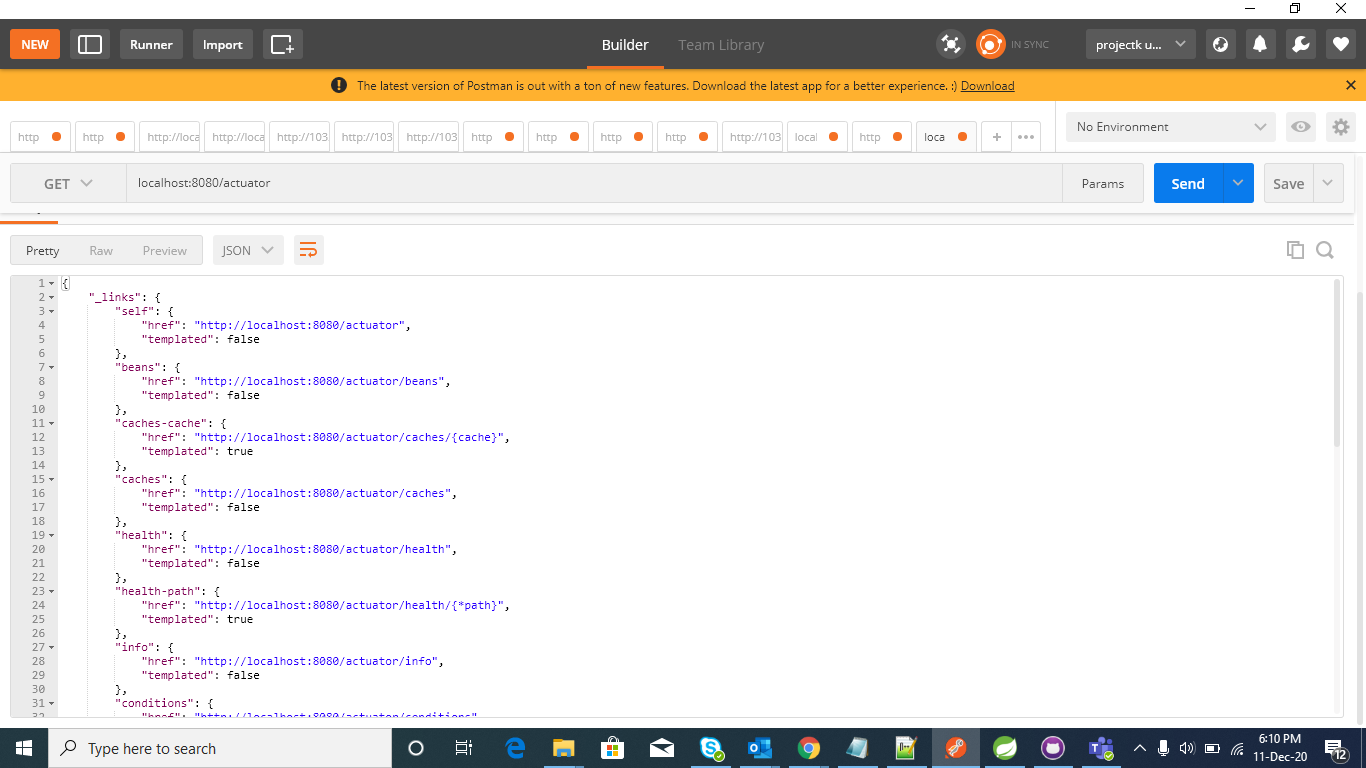


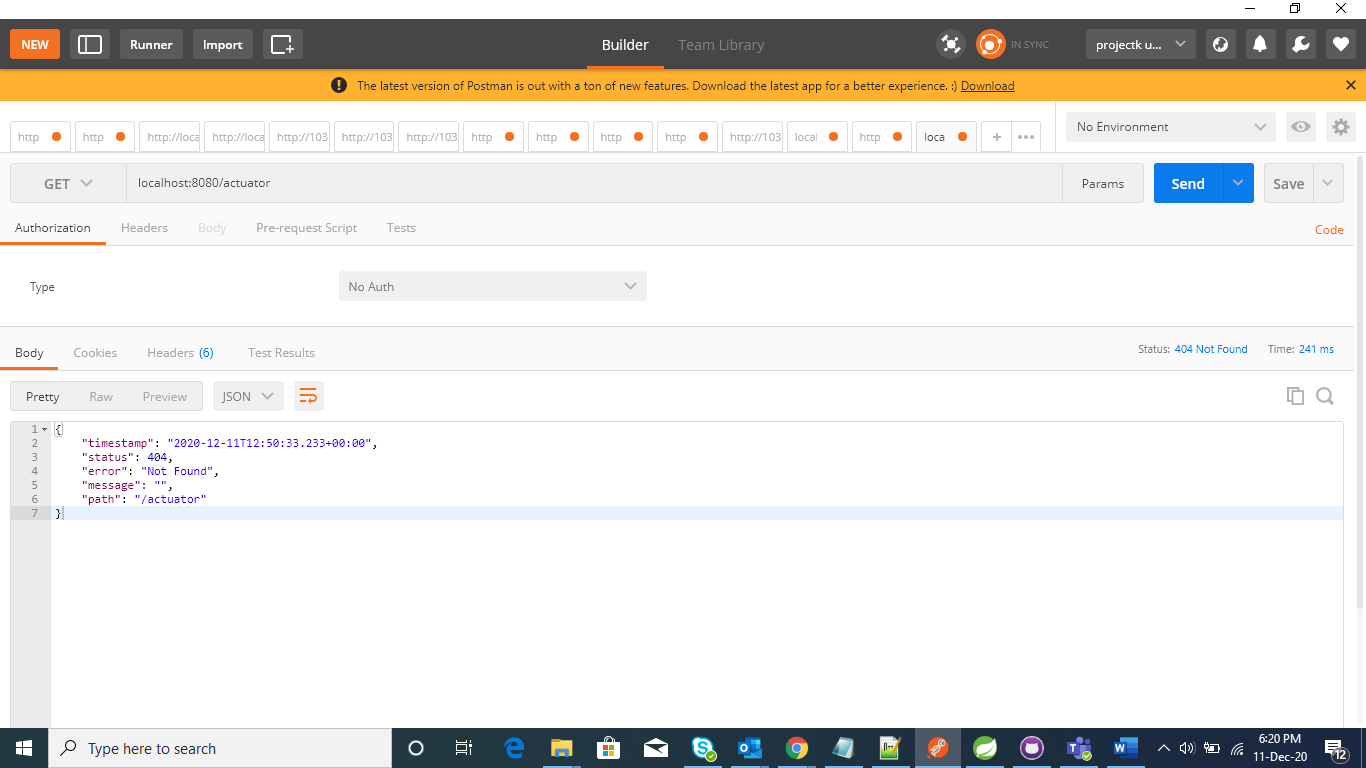
application.properties

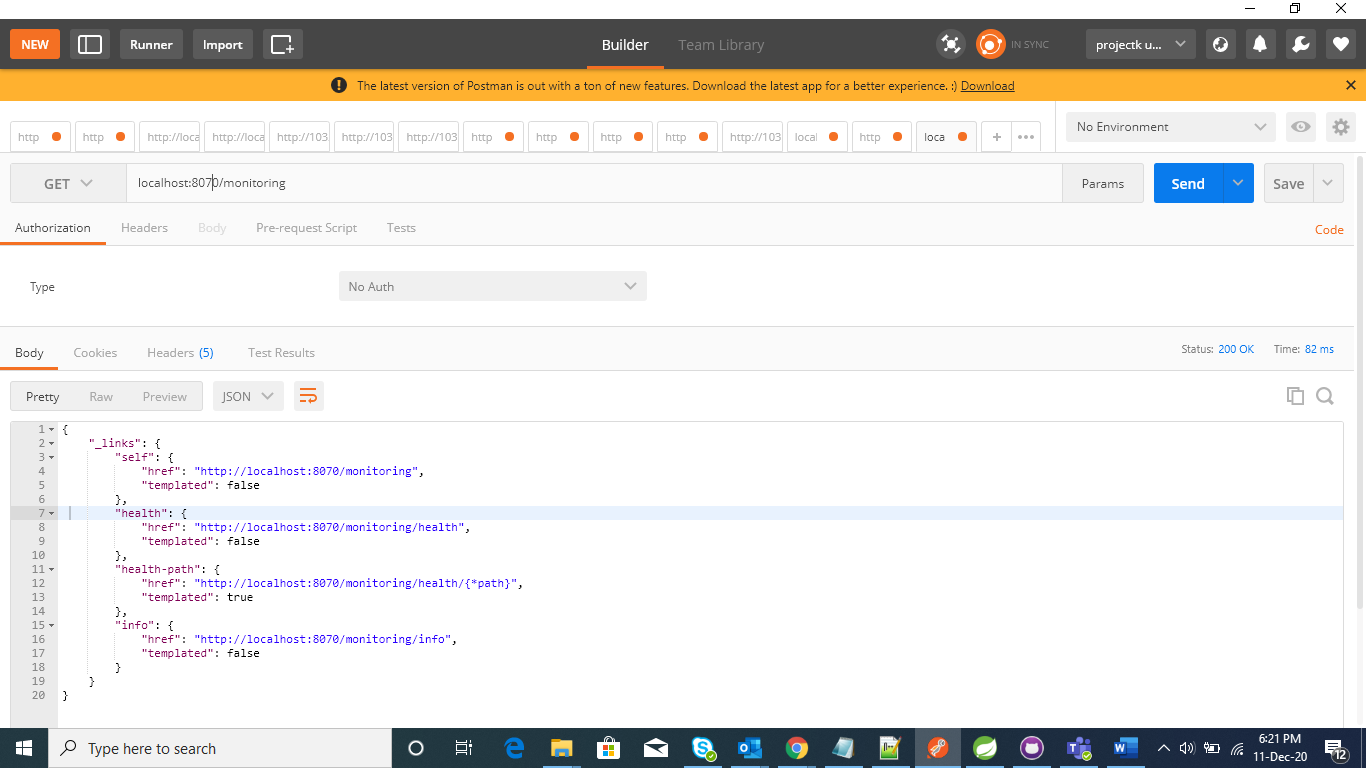
management.endpoints.web.exposure.include=\*

The Spring Boot Actuator module provides the facility to change the management port and path by adding some property in application.properties file.

management.endpoints.web.base-path=/monitoring

management.server.port=8070



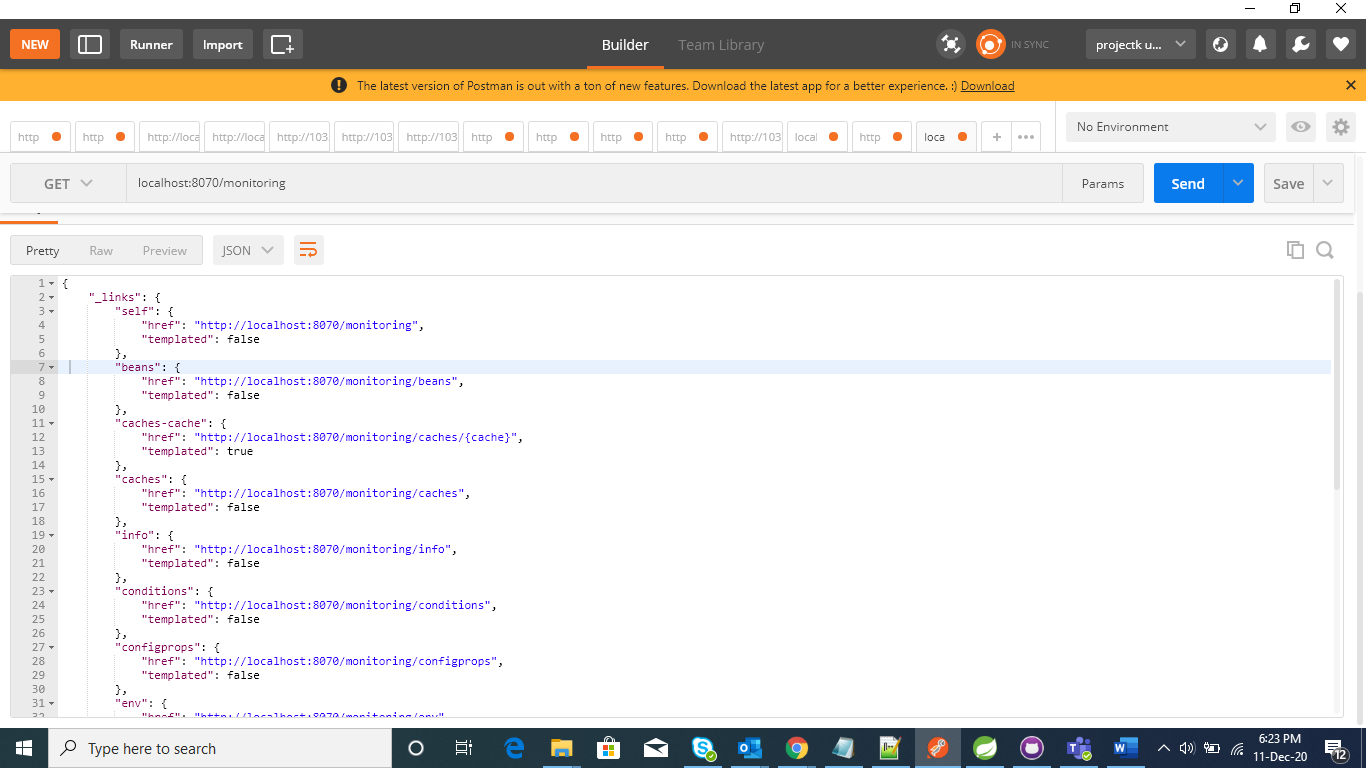


management.endpoints.jmx.exposure.include=\*

management.endpoints.jmx.exposure.exclude=health, metrics

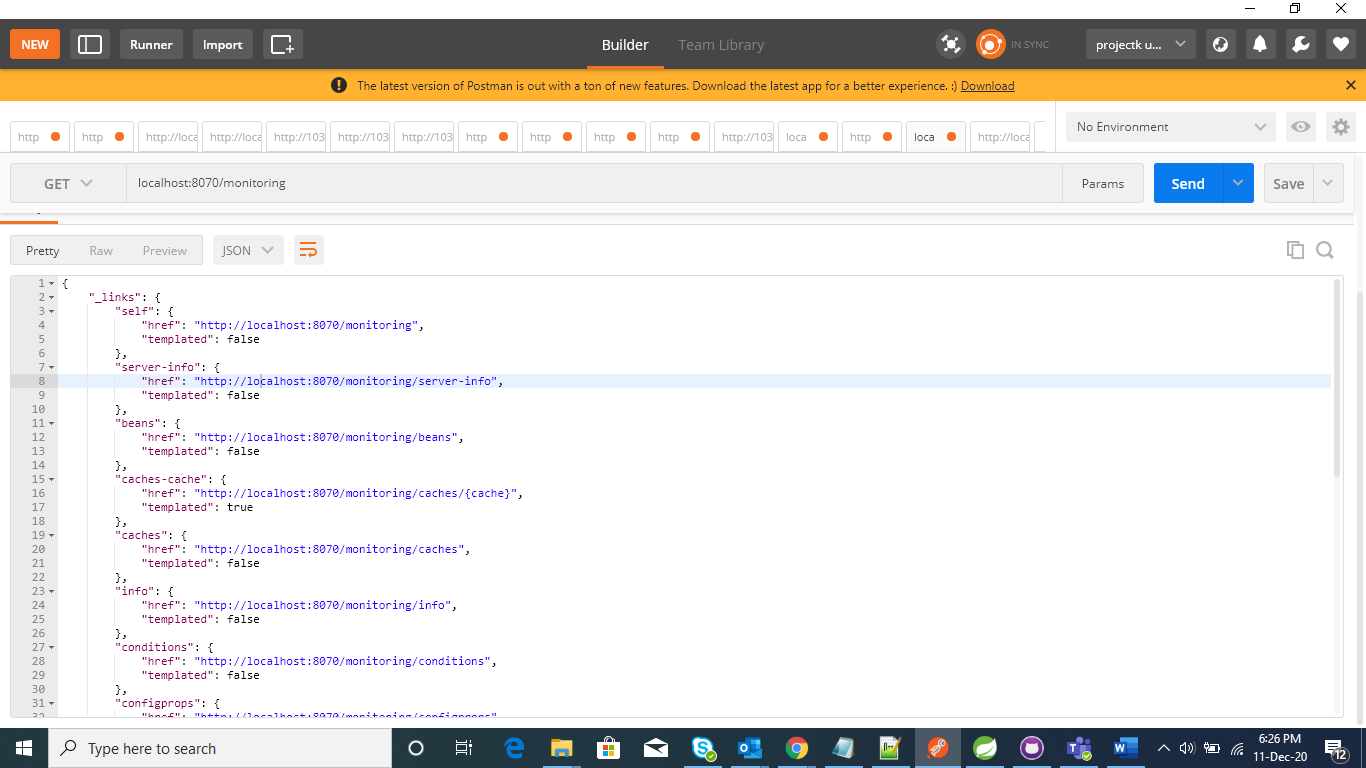
management.endpoints.web.exposure.include=\*

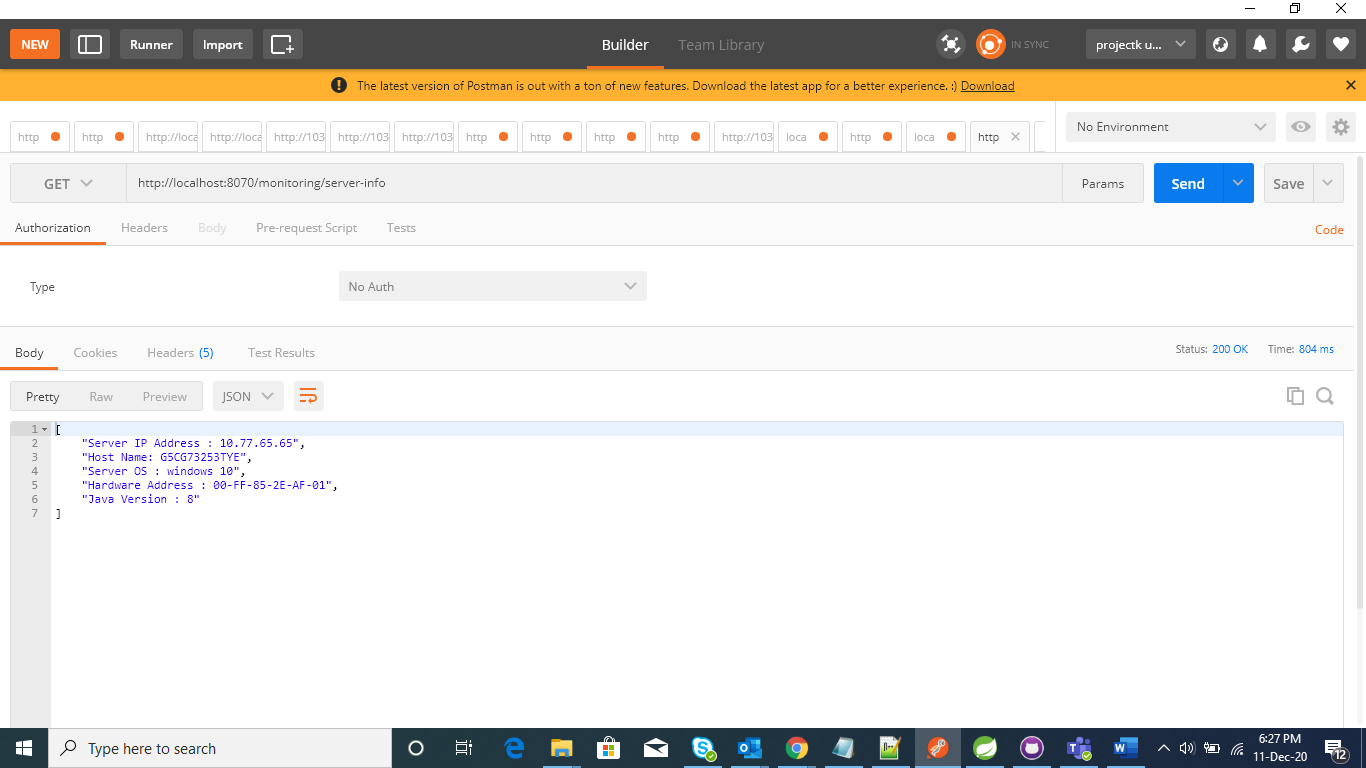
management.endpoints.web.exposure.exclude=health, metrics



**Add Custom Endpoints**

Spring Boot Actuator provides the facility to write custom endpoints from where we can see our own customized application data. For example, I am using a custom endpoint to check my server address, server deployed in which OS, MAC address of the server and which java version installed in the server machine.

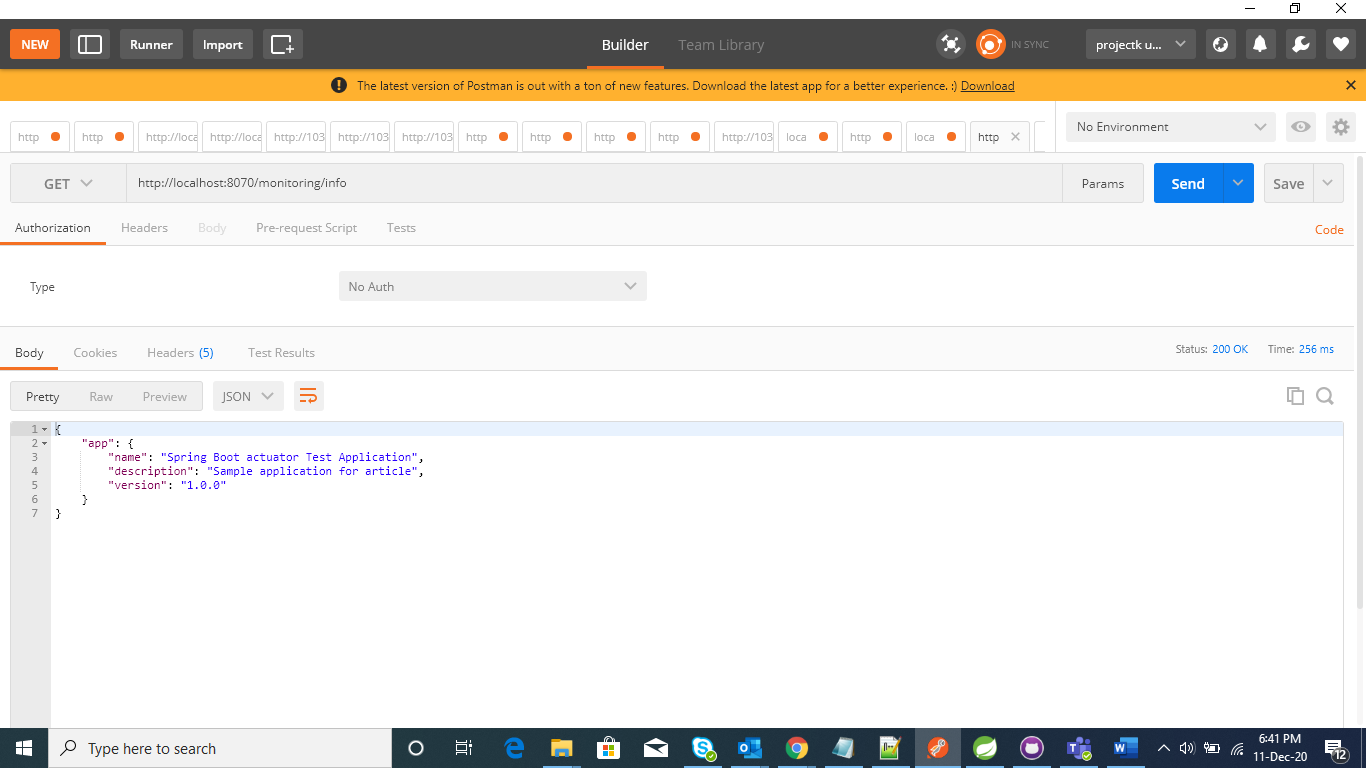




info.app.name=Spring Boot actuator Test Application

info.app.description=Sample application for article

info.app.version=1.0.0



We can also add other information like build info, git info of application.

For build info, we have to add

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

<version>1.5.9.RELEASE</version>

<executions>

<execution>

<goals>

<goal>build-info</goal>

</goals>

<configuration>

<additionalProperties>

<encoding.source>UTF-8</encoding.source>

<encoding.reporting>UTF-8</encoding.reporting>

<java.source>${maven.compiler.source}</java.source>

<java.target>${maven.compiler.target}</java.target>

</additionalProperties>

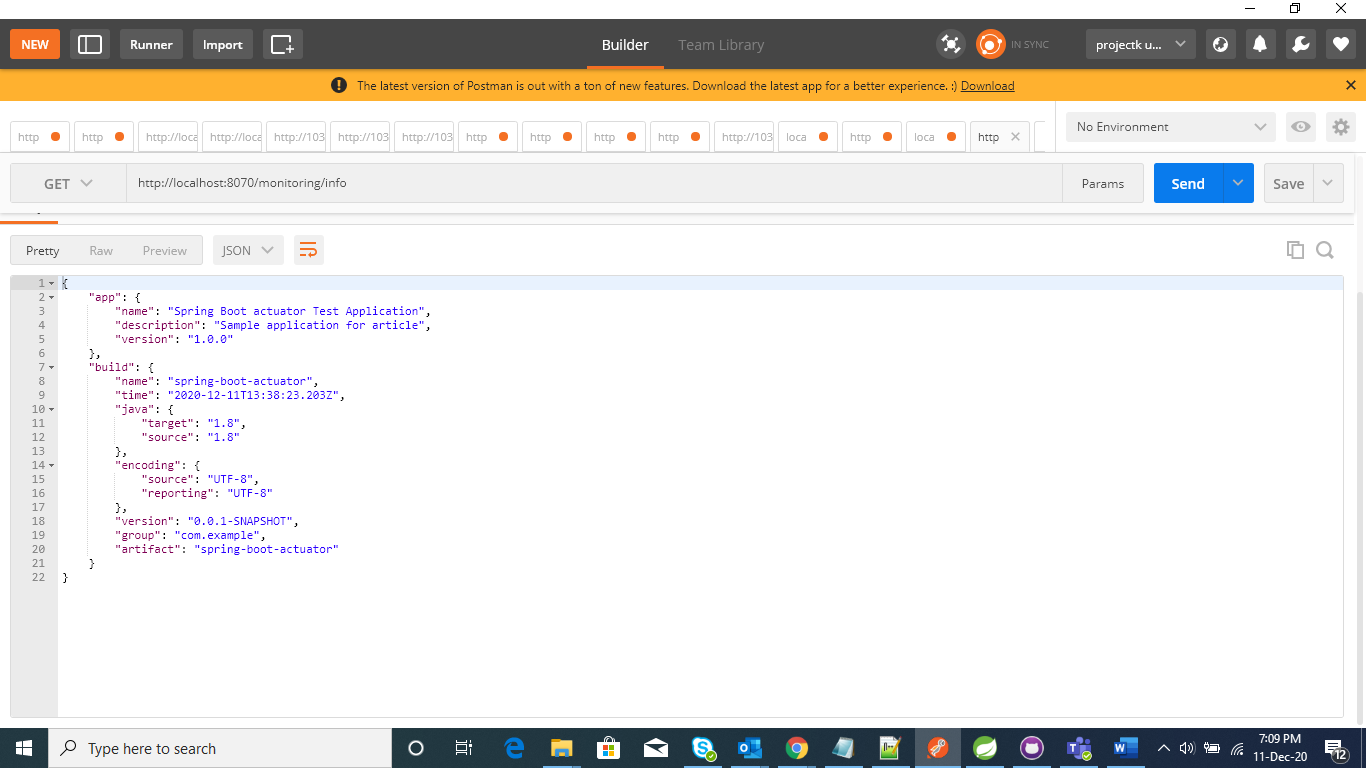
</configuration>

</execution>

</executions>

...

</plugin>



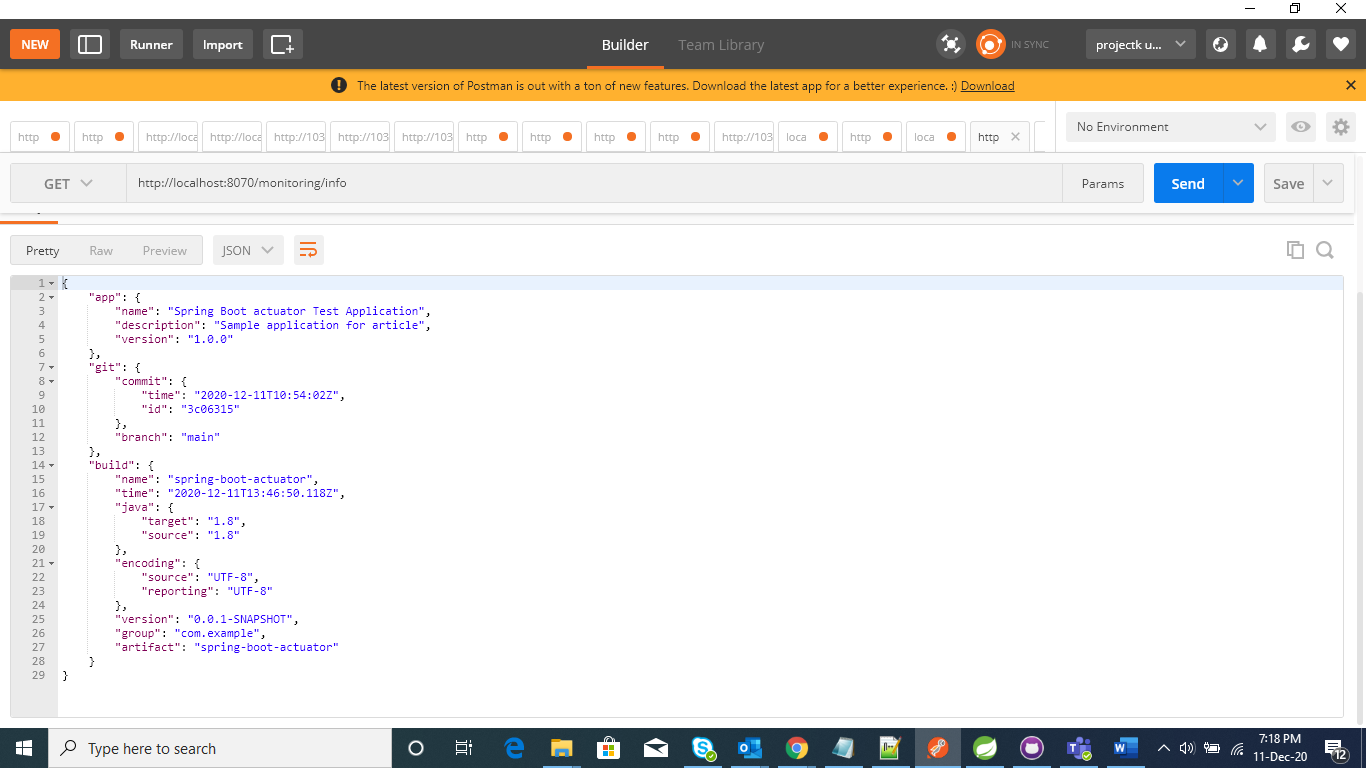
Add below code to get git information

<plugin>

<groupId>pl.project13.maven</groupId>

<artifactId>git-commit-id-plugin</artifactId>

</plugin>



We can see the custom information of the application by using the InfoContributor interface.

We will create CustomInfoContributor.java by implementing InfoContributor.

@Component

**public** **class** CustomInfoContributor **implements** InfoContributor {

@Override

**public** **void** contribute(Info.Builder builder) {

builder.withDetail("customInfo", "This is custom info indicator. You can add your application data. "

+ "You can share application persistent data from here");

}

}

