**Rest with Spring**

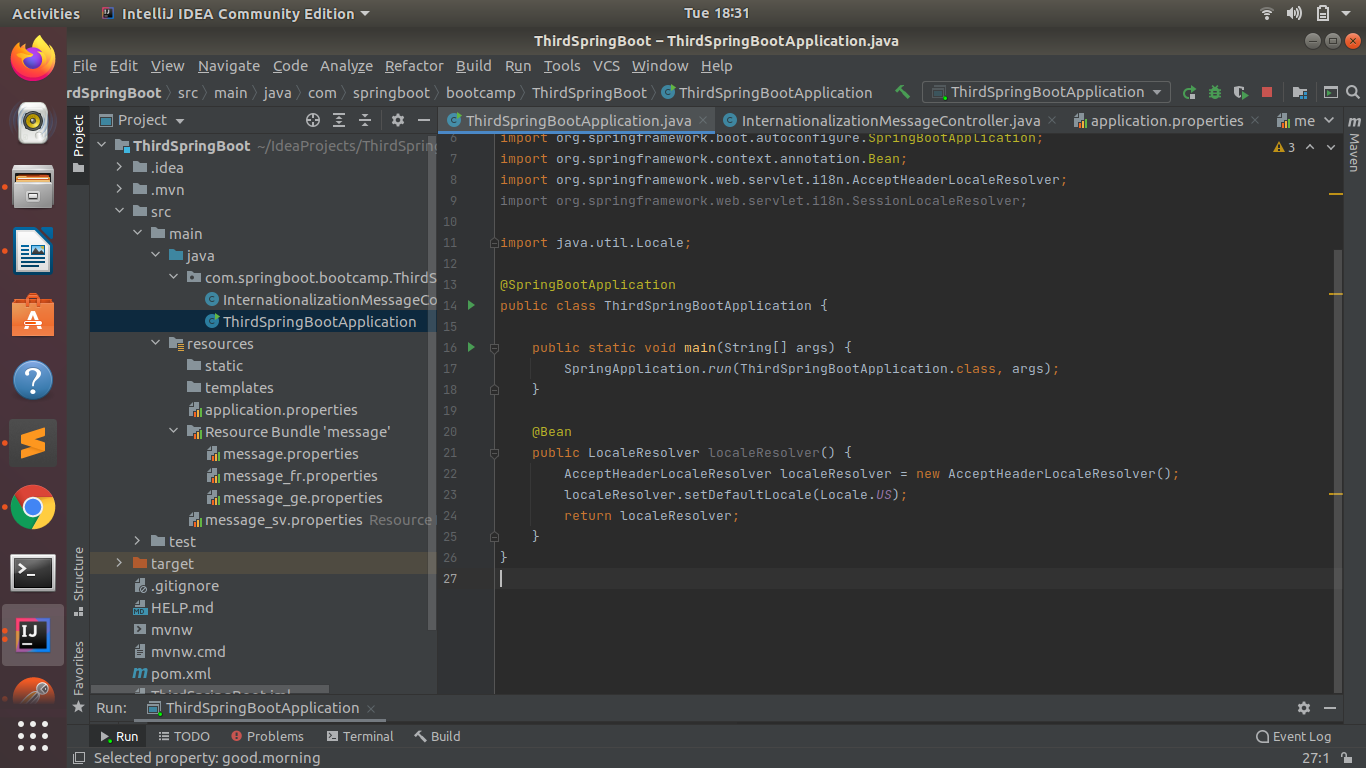
**Date 08/07/2021**

**Google doc link**: <https://docs.google.com/document/d/1EQvkCfwp8-aG3cNdJaE0WCfB-WGeMsP0pnbxv8sJvmU/edit?usp=sharing>

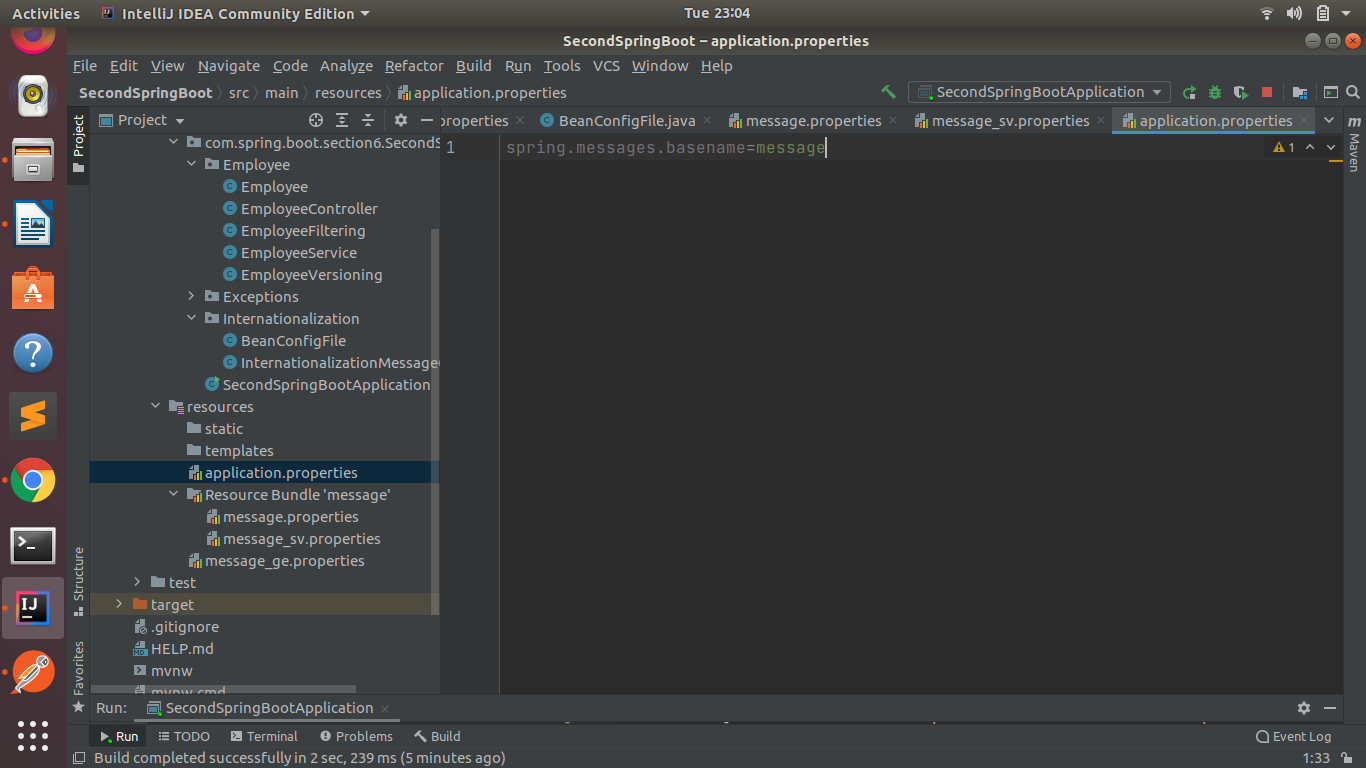
**\*Internationalization**

1. Add support for Internationalization in your application allowing messages to be shown in English, German and Swedish, keeping English as default.

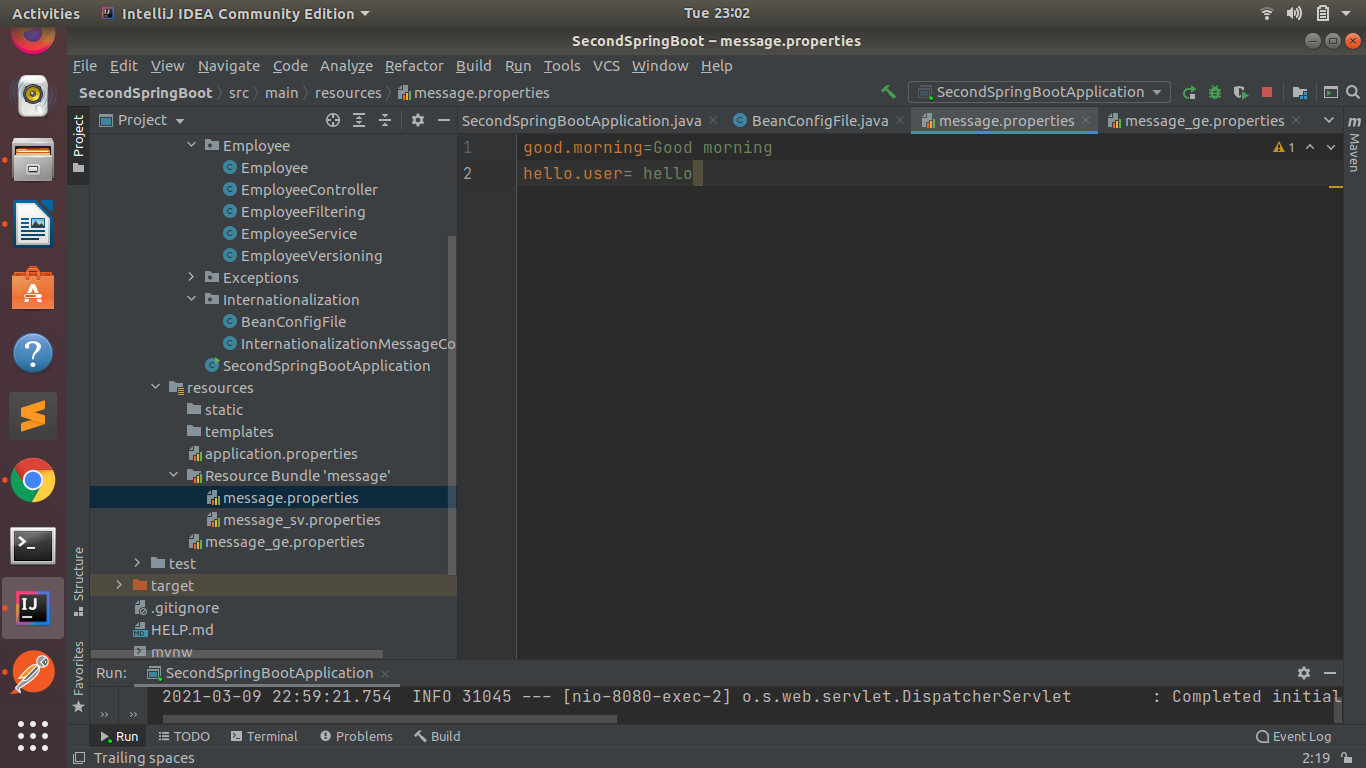


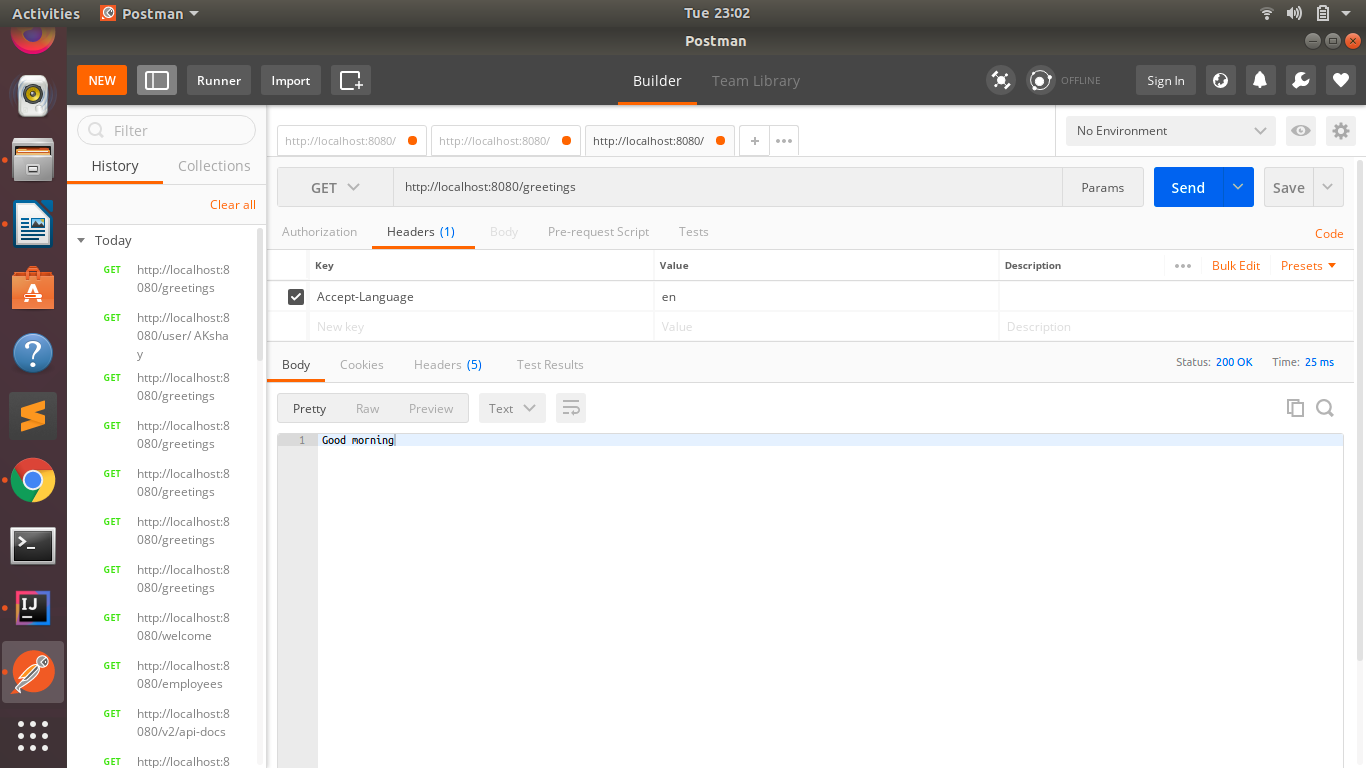


Application.properties file:

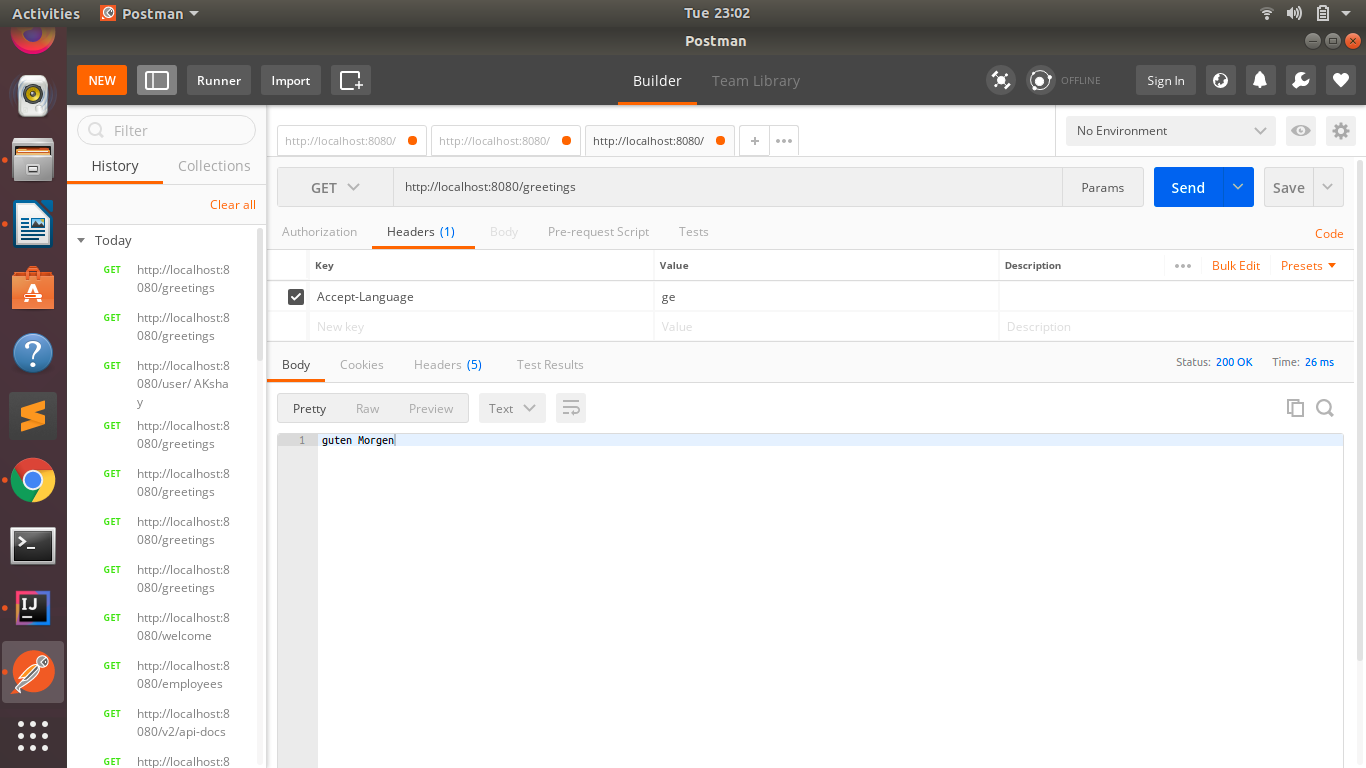
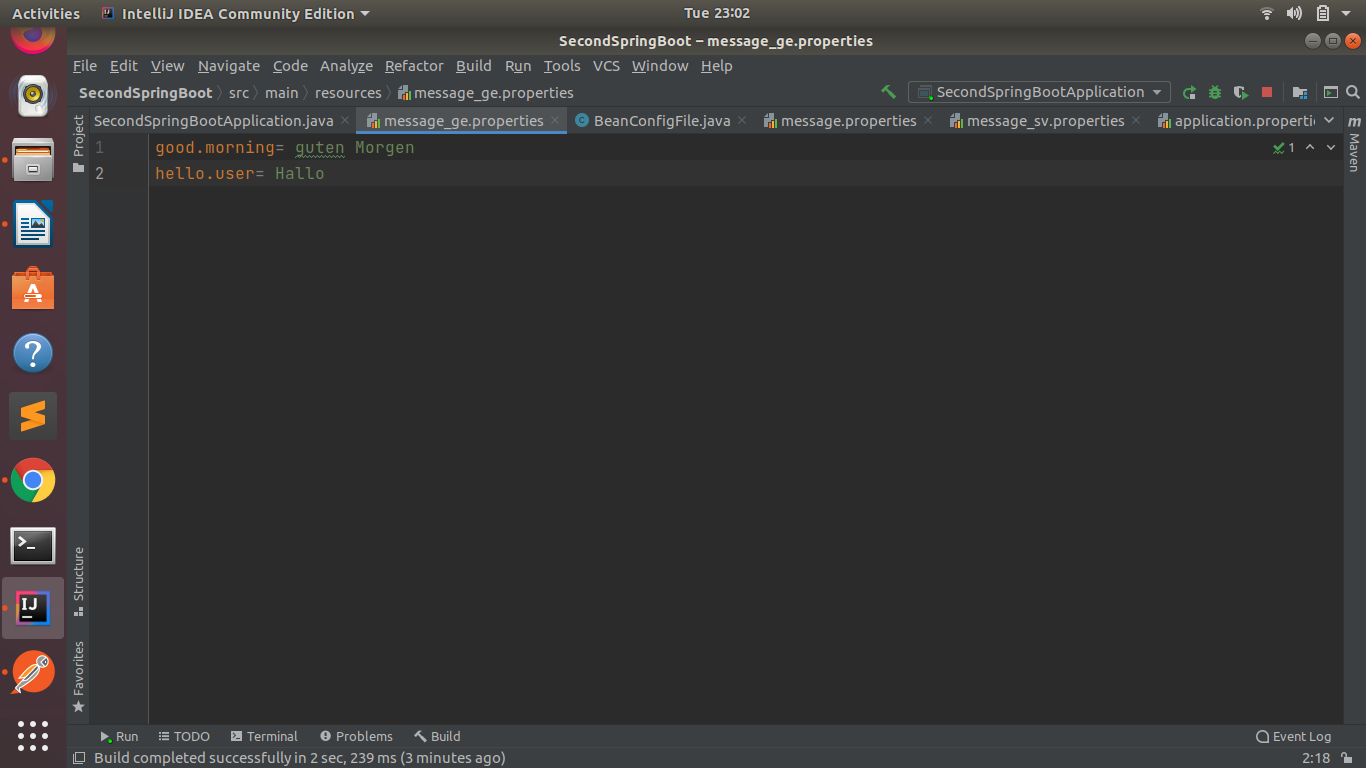


English:

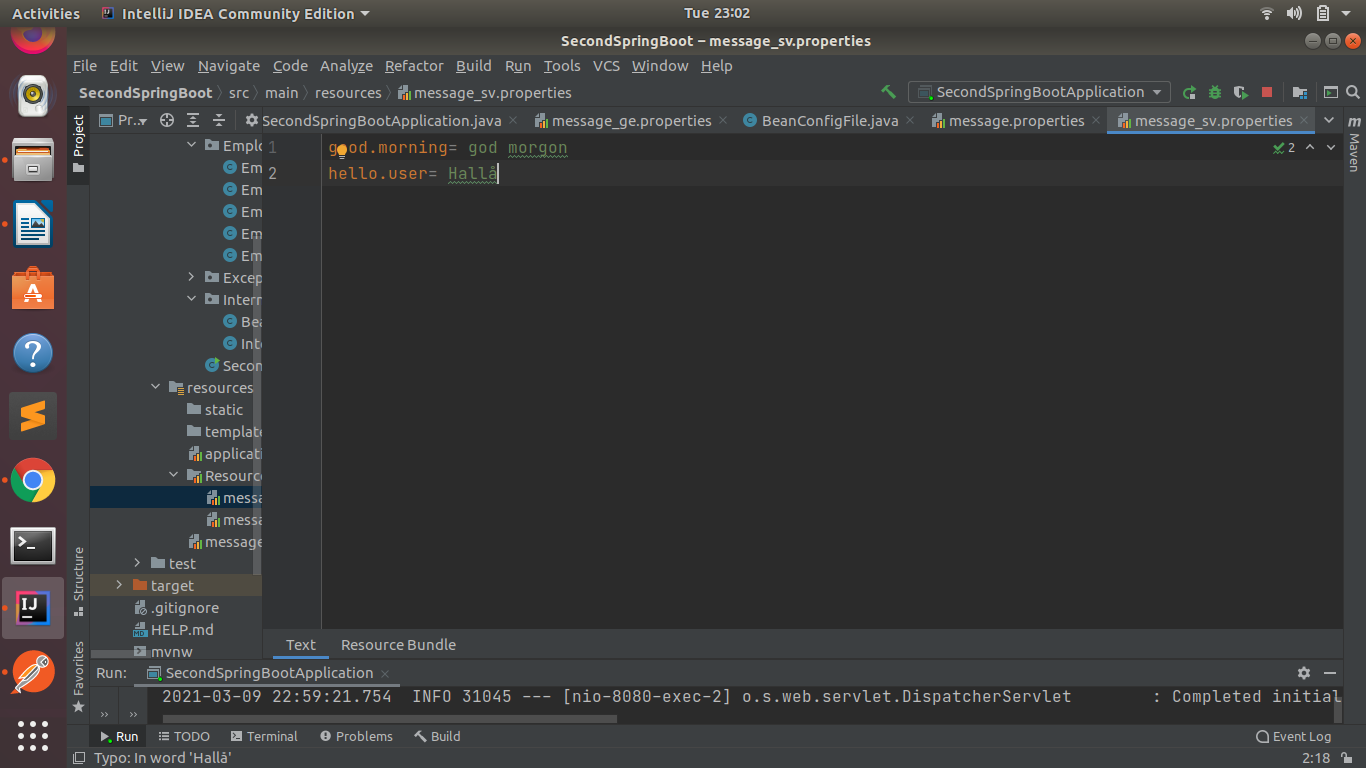




German:



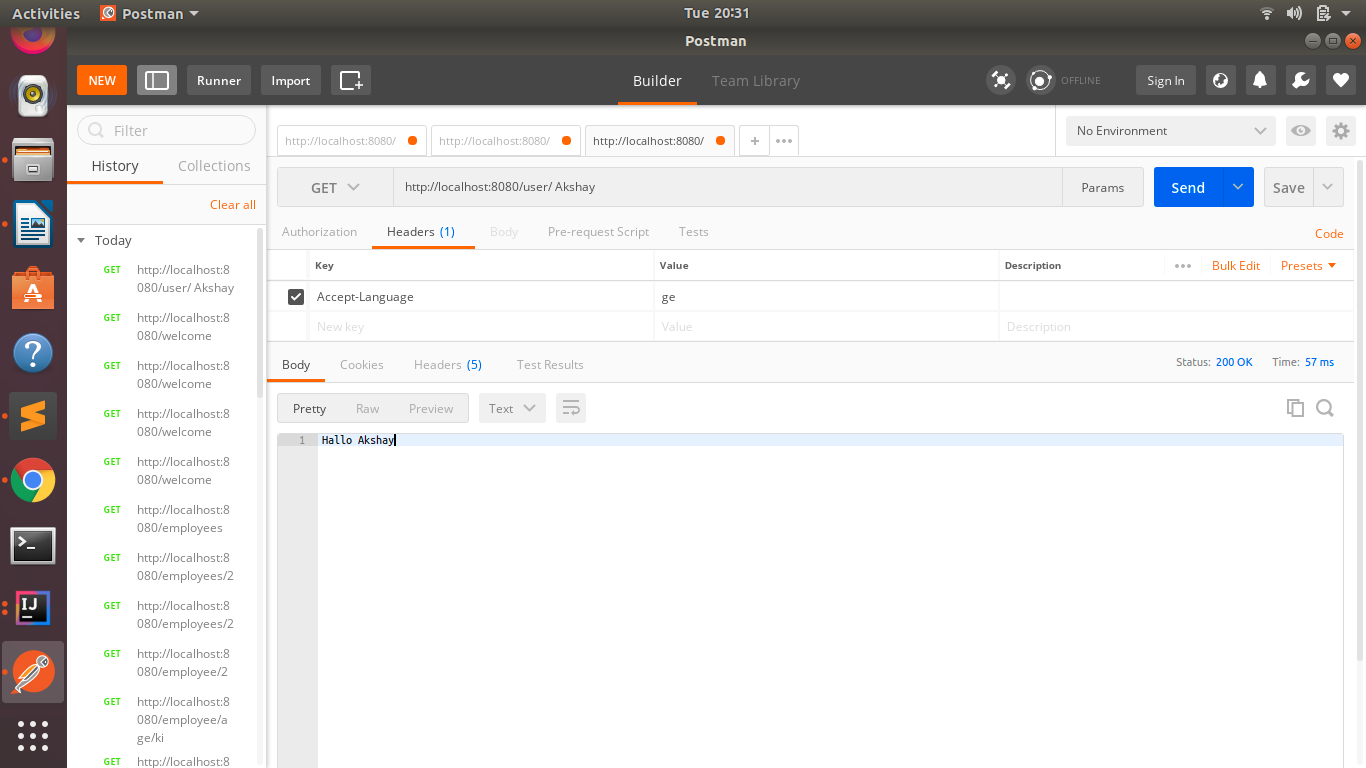
Swedish:

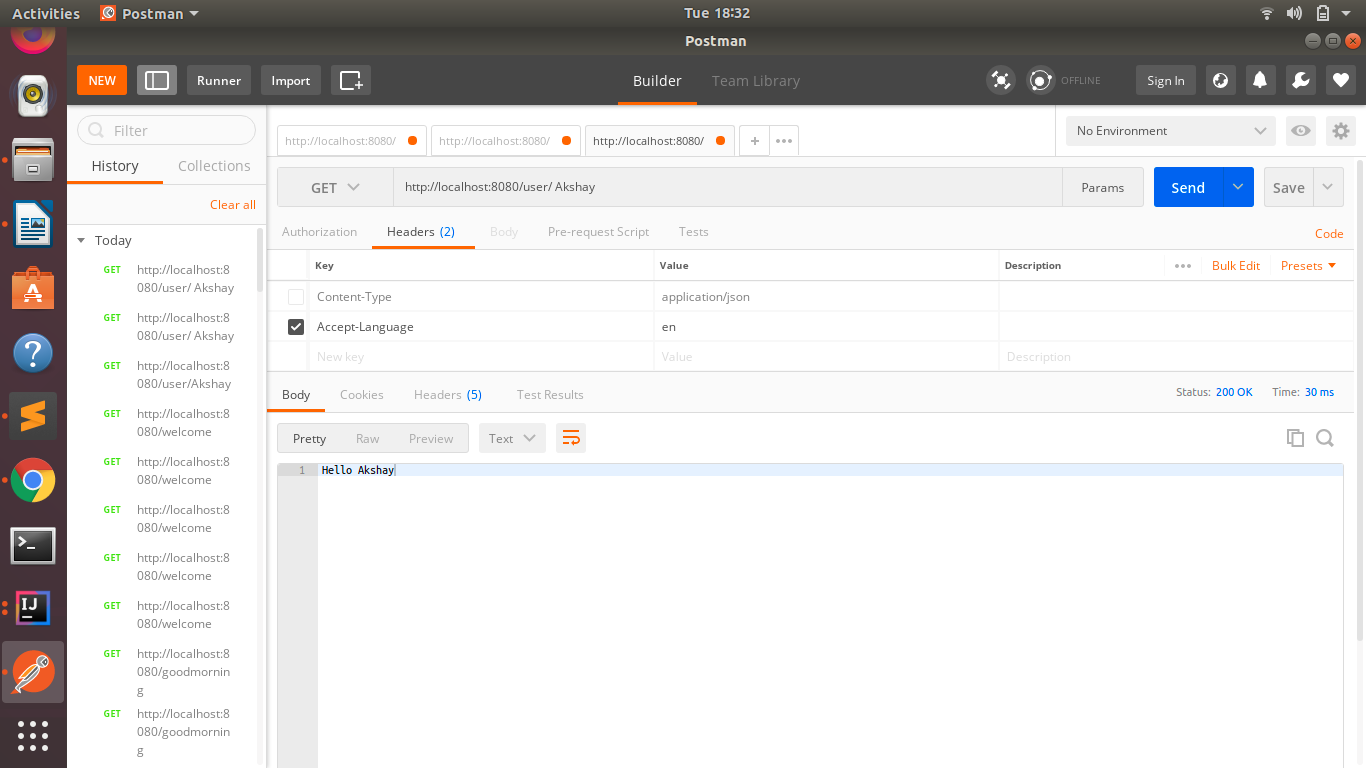




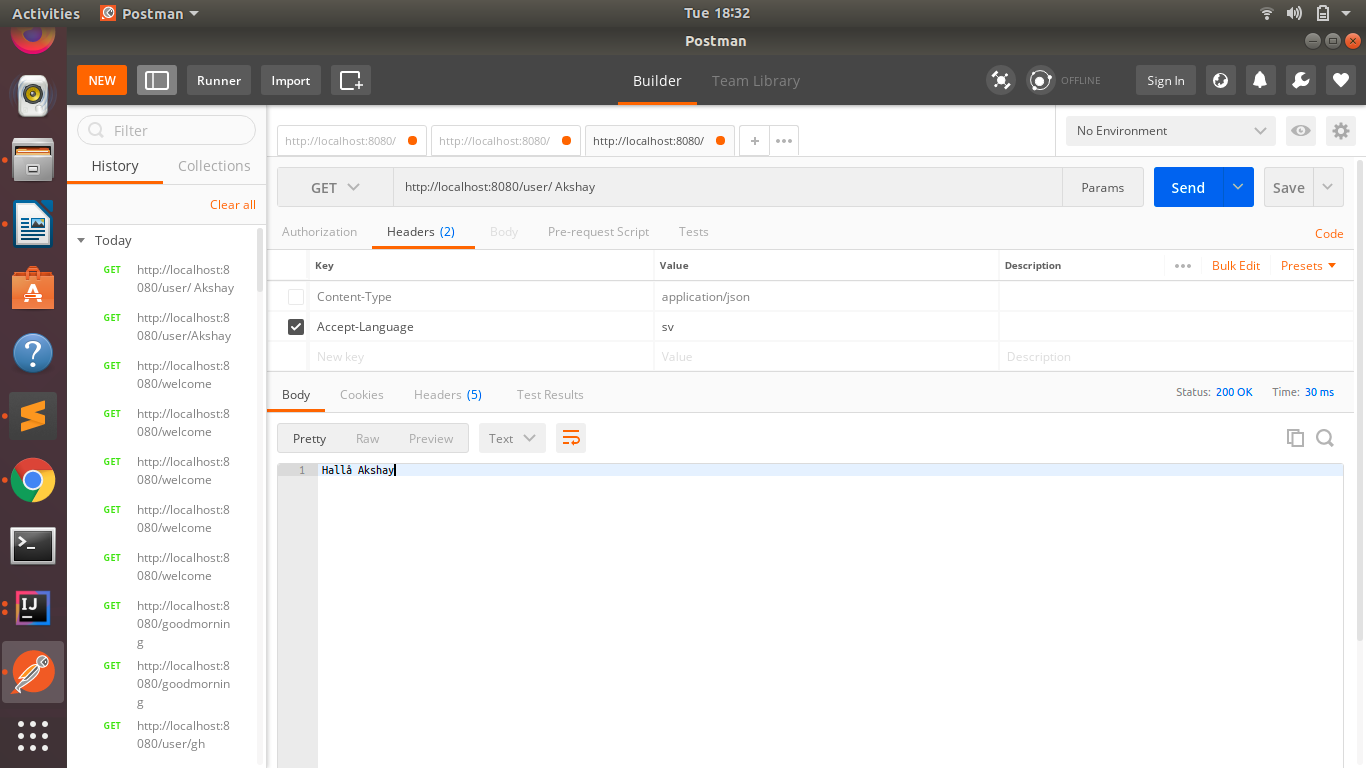
1. Create a GET request which takes "username" as param and shows a localized message "Hello Username". (Use parameters in message properties)

German:



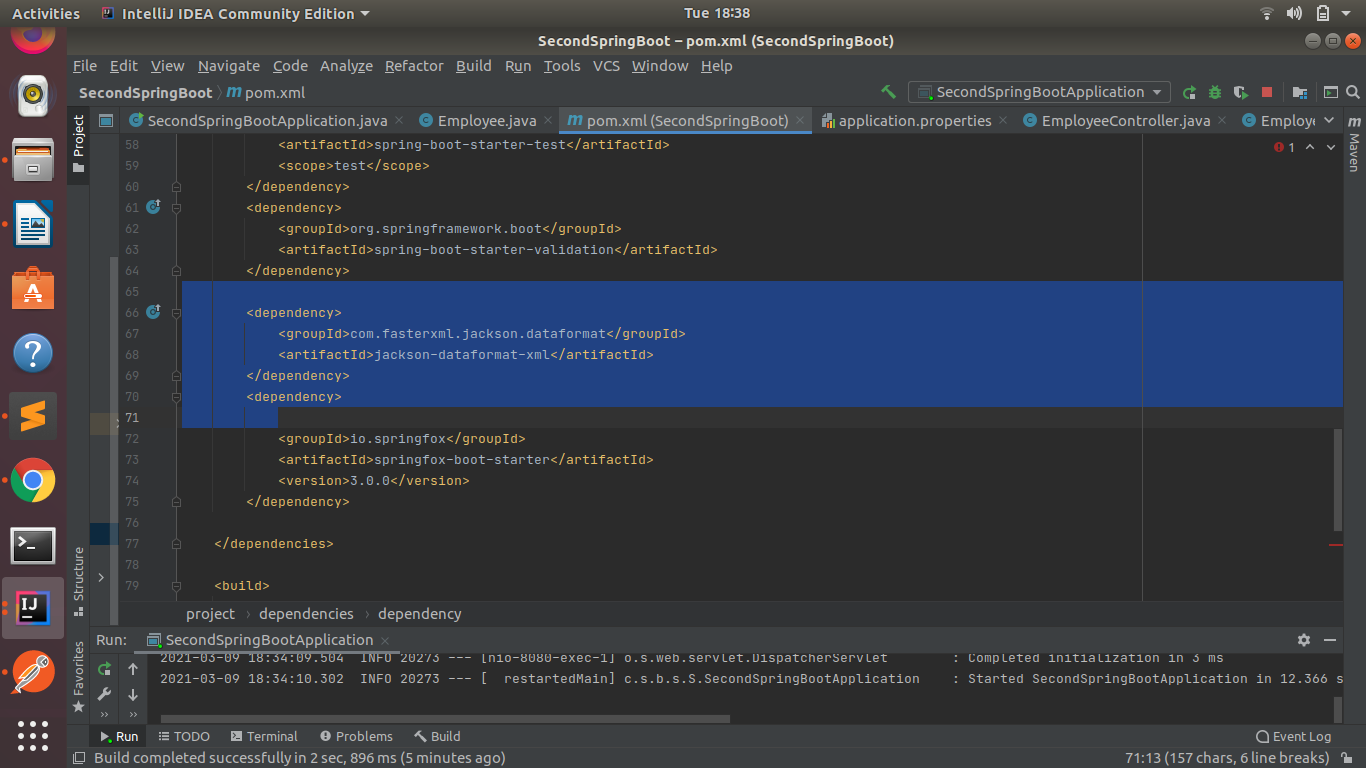
English:

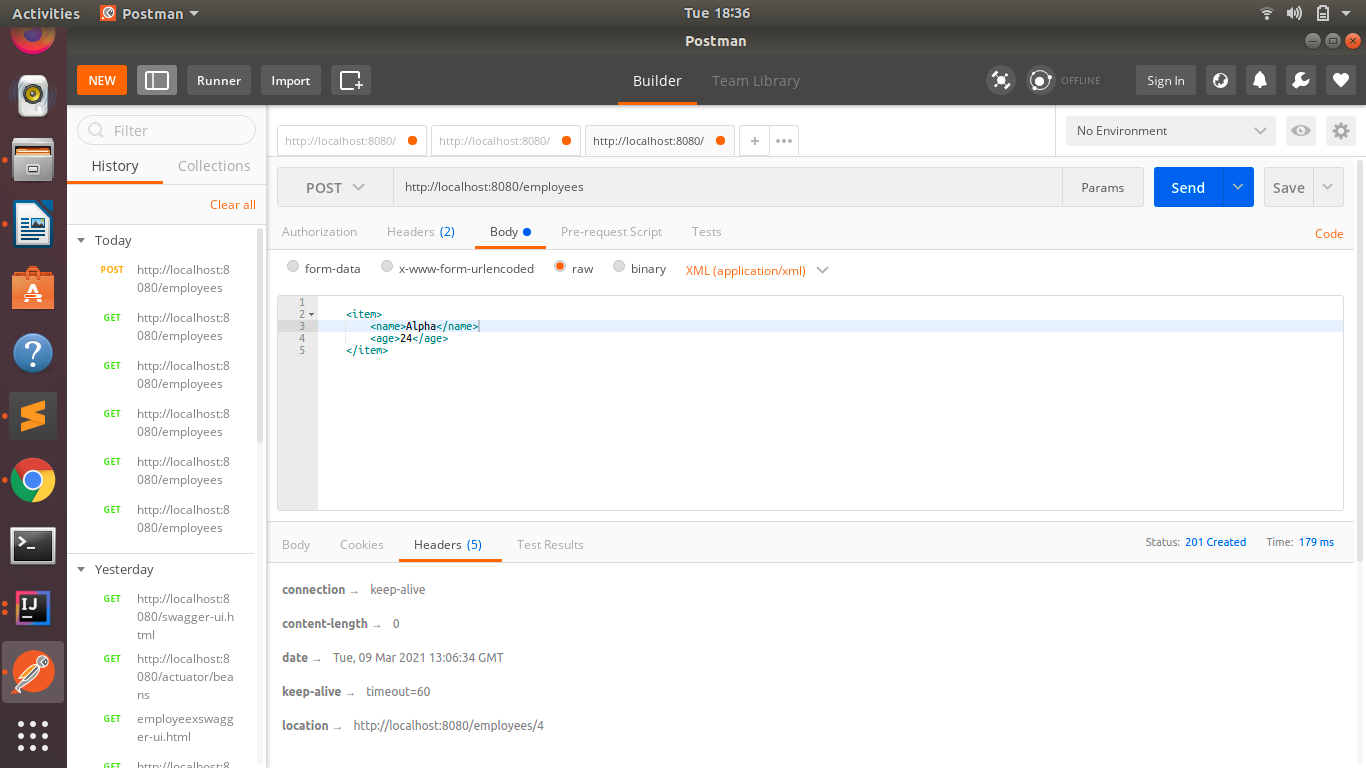
Swedish:



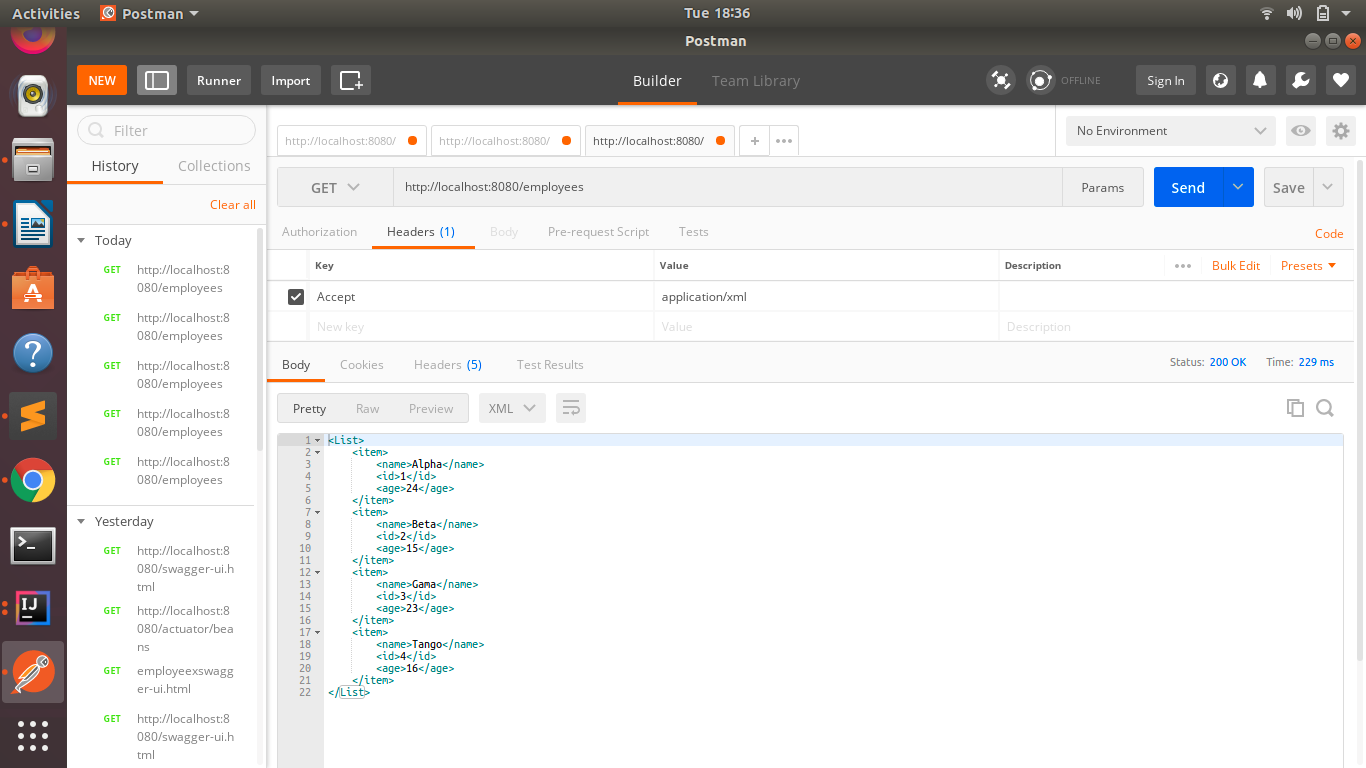
\*Content Negotiation

3. Create POST Method to create user details which can accept XML for user creation.



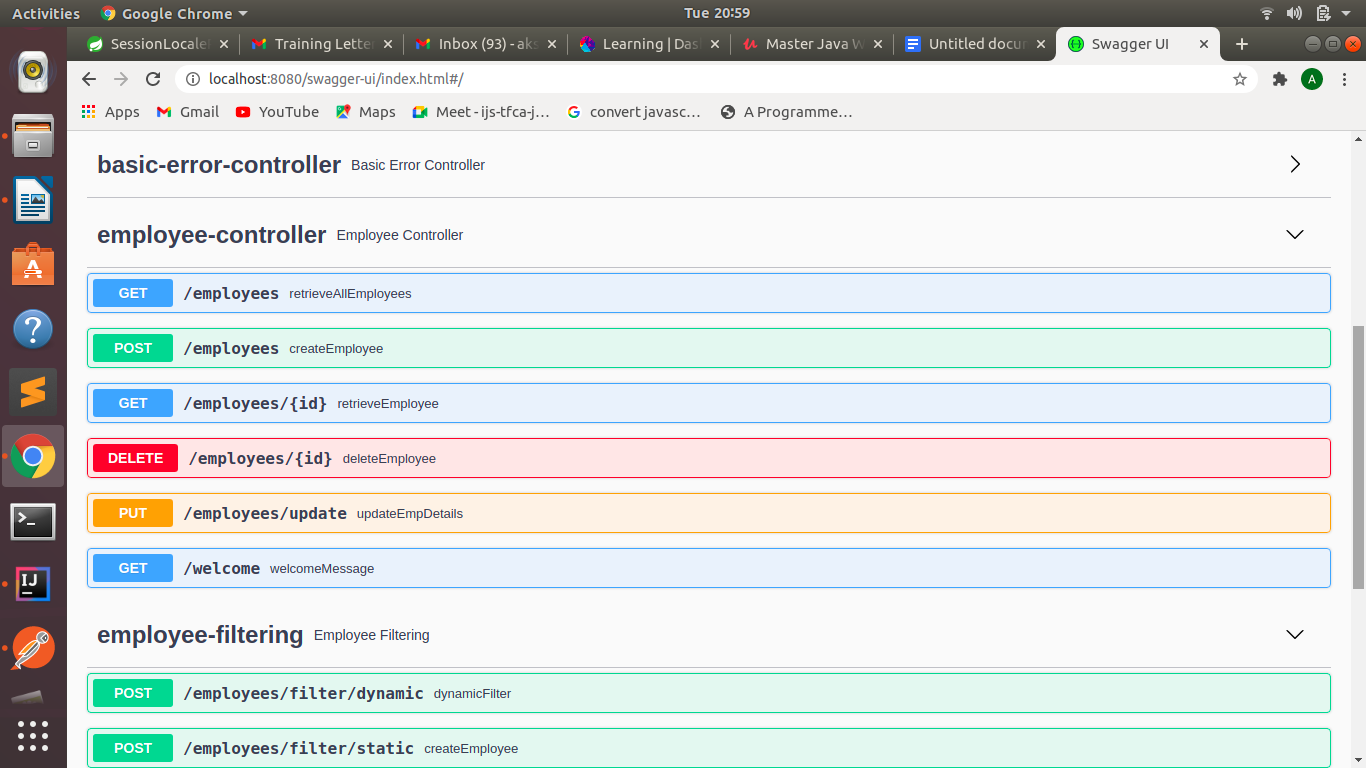


4. Create GET Method to fetch the list of users in XML format.

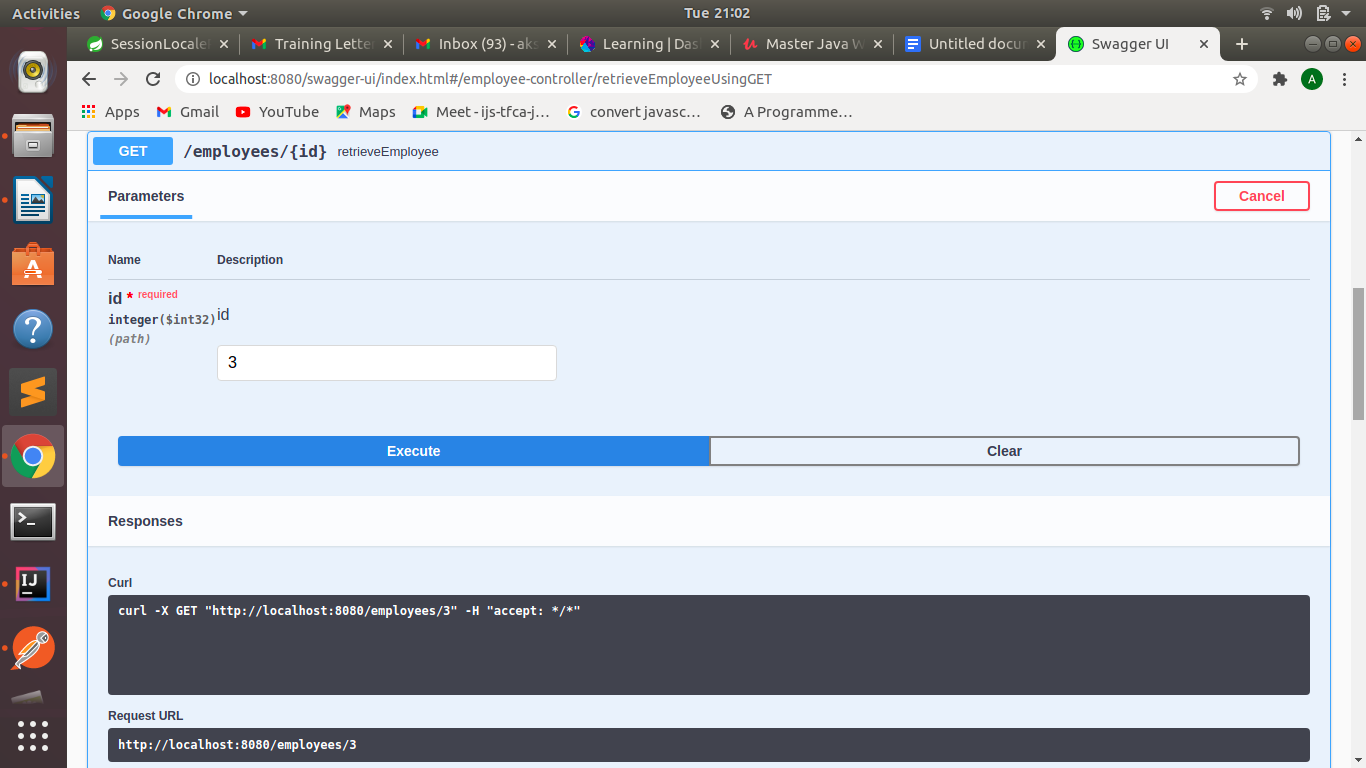


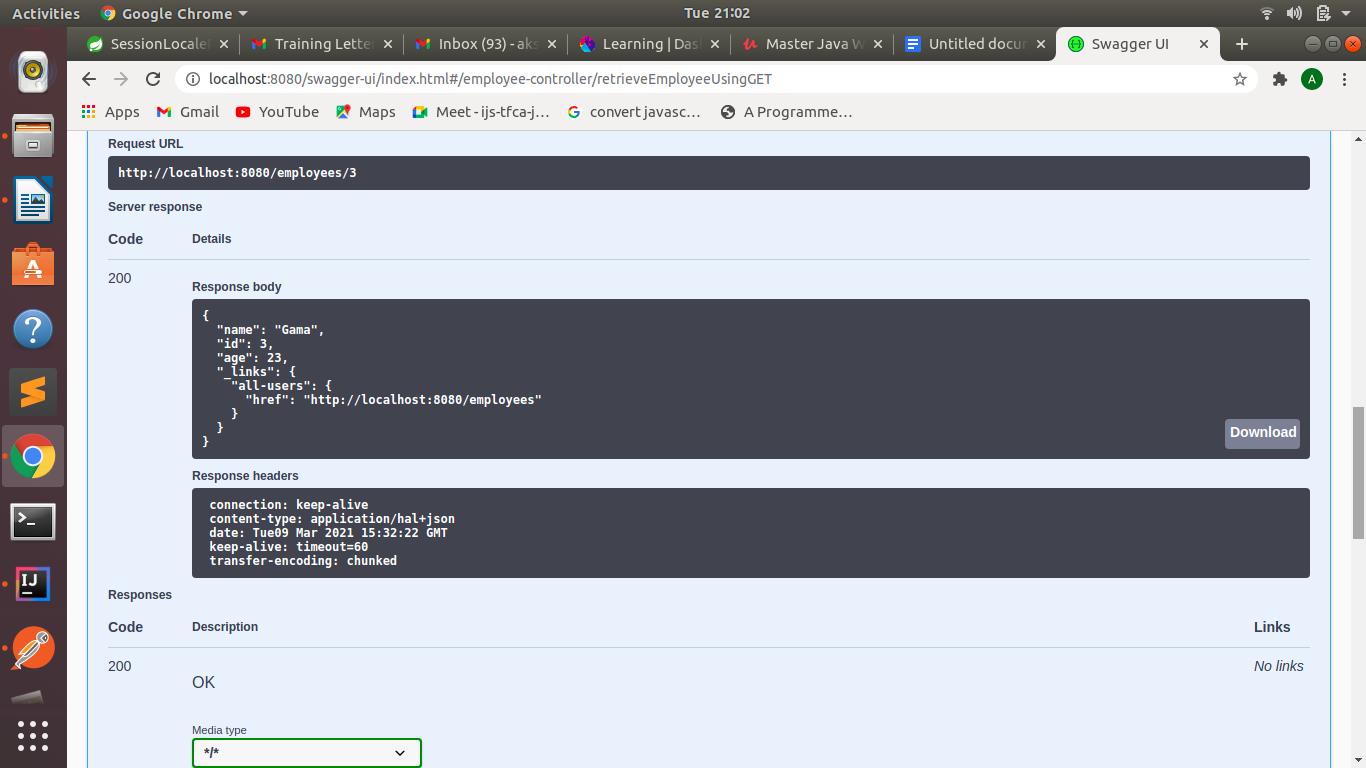
\***Swagger**

5. Configure swagger plugin and create document of following methods:

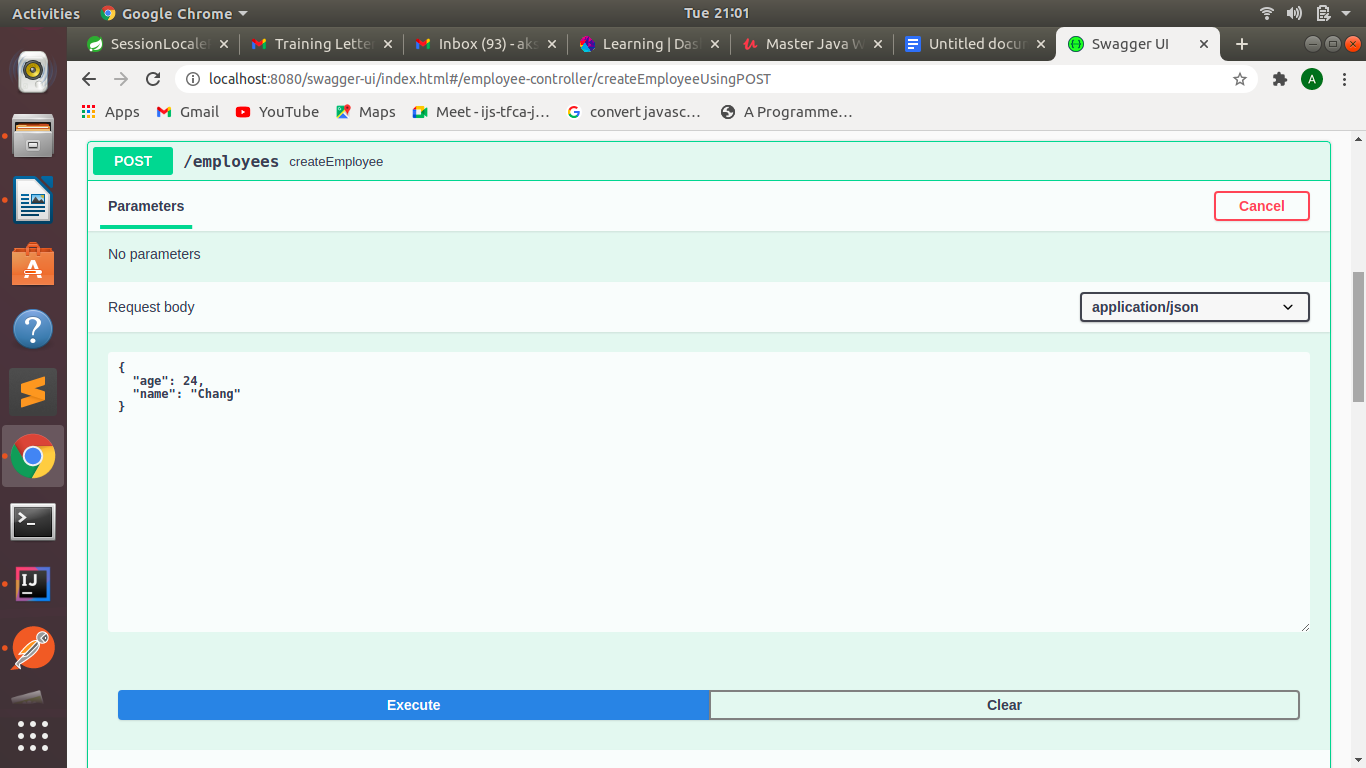


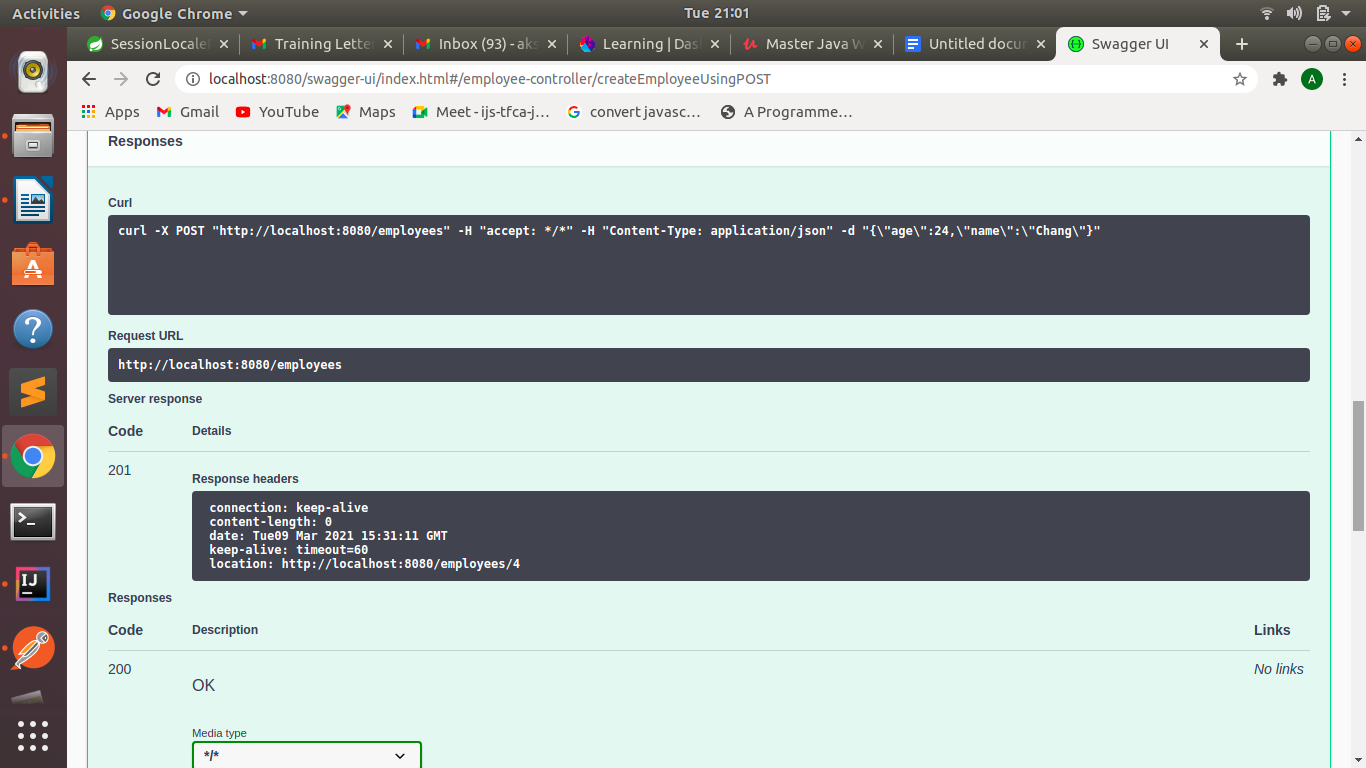
Get details of User using GET request.



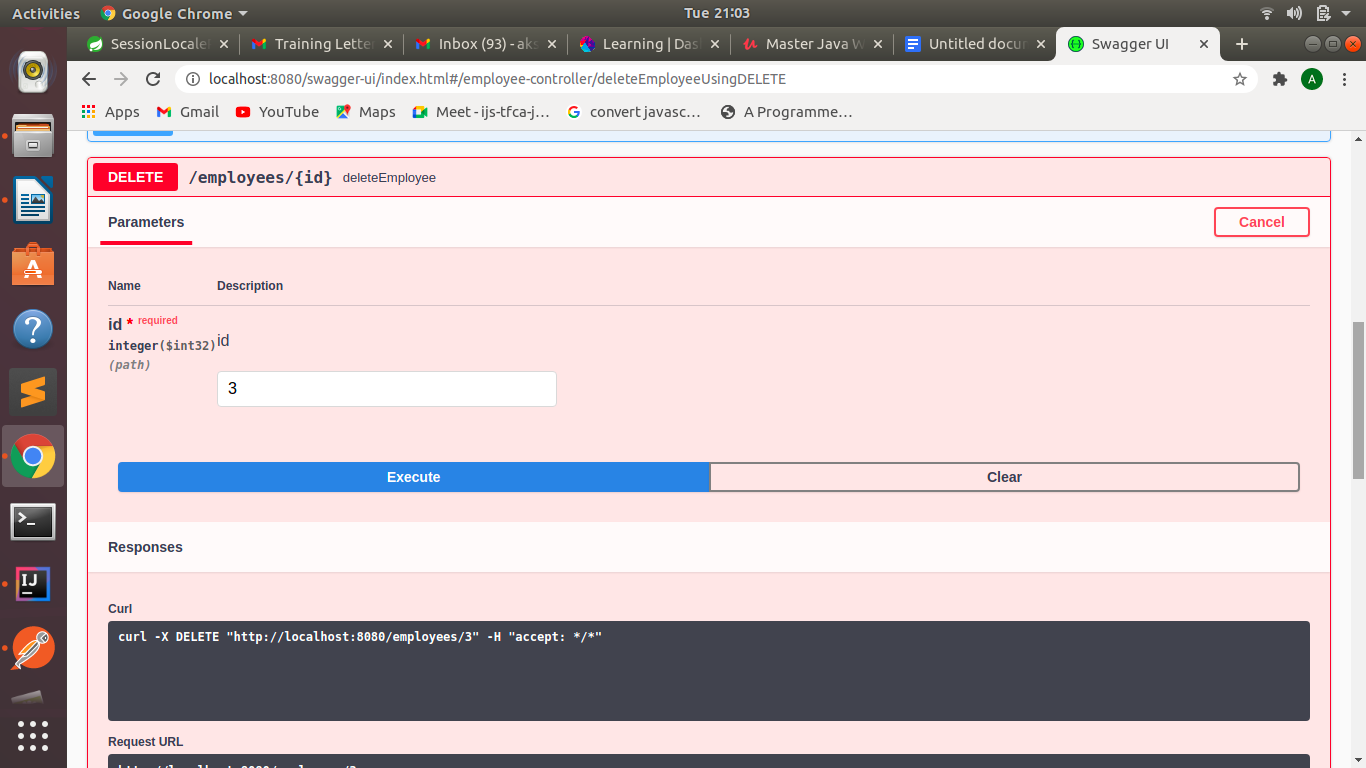


Save details of the user using POST request.

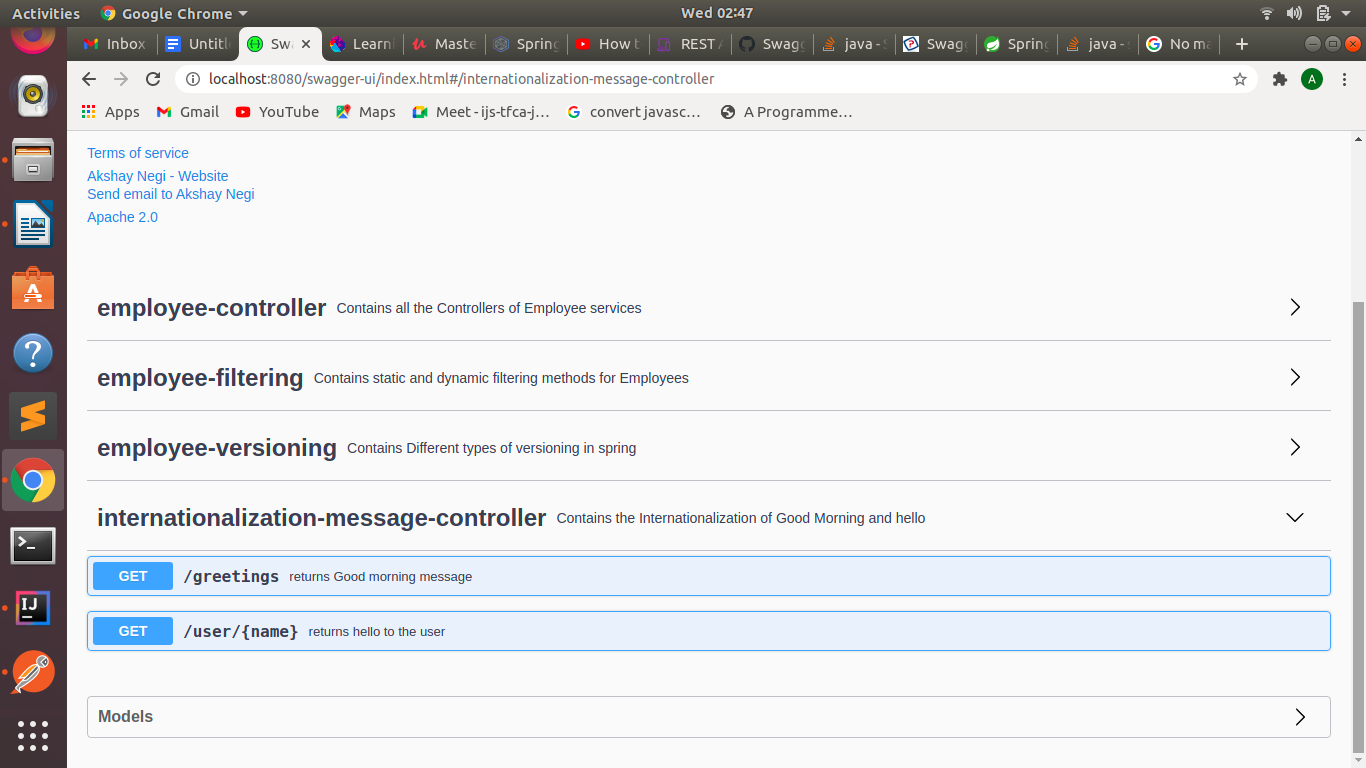
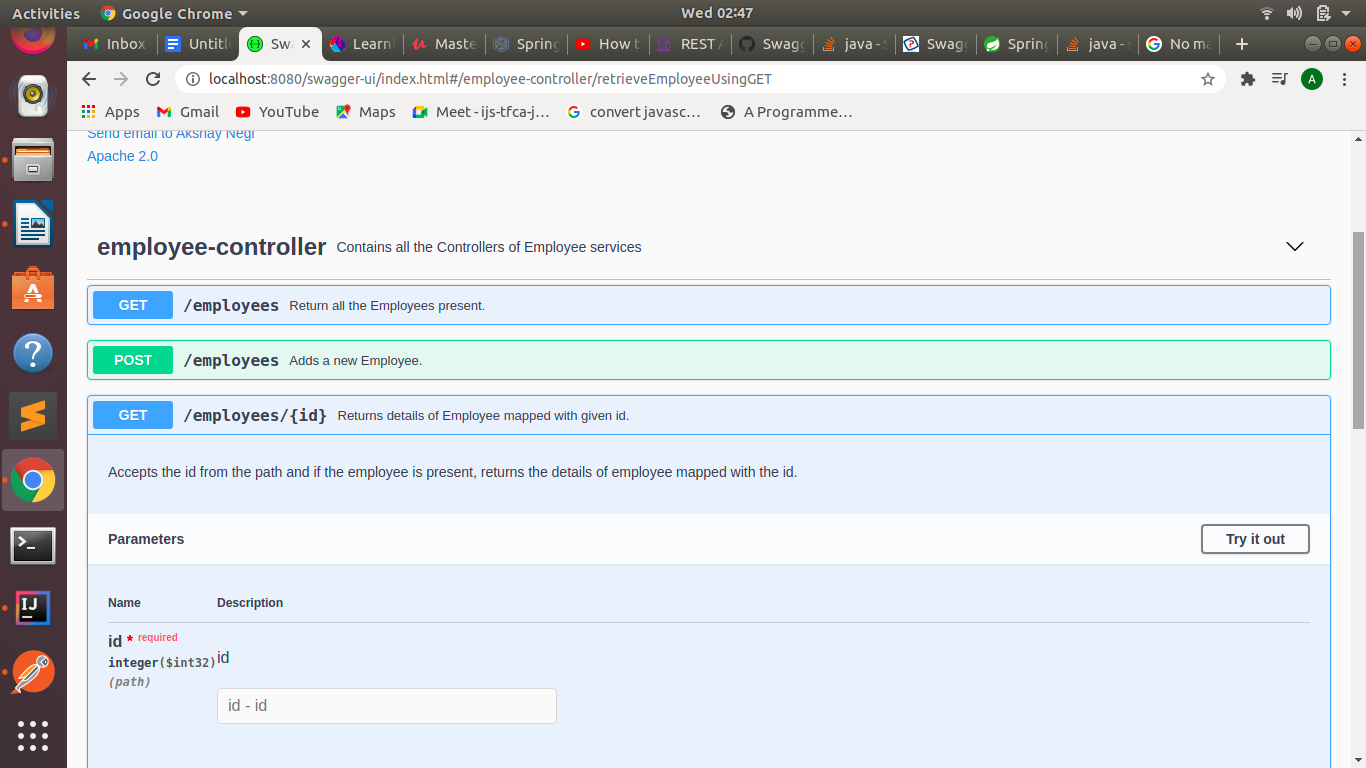
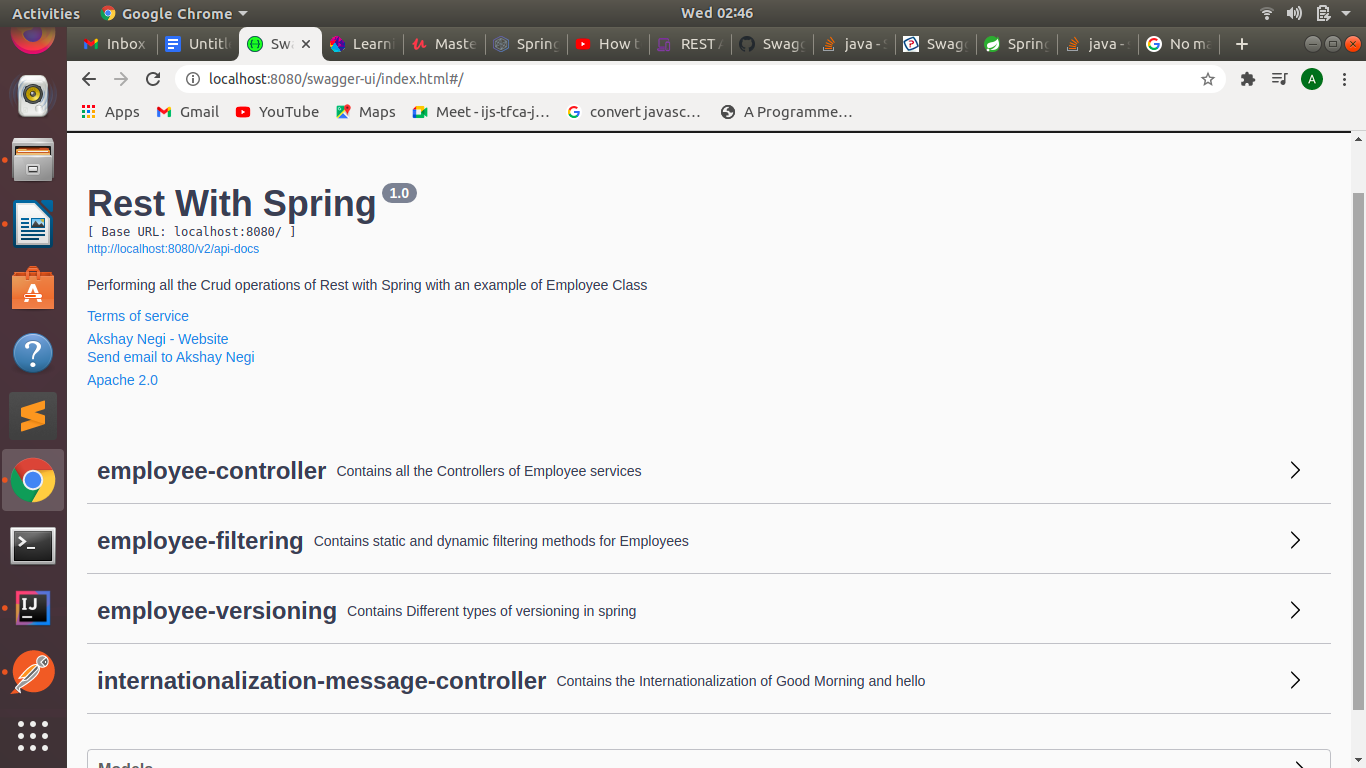




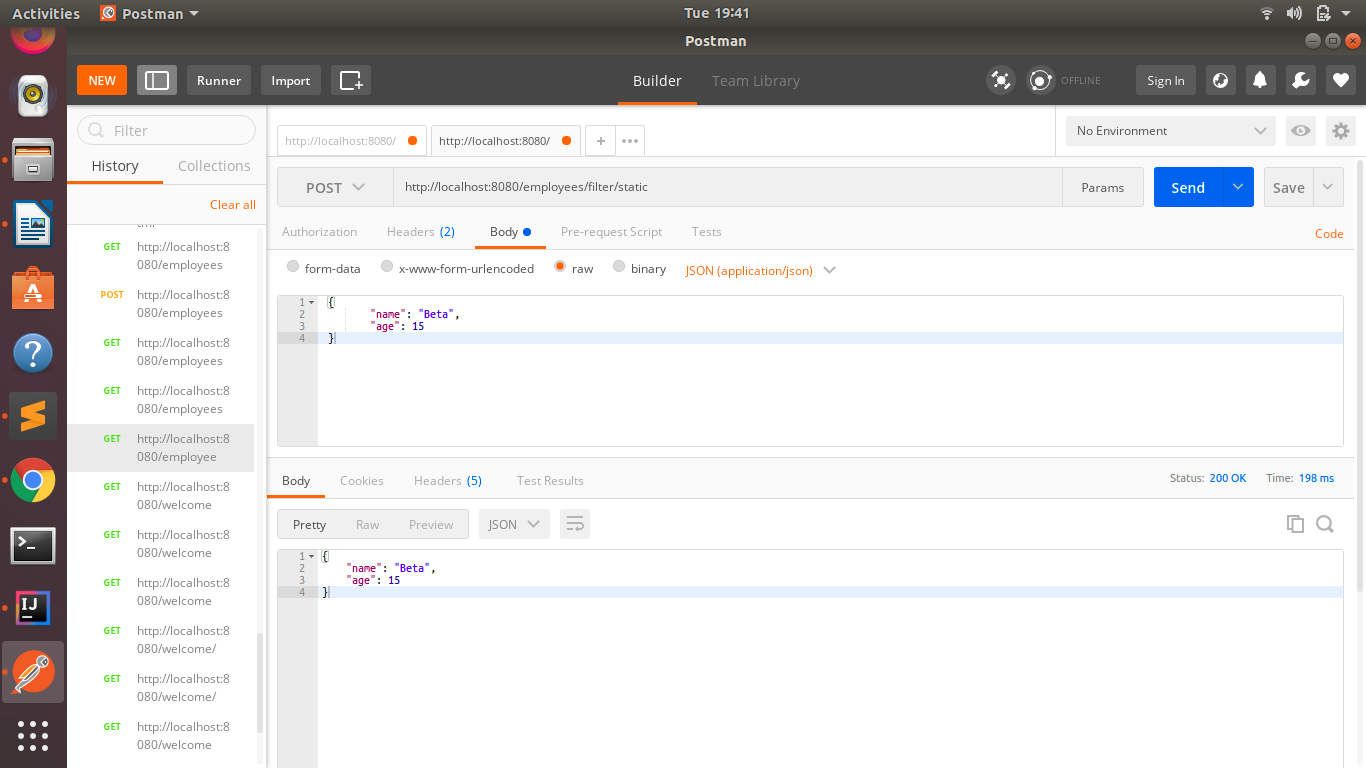
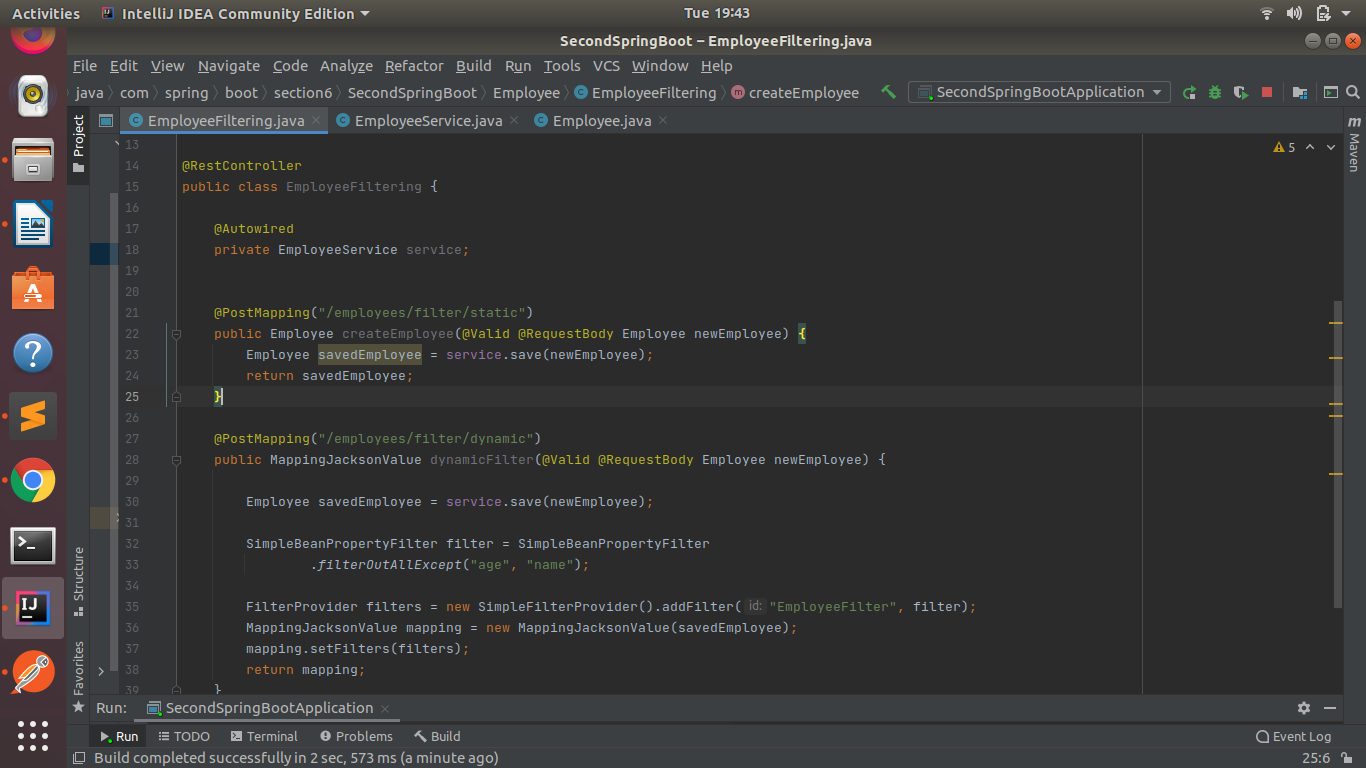
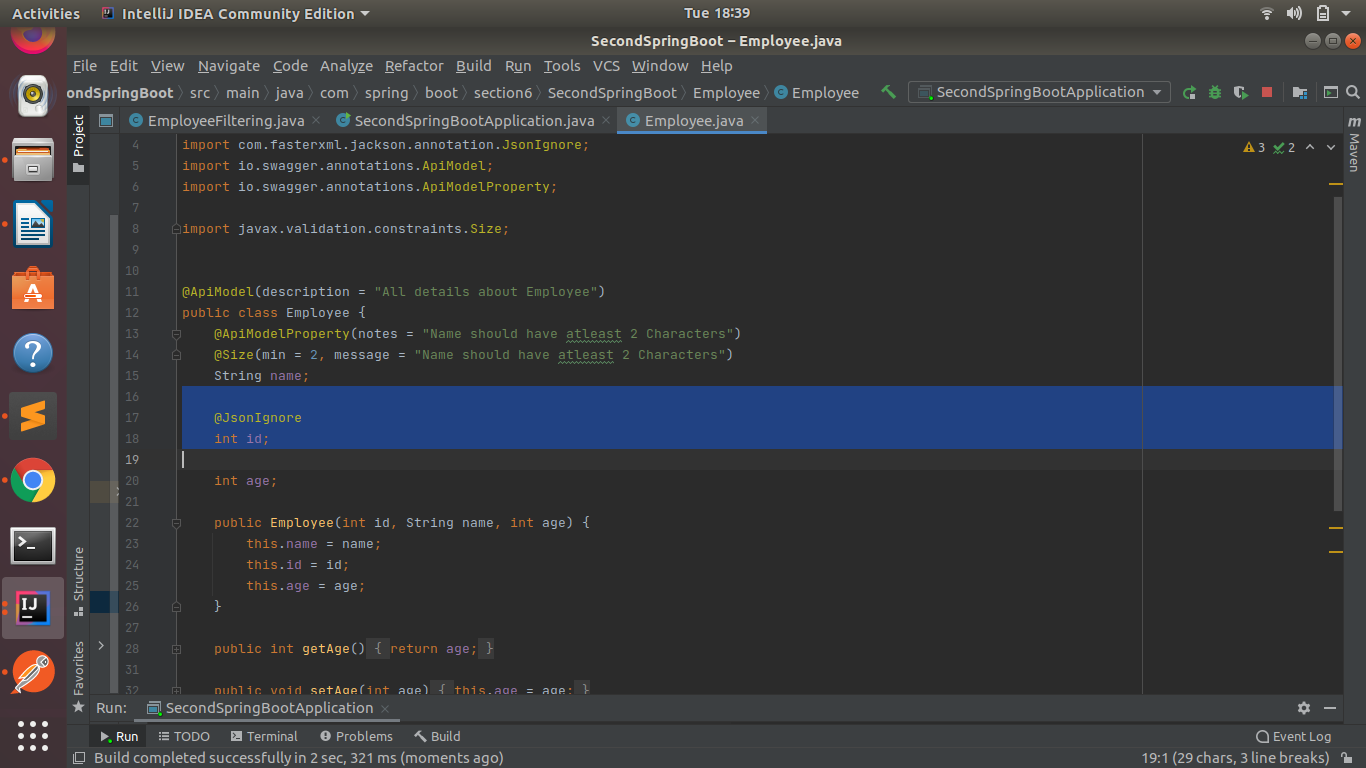
Delete a user using DELETE request.



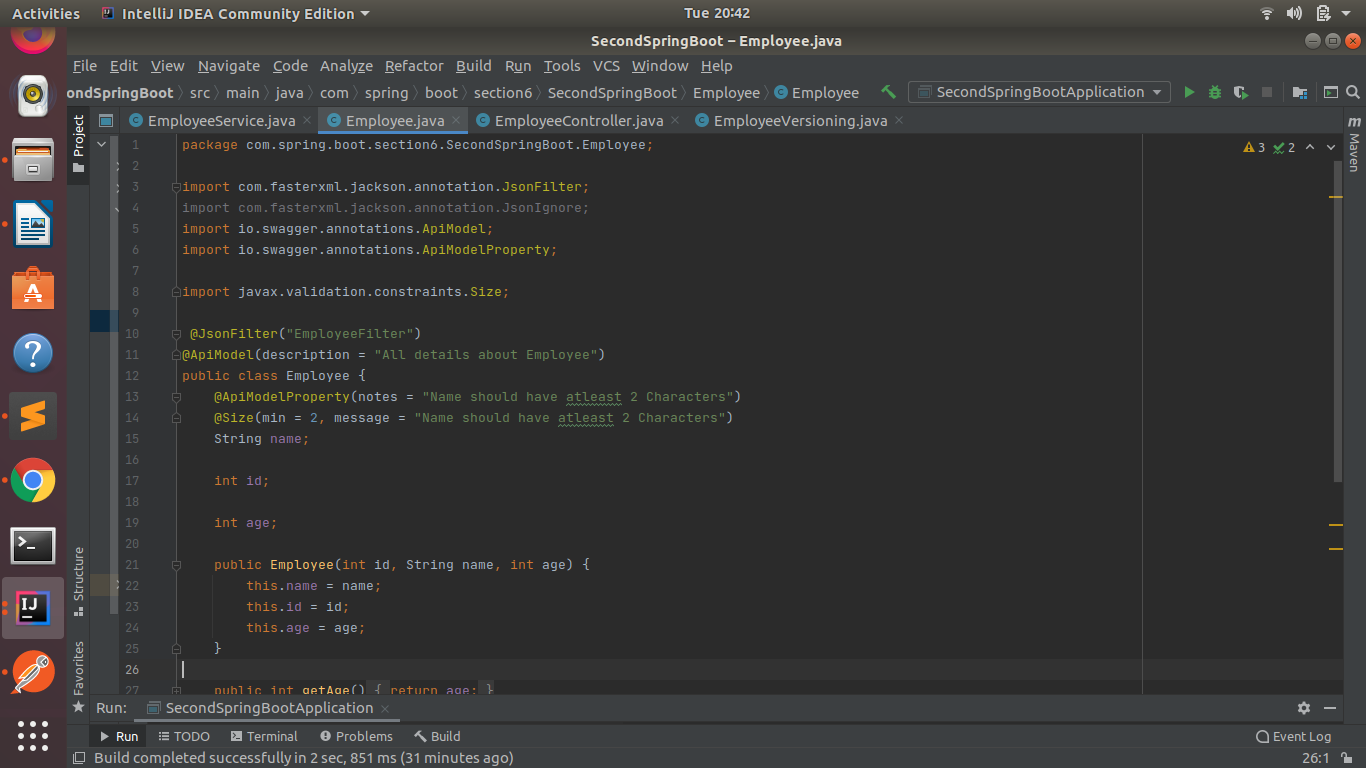


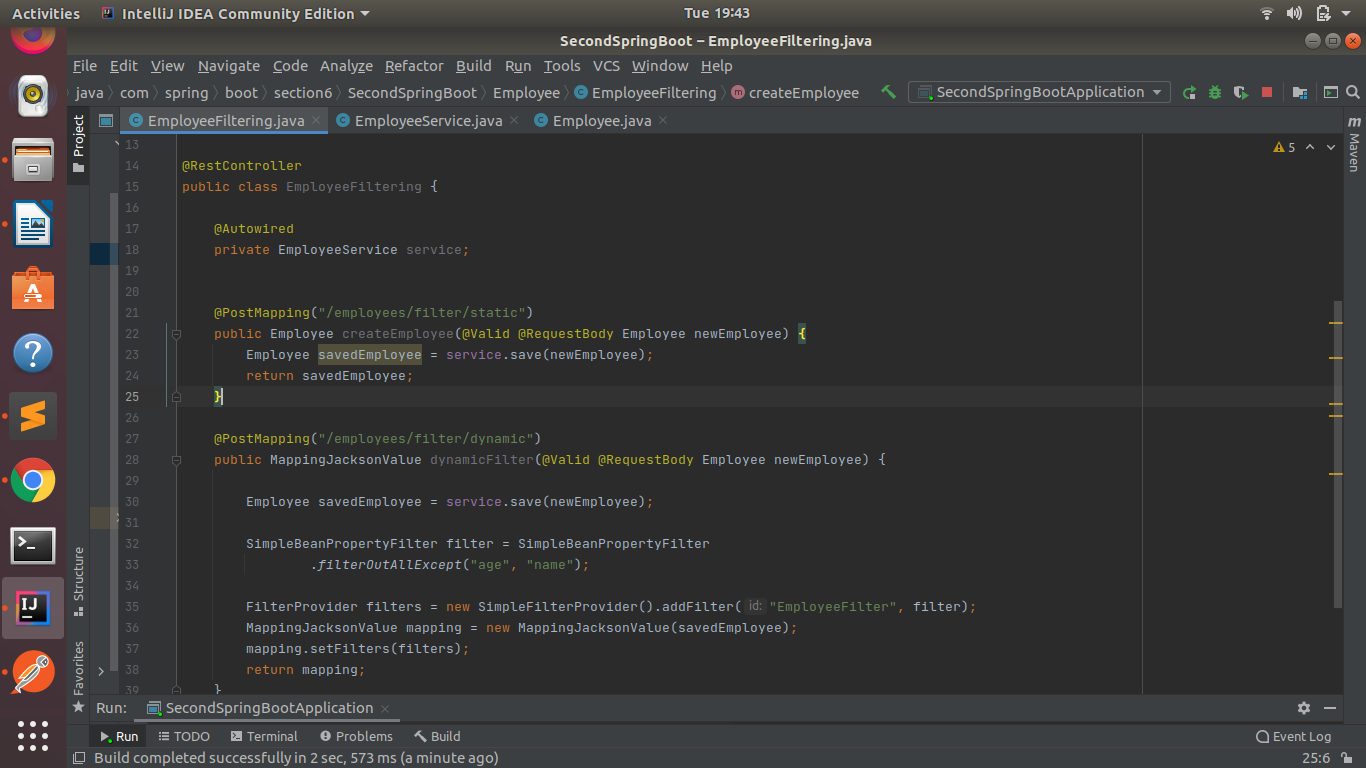
7. In swagger documentation, add the description of each class and URI so that in swagger UI the purpose of class and URI is clear.

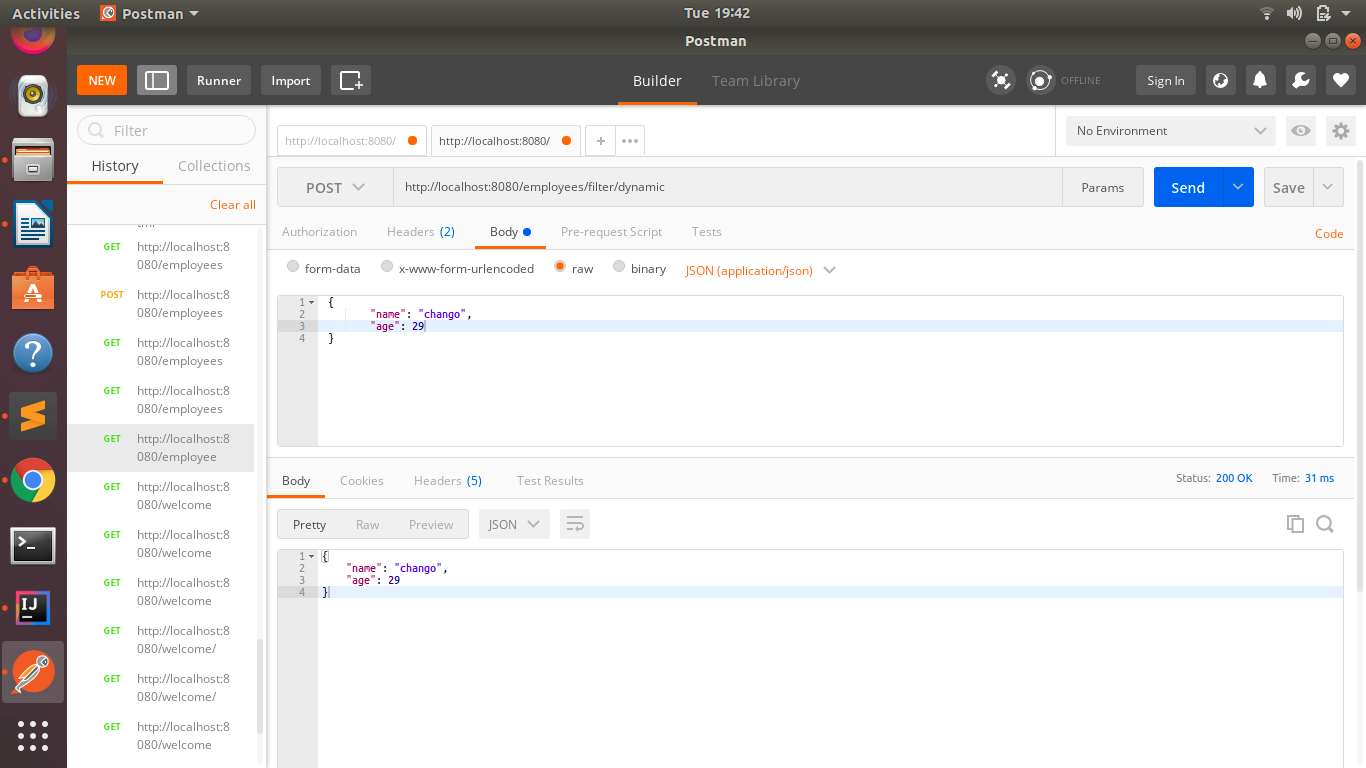
\***Static and Dynamic filtering**

8. Create API which saves details of User (along with the password) but on successfully saving returns only non-critical data. (Use static filtering)

9. Create another API that does the same by using Dynamic Filtering.



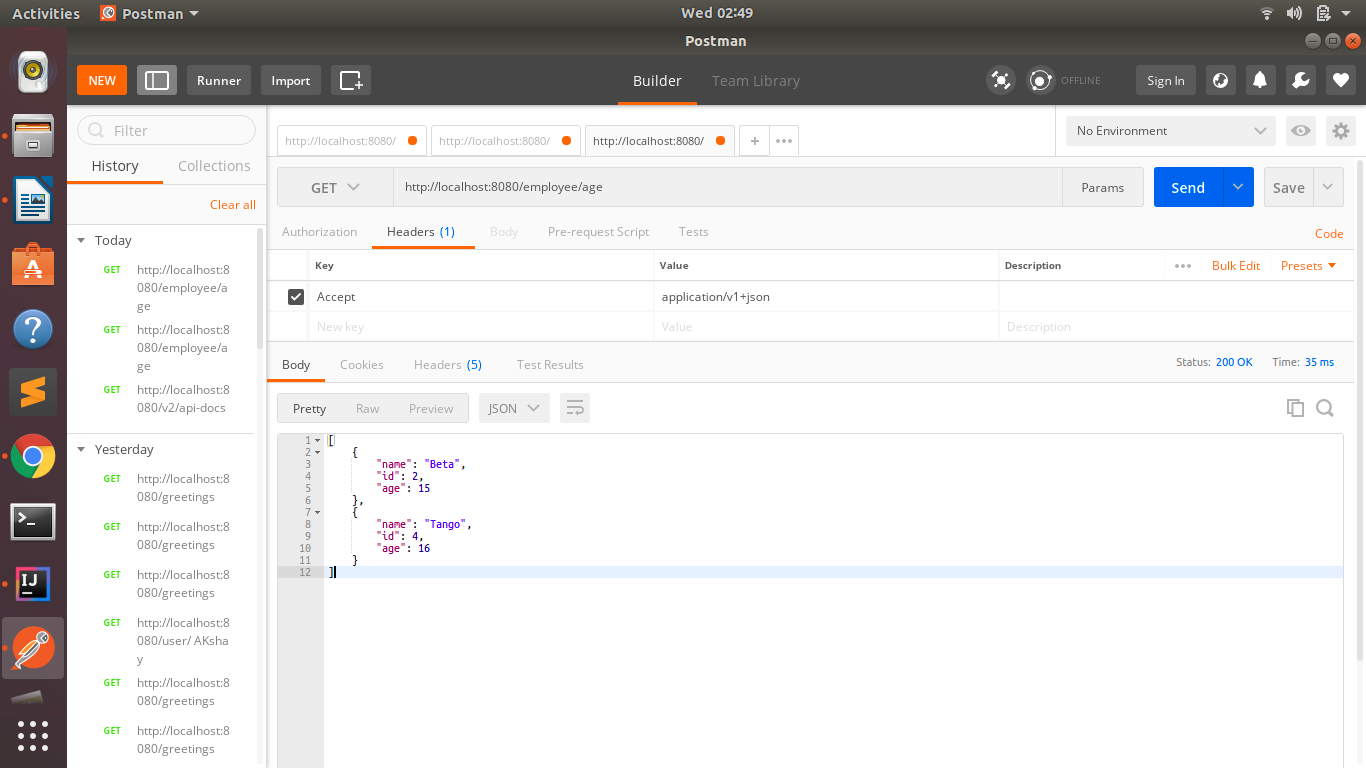
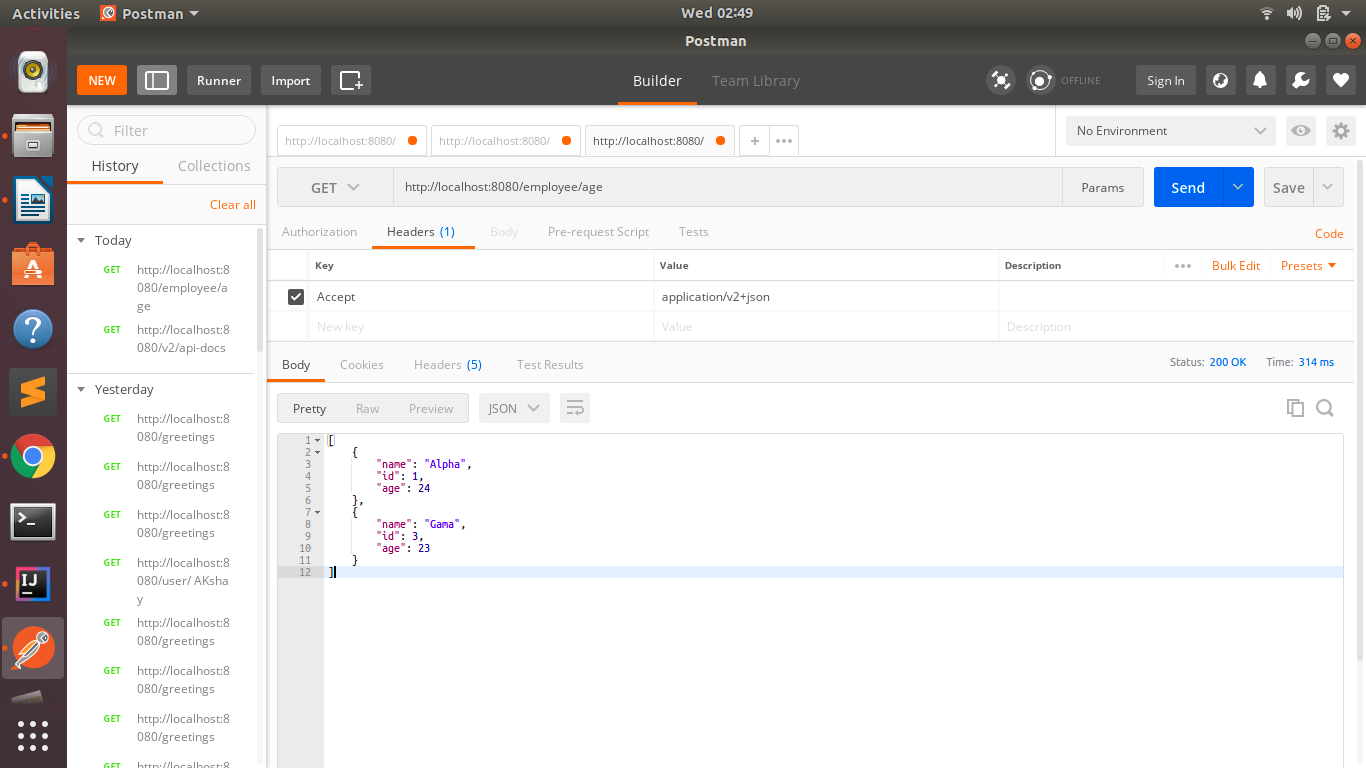
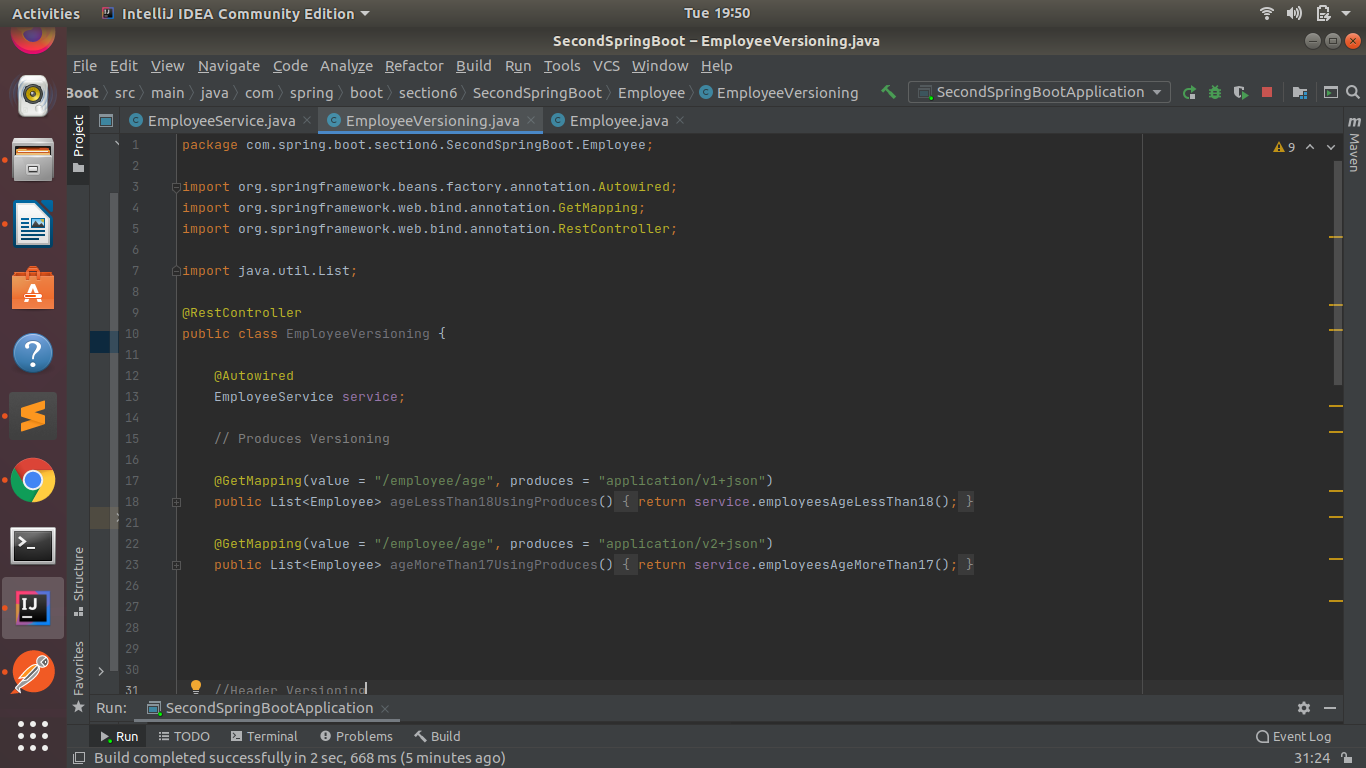


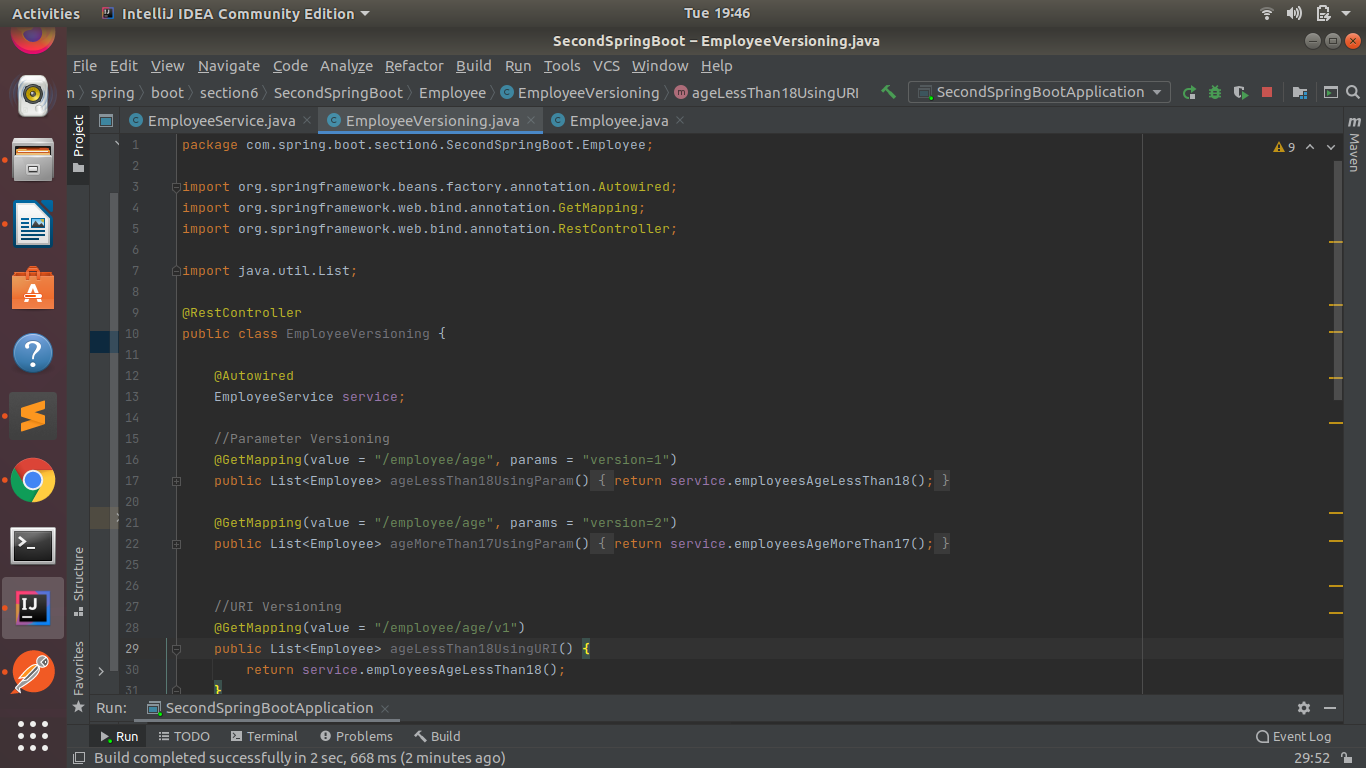


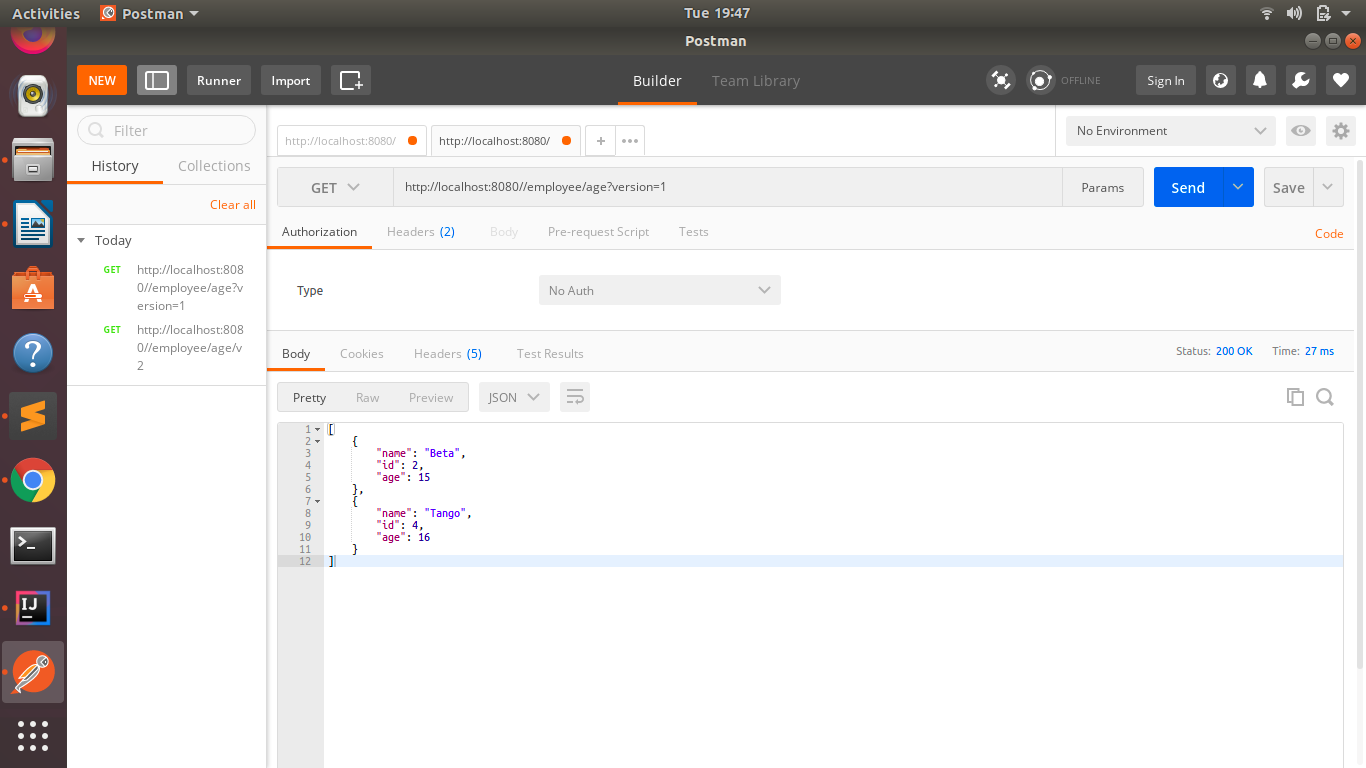
**\*Versioning Restful APIs**

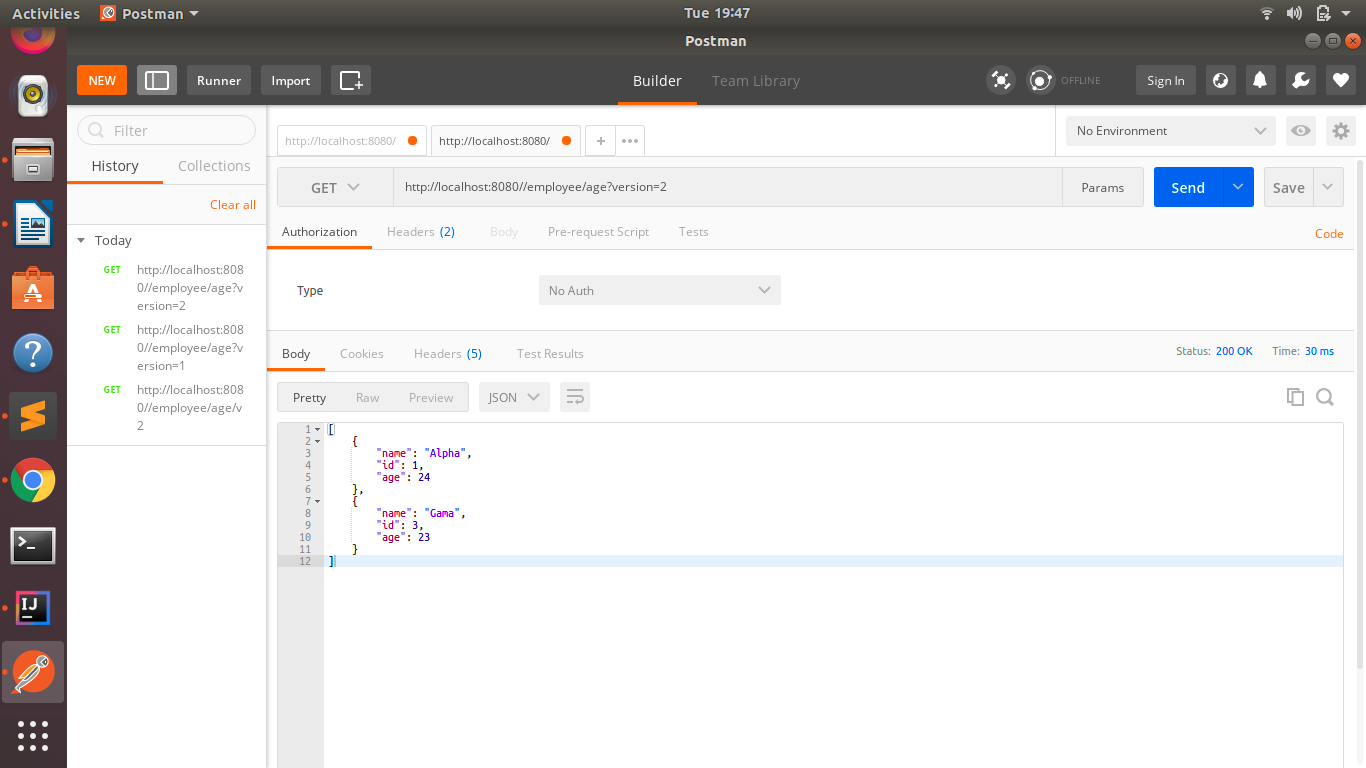
10. Create 2 API for showing user details. The first api should return only basic details of the user and the other API should return more/enhanced details of the user,

Now apply versioning using the following methods:

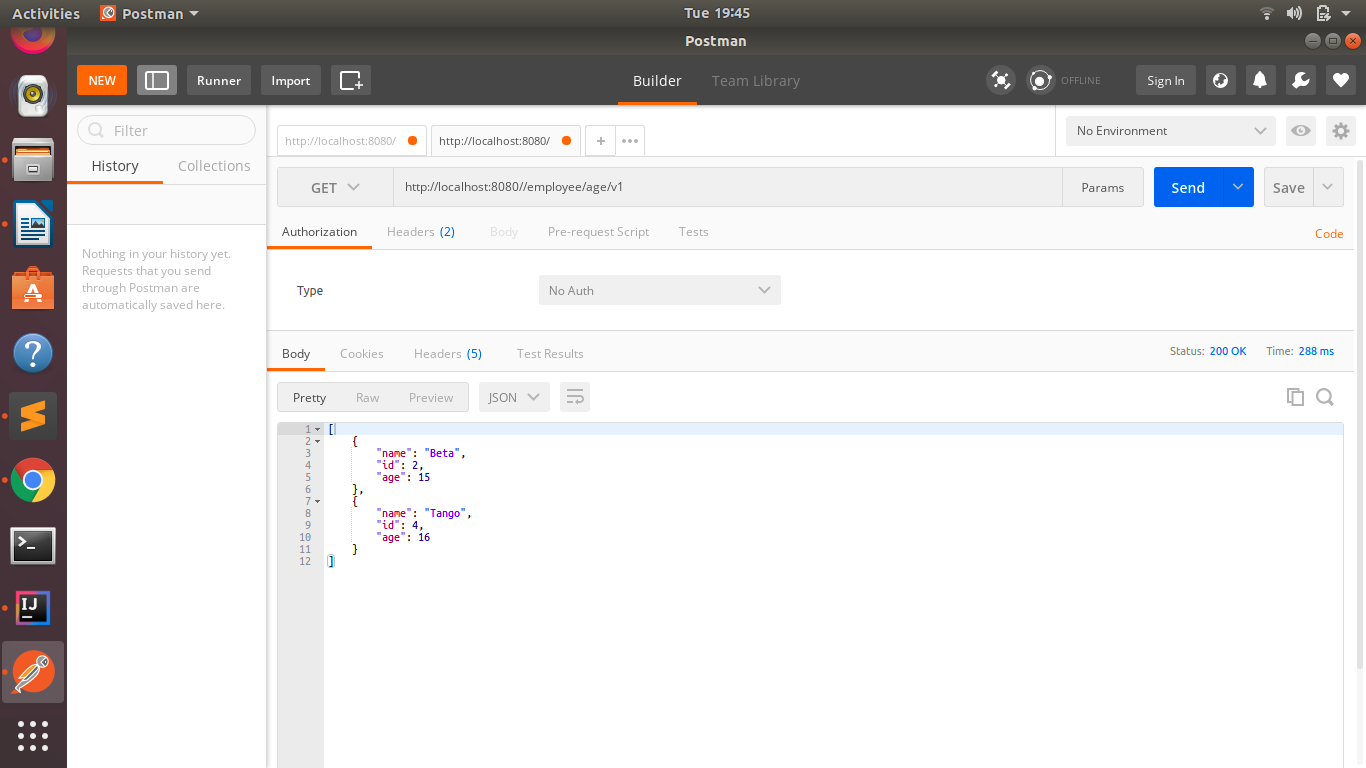
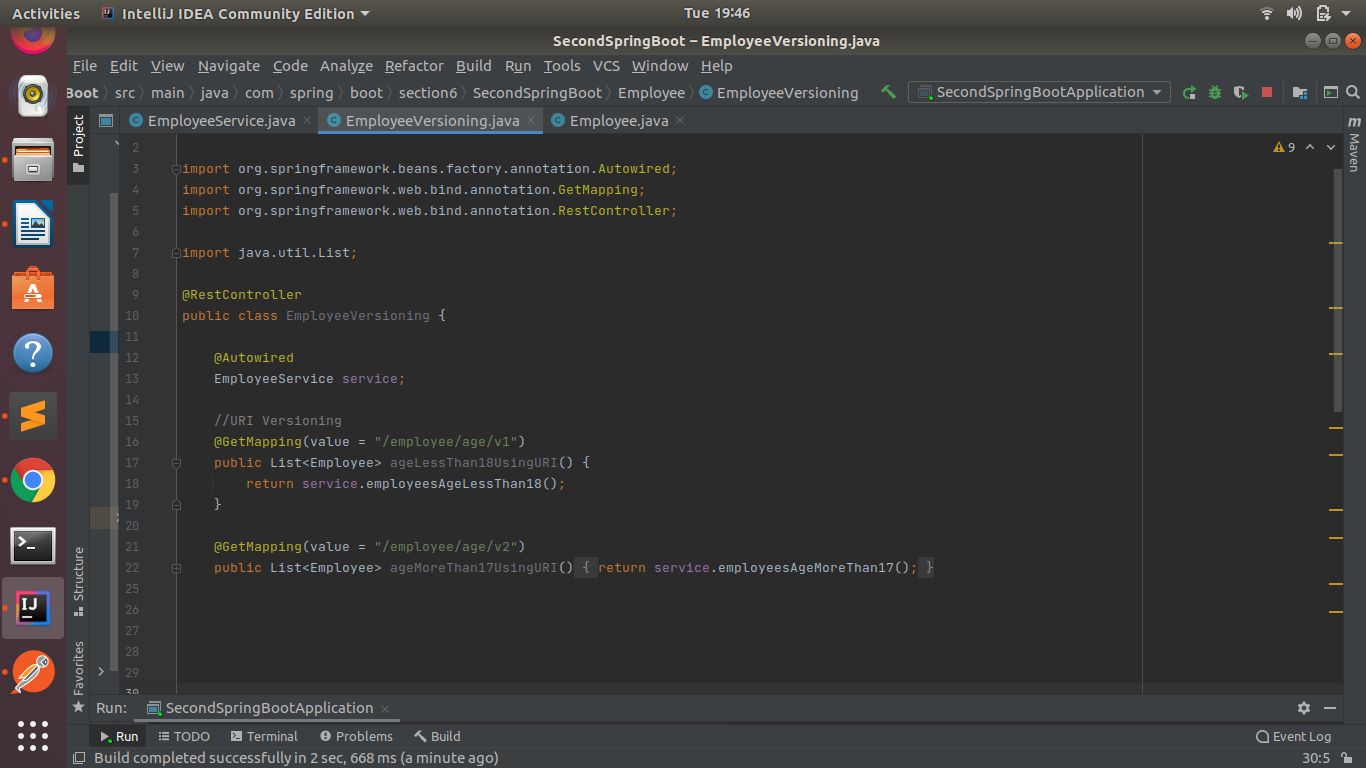
* MimeType Versioning
* Request Parameter versioning

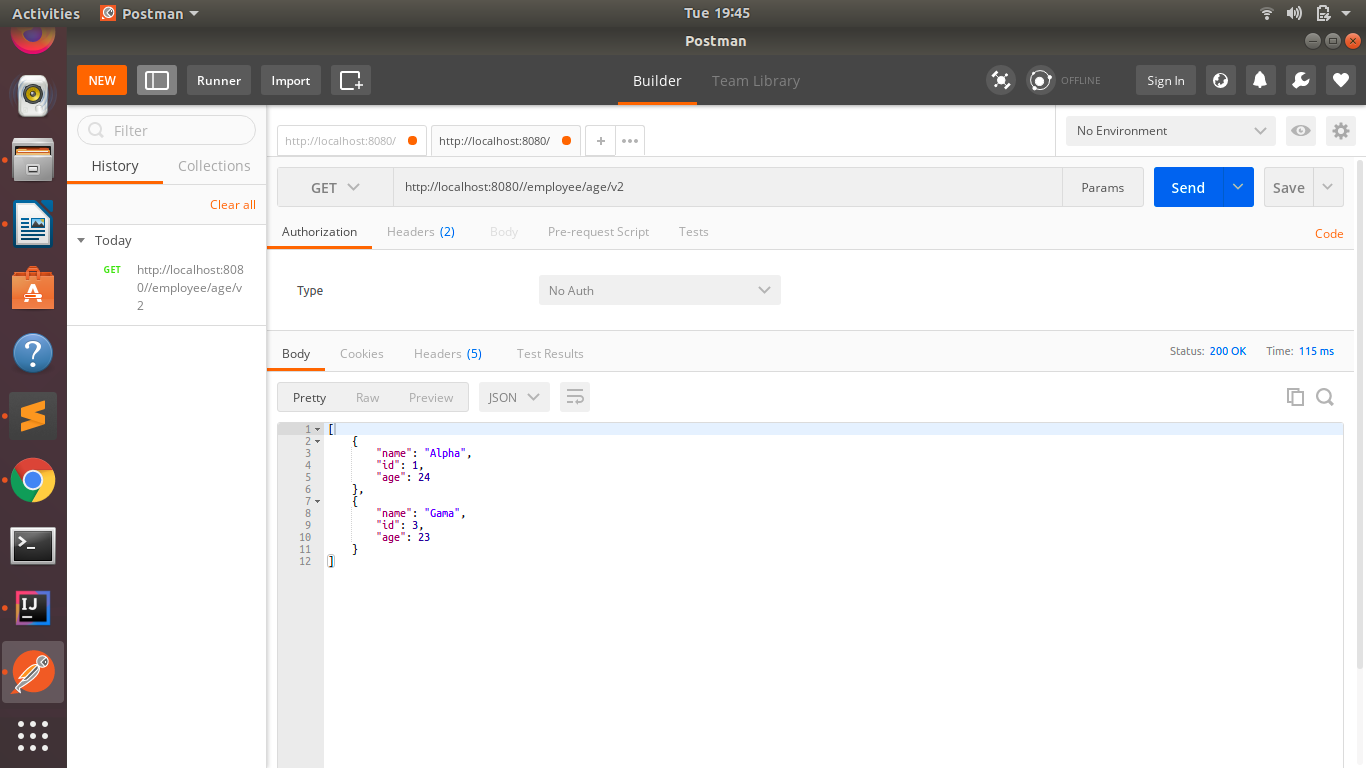




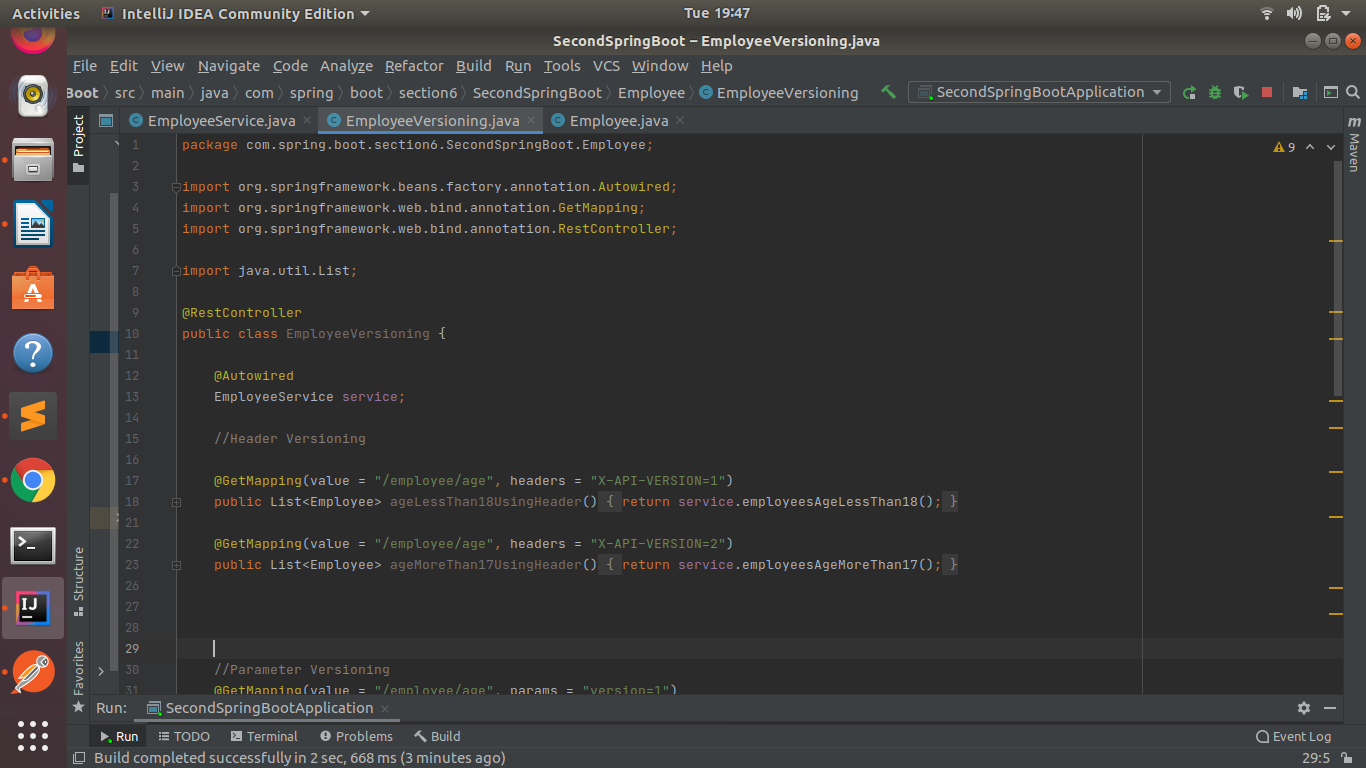


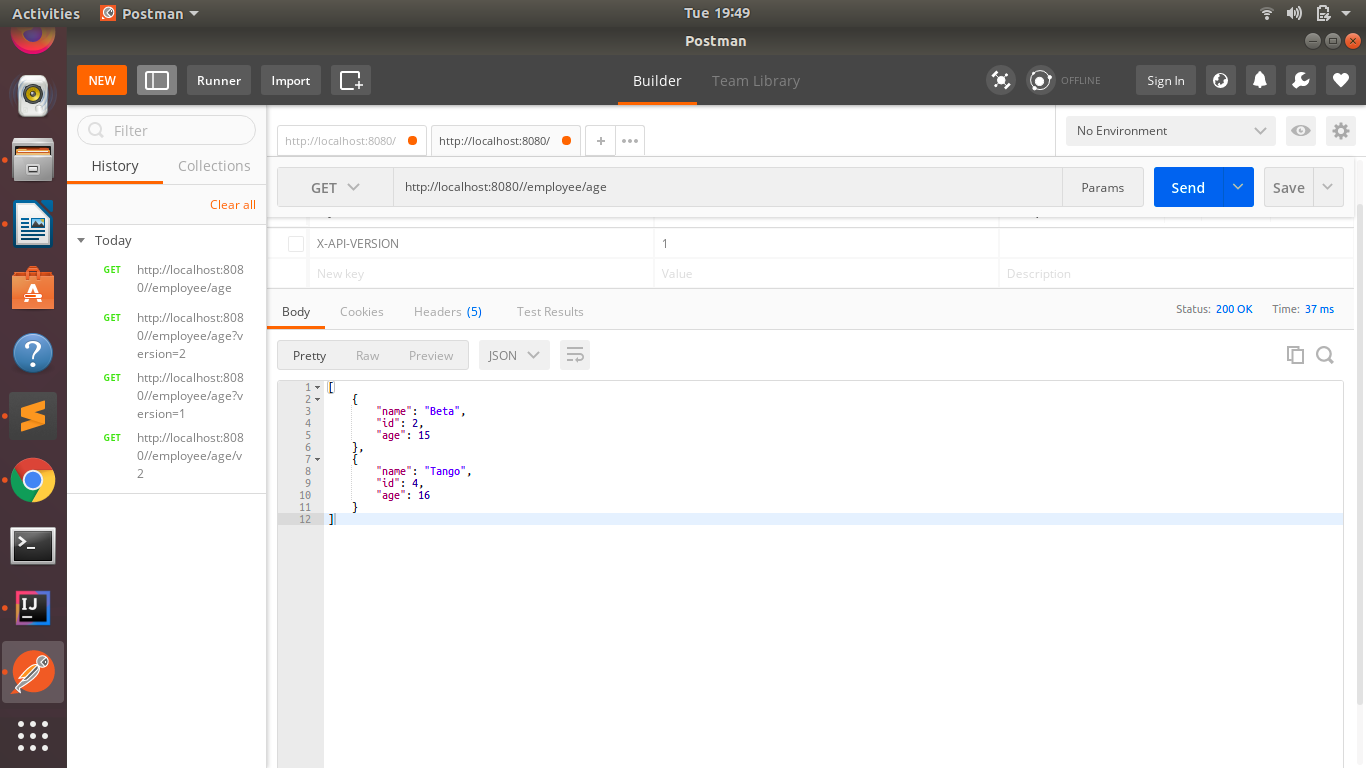
* URI versioning

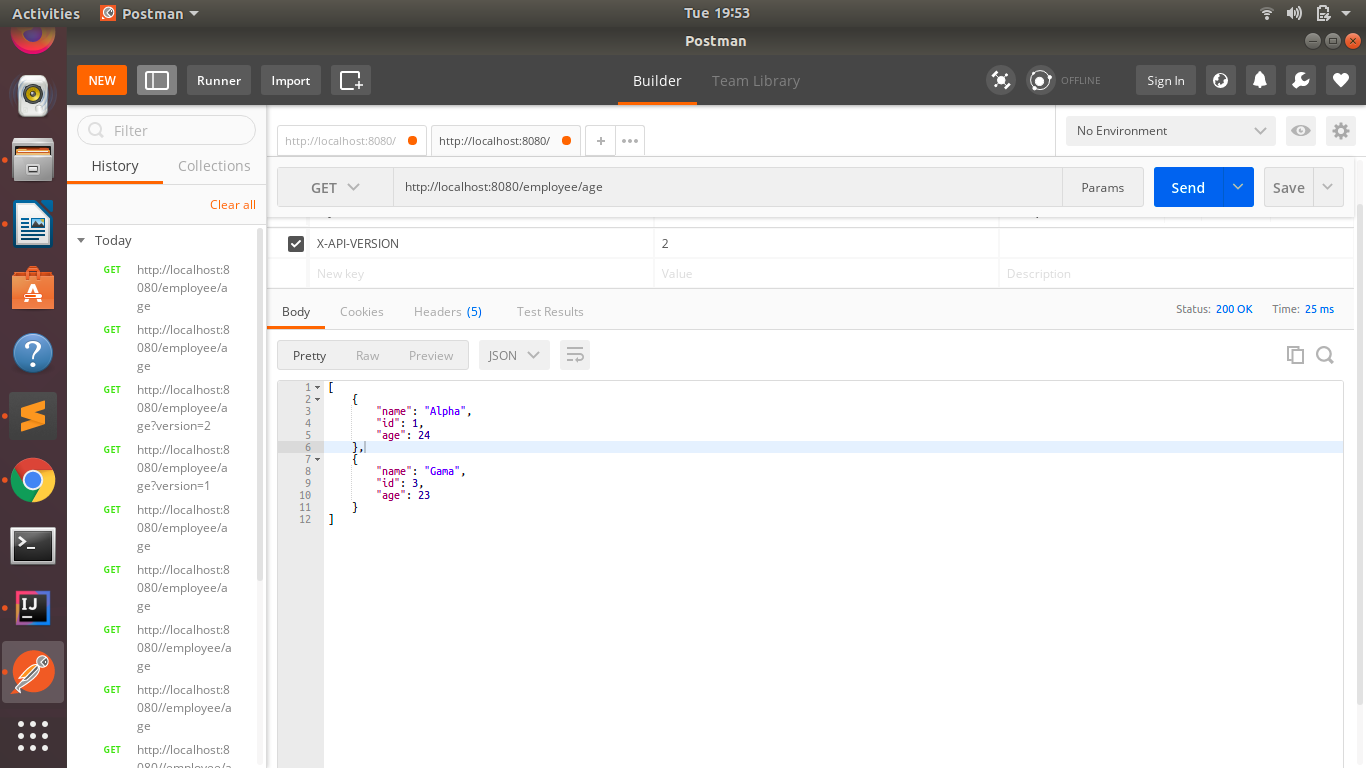




* Custom Header Versioning







**\*HATEOAS**

11. Configure hateoas with your springboot application. Create an api which returns User Details along with url to show all topics.

