**Master SpringBoot**

**11-SpringBoot Versioning and Content Negotiation:**

**Step-00: Introduction**

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

* **Git Branch name: 11-01-SpringBoot-Versioning**
* **Create new local branch**

**Step-02: Create 2 DTO’s and add one additional field in Entity named “address”**

* **Entity Layer**
  + **Add new field named city**
  + **Update the data.sql (import.sql) with address**
* **DTO Layer**
  + **Create two DTO’s**
    - **UserDtoV1 – contains all other fields except “address”**
    - **UserDtoV2 – contains all other fields plus “address” field**

**Step-03: Implement URI Versioning**

* **Controller Layer**
  + **Create UserUriVersioningController by copying UserModelMapperController**
  + **We are going to use ModelMapper to transform Entity to DTO**
  + **Implement getUserByIdv1 and getUserByIdv2 methods with URI’s**
  + **URI Versions**
    - **@GetMapping({“/v1.0/{id}”, “/v1.1/{id}”})**
    - **@GetMapping(“/v2.0/{id}”)**
* **Test using Postman**
  + - **Version V1.0 - GET** [**http://localhost:8080/versioning/uri/users/v1.0/101**](http://localhost:8080/versioning/uri/users/v1.0/101)
    - **Version V1.0 - GET** [**http://localhost:8080/versioning/uri/users/v1.1/101**](http://localhost:8080/versioning/uri/users/v1.1/101)
    - **Version V1.0 - GET http://localhost:8080/versioning/uri/users/v2.0/101**

**Step-04: Implement Request Parameter Versioning**

* **Controller Layer**
  + **Create UserRequestParameterVersioningController by copying UserUriVersioningController**
  + **We are going to use ModelMapper to transform Entity to DTO**
  + **Implement getUserByIdv1 and getUserByIdv2 methods with Request Parameters**
  + **Request Parameters**
    - **@GetMapping(value = “/{id}”, params=”version=1”)**
    - **@GetMapping(value = “/{id}”, params=”version=2”)**
    - **GET** [**http://localhost:8080/versioning/params/users/101?version=1**](http://localhost:8080/versioning/params/users/101?version=1)
    - **GET** [**http://localhost:8080/versioning/params/users/101?version=2**](http://localhost:8080/versioning/params/users/101?version=2)
* **Test using Postman**
  + **Test using Postman**
    - **Version V1**
      * **GET** [**http://localhost:8080/versioning/params/users/101?version=1**](http://localhost:8080/versioning/params/users/101?version=1)
    - **Version V2**
      * **GET** [**http://localhost:8080/versioning/params/users/101?version=2**](http://localhost:8080/versioning/params/users/101?version=2)

**Step-05: Implement Custom Header Versioning**

* **Controller Layer**
  + **Create UserCustomHeaderVersioningController by copying UserRequestParameterVersioningController**
  + **We are going to use ModelMapper to transform Entity to DTO**
  + **Implement getUserByIdv1 and getUserByIdv2 methods with Custom Headers**
  + **Media Type**
    - **@GetMapping(value = “/{id}”, params=”version=1”)**
    - **@GetMapping(value = “/{id}”, params=”version=2”)**
* **Test using Postman**
  + **Test using Postman**
    - **Version V1**
      * **GET** [**http://localhost:8080/versioning/header/users/101?version=1**](http://localhost:8080/versioning/header/users/101?version=1)
      * **Header: API-VERSION = 1**
    - **Version V2**
      * **GET** [**http://localhost:8080/versioning/header/users/101?version=2**](http://localhost:8080/versioning/header/users/101?version=2)
      * **Header: API-VERSION = 2**

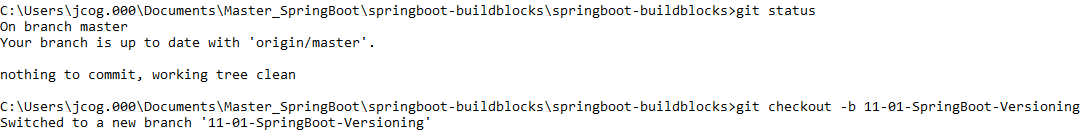
**Step-06: Implement Media Type Versioning**

* **Controller Layer**
  + **Create UserCustomHeaderVersioningController by copying UserRequestParameterVersionController**
  + **We are going to use ModelMapper to transform Entity to DTO**
  + **Implement getUserByIdv1 and getUserByIdv2 methods with Custom Headers**
* **Media Type**
* **@GetMapping(value = “/{id}”, produces=”application/vnd.stacksimplify.app-v1+json)**
* **@GetMapping(value = “/{id}”, produces=”application/vnd.stacksimplify.app-v2+json)**
  + **Test using Postman**
    - **Version V1**
      * **GET** [**http://localhost:8080/versioning/header/users/101**](http://localhost:8080/versioning/header/users/101)
      * **Header: Accept = application/vnd.stacksimplify.app-v1+json**
    - **Version V2**
      * **GET** [**http://localhost:8080/versioning/header/users/101**](http://localhost:8080/versioning/header/users/101)
      * **Header: Accept = application/vnd.stacksimplify.app-v2+json**

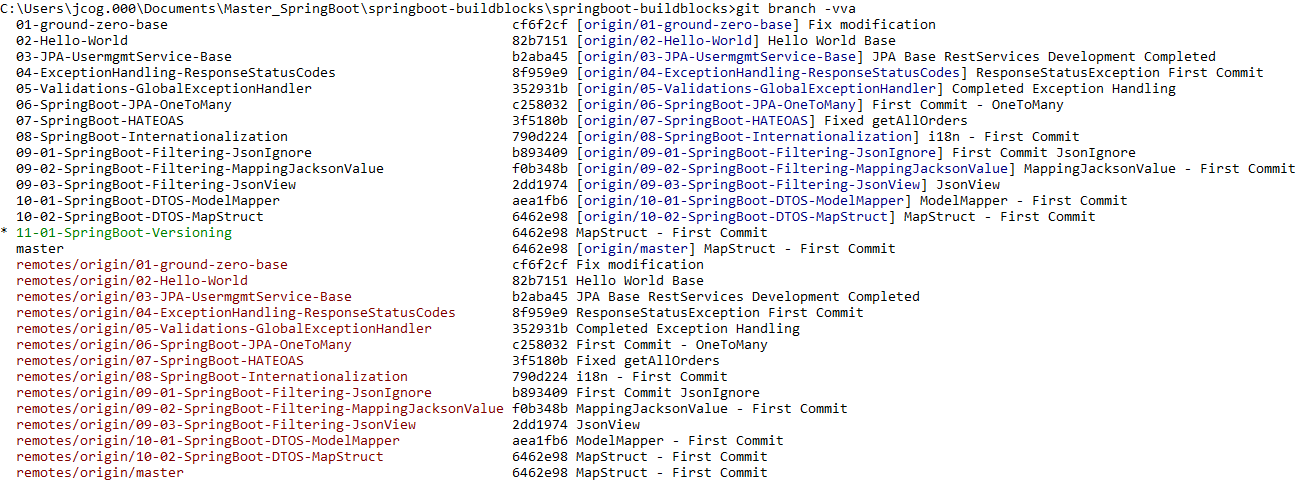
**Step-07: Commit & Push code via IDE**

**Create new branch:**

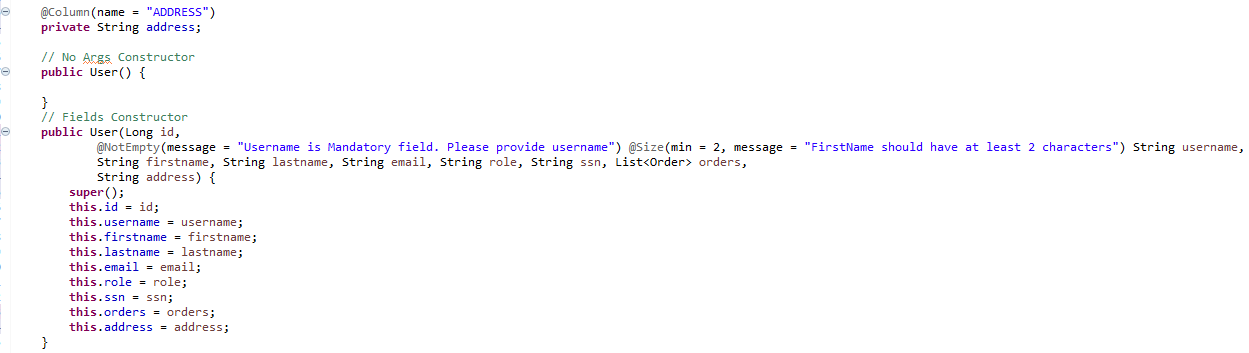
**git status git checkout –b 11-01-SpringBoot-Versioning**



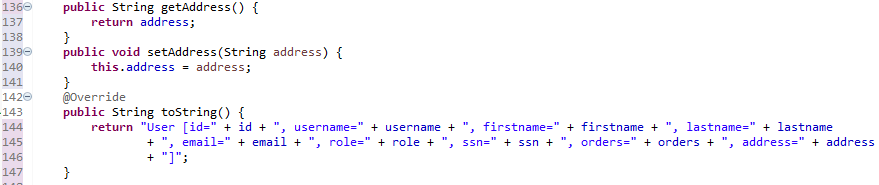
**git branch -vva**



**Add address and regenerate the Fields Constructor for User Entity:**



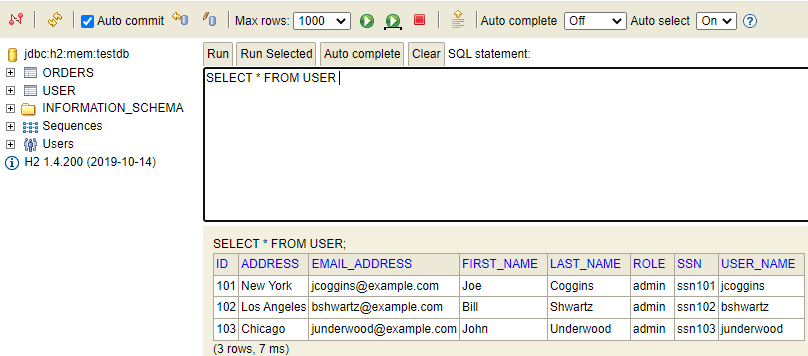
**Generate the GETTER and SETTER for address and toString method for User Entity:**



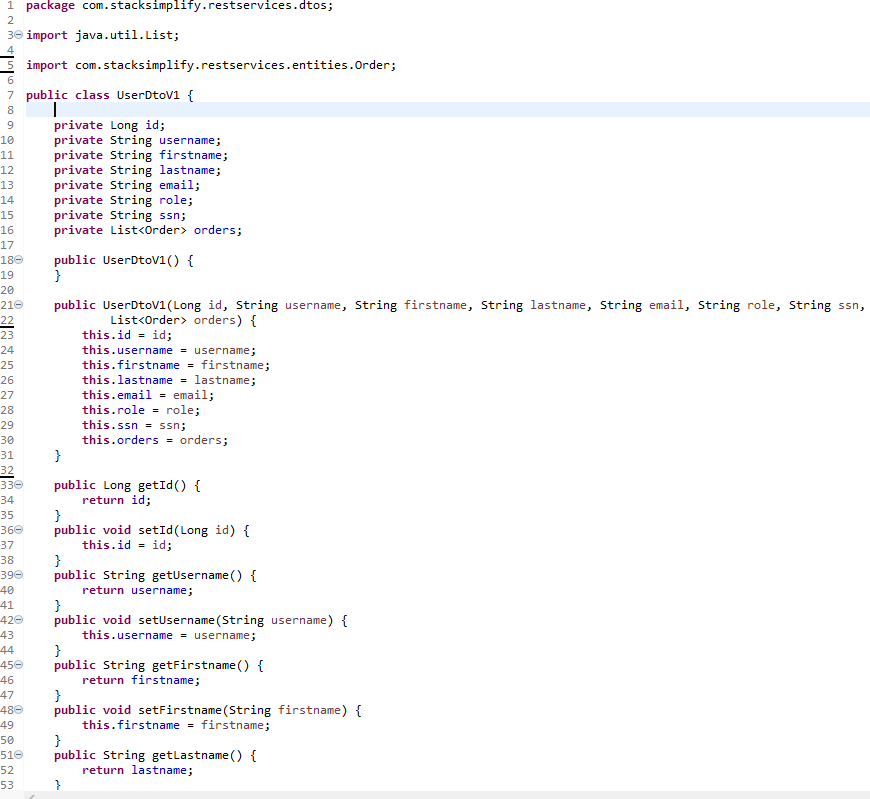
**Add 3 cities as addresses for import.sql (Support for H2 database)**

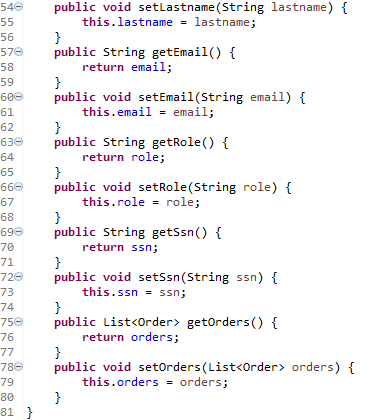


**Verify in H2 DB:**

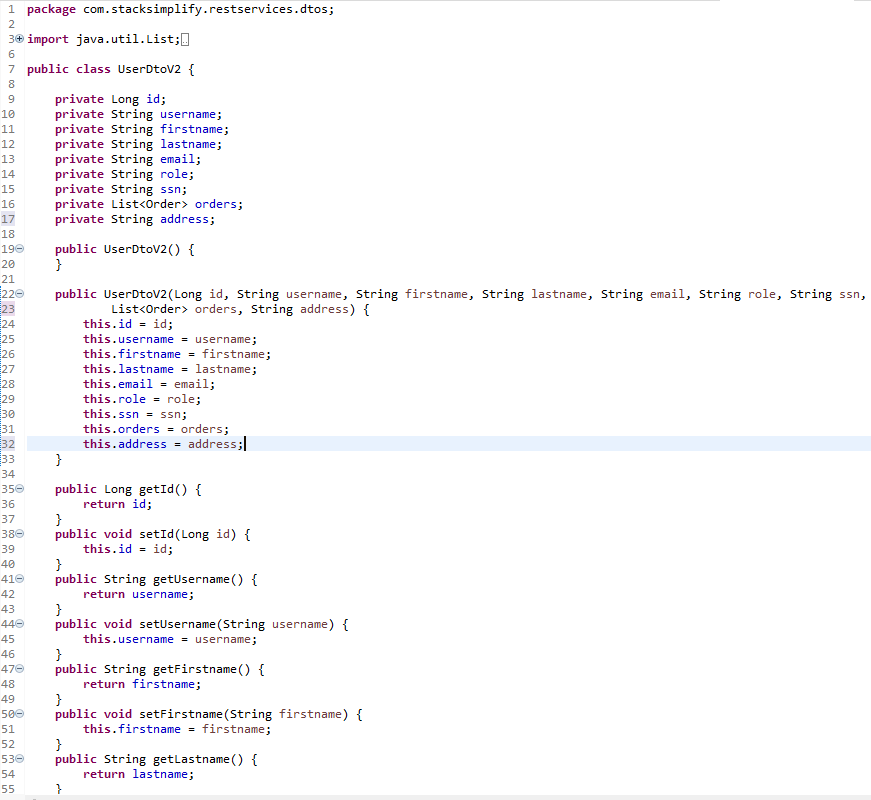


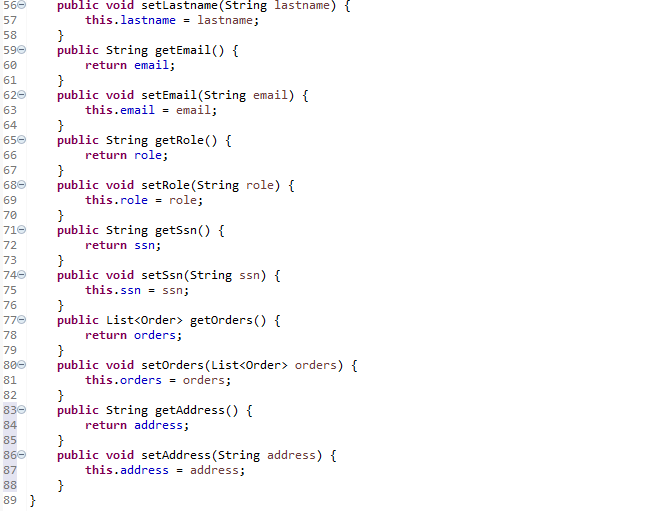
**Create UserDtoV1:**



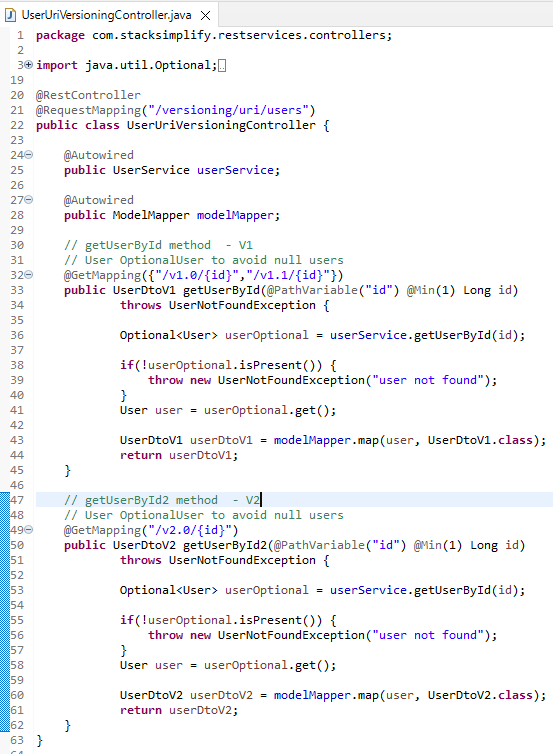


**Create UserDtoV2:**

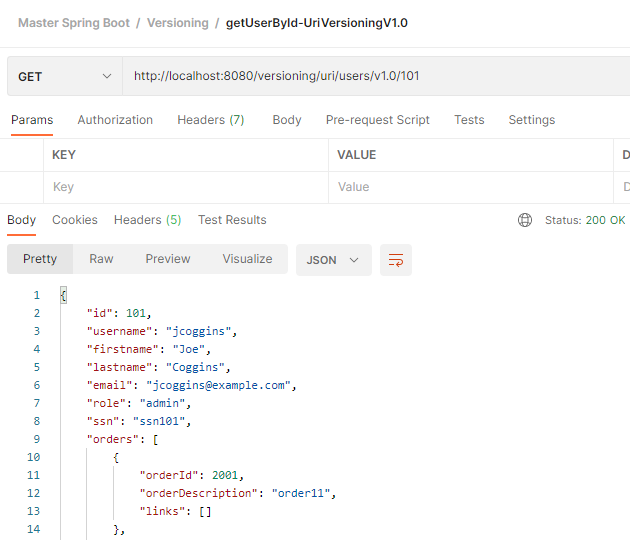




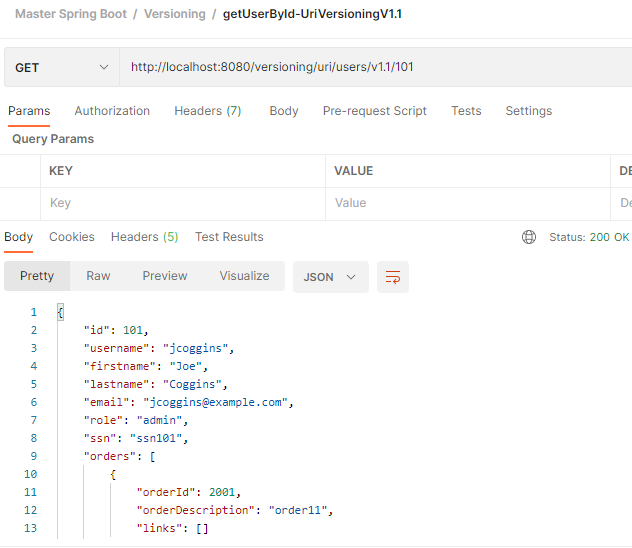
**Create UserUriVersioningController from UserModelMapperController**



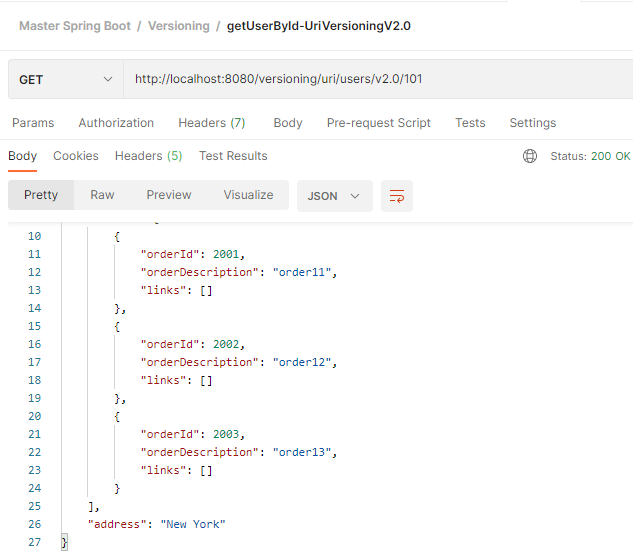
**Test using Postman: v1.0**



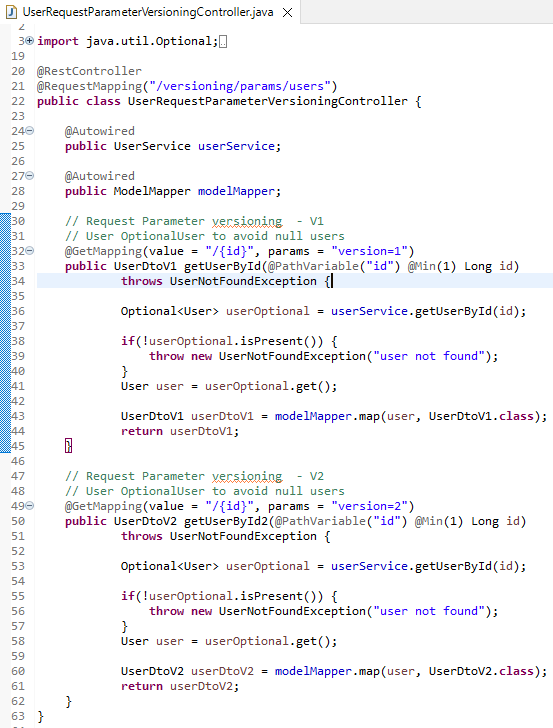
**Test using Postman: v1.1**



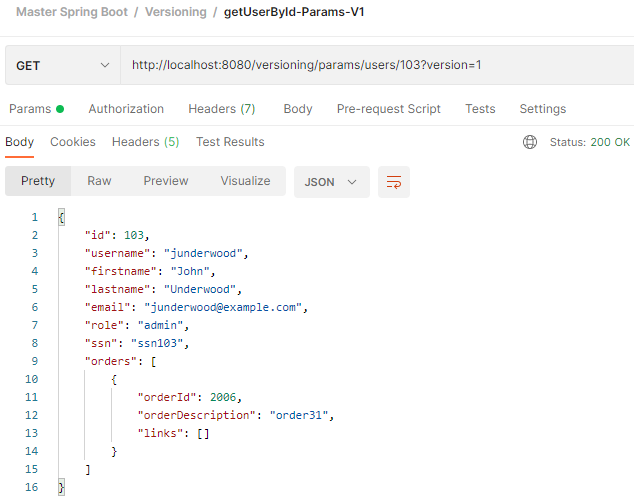
**Test using Postman: v2.0 (address appears)**



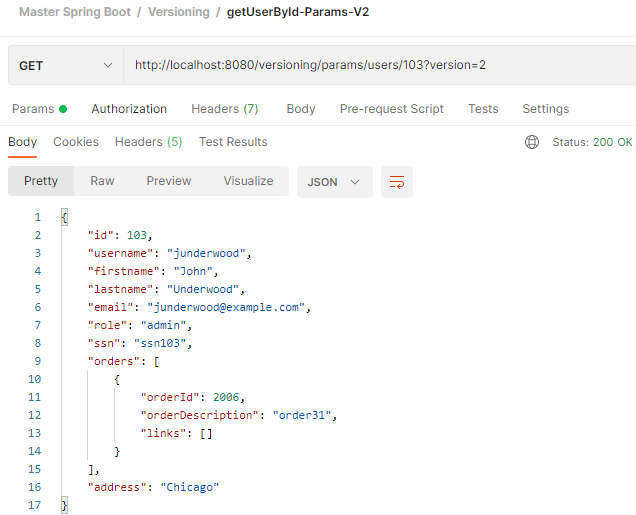
**Create UserRequestParameterVersioningController:**



**Test using Postman: v1**



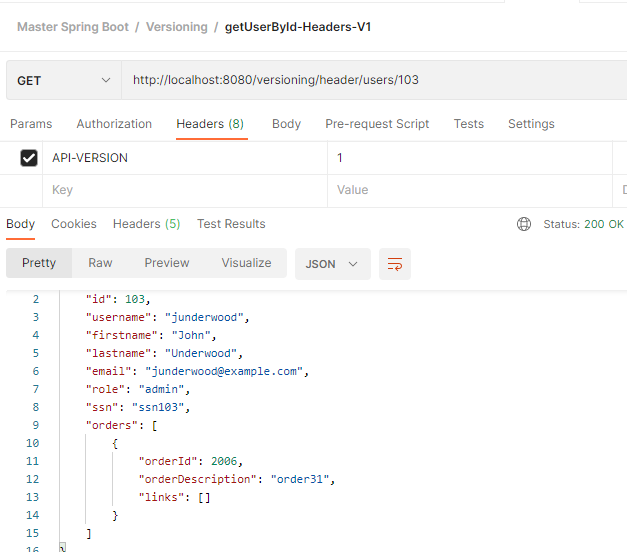
**Test using Postman: v2 (address appears)**



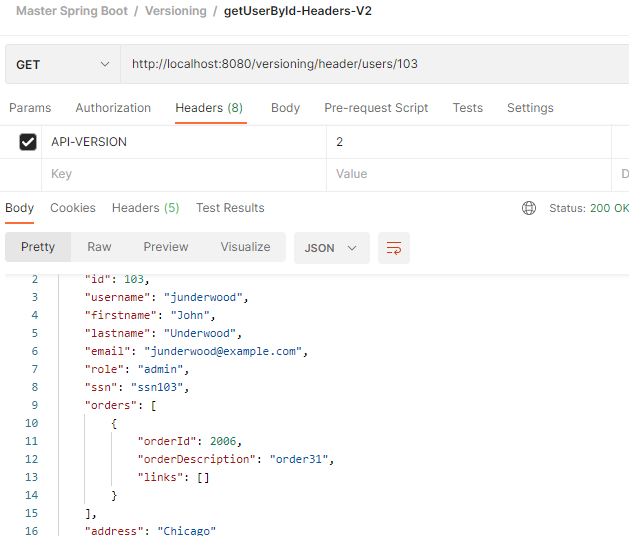
**Create UserCustomHeaderVersioningController:**



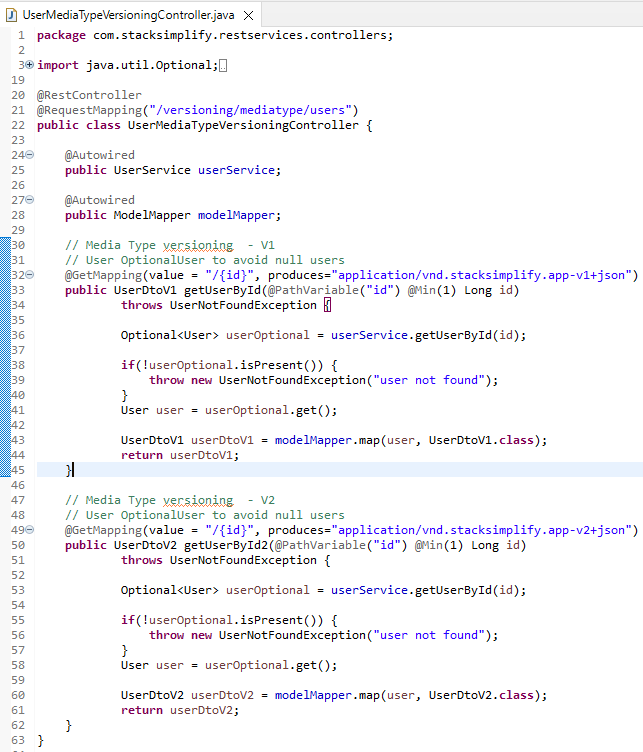
**Test using Postman: v1**



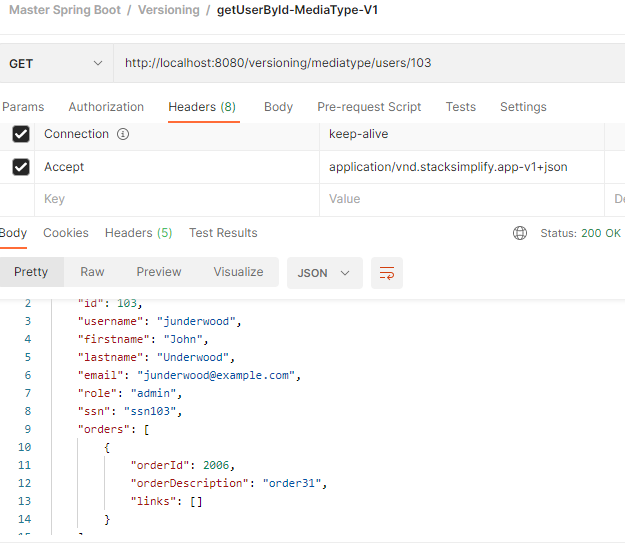
**Test using Postman: v2 (address appears)**



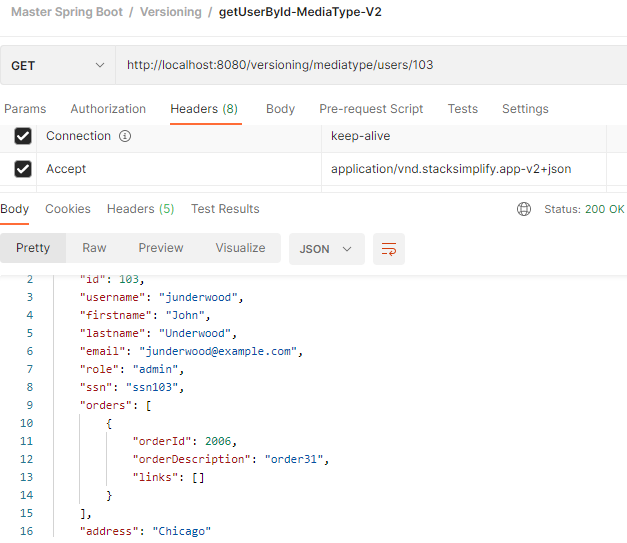
**Create UserMediaTypeVersioningController:**



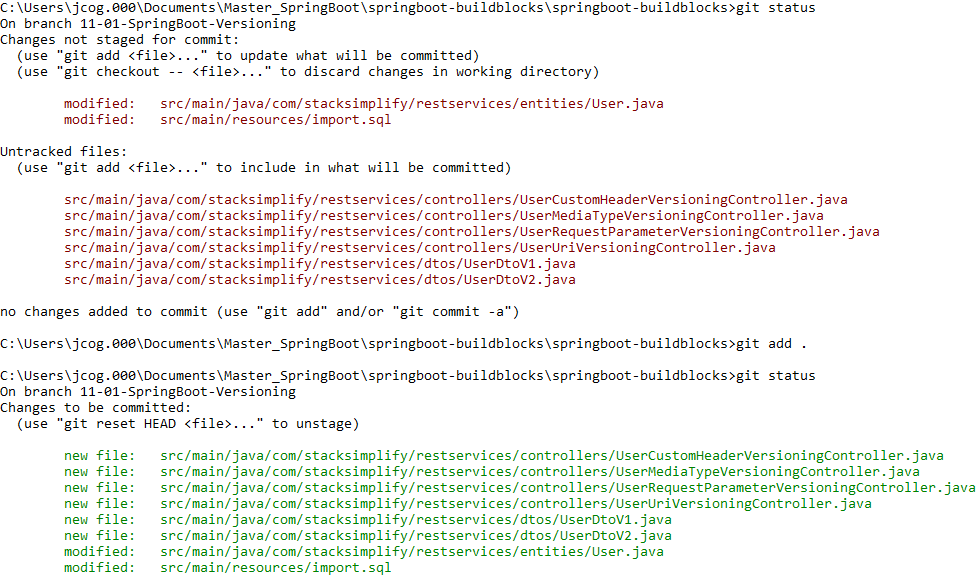
**Test using Postman: v1**



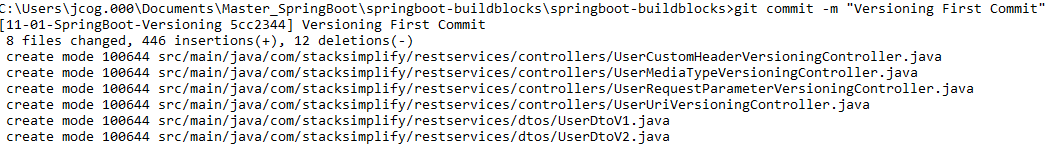
**Test using Postman: v2 (address appears)**



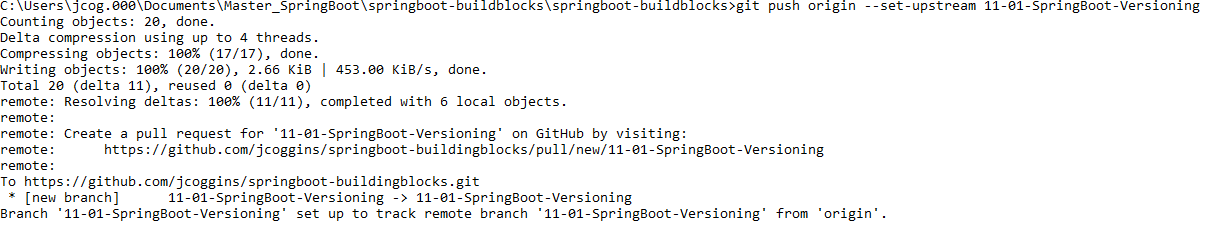
**git status git add . git status**



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



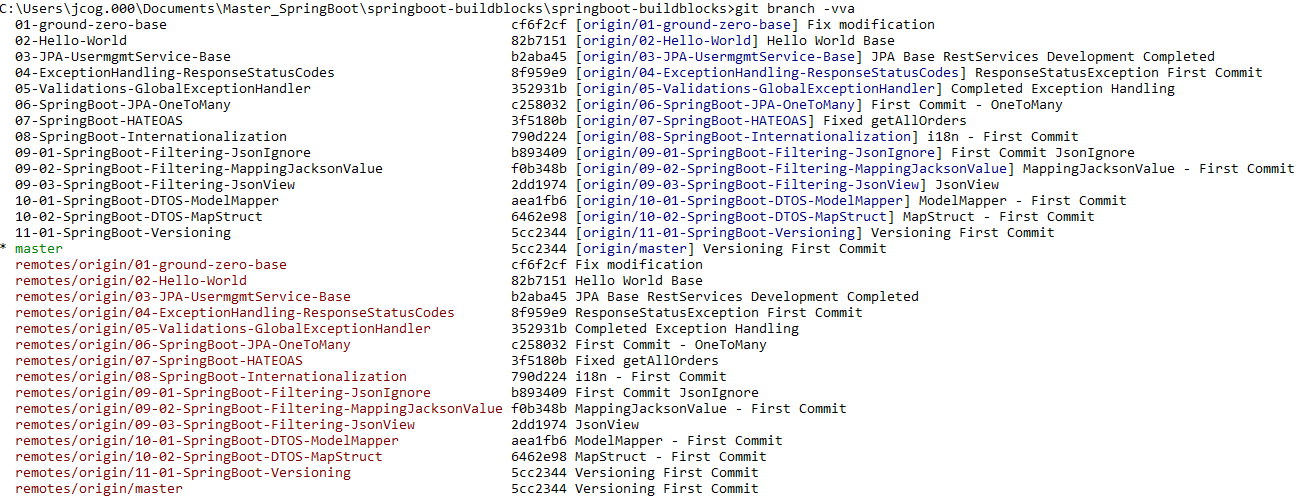
**git push origin –set-upstream 11-01-SpringBoot-Versioning**



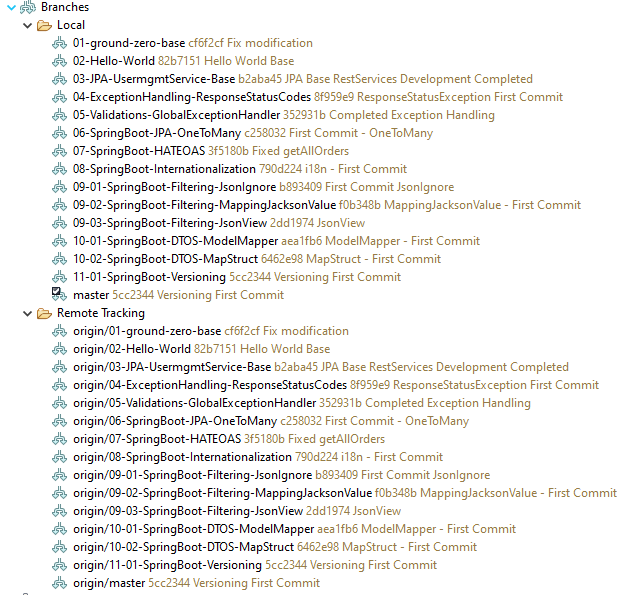
**git checkout master git merge 11-01-SpringBoot-Versioning git push**



**git branch –vva**



**IDE Branches:**



**Step-00: Introduction Content Negotiation**

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

* **git Branch name: 11-02-SpringBoot-ContentNegotiation**
* **Create new local branch**

**Step-02: Test using Postman**

* **getAllUsers (GET)**
  + **GET /users**
  + **Request Header**
    - **Accept: application/xml**
* **Create User (POST)**
  + **POST /users**
  + **Request Header**
    - **Content-Type: application/xml**

**Step-03: Add Dependency in pom.xml & Restart Embedded Tomcat**

**<dependency>**

**<groupId>com.fasterxml.jackson.dataformat</groupId>**

**<artifactId>jackson-dataformat-xml</artifactId>**

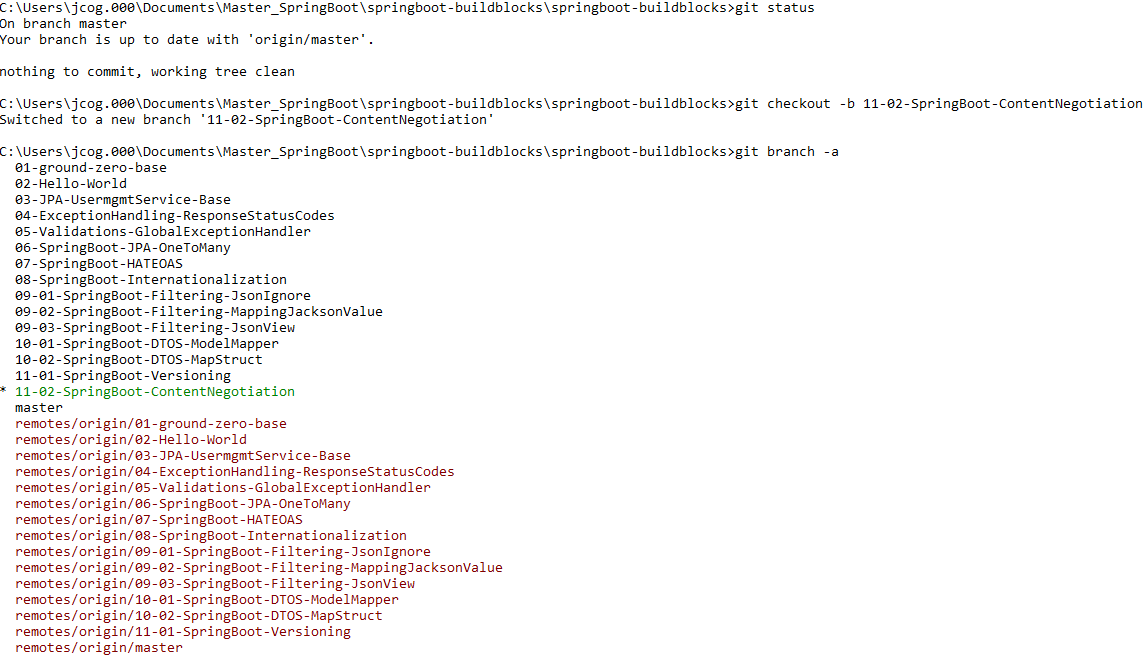
**</dependency>**

**Step-04: Test using postman**

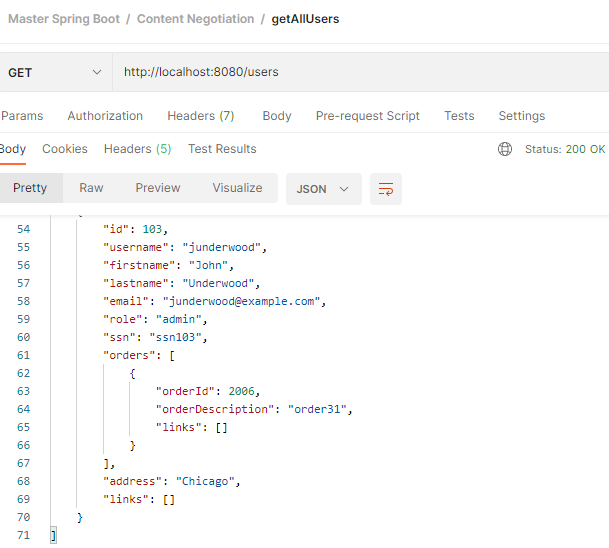
* **getAllUsers (GET)**
* **Request Header**
  + **Accept: application/xml**
* **Create User (POST)**
  + **POST /users**
  + **Request Header**
    - **Content-Type: application/xml**

**Step-05: Commit & Push code via IDE**

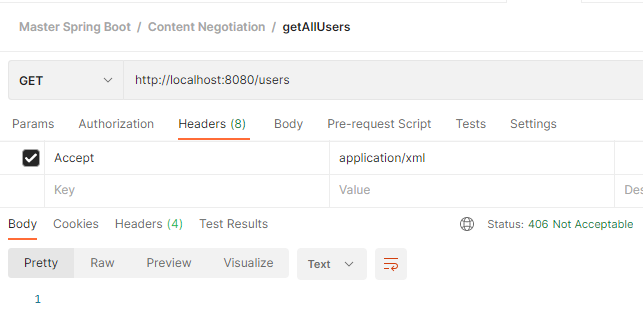
**git status git checkout –b 11-02-SpringBoot-ContentNegotiation git branch -a**



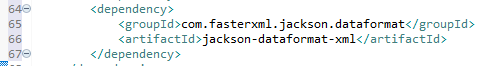
**Test Json output for getAllUsers:**



**Change the accept to xml:**

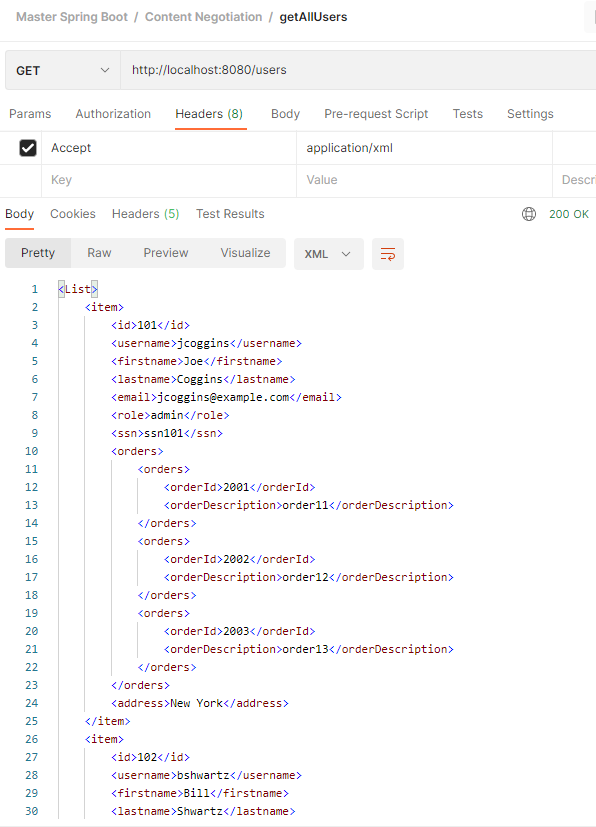


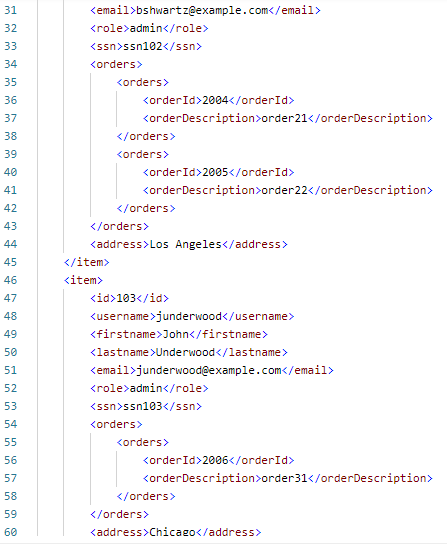
**Add dependency:**



**Stop and Start JVM**

**Rerun the accept: application/xml**

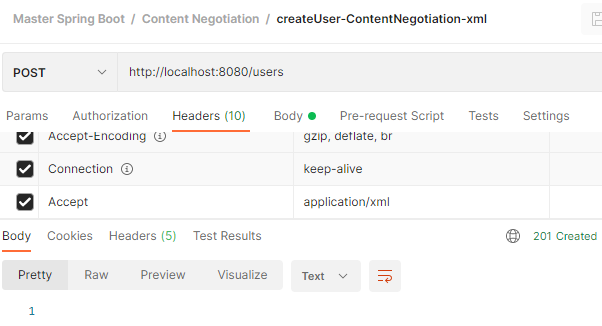




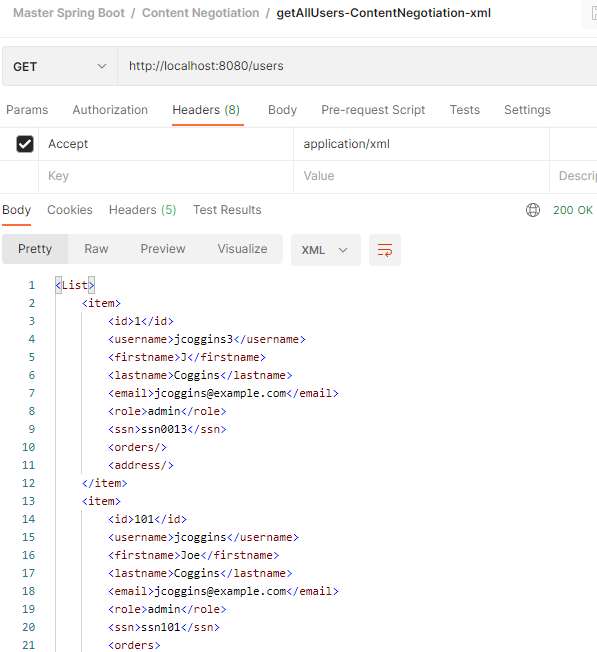


**So, we now see that json has been replaced with xml!**

**Test creatUser method:**

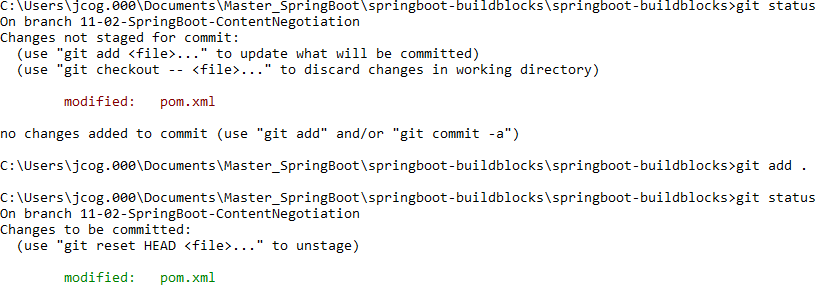


**Check user id=1 is created:**



**It works fine! I did not need to use userId as a property!!**

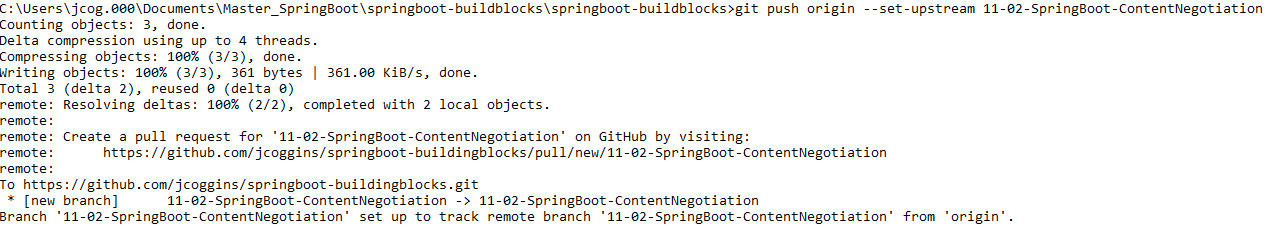
**git status git add . git status**



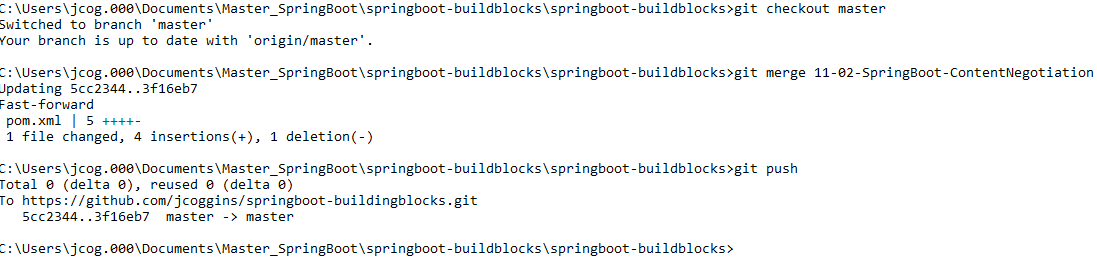
**git commit –m “Content Negotiation First Commit”**



**git push origin –set-upstream 11-02-SpringBoot-ContentNegotiation**



**git checkout master git merge 11-01-SpringBoot-Versioning git push**



**git branch –vva**

