**Master SpringBoot**

**03-Exception Handling:**

**Step-00: Create new git branch in local git repo and remote github repo**

* **Verify we are in master branch**
  + **git status**
* **Create new branch**
  + **git checkout –b 04-ExceptionHandling-ResponseStatusCodes**
* **Create new branch in remote github and setup upstream**
  + **git push –set-upstream origin 04-ExceptionHandling-ResponseStatusCodes**
* **Verify new branch in remote github GIT HUB & IDE Perspective**
  + [**https://github.com/stacksimplify/springboot-buildingblocks**](https://github.com/stacksimplify/springboot-buildingblocks)
  + [**http://github.com/jcoggins/springboot-buildingblocks**](http://github.com/jcoggins/springboot-buildingblocks)

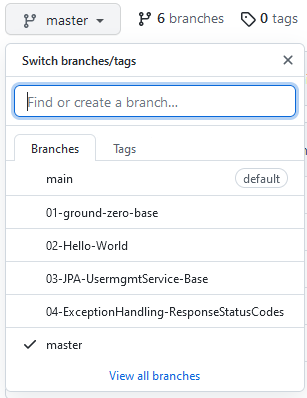
**From the command line, issue 3 git commands: git status, git checkout –b …, git push …**



**Verify in the IDE**



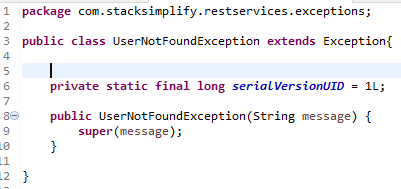
**Verify on GitHub**



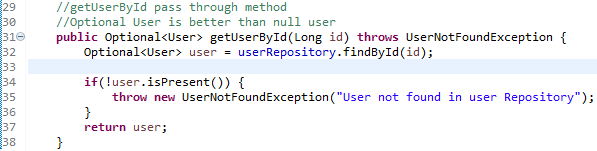
**Step-01: Implement “ResponseStatusException” for getUserById**

* **Custom Exceptions Layer:**
  + **Create “UserNotFoundException” which extends Exception**
* **Service Layer:**
  + **Update the getUserById method with throws Exception**
  + **Check for user and if not exists throw exception**
* **Controller Layer:**
  + **Update the getUserById method with try catch block**
  + **In catch block, implement “ResponseStatusException”**
* **Test using postman**
  + **Method: GET**
  + **URI:** [**http://localhost:8080/users/1001**](http://localhost:8080/users/1001)
* **Remove Trace in Exception**
  + **When using DevTools, “server.error.include-stacktrace” will be set to always**
  + **Change to never or on-trace-param**
* **Verify the response again.**
  + **Exception Message**
  + **HTTP Status Code**

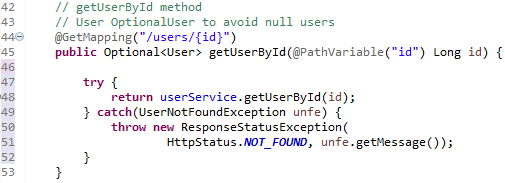
**Create UserNotFoundException custom exception**



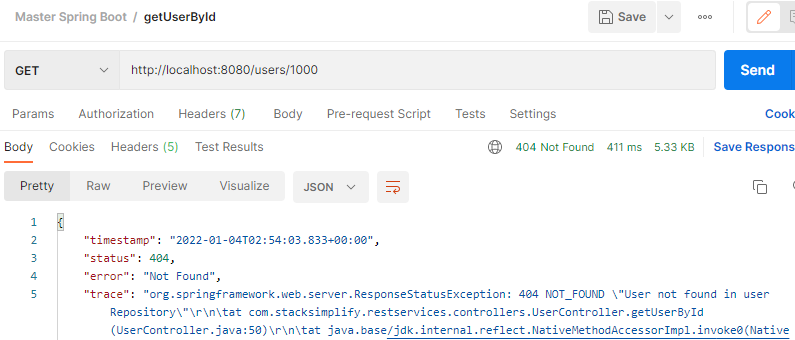
**Update the getUserById Method in the Service Layer:**



**Update the getUserById Method in the Controller Layer:**



**Test with POSTMAN (Restart the application first!)**

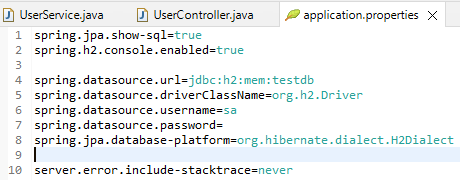


**Notice that the default Response:**

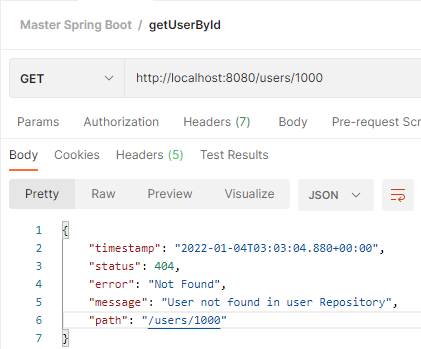
**status: 404, error: Not Found, msg:“User not found in user Repository”**

**Use DevTools to customize the Postman output with an application.properties entry:**

**server.error.include-stacktrace=never**



**Retest Postman**

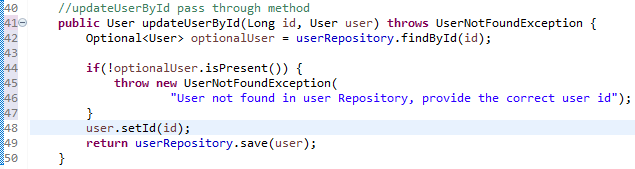


**It works!!**

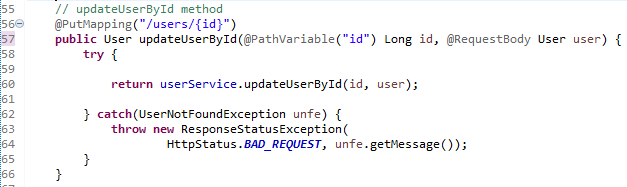
**Step-02: Implement “ResponseStatusException” for updateUserById**

* **Service Layer:**
  + **Update the updateUserById method with throws Exception**
  + **Check for user and if not exists throw exception**
* **Controller Layer:**
  + **Update the updateUserById method with try catch block**
  + **In catch block, implement “ResponseStatusException”**
* **Test using postman**
  + **Method: PUT**
  + **URI:** [**http://localhost:8080/users/1001**](http://localhost:8080/users/1001)
  + **Request Body:**
  + **Verify the response.**
    - **Exception Message**
    - **HTTP Status Code**

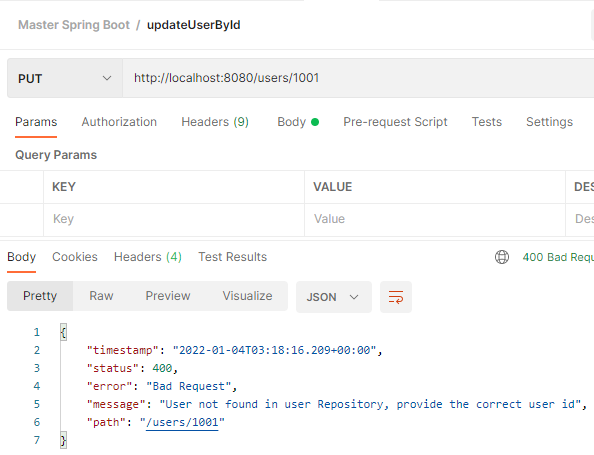
**Update the updateUserById Method in the Service Layer:**



**Update the updateUserById Method in the Controller Layer:**



**Test with POSTMAN (Restart the application first!)**

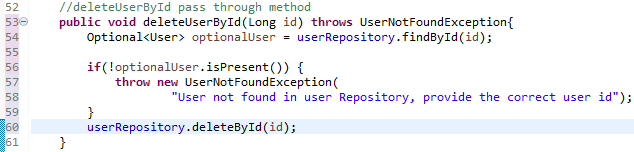


**Step-03: Implement “ResponseStatusException” for deleteUserById at Service Layer**

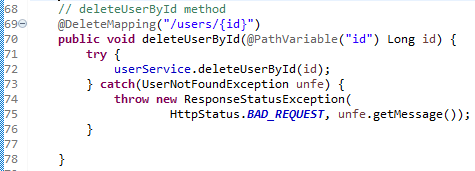
* **Service Layer:**
  + **Check for user and if not exists throw ResponseException**
* **Controller Layer:**
  + **No Changes**
* **Test using postman**
  + **Method: DELETE**
  + **URI:** [**http://localhost:8080/users/1001**](http://localhost:8080/users/1001)
  + **Request Body:**
  + **Verify the response.**
    - **Exception Message**
    - **HTTP Status Code**

**Production Pattern:**

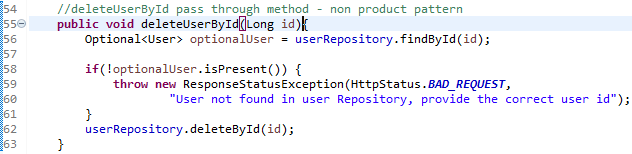
**Update the deleteUserById Method in the Service Layer:**



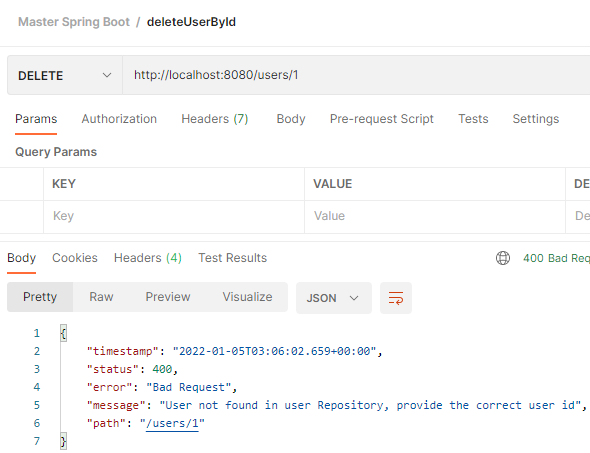
**Update the deleteUserById Method in the Controller Layer:**



**NonProduction Pattern: Service Layer only changes!**



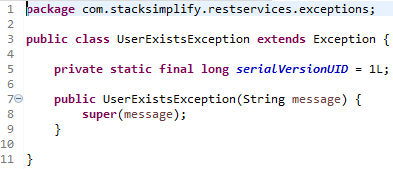
**Test with POSTMAN (Restart the application first!) Both output the same thing!**



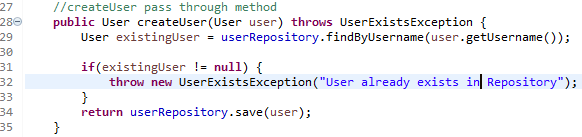
**Step-04: Implement “ResponseStatusException” for createUser**

* **Exception Layer**
  + **Create the “UserExistsException” class**
* **Service Layer:**
  + **Update the createUser method with throws Exceptions**
  + **Verify if user exists based on username (Username is unique constraint)**
  + **If not null, throw exception**
* **Controller Layer:**
  + **Update the createUser method with try catch block**
  + **In catch block, implement “ResponseStatusException”**
* **Test using postman**
  + **Method: POST**
  + **URI: http://localhost:8080/users**
  + **Request Body:**
  + **Verify the response.**
    - **Exception Message**
    - **HTTP Status Code**

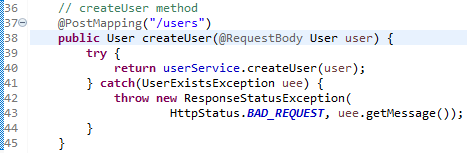
**Create UserExistsException class**



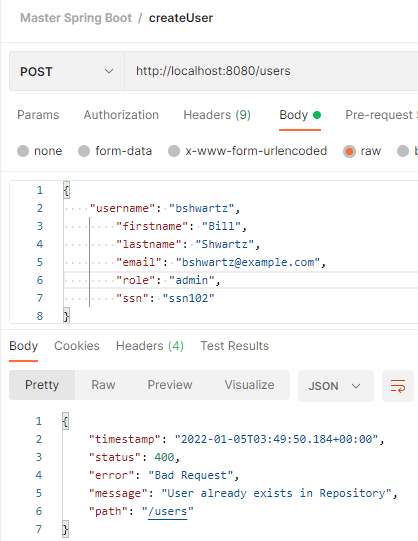
**Update the createUser Method in the Service Layer:**



**Update the createUser Method in the Controller Layer:**



**Test with POSTMAN**

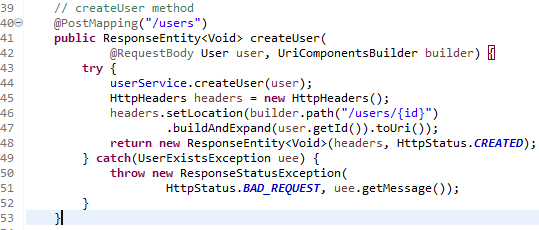


**Step-05: For createUser Method, implement HTTP Status code 201 & Location Header path**

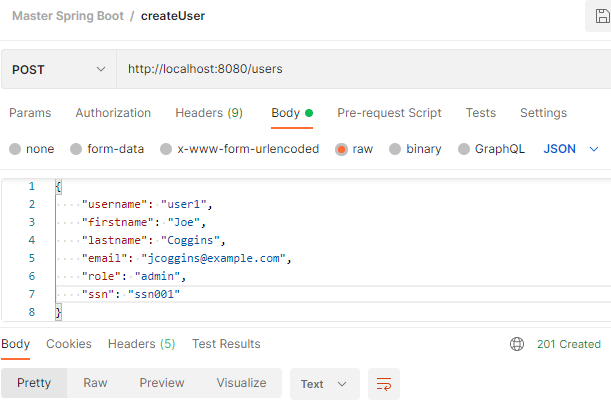
**– Http Status Code 201 & Location Header with User URI path**

* **Controller Layer:**
  + **Implement Service to return HTTP Status code 201**
  + **Implement Service to return Location Header as user path**
* **Test using postman**
  + **Method: POST**
  + **URI: http://localhost:8080/users**
  + **Request Body:**
  + **Verify the response.**
    - **HTTP Status Code**
    - **Location Header in Response Headers**

**Implement the Controller Layer**



**Test with POSTMAN:**



**Notice 201 status code is returned!!**

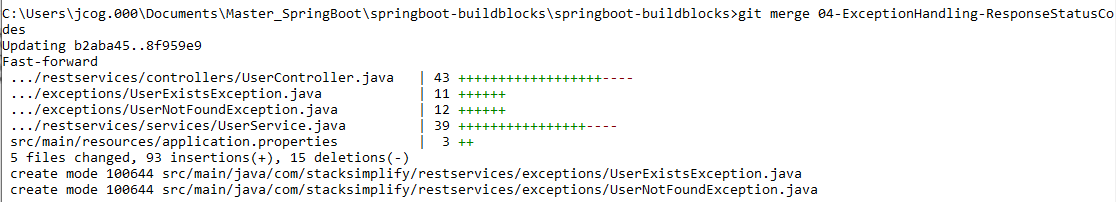
**git stage with git add . git commit git push to remote branch**

**(git push should have been – git push –set-upstream 04-ExceptionHandling-ResponseStatus)**

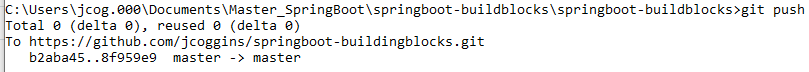


**git checkout master and git merge**





**git push to master**



**Now check the GitHub repository:**

