CHEKKAPALLI TANUSHA

20EM1A0515

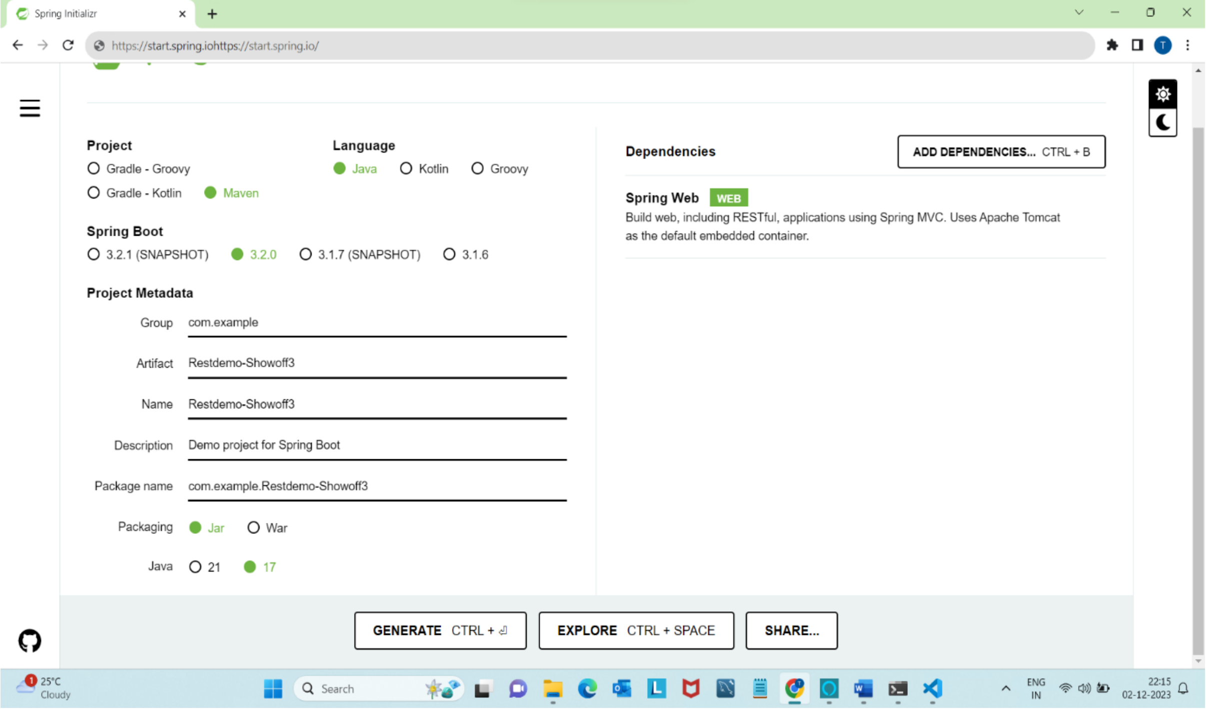
SWARNANDHRA INSTITUTE OF ENG& TECHNOLOGY

Showoff3_documentation

Initializing the Spring Boot Application :

1.Go to the https://start.spring.io/ this website .

2.Then choose lang, and Add Dependencies as shown below;



Project
○ Gradle - Groovy
○ Gradle - Kotlin
● Maven

Language
● Java ○ Kotlin ○ Groovy

Spring Boot
○ 3.2.1 (SNAPSHOT) ● 3.2.0 ○ 3.1.7 (SNAPSHOT) ○ 3.1.6

Project Metadata

Group com.example

Artifact Restdemo-Showoff3

Name Restdemo-Showoff3

Description Demo project for Spring Boot

Package name com.example.Restdemo-Showoff3

Packaging ● Jar ○ War

Java ○ 21 ● 17

Dependencies

Spring Web WEB
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

3. After Generate this ,and open the downloaded zip file and right click on it,,,,,

Then tap Extract all and click “**EXTRACT**”.

4. And go to search bar, then select the COPY ADD and REMOVE IT

And type “*CMD”.*

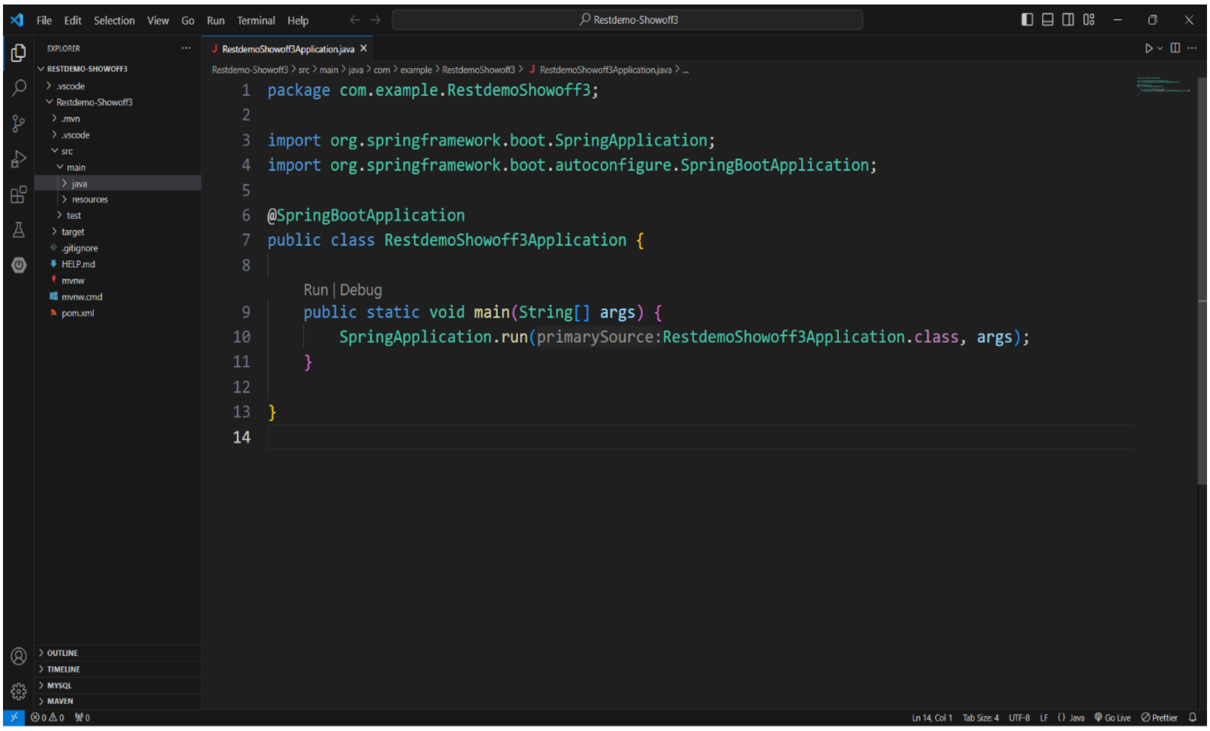
5. and COMMAND PROMPT app will open, then type <<< code .

6. It directly goes to **VSCODE APP.**

7.And the selected folder will directly Opened…

As Shown below…and next click on RUN and “START DEBBUGGING”.

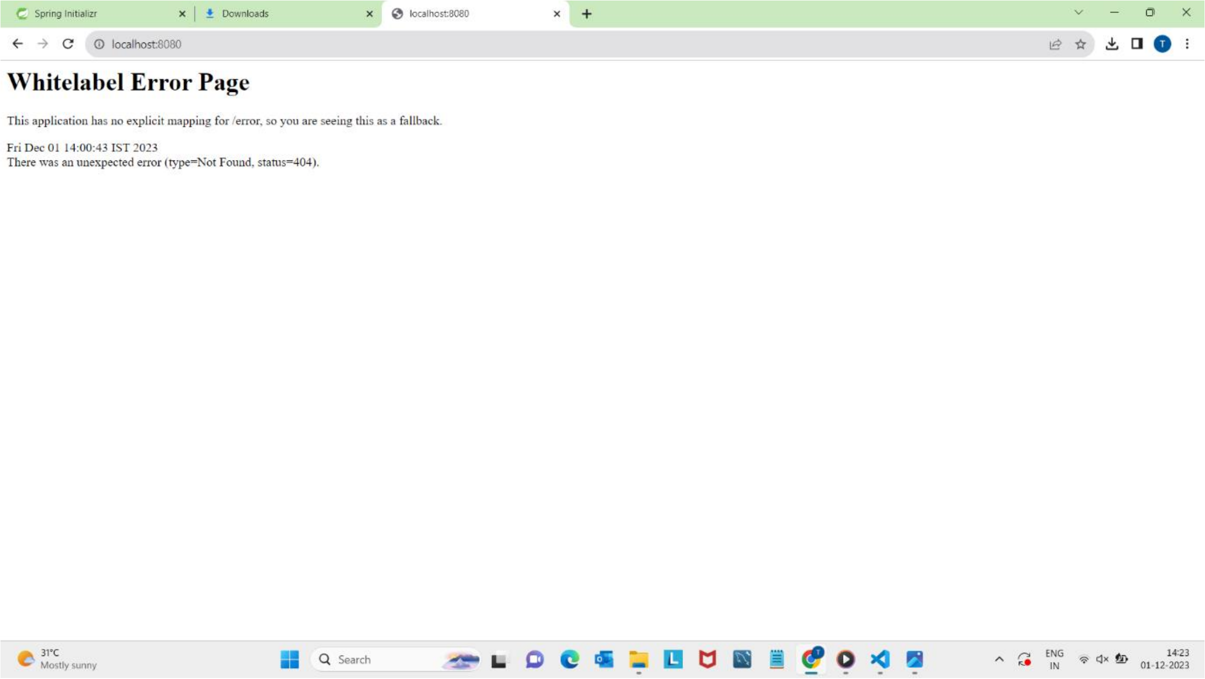
This shows a TOMCAT webserver started at port **8080.**



Restdemo-Showoff3
EXPLORER
RESTDEMO-SHOWOFF3
.vscode
Restdemo-Showoff3
.mvn
.vscode
src
main
java
resources
test
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
OUTLINE
TIMELINE
MYSQL
MAVEN
RestdemoShowoff3Application.java
Restdemo-Showoff3 > src > main > java > com > example > RestdemoShowoff3 > RestdemoShowoff3Application.java > ...
Ln 14, Col 1 Tab Size: 4 UTF-8 LF {} Java Go Live Prettier

```java
package com.example.RestdemoShowoff3;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class RestdemoShowoff3Application {

    Run | Debug
    public static void main(String[] args) {
        SpringApplication.run(primarySource:RestdemoShowoff3Application.class, args);
    }

}
```

8.It shows a Whitelabel Error Page



Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

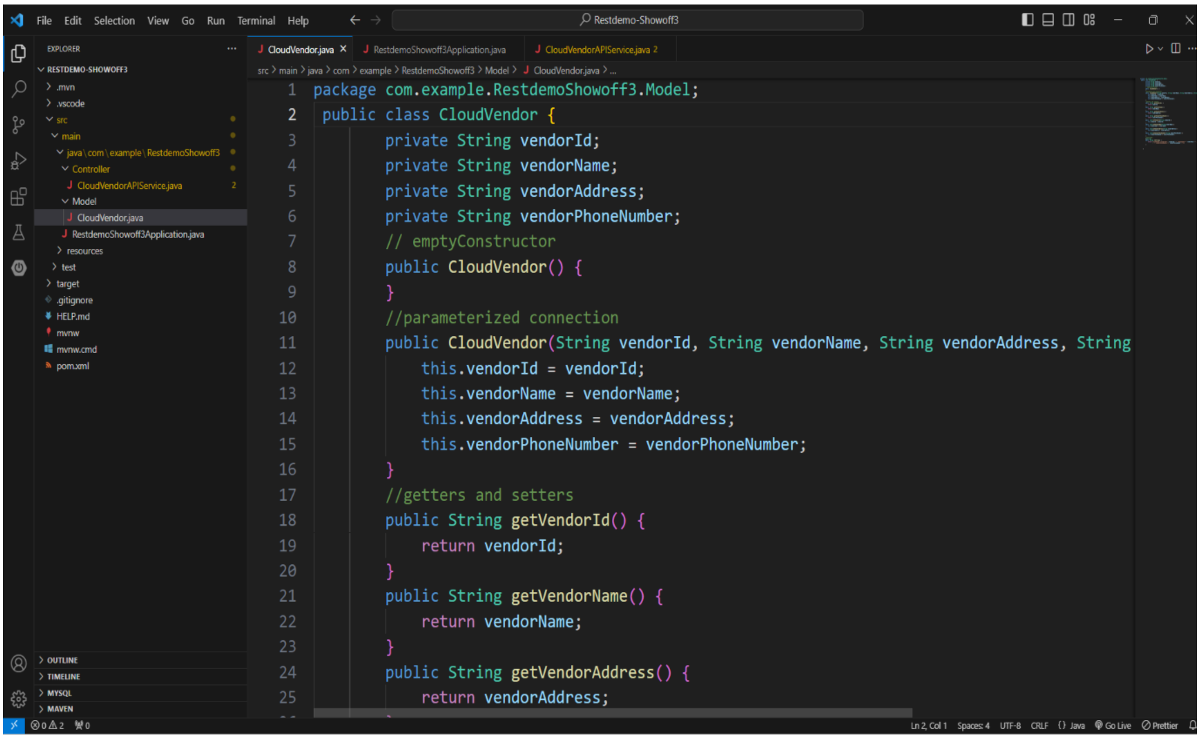
Fri Dec 01 14:00:43 IST 2023
There was an unexpected error (type=Not Found, status=404).

9. go to src < main < java/com/ example/ RestdemoShowoff3 , and click that highlighted one .

10. In that That RestdemoShowoff3 create a new folder named as "Model",

11. In that folder create a File named as "CloudVendor.java" as shown below….

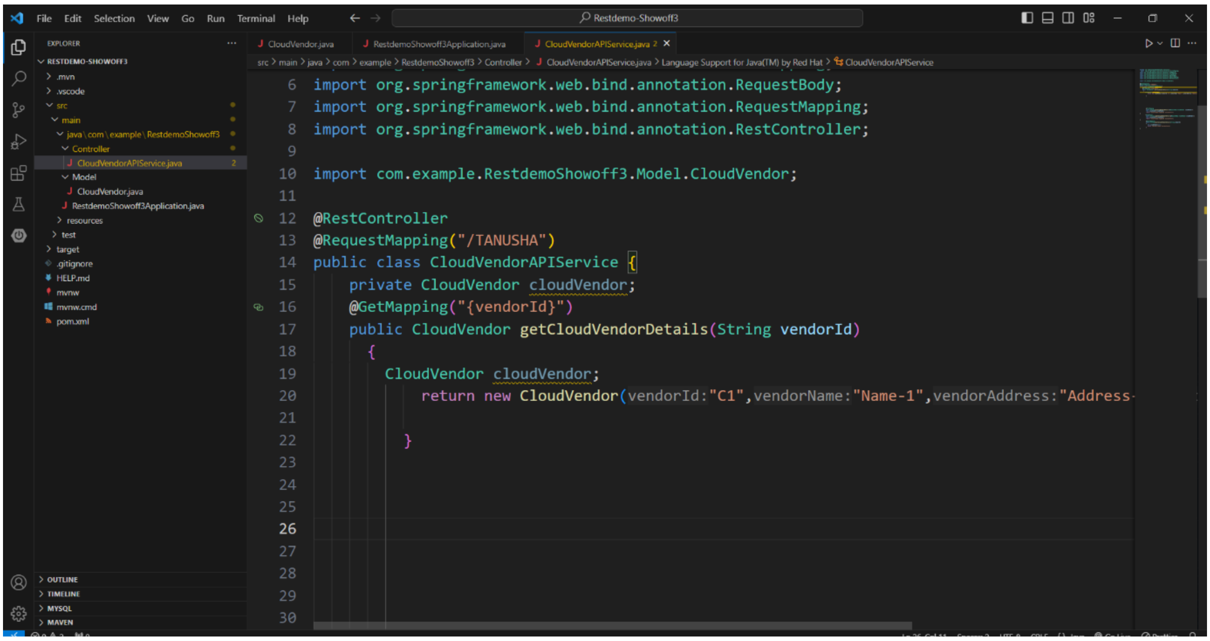
12. And create a another Folder named as "Controller"..



```java
package com.example.RestdemoShowoff3.Model;
 public class CloudVendor {
        private String vendorId;
        private String vendorName;
        private String vendorAddress;
        private String vendorPhoneNumber;
        // emptyConstructor
        public CloudVendor() {
        }
        //parameterized connection
        public CloudVendor(String vendorId, String vendorName, String vendorAddress, String
            this.vendorId = vendorId;
            this.vendorName = vendorName;
            this.vendorAddress = vendorAddress;
            this.vendorPhoneNumber = vendorPhoneNumber;
        }
        //getters and setters
        public String getVendorId() {
            return vendorId;
        }
        public String getVendorName() {
            return vendorName;
        }
        public String getVendorAddress() {
            return vendorAddress;
```

13. And In that folder create one File named as “CloudVendorAPIService.java”

Aftr completion of the code

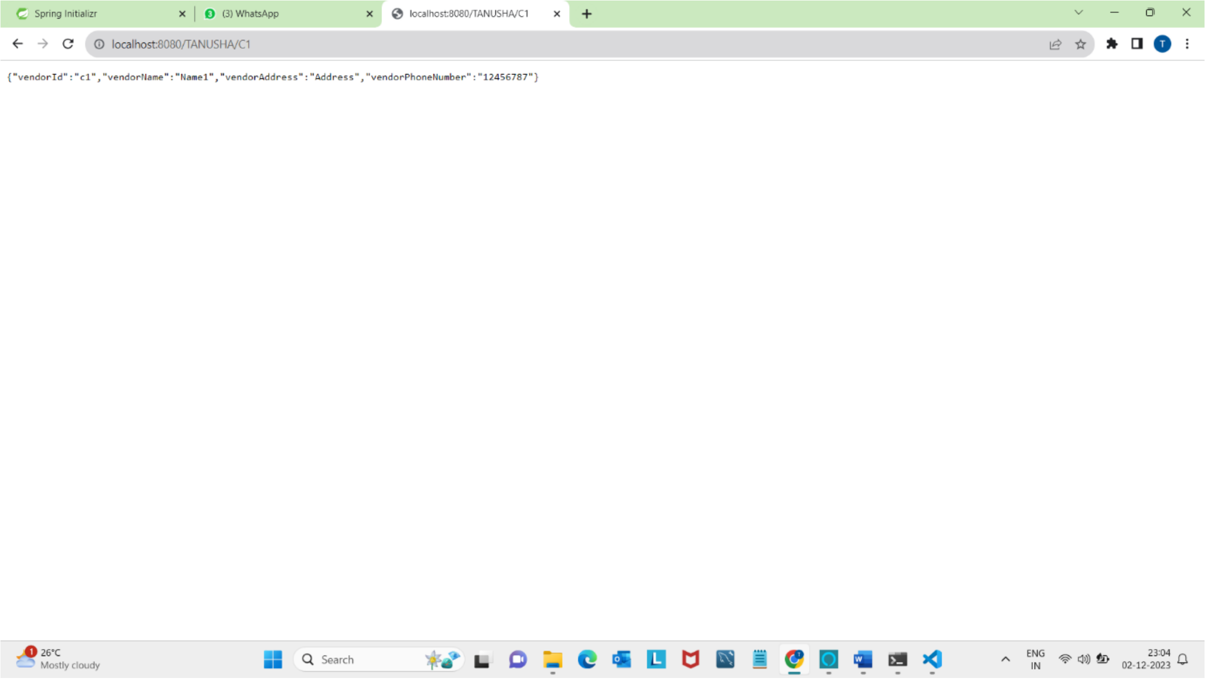


File
Edit
Selection
View
Go
Run
Terminal
Help
Restdemo-Showoff3
EXPLORER
RESTDEMO-SHOWOFF3
CloudVendor.java
RestdemoShowoff3Application.java
CloudVendorAPIService.java 2
src > main > java > com > example > RestdemoShowoff3 > Controller > CloudVendorAPIService.java > Language Support for Java(TM) by Red Hat > CloudVendorAPIService
OUTLINE
TIMELINE
MYSQL
MAVEN

```java
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.example.RestdemoShowoff3.Model.CloudVendor;

@RestController
@RequestMapping("/TANUSHA")
public class CloudVendorAPIService {
    private CloudVendor cloudVendor;
    @GetMapping("{vendorId}")
    public CloudVendor getCloudVendorDetails(String vendorId)
     {
       CloudVendor cloudVendor;
           return new CloudVendor(vendorId:"C1",vendorName:"Name-1",vendorAddress:"Address-

          }
```

14. and next click on RUN and “START DEBBUGGING”. This shows a TOMCAT webserver strtd at port **8080.**

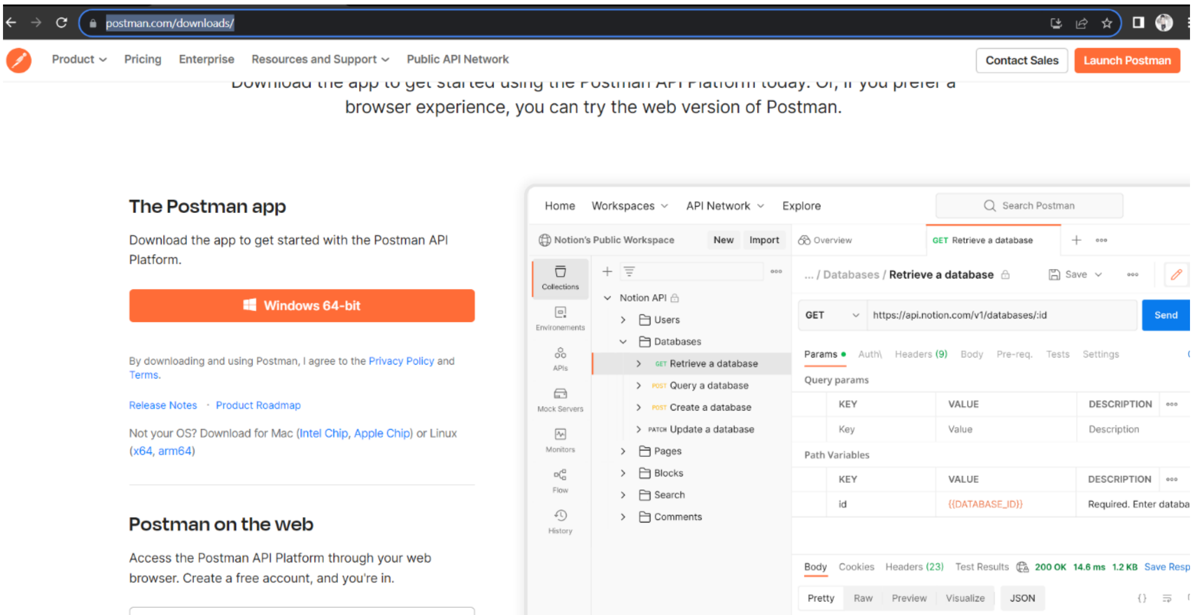
And IT SHOWS THE OUTPUT OF **VENDOR DETAILS**……AS BELOW



```
{"vendorId":"c1","vendorName":"Name1","vendorAddress":"Address","vendorPhoneNumber":"12456787"}
```

1. Lets continue the process of showoff3 ,, first download the App of **Postman**

2.Go to this url https://www.postman.com/downloads/ to download the App..



Download the app to get started using the Postman API Platform today. Or, if you prefer a browser experience, you can try the web version of Postman.

The Postman app

Download the app to get started with the Postman API Platform.

Windows 64-bit

By downloading and using Postman, I agree to the Privacy Policy and Terms.

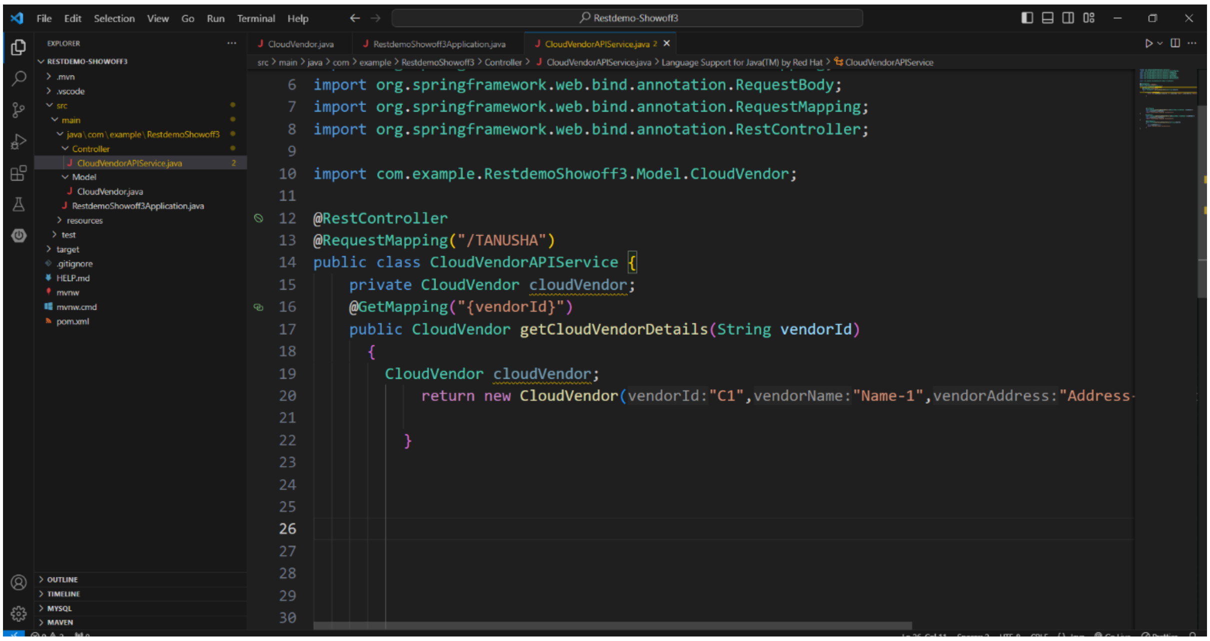
Release Notes · Product Roadmap

Not your OS? Download for Mac (Intel Chip, Apple Chip) or Linux (x64, arm64)

Postman on the web

Access the Postman API Platform through your web browser. Create a free account, and you're in.

3. Now open previous code in Visualstudioapp, and open <u>CloudVendorAPIService.java module.</u>


File
Edit
Selection
View
Go
Run
Terminal
Help
Restdemo-Showoff3
EXPLORER
RESTDEMO-SHOWOFF3
.mvn
.vscode
src
main
java \ com \ example \ RestdemoShowoff3
Controller
CloudVendorAPIService.java
Model
CloudVendor.java
RestdemoShowoff3Application.java
resources
test
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
OUTLINE
TIMELINE
MYSQL
MAVEN
CloudVendor.java
RestdemoShowoff3Application.java
CloudVendorAPIService.java
src > main > java > com > example > RestdemoShowoff3 > Controller > CloudVendorAPIService.java > Language Support for Java(TM) by Red Hat > CloudVendorAPIService

```java
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.example.RestdemoShowoff3.Model.CloudVendor;

@RestController
@RequestMapping("/TANUSHA")
public class CloudVendorAPIService {
    private CloudVendor cloudVendor;
    @GetMapping("{vendorId}")
    public CloudVendor getCloudVendorDetails(String vendorId)
     {
       CloudVendor cloudVendor;
          return new CloudVendor(vendorId:"C1",vendorName:"Name-1",vendorAddress:"Address-

          }
```

4.Open postman app and create Blank collection and give name to that blankcollection as *******.

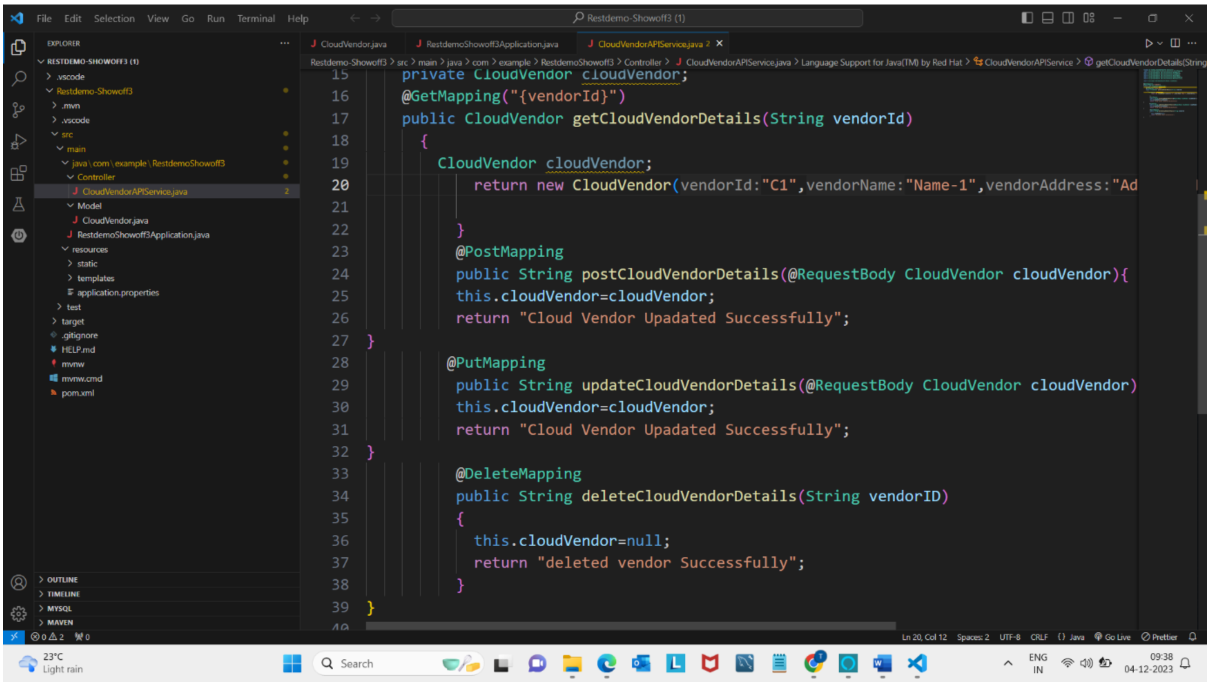
5. In the ********** collection add a request and assign the name to that request as Get and save it.

6.Copy the url ************ from the browser and past it in postman and click Send button.

7.It will display the following details…..

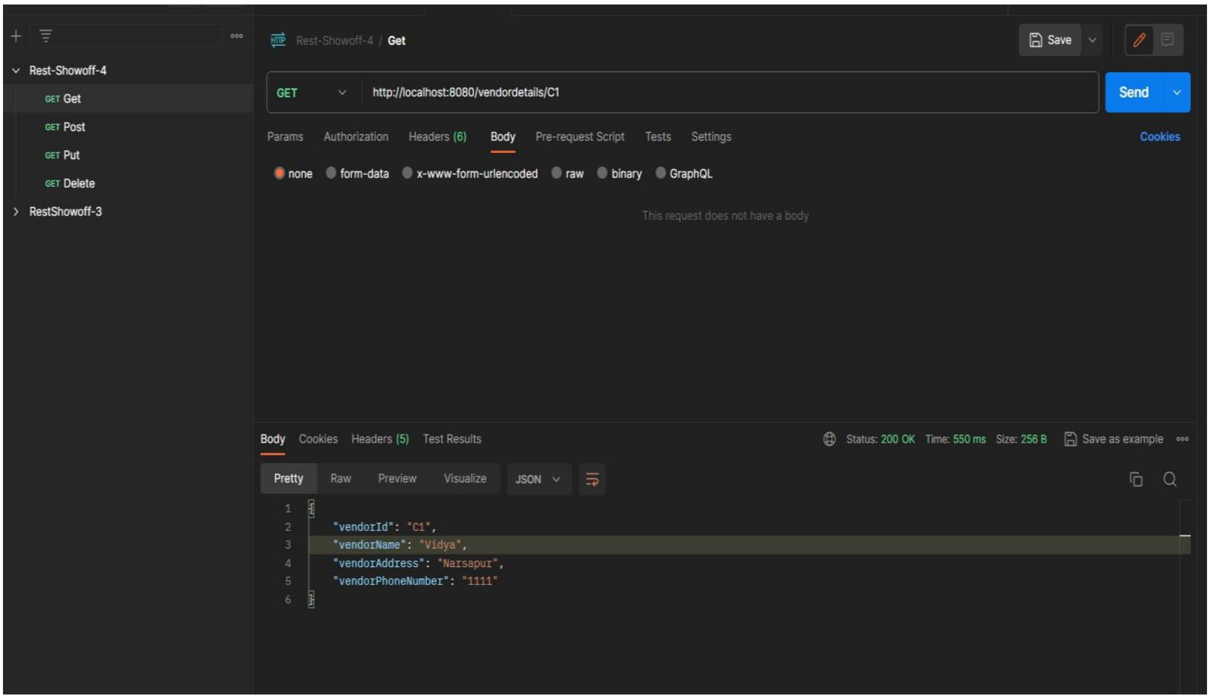
8.Again go to VScode and this methods

> @PostMapping
> @PutMapping
> @DeleteMapping


File Edit Selection View Go Run Terminal Help
Restdemo-Showoff3 (1)
EXPLORER
RESTDEMO-SHOWOFF3 (1)
.vscode
Restdemo-Showoff3
.mvn
.vscode
src
main
java \ com \ example \ RestdemoShowoff3
Controller
CloudVendorAPIService.java
Model
CloudVendor.java
RestdemoShowoff3Application.java
resources
static
templates
application.properties
test
target
.gitignore
HELP.md
mvnw
mvnw.cmd
pom.xml
OUTLINE
TIMELINE
MYSQL
MAVEN
CloudVendor.java
RestdemoShowoff3Application.java
CloudVendorAPIService.java
Restdemo-Showoff3 > src > main > java > com > example > RestdemoShowoff3 > Controller > CloudVendorAPIService.java > Language Support for Java(TM) by Red Hat > CloudVendorAPIService > getCloudVendorDetails(String
Ln 20, Col 12 Spaces: 2 UTF-8 CRLF Java Go Live Prettier
23°C Light rain
Search
ENG IN
09:38 04-12-2023

```
    private CloudVendor cloudVendor;
    @GetMapping("{vendorId}")
    public CloudVendor getCloudVendorDetails(String vendorId)
      {
        CloudVendor cloudVendor;
          return new CloudVendor(vendorId:"C1",vendorName:"Name-1",vendorAddress:"Ad

        }
        @PostMapping
        public String postCloudVendorDetails(@RequestBody CloudVendor cloudVendor){
        this.cloudVendor=cloudVendor;
        return "Cloud Vendor Upadated Successfully";
}
       @PutMapping
        public String updateCloudVendorDetails(@RequestBody CloudVendor cloudVendor)
        this.cloudVendor=cloudVendor;
        return "Cloud Vendor Upadated Successfully";
}
        @DeleteMapping
        public String deleteCloudVendorDetails(String vendorID)
        {
          this.cloudVendor=null;
          return "deleted vendor Successfully";
        }
}
```

9.get output:


Rest-Showoff-4 / Get
Save
Rest-Showoff-4
GET Get
GET Post
GET Put
GET Delete
RestShowoff-3
Send
Params Authorization Headers (6) Body Pre-request Script Tests Settings
Cookies
○ none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL
Body Cookies Headers (5) Test Results
Save as example
Pretty Raw Preview Visualize JSON

GET http://localhost:8080/vendordetails/C1

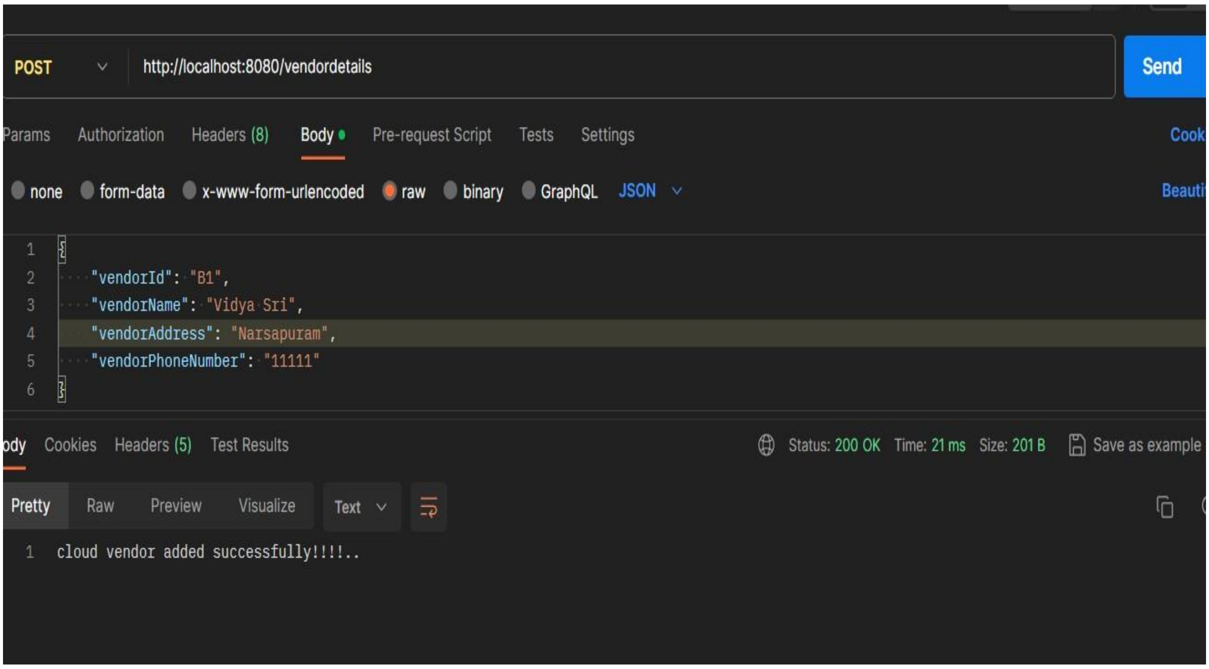
Body: ● none

This request does not have a body

Status: 200 OK Time: 550 ms Size: 256 B

```
{
    "vendorId": "C1",
    "vendorName": "Vidya",
    "vendorAddress": "Narsapur",
    "vendorPhoneNumber": "1111"
}
```

10. And past the copied url ********* in postrequest , Enter the details and click Body -> raw->Text-> JSON. click Send button.


Send
Params Authorization Headers (8) Body Pre-request Script Tests Settings
Cook
○ none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL
Beauti
ody Cookies Headers (5) Test Results
Save as example
Pretty Raw Preview Visualize Text

POST http://localhost:8080/vendordetails

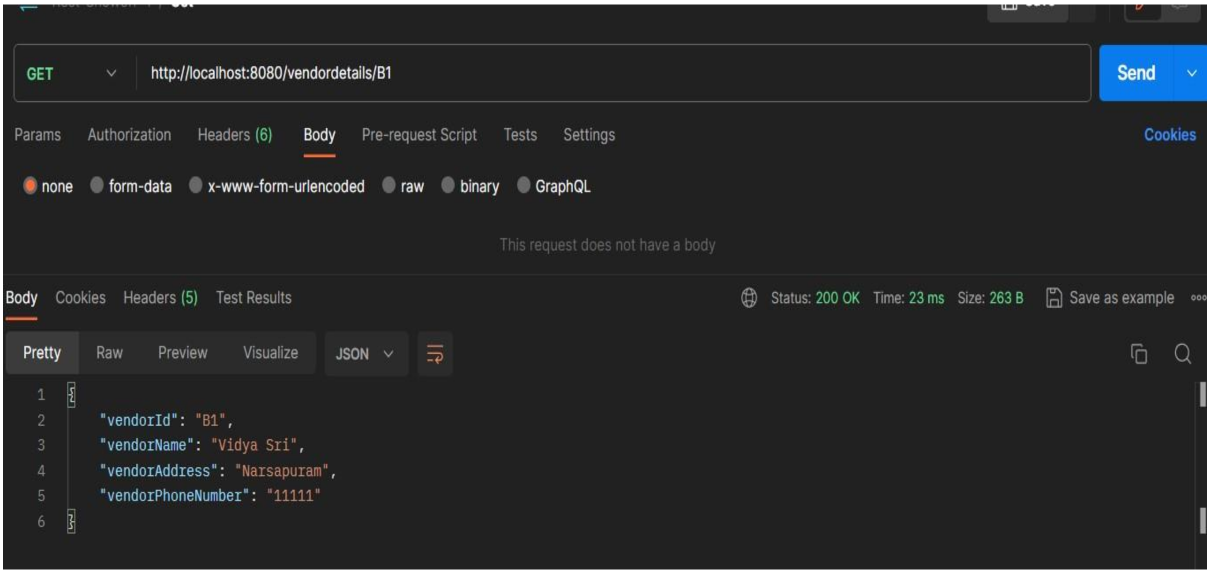
Body: ● raw JSON

```
{
    "vendorId": "B1",
    "vendorName": "Vidya Sri",
    "vendorAddress": "Narsapuram",
    "vendorPhoneNumber": "11111"
}
```

Status: 200 OK Time: 21 ms Size: 201 B

```
cloud vendor added successfully!!!!..
```

@GetMapping after performing @PostMapping:



GET http://localhost:8080/vendordetails/B1

Body: ● none

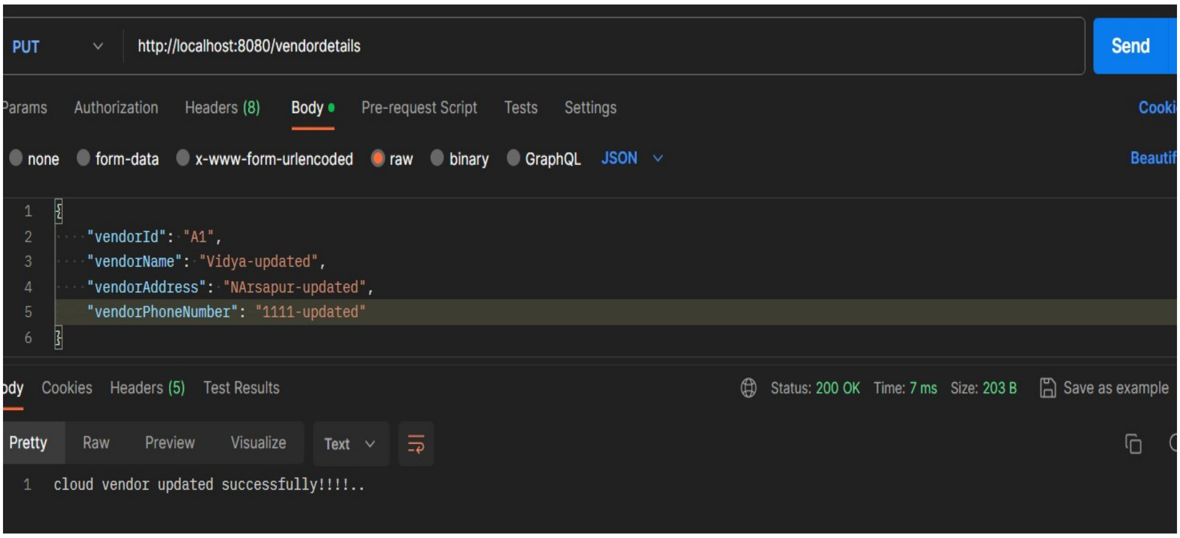
This request does not have a body

Body — Status: 200 OK Time: 23 ms Size: 263 B

Pretty JSON

```
{
    "vendorId": "B1",
    "vendorName": "Vidya Sri",
    "vendorAddress": "Narsapuram",
    "vendorPhoneNumber": "11111"
}
```

11. Create a newrequest file in ****** and name it as Put .Then the details are updated.



PUT http://localhost:8080/vendordetails

Body: ● raw JSON

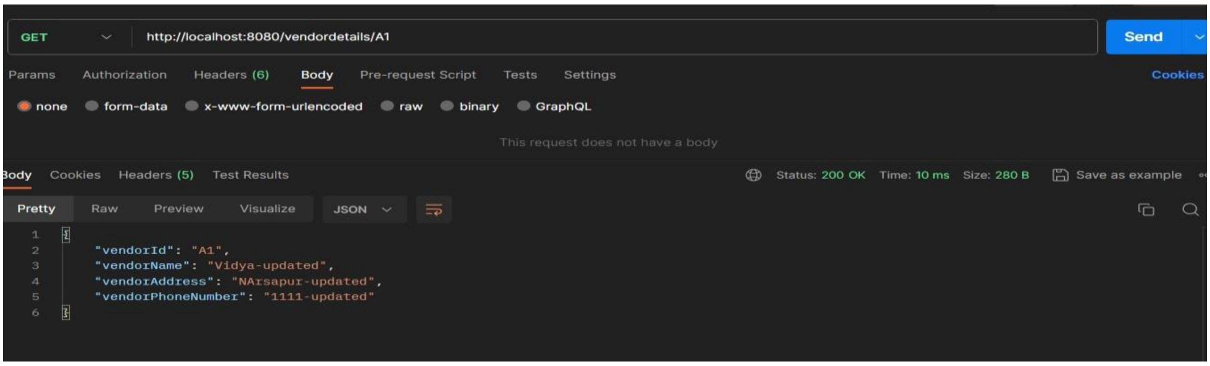
```
{
    "vendorId": "A1",
    "vendorName": "Vidya-updated",
    "vendorAddress": "NArsapur-updated",
    "vendorPhoneNumber": "1111-updated"
}
```

ody — Status: 200 OK Time: 7 ms Size: 203 B

Pretty Text

```
cloud vendor updated successfully!!!!..
```

@GetMapping after performing @PutMapping :



GET http://localhost:8080/vendordetails/A1

Body: ● none

This request does not have a body

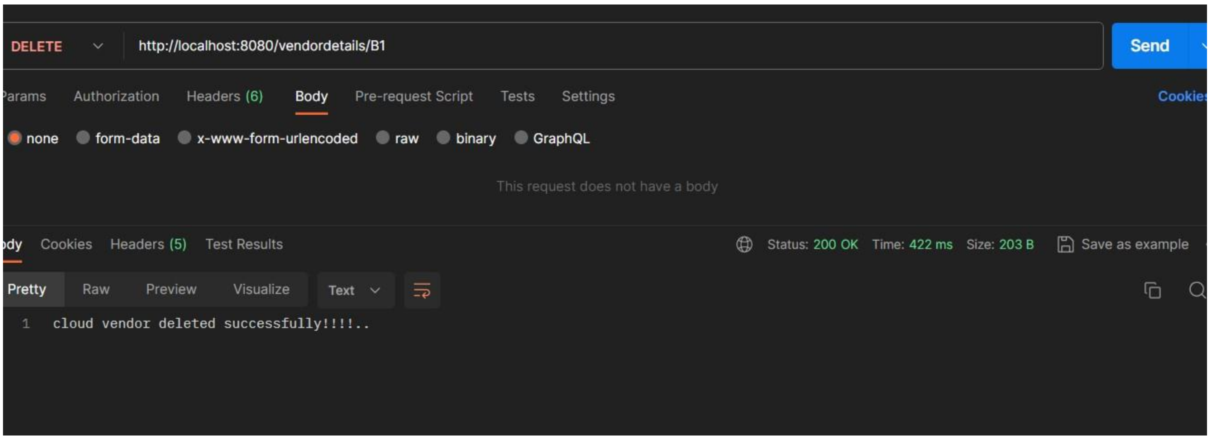
Body — Status: 200 OK Time: 10 ms Size: 280 B

Pretty JSON

```
{
    "vendorId": "A1",
    "vendorName": "Vidya-updated",
    "vendorAddress": "NArsapur-updated",
    "vendorPhoneNumber": "1111-updated"
}
```

@DeleteMapping :

- An HTTP DELETE method that deletes the cloud vendor details associated with the provided vendorId path variable. The method sets the class variable to null and returns a success message.
- **OUTPUT:**



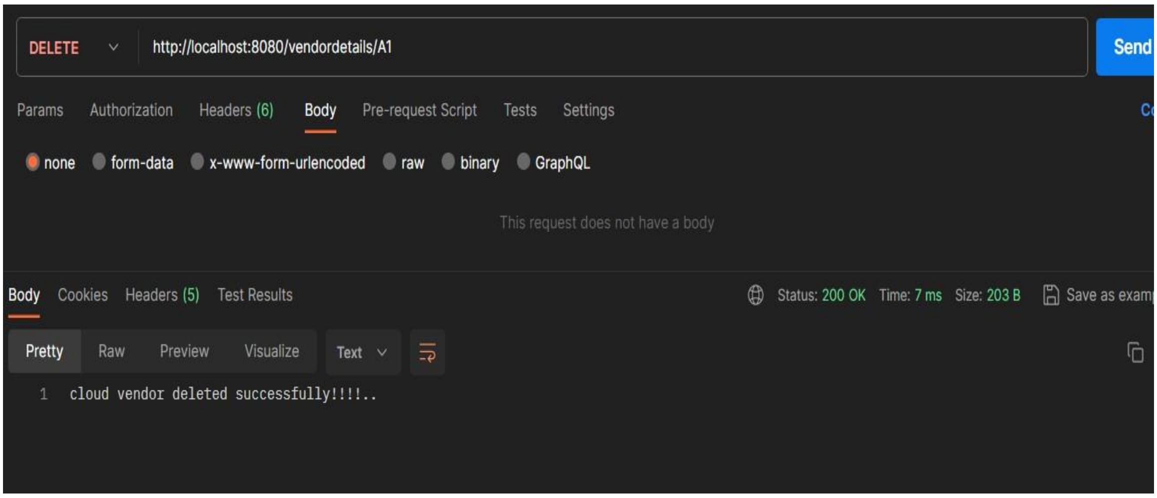
DELETE http://localhost:8080/vendordetails/B1

Body: none

This request does not have a body

Status: 200 OK Time: 422 ms Size: 203 B

```
cloud vendor deleted successfully!!!!..
```



DELETE http://localhost:8080/vendordetails/A1

Body: none

This request does not have a body

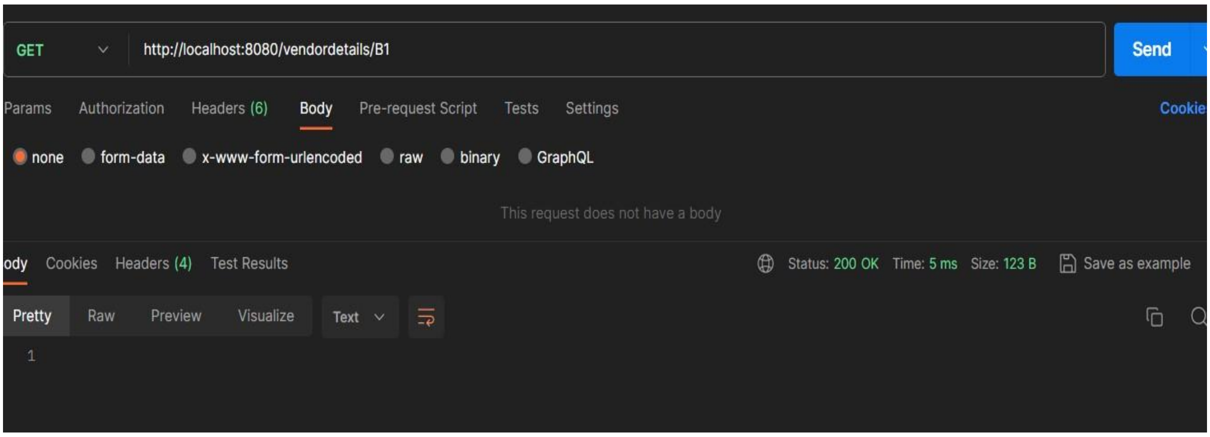
Status: 200 OK Time: 7 ms Size: 203 B

```
cloud vendor deleted successfully!!!!..
```

14. Check in the Get request to confirm whether the details or deleted or not.

@GetMapping after performing DeleteMapping :

OUTPUT :



GET http://localhost:8080/vendordetails/B1

Body: ● none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

This request does not have a body

Status: 200 OK Time: 5 ms Size: 123 B

```
1
```