**Master SpringBoot**

**09-SpringBoot DTO:**

**Step-00: Introduction**

**Step-01: New GIT branch (using IDE)**

* **Git Branch name: 10-01-SpringBoot-DTOS-ModelMapper**
* **Create new local branch**

**Step-02: Implement DTOS for User Entity using ModelMapper**

* **Update pom.xml**

**<dependency>**

**<groupId>org.modelmapper</groupId>**

**<artifactId>modelmapper</artifactId>**

**<version>2.3.5</version>**

**</dependency>**

**Step-03: Define ModelMapper Bean**

* **Config Layer**
  + **Create a config package**
  + **Create AppConfig class**
  + **Define ModelMapper bean in configuration class**

**Step-04: Create a DTO class with name as UserMmDTO**

* **DTO Layer**
  + **Create UserMmDto with fields id, username**

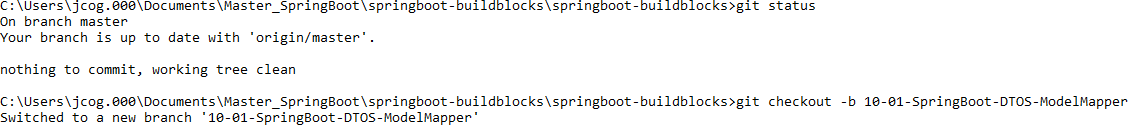
**Step-05: Create getUserBtoById method with Entity to DTO Conversion logic**

* **Controller layer**
  + **Create new controller named UserModelMapperController**
  + **Annotate with @RestController**
  + **Annotate with @RequestMapping(“modelmapper/users”)**
  + **Create getUserDtoById method**
    - **GET /{id}**
  + **Implement ModelMapper conversion logic**
* **Test using Postman**

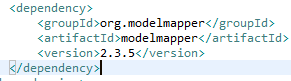
**Step-06: Commit Code (using IDE)**

* **Commit code**
* **Push branch to Remote repo**

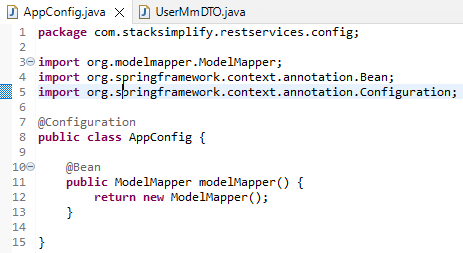
**git status (Make sure on local master) git checkout –b 10-01-SpringBoot-DTOS-ModelMapper**



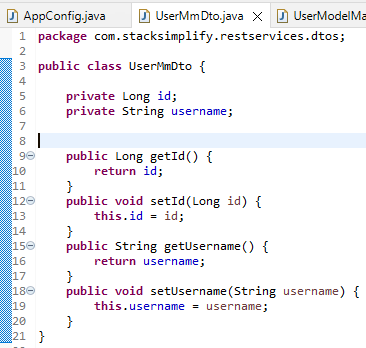
**Add dependency to pom.xml:**



**Create the AppConfig file:**



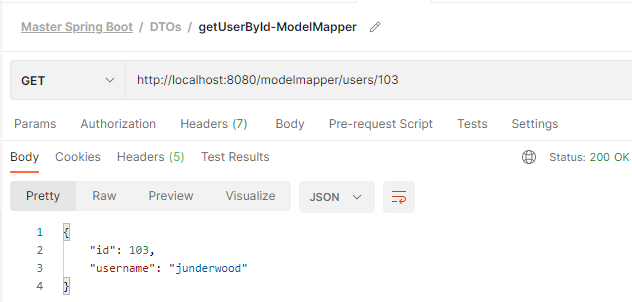
**Create the modelMapper, UserModelDto class:**



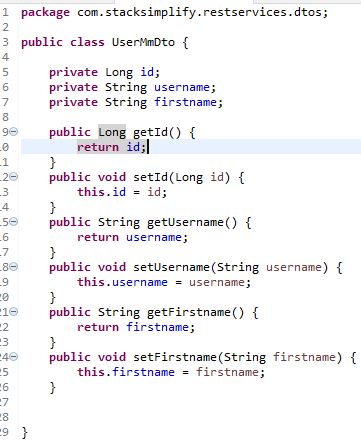
**Create the UserModelMapperController:**



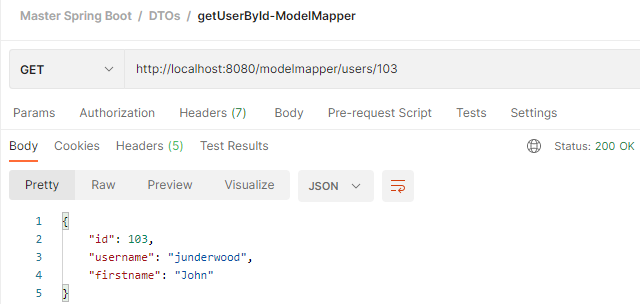
**Test using postman:**



**Let us add firstname to DTO:**

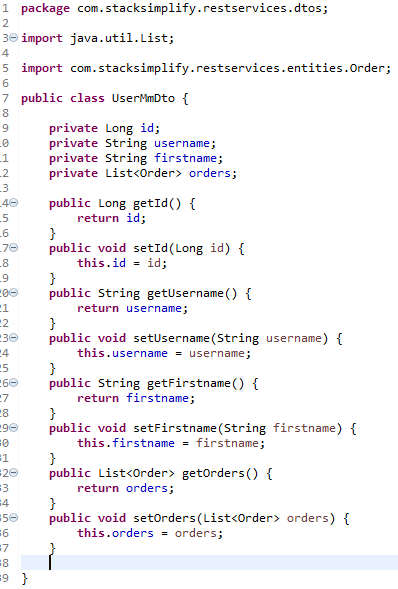


**Test using postman:**

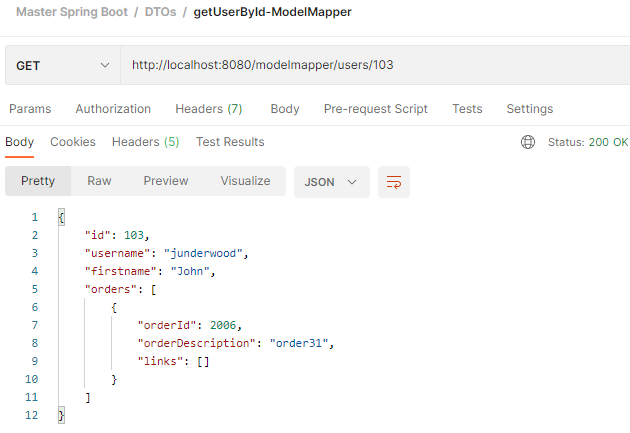


**So, only the DTO needs testing for any changes**

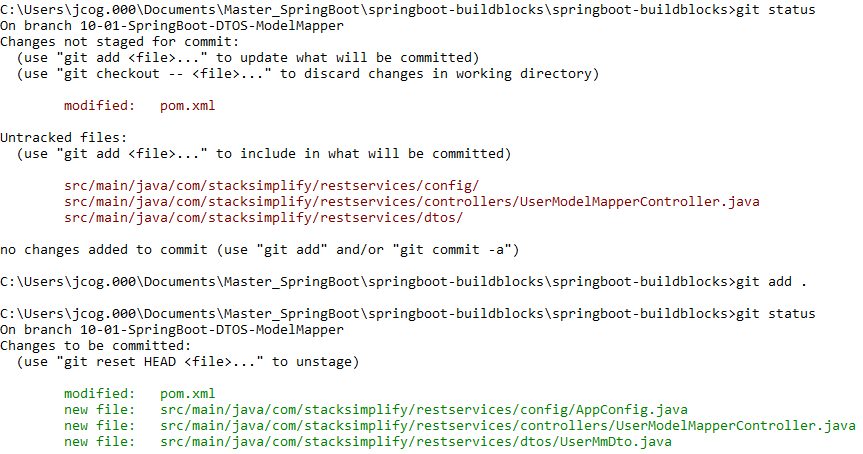
**Let us add orders:**



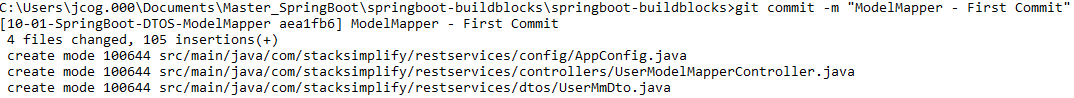
**Test using postman:**



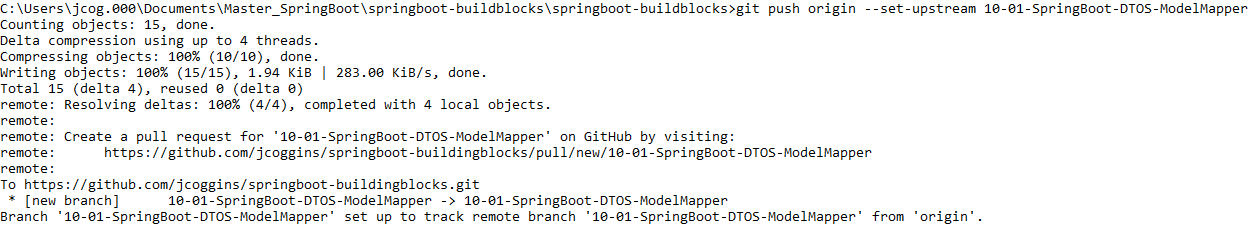
**Now, git stage and update the branch before creating a new branch:**



**git commit –m “ModelMapper – First Commit”**



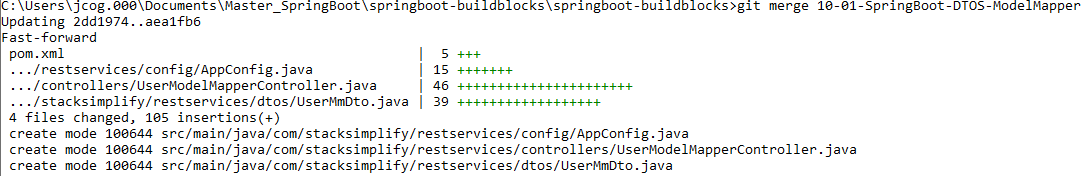
**git push origin –set-upstream 10-01-SpringBoot-DTOS-ModelMapper**



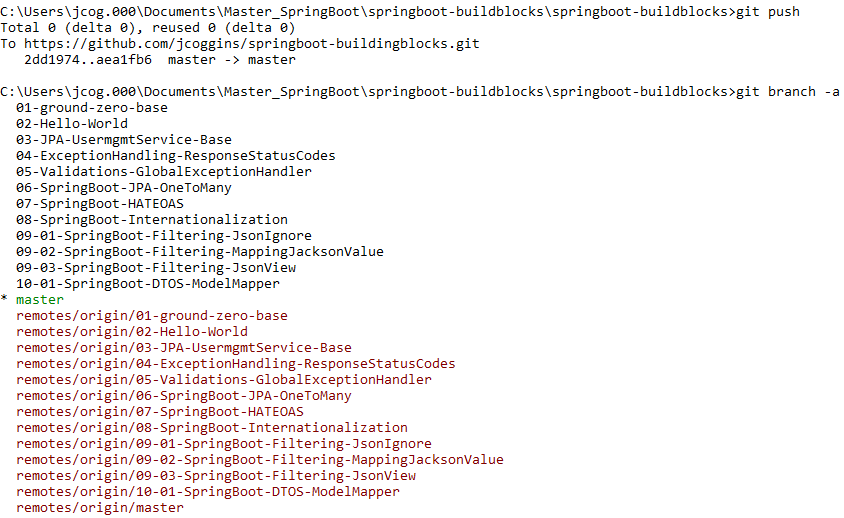
**git checkout master**



**git merge 10-01-SpringBoot-DTOS-ModelMapper**



**git push git branch –a**



**Step-00: Introduction MapStruct**

**Step-01: New GIT branch (using IDE)**

* **git Branch name: 10-02-SpringBoot-DTOS-MapStruct**
* **Create new local branch**

**Step-02: Update pom.xml with necessary dependencies for MapStruct**

* **pom.xml**
* **Change#1: Add Properties**

**<properties>**

**<org.mapstruct.version>1.3.0.Final</org.mapstruct.version>**

**<org.apache.maven.plugins.version>3.8.1</org.apache.maven.plugins.version>**

**</properties>**

* **Change#2: Add mapstruct Dependency**

**<dependency>**

**<groupId>org.mapstruct</groupId>**

**<artifactId>mapstruct-jdk8</artifactId>**

**<version>${org.mapstruct.version}</version>**

**</dependency>**

* **Change#3: Add MapStruct Processor Plugin**

**<plugin>**

**<groupId>org.apache.maven.plugins</groupId>**

**<artifactId>maven-compiler-plugin</artifactId>**

**<version>${org.apache.maven.plugins.version}</version>**

**<configuration>**

**<source>${java.version}</source>**

**<target>${java.version}</target>**

**<annotationProcessorPaths>**

**<path>**

**<groupId>org.mapstruct</groupId>**

**<artifactId>mapstruct-processor</artifactId>**

**<version>${org.mapstruct.version}</version>**

**</path>**

**</annotationProcessorPaths>**

**</configuration>**

**</plugin>**

**Step-03: Create UserMsDTO class required for MapStruct Implementation**

* **DTO Layer**
  + **Create UserDto in dtos package**
  + **Define fields that need to be exposed via UserMsDto**
  + **Generate No Argument and Field Constructors**
  + **Generate Getters and Setters**

**Step-04: Create the MapStruct Mapper Interface**

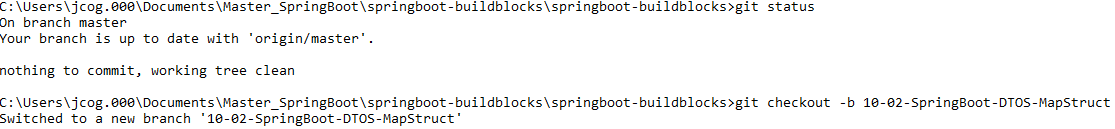
* **Mapper Layer**
  + **Create an interface with methods for mapping between objects(User to UserMsDto).**
  + **Add a @Mapper annotation to the interface**
  + **Configure MapStruct to user Spring’s dependency injection.**
  + **Add a ‘componentModel’ attribute with the value ‘spring’ to the @Mapper annotation**
  + **Create methods**
    - **userToUserDto**
      * **Input Object: User**
      * **Output Object: UserMsDto**
      * **@Mapping(source = “email”, target = “emailaddress”)**
    - **usersToUserDtos**
      * **Input Object: List<User>**
      * **Output Object: List<User>**

**Step-05: Create the REST services by calling methods defined in MapStruct Mapper.**

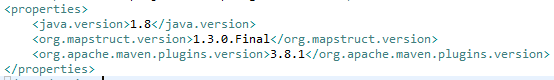
* **Controller Layer**
  + **Create new Controller named UserMapStructController**
  + **Copy getUserById and getAllUsers methods from UserController**
  + **getAllUserMsDtos**
    - **Implement this method to return DTO converted by MapStruct**
    - **Test using postman**
  + **getUserByIdMsDtos**
    - **Implement this method to return DTO converted by MapStruct**
    - **Test using postman**

**Step-06: Commit & Push code via IDE**

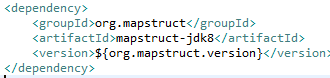
**git status (Make sure on local master) git checkout –b 10-01-SpringBoot-DTOS-ModelMapper**



**Add properties to pom.xml:**



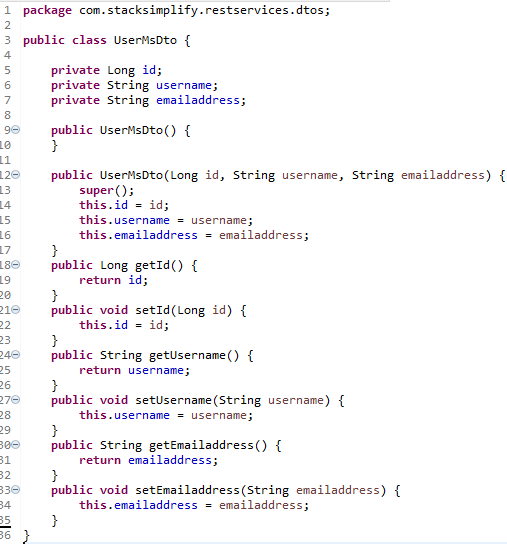
**Add dependencies to pom.xml:**



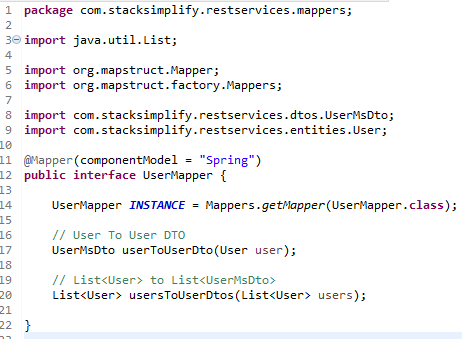
**Add processor plugin to pom.xml:**



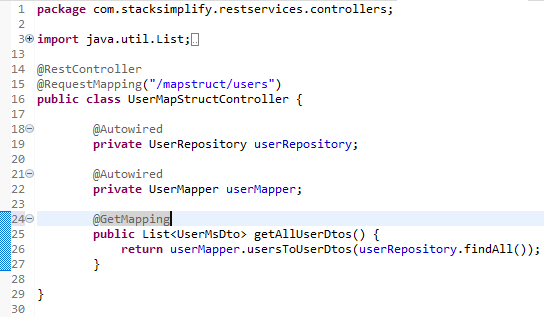
**Create UserMsDto class**



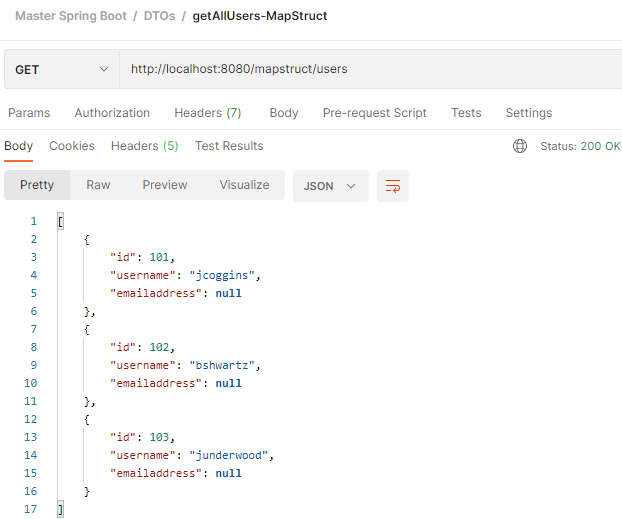
**Create the MapStruct interface:**



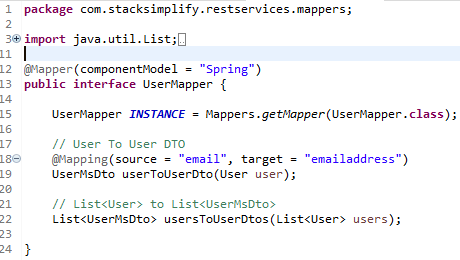
**Create the UserMapStructController**



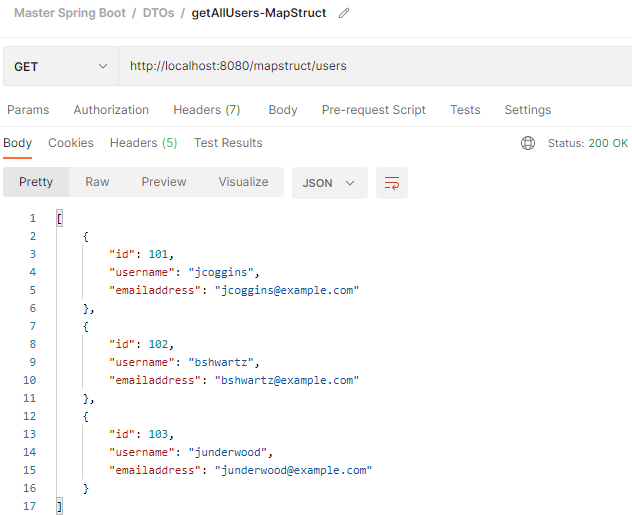
**Test using postman**



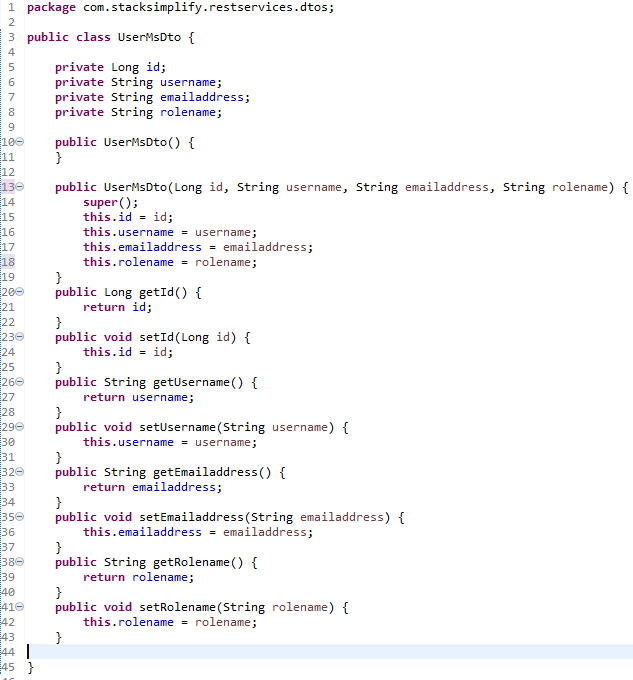
**Now, fix the emailaddress by mapping it:**



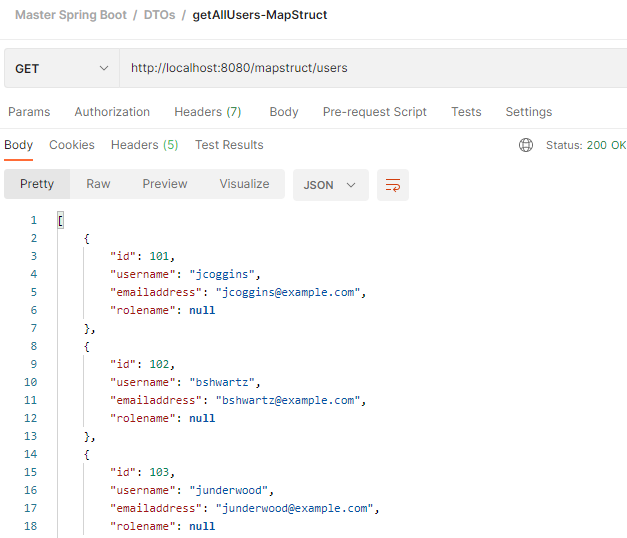
**Test using postman to see emailaddress:**



**Multiple mappings: (added rolename)**



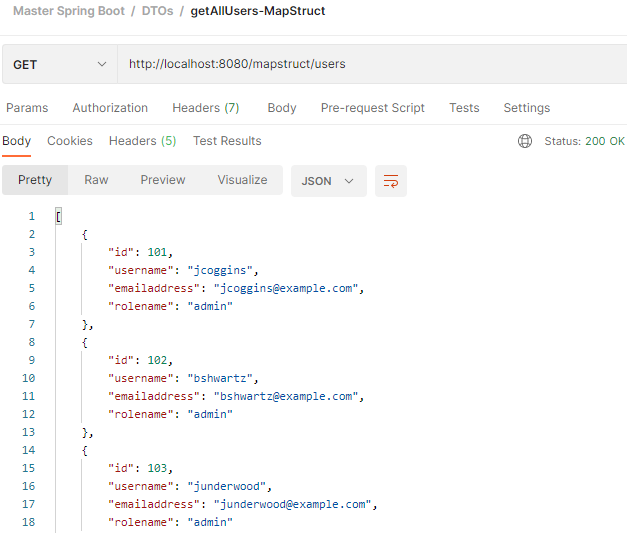
**Test using postman:**



**Fix rolename using mapper:**



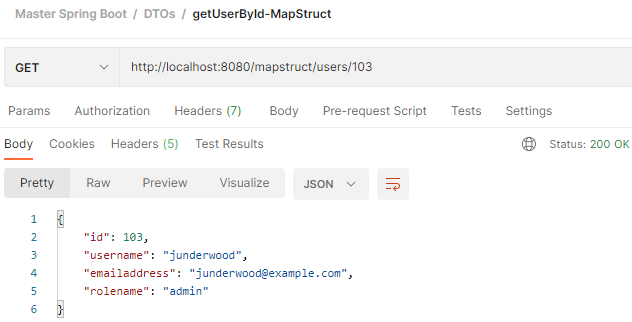
**Test using postman**



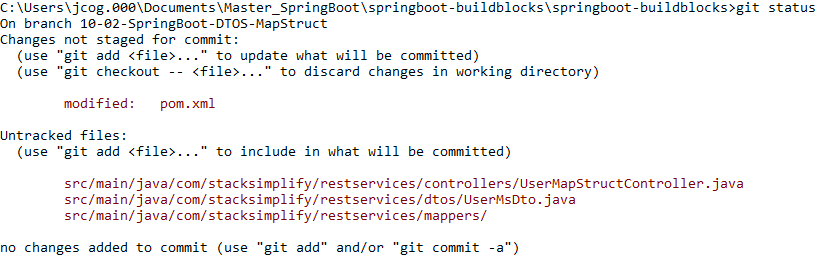
**Implement getUserById method:**



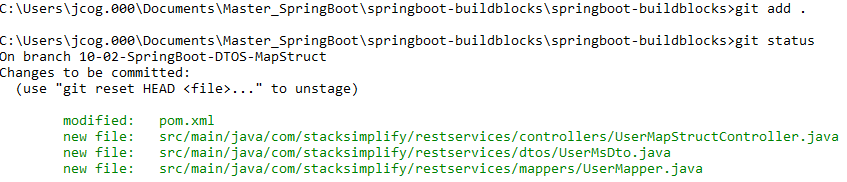
**Test using postman**



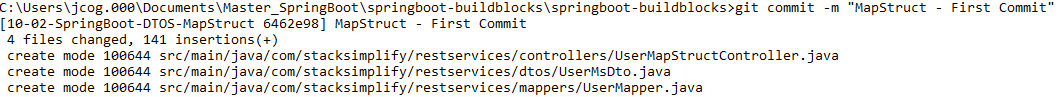
**git status**



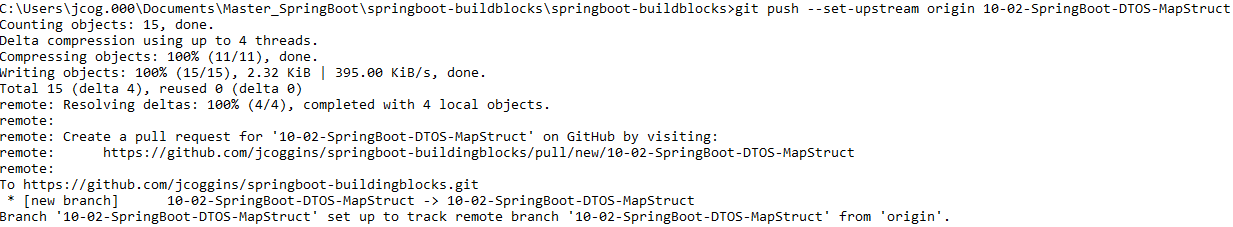
**git add . git status**



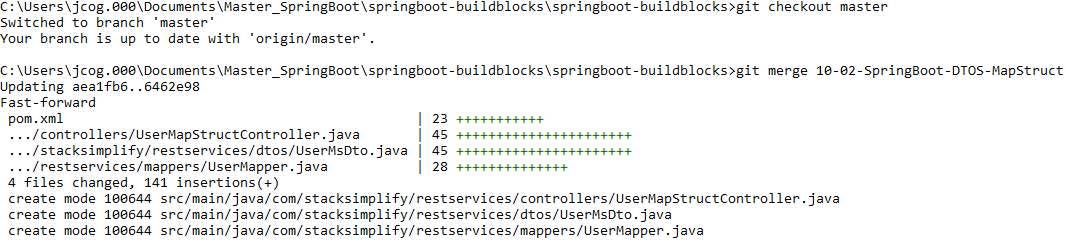
**git commit –m “MapStruct – First Commit”**



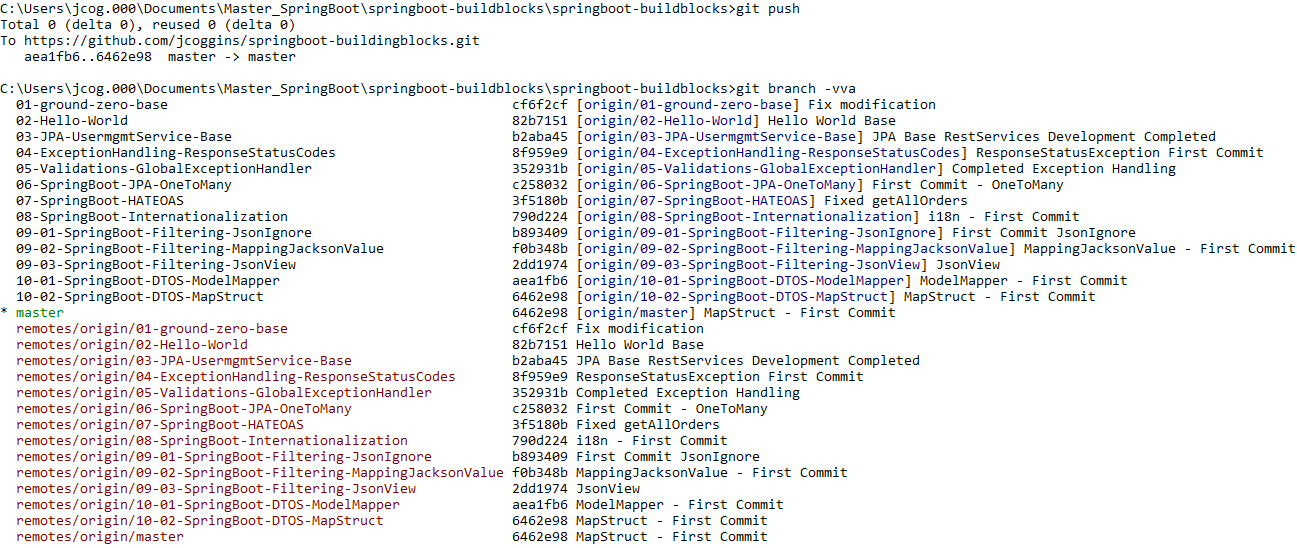
**git push –set-upstream origin 10-02-SpringBoot-DTOS-MapStruct**



**git checkout master git merge 10-02-SpringBoot-DTOS-MapStruct**



**git push git branch –vva**



**GUI Branches**

