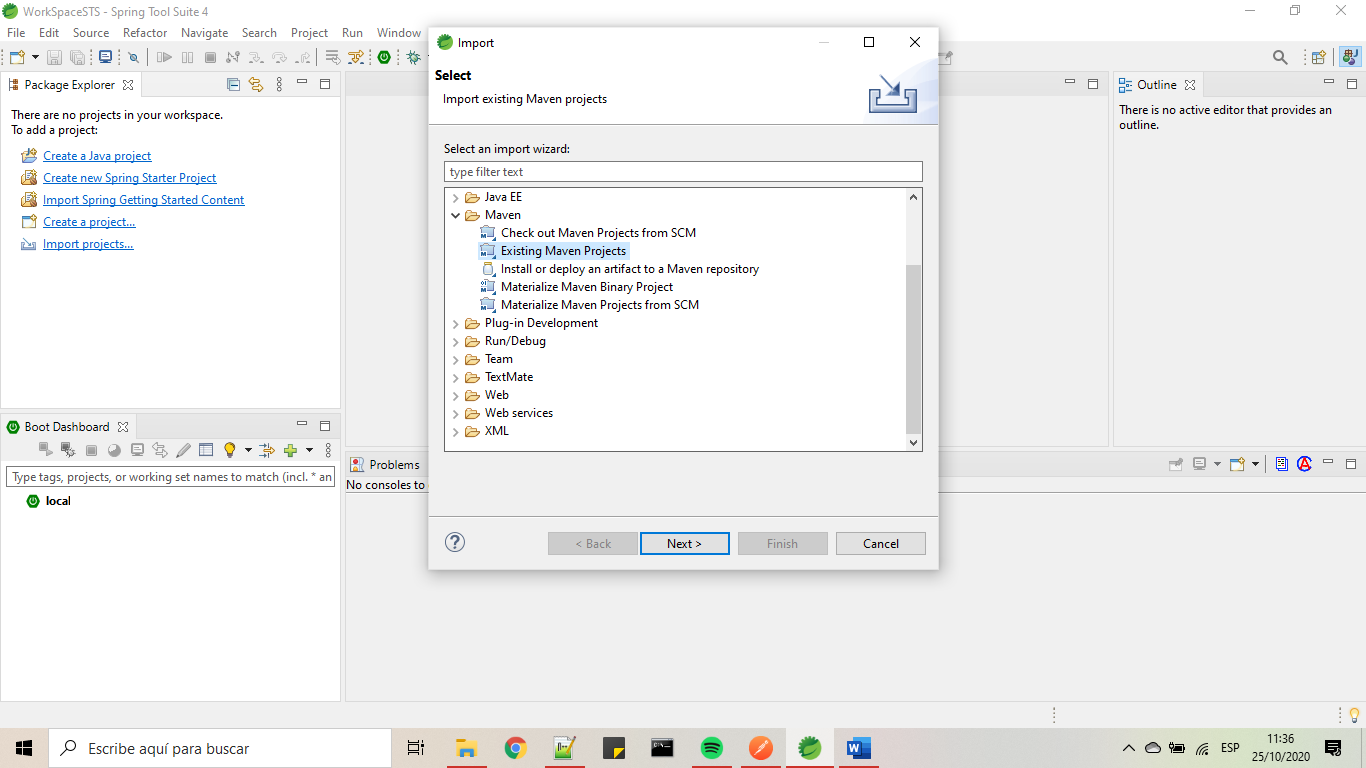
You can clone the project using git or you can directly download the project from GitHub and then unzip the file.

## Running the backend API

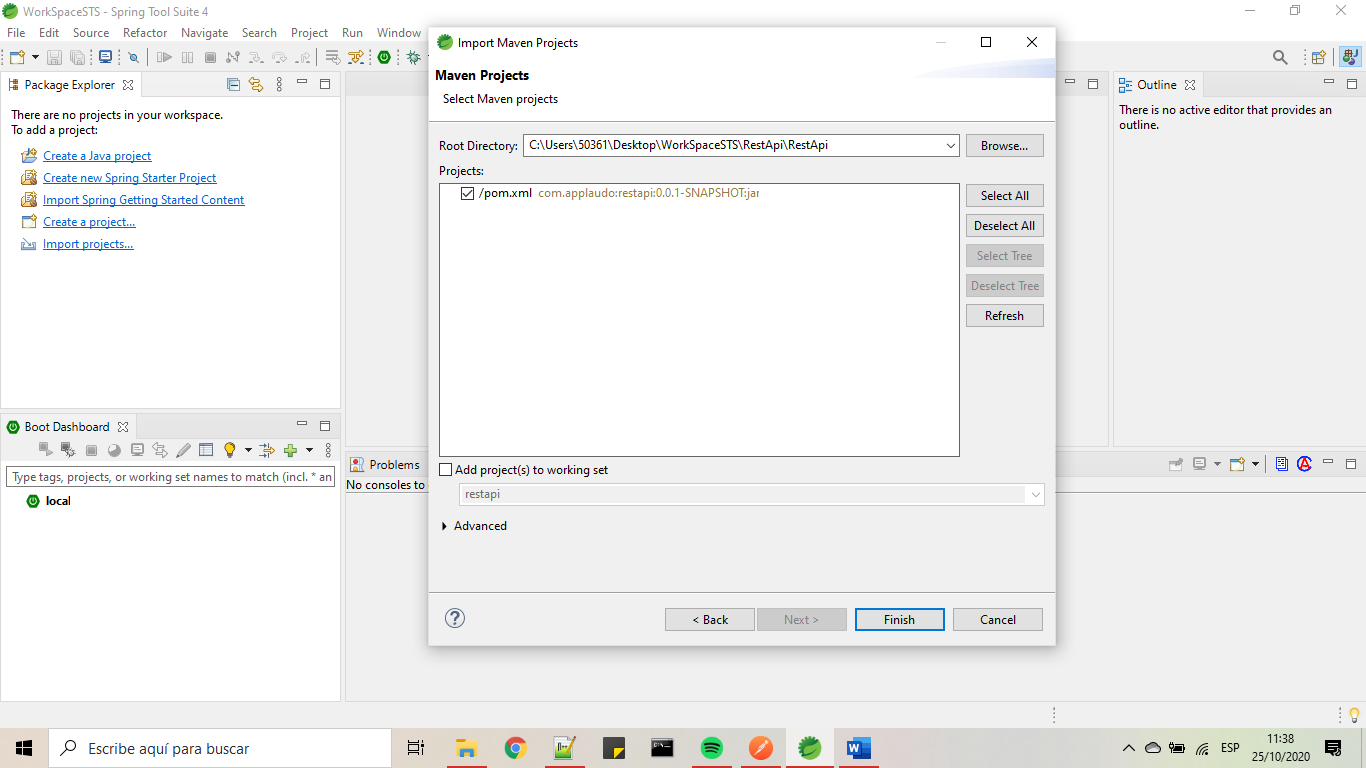
Open Spring Tool Suite application, you will see the workspace like this:



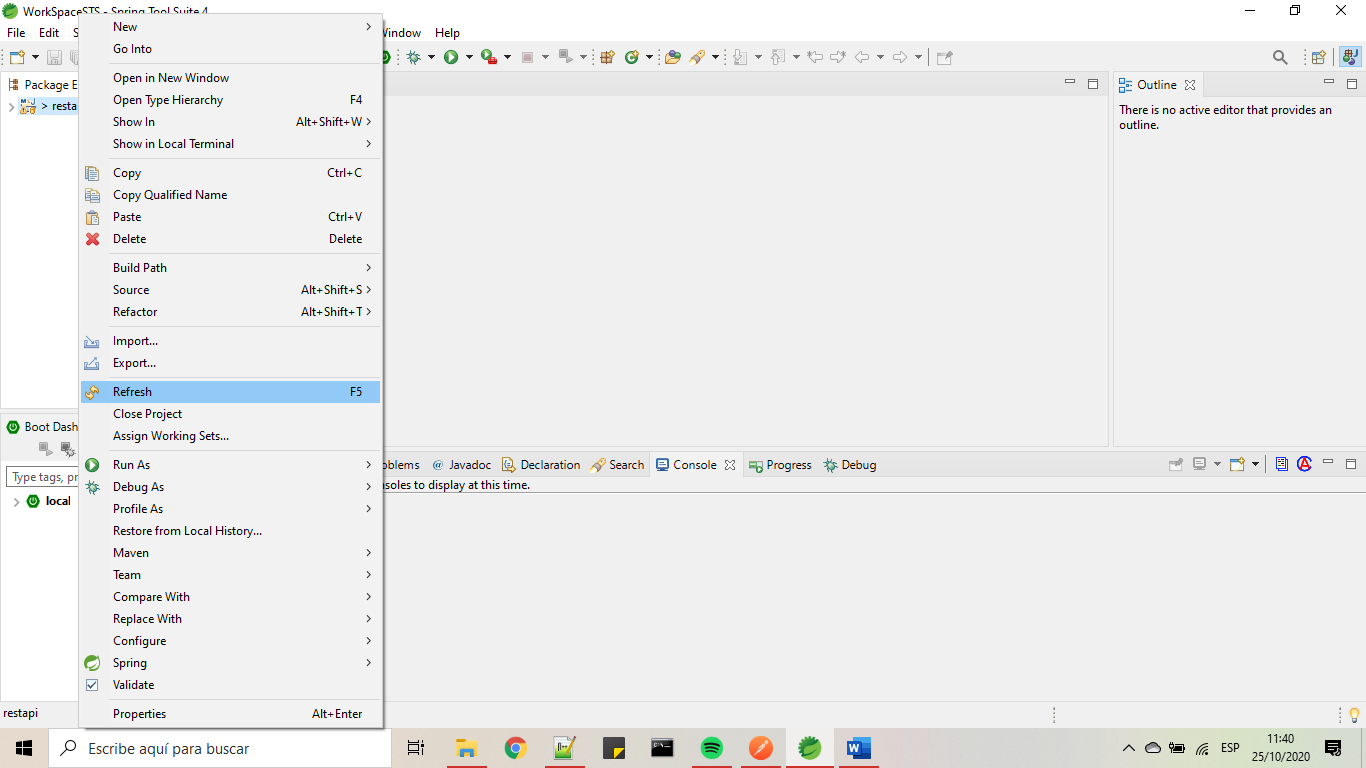
Click on File option then select the option import and then select existing maven projects like this:



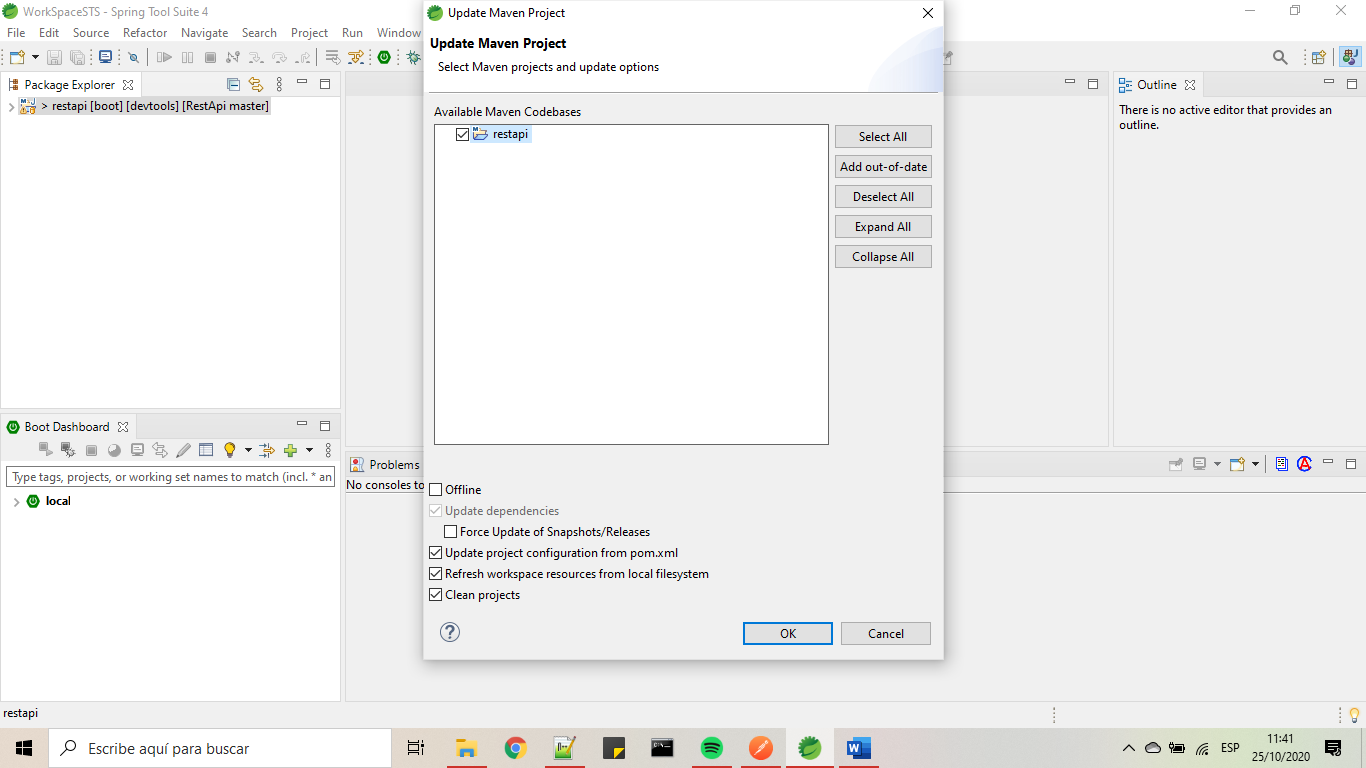
Click on next and then search for the RestAPI folder that previously you download from GitHub. After of that, select the RestAPI folder do click on finish button.



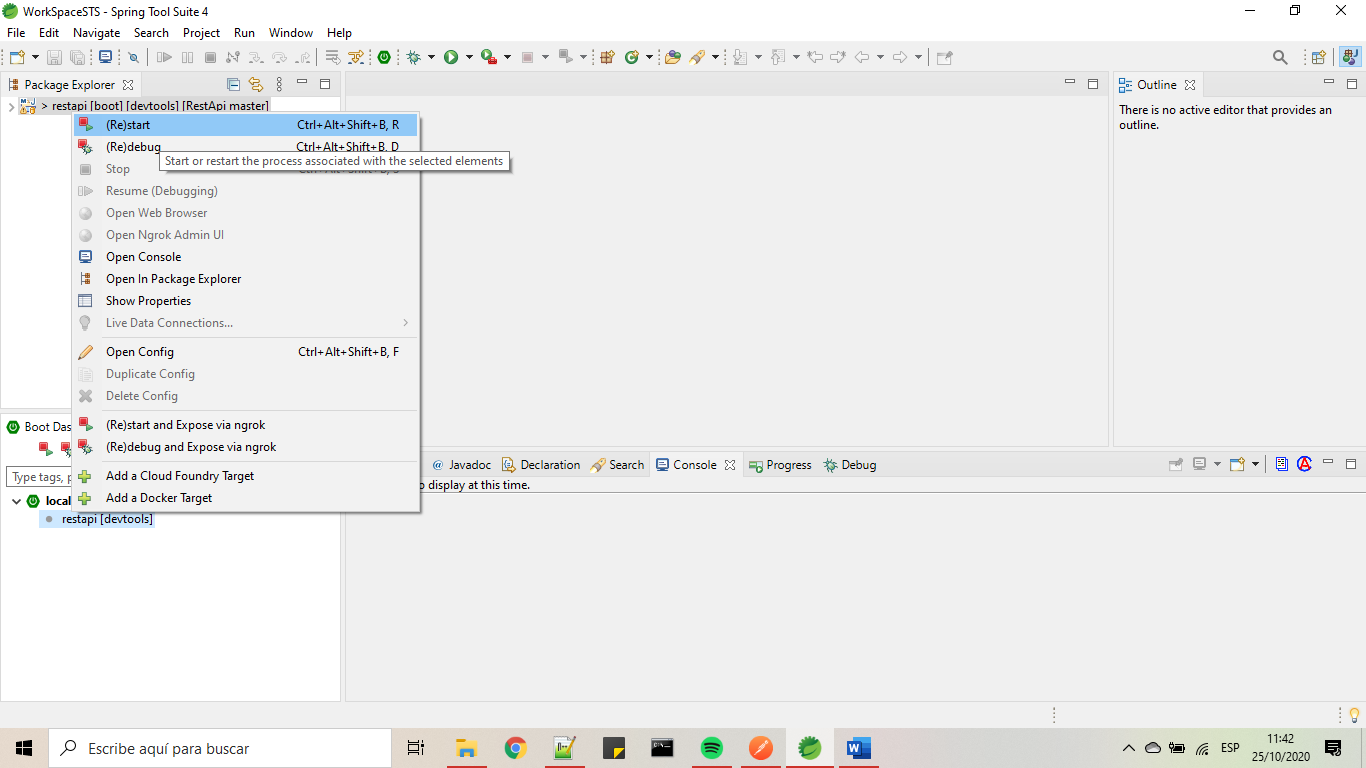
After of that you will see the project in spring tool suite: please do right click on the project and select refresh option:

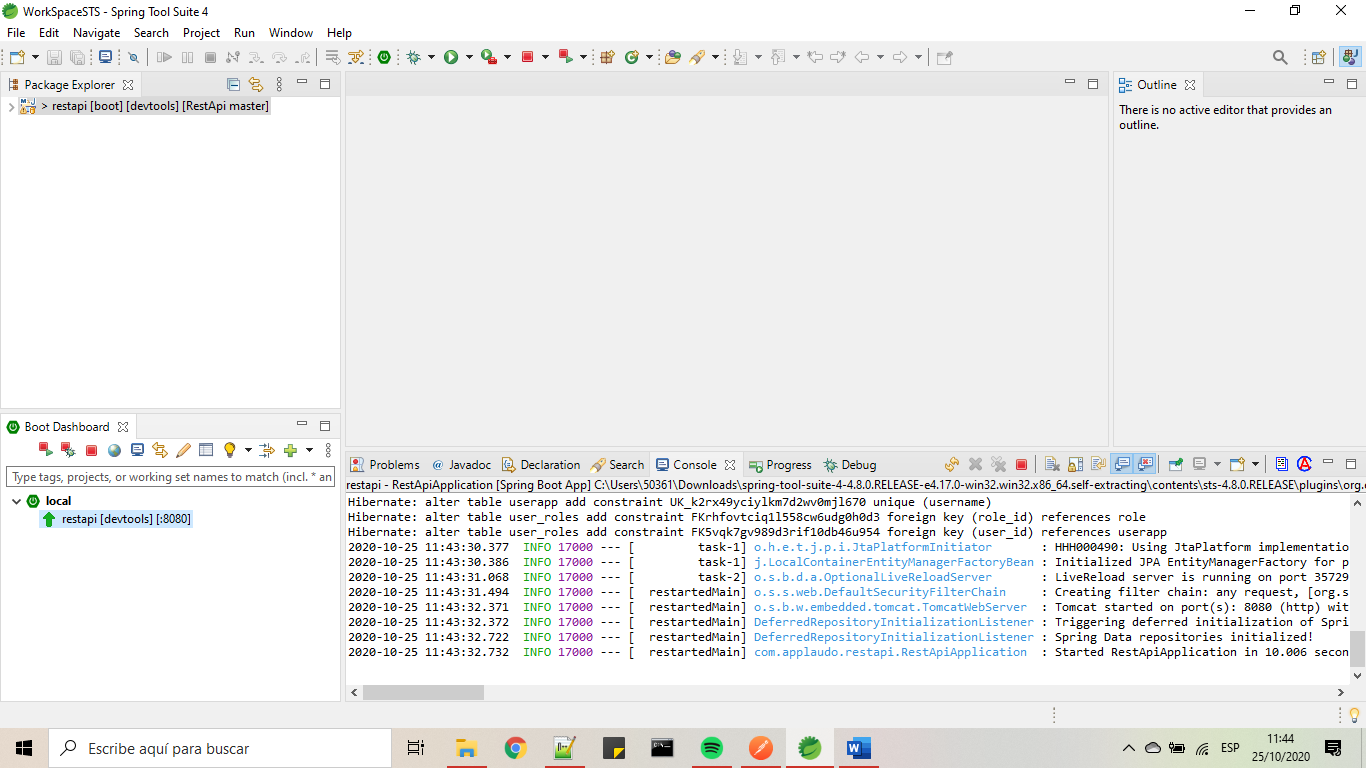
7

After of finish the refresh you have to do a right click again and select the option maven, and then select the option update Project, it will appear a window, select the project and do click on ok button.



Then look for Boot Dashboard on Spring tool suite and you will see the application with name restapi, do right click on that and select the option (Re)start. Wait for a moment and the application it will running.



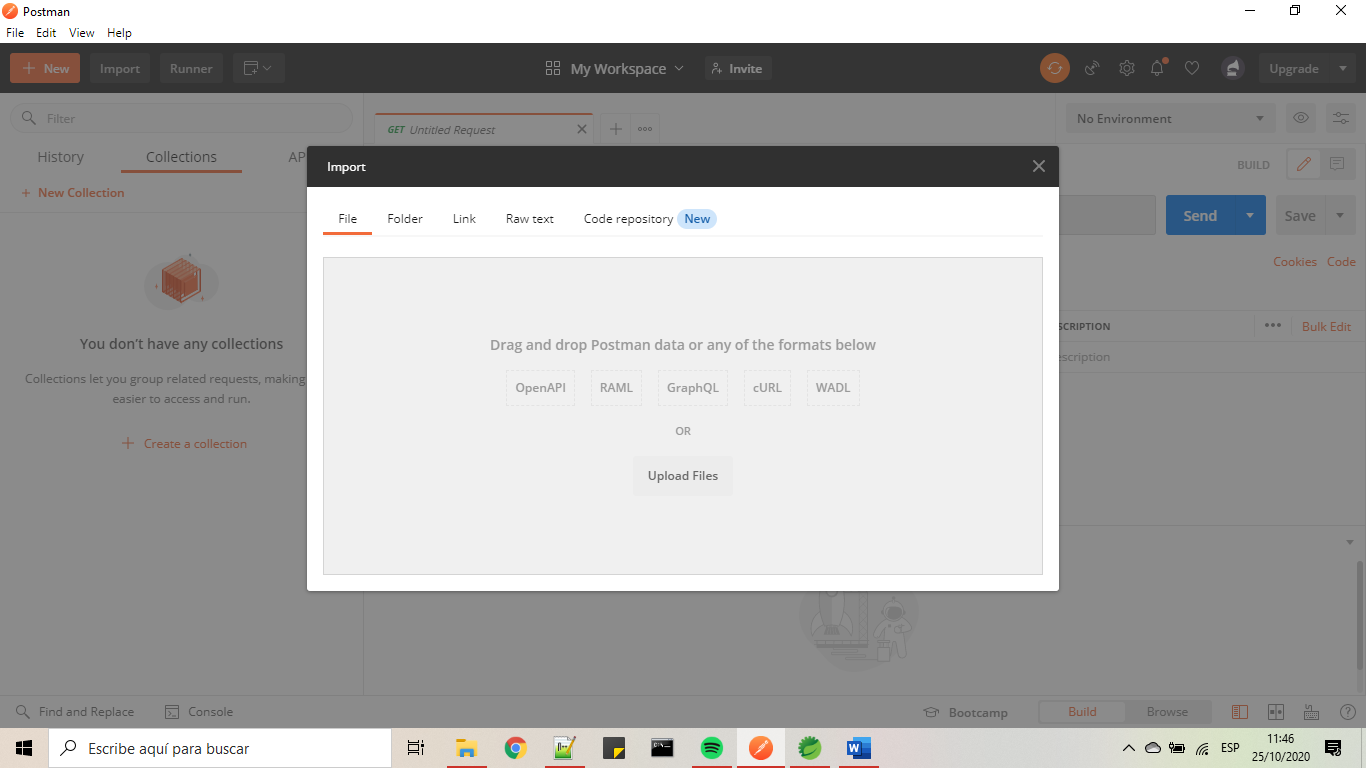


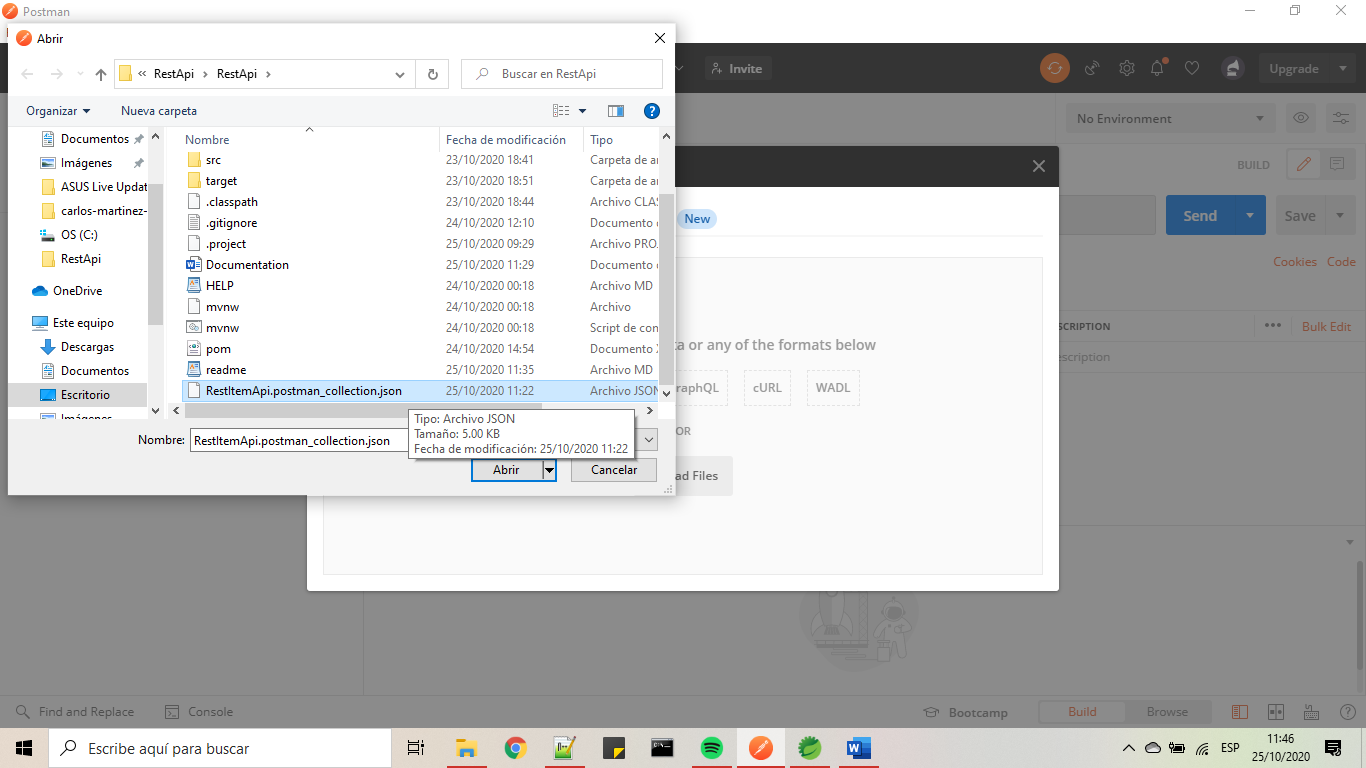
¡Now, the Item Rest API is Running! we can do several requests to this API using Postman.

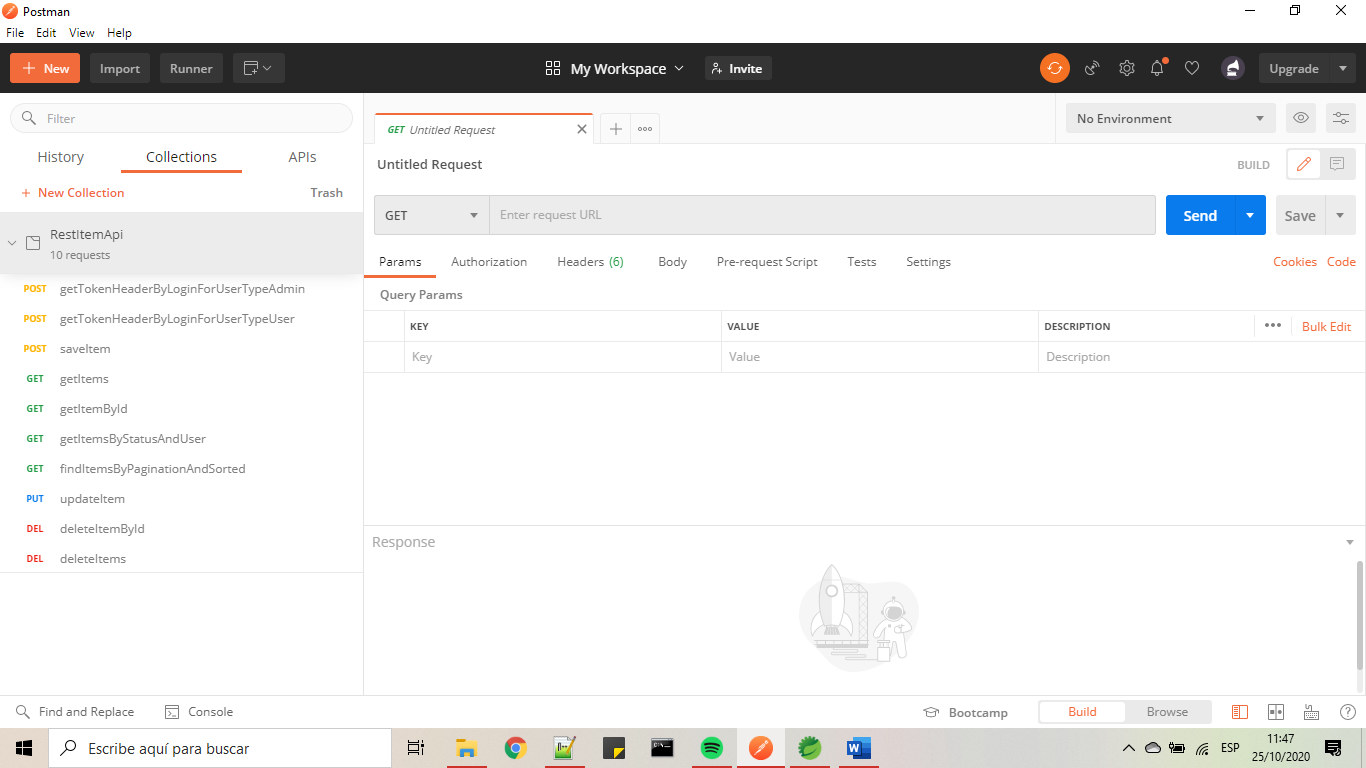
Important Note: As is required, we don’t need to install any database because the project use H2 database. Also keep in mind that with this kind of DB the registers will be deleted when we stop the application.

## Configuring Postman

Open PostMan application and press the button import and select the file which is located under the folder restAPI with name RestItemApi.postman\_collection.json. This file contains all the requests of the API.





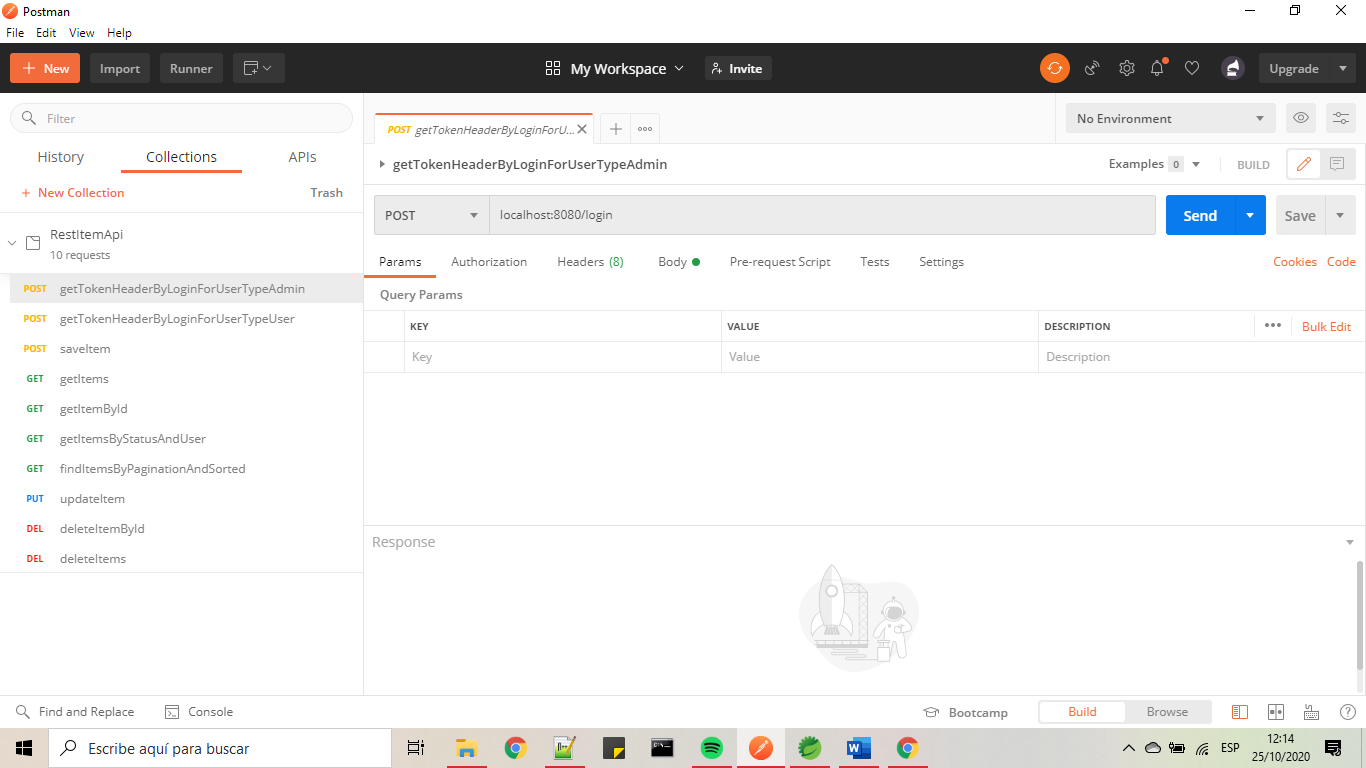


## Testing the backend application with Postman

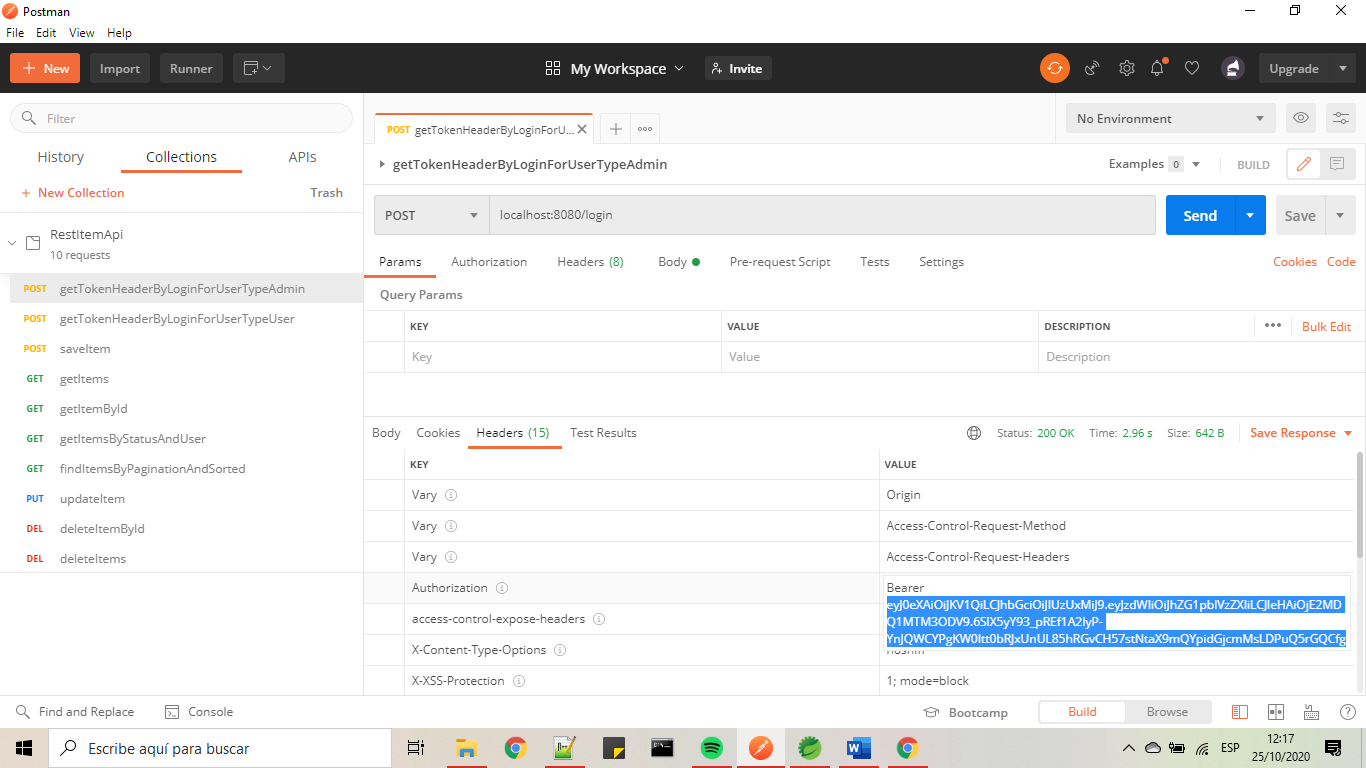
First, we need to understand that some endpoints of the API require not only authentication but also needs authorization and other endpoints required nothing. So, for achieve that I have created two users, one user with username **adminUser** which have **ADMIN** role and another user with username **customerUser** which have **USER** role.

### Getting the token for the adminUser

On postman select the request getTokenHeaderByLoginForUserTypeAdmin



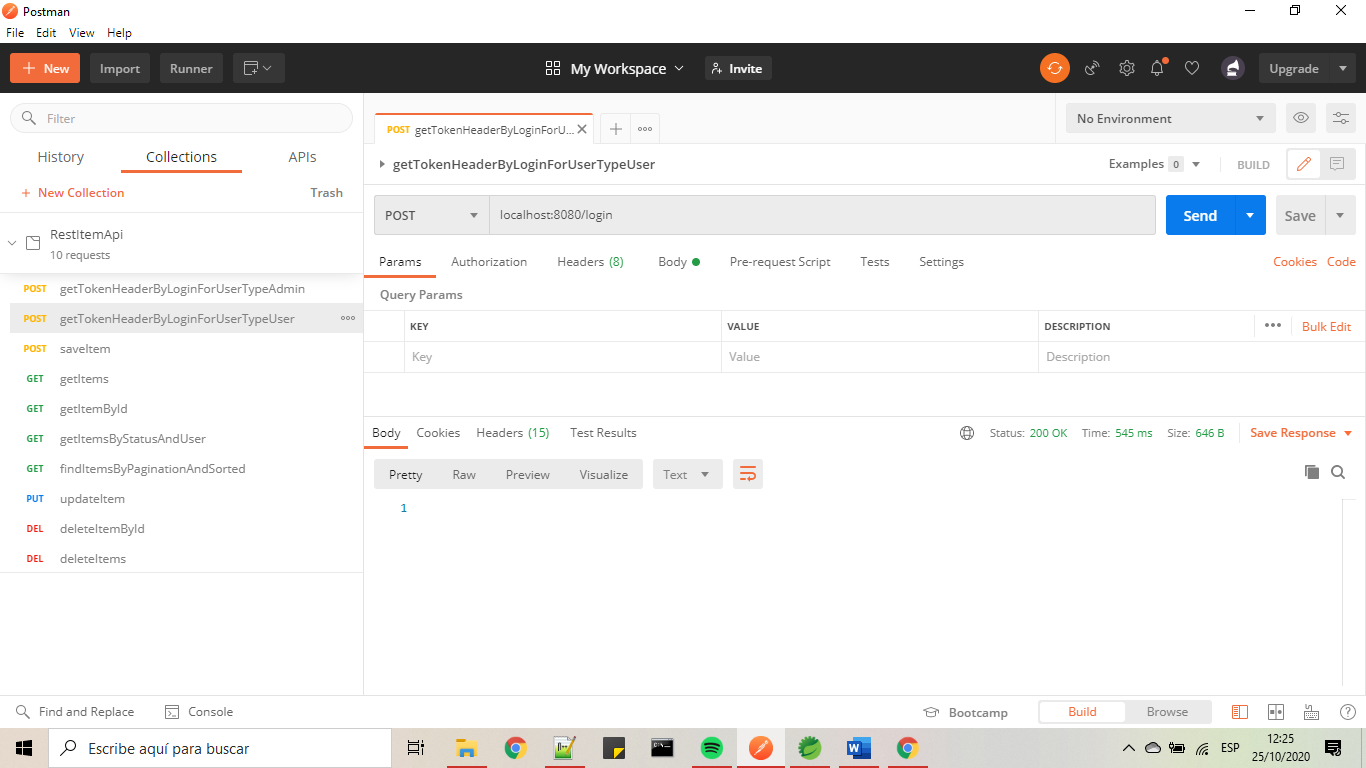
Click on send and look for a response header and copy the token in the key Authorization



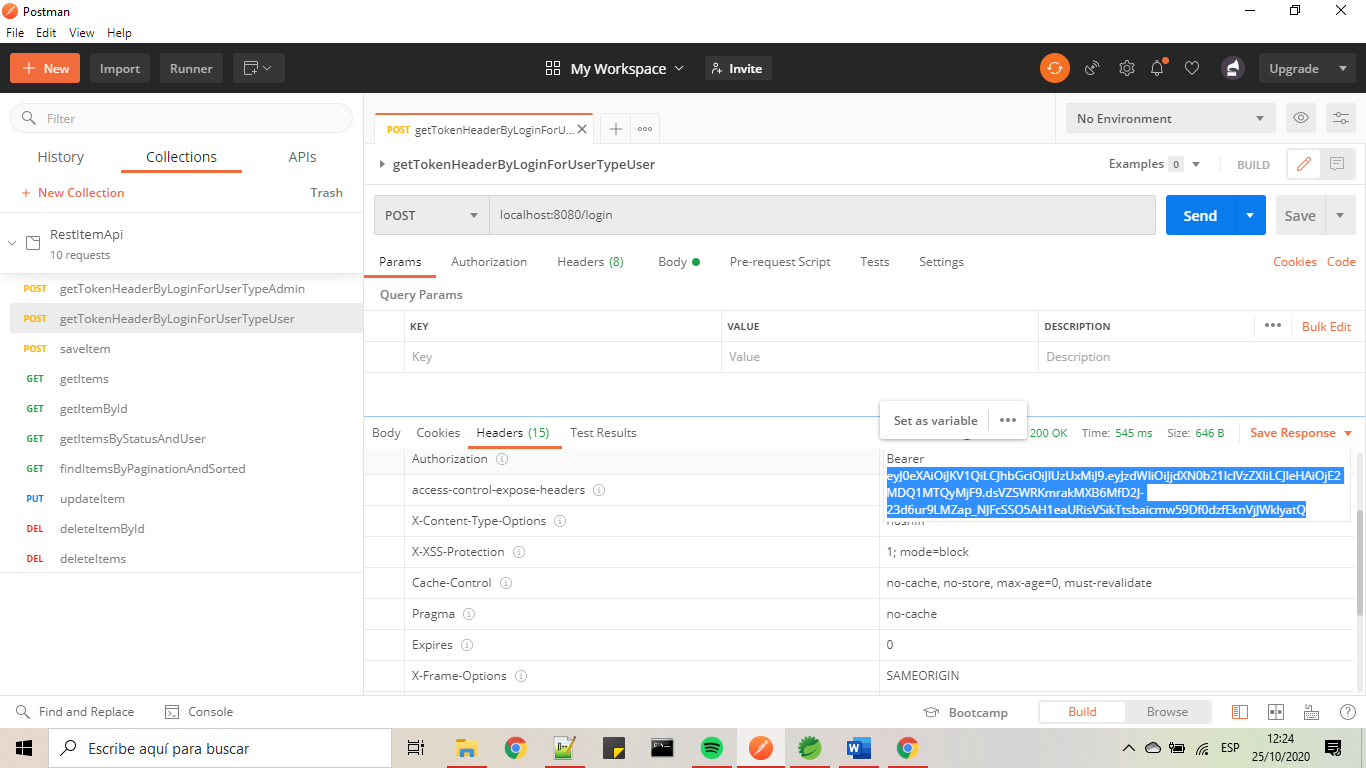
Save the token because we will use it then. This token has a duration of 10 days, this duration is defined in the application, so it can be change.

### Getting the token for the customerUser

On postman select the request getTokenHeaderByLoginForUserTypeUser



Click on send and look for a response header and copy the token of the key Authorization



Save the token because we will use it then. This token has a duration of 10 days, this duration is defined in the application, so it can be change.

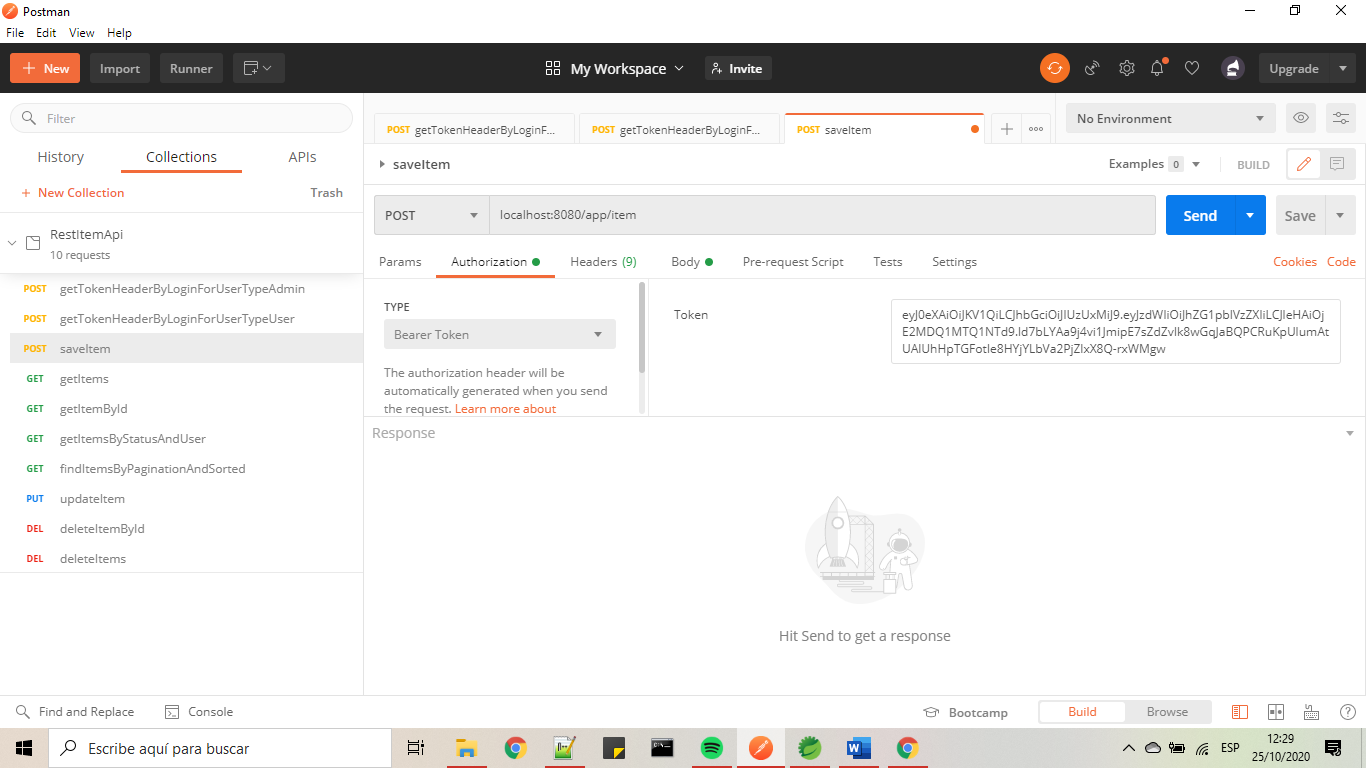
For a better explanation:

|  |  |  |
| --- | --- | --- |
| EndpointName | ¿Require Authentication? | ¿Require Authorization? |
| saveItem | Yes | Yes (‘ADMIN’) |
| getItems | No | No |
| getItemById | No | No |
| getItemByStatusAndUser | No | No |
| findItemByPaginationAndSorted | No | No |
| updateItem | Yes | No |
| deleteItems | Yes | Yes (‘ADMIN’) |
| deleteItemById | Yes | Yes (‘ADMIN’) |

Based on that table, we need to add the token on the header for the request saveItem, updateItem, deleteItems and deleteItemById because if we don’t add the token to that requests, we are going to have errors of authentication.

### Example of adding token to the saveItem request:

In this case we just need to paste the token of the admin that we create before. Search in a section Authorization of the request saveItem and selecting on Type Bearer Token and add the value of the token. May already exist a previously token in the request but is better to use a new (like the token that we create before) because of the duration of the token that exist in the request may has expired when you are testing the API.



If we try to use for example the token of the user with role USER in the saveItem endpoint, we will get a http response with status 403 because only user with role ADMIN can save Items. O for example if we try to use an invalid token, we will get a http response with status 401 because only authenticated users can save Items. All this configuration was reached thanks to Spring Security.

**Remember to add or update the token to the rest of the endpoints; updateItem, deleteItems and deleteItemById.**

After that, you are be able to test all the endpoints of the API.