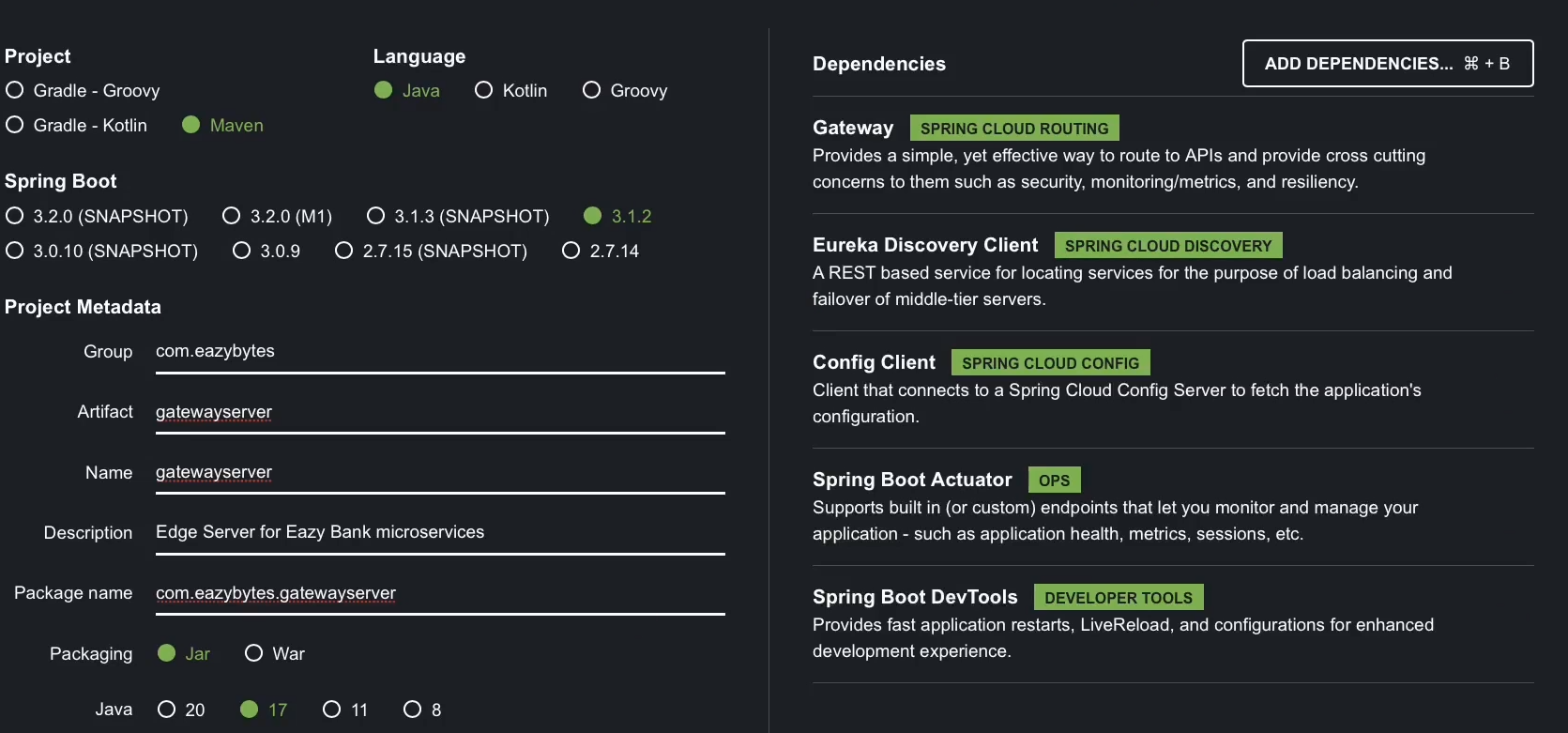
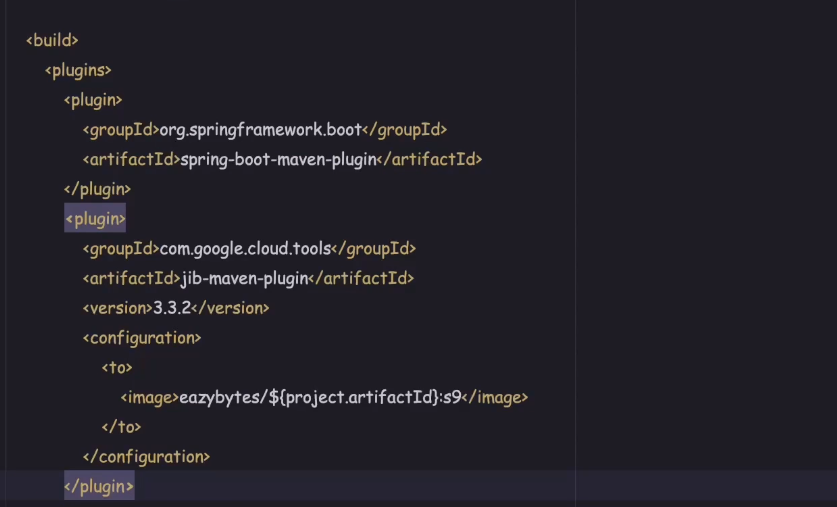
Spring cloud API gateway configuration

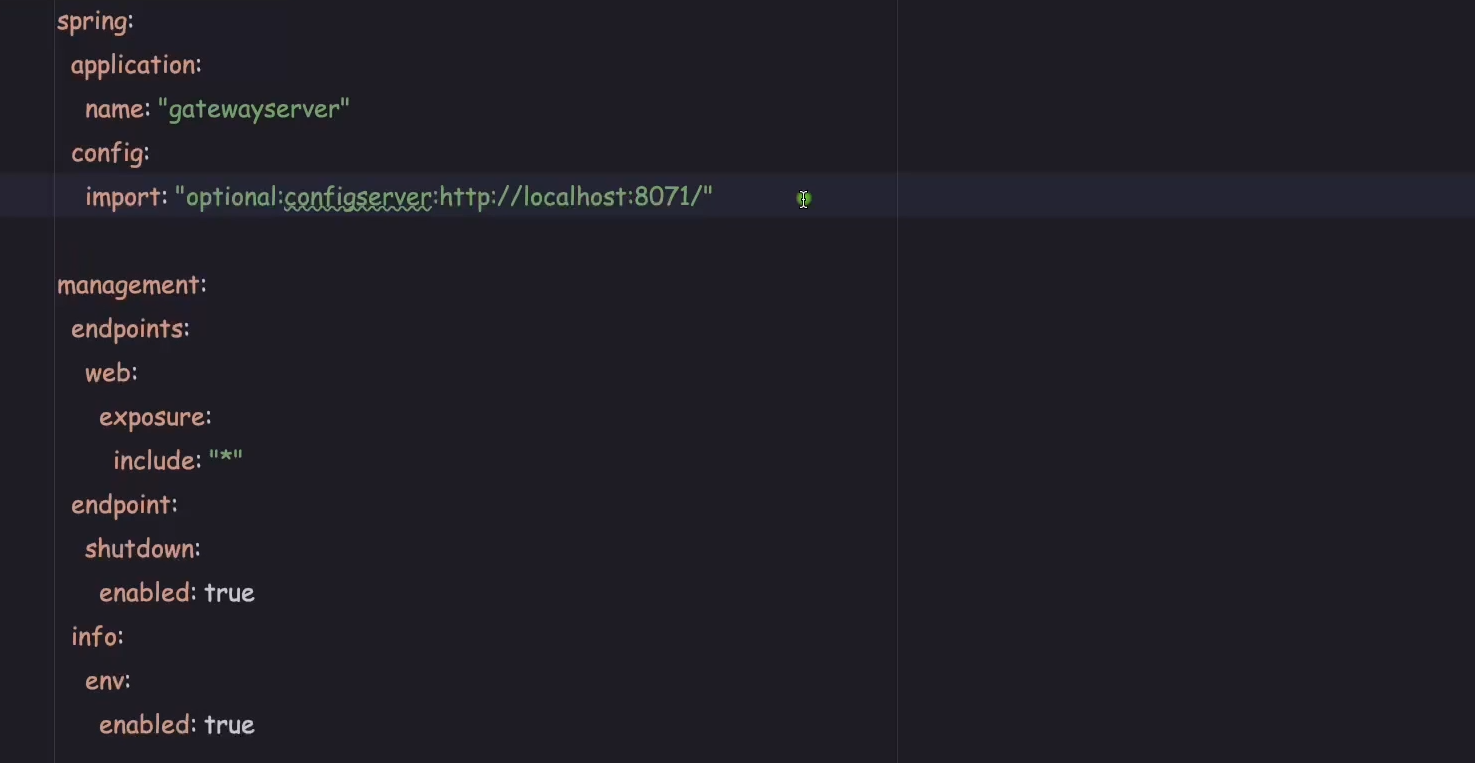
Create a spring cloud gateway server

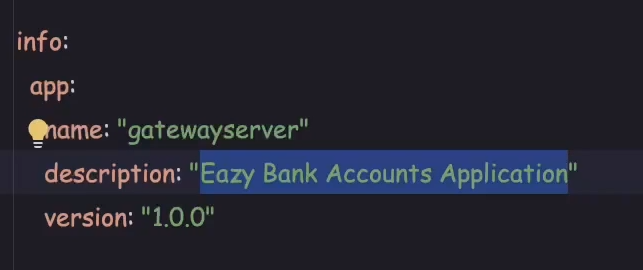


Add google jib to build docker image in pom.xml



Application.properties config

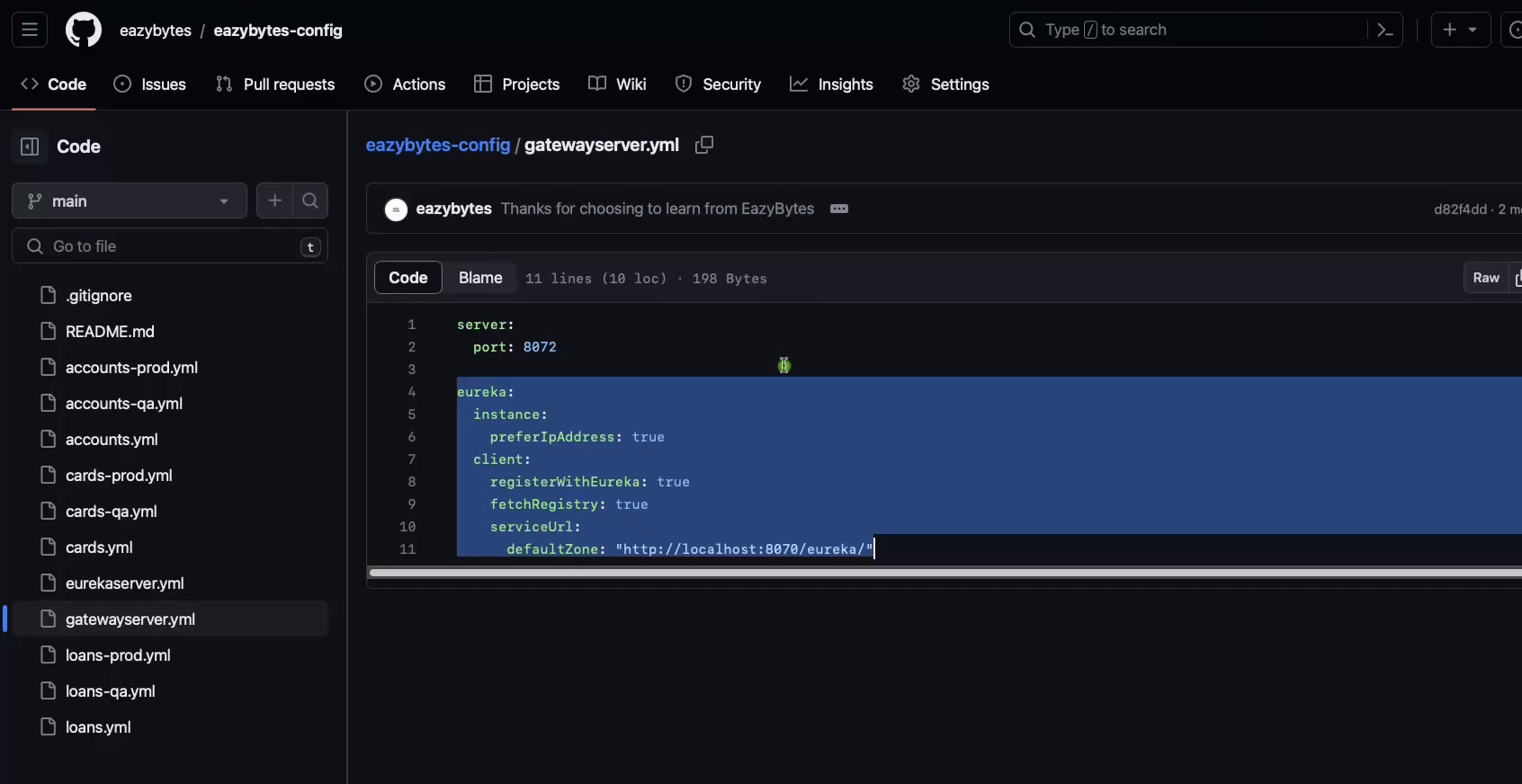




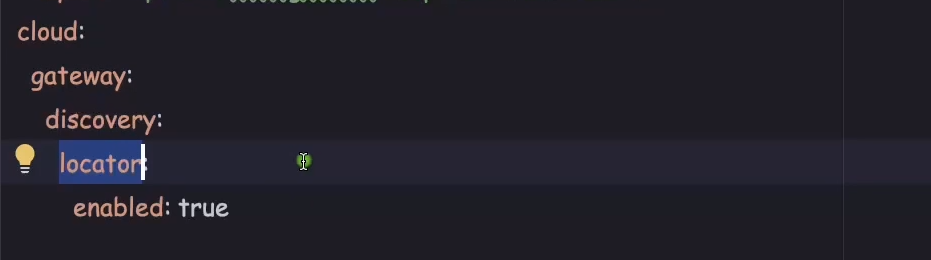
Enable gateway related end points



Gate way server config properties in git hub fetched using spring cloud config server



Add below properties in pom.xml so that our gateway server communicates with service registry eureka server



First start config server

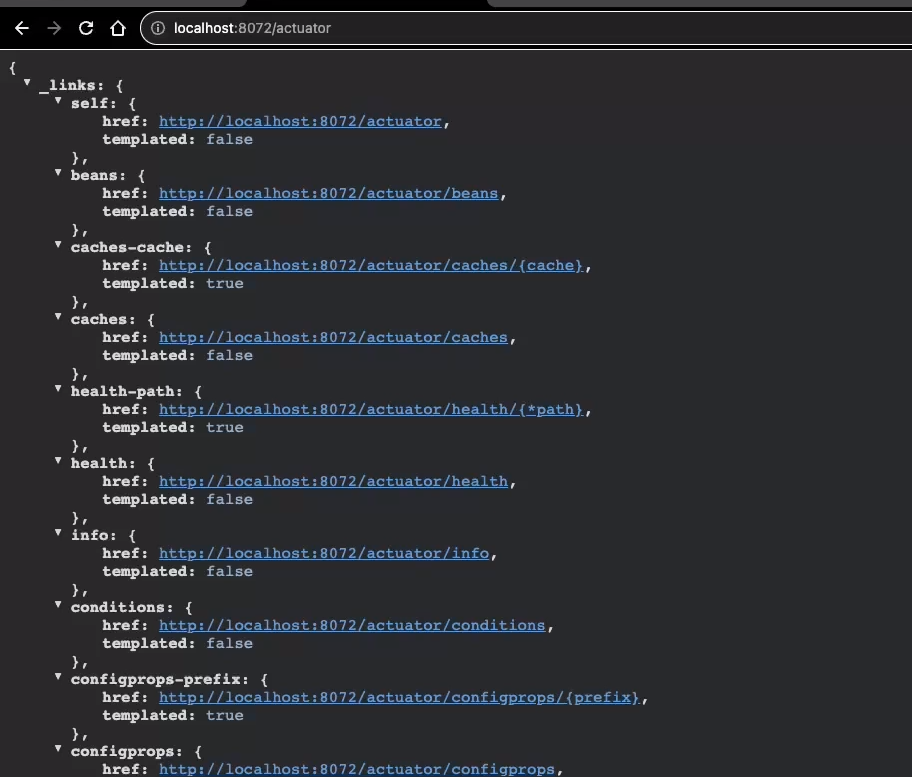
Then start eureka server

Then start all the microservices

Then start the API cloud gateway

Then open actuator

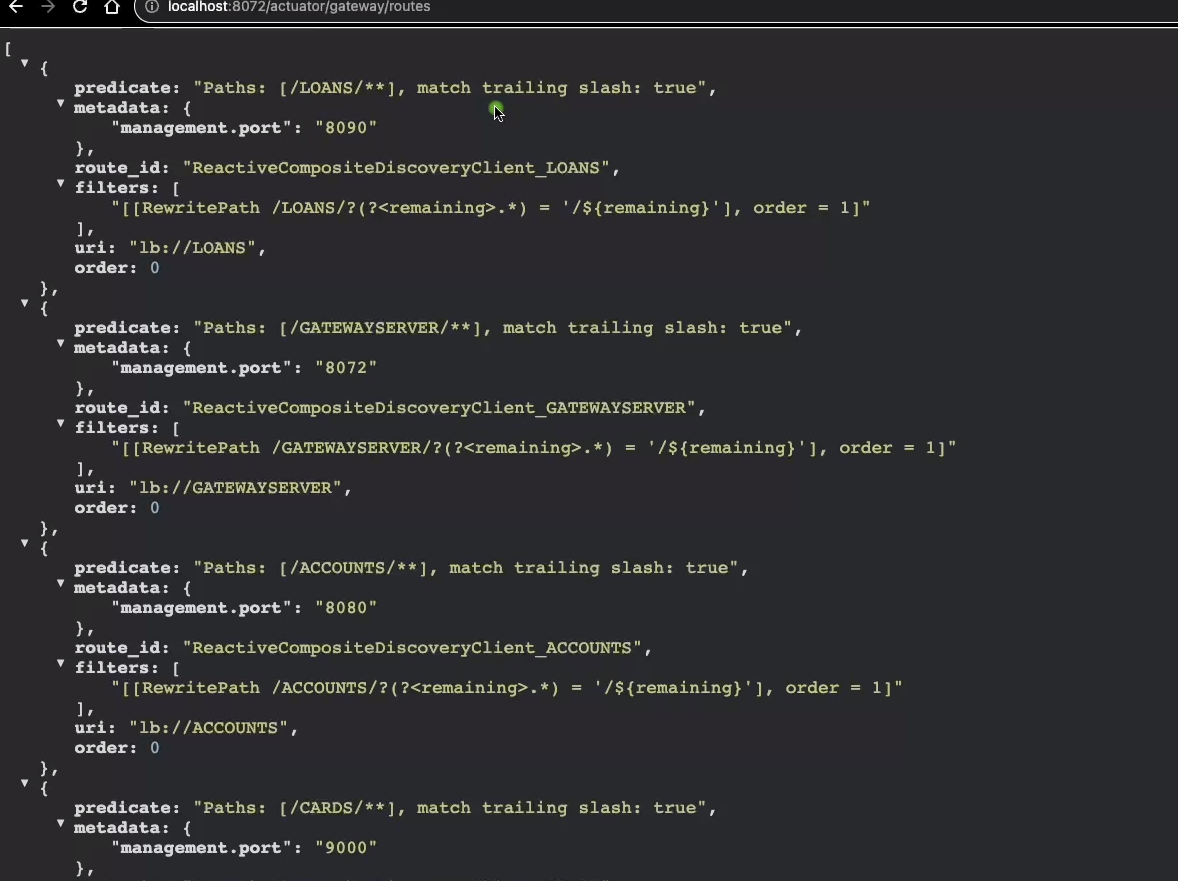
<http://localhost:8072/actuator>



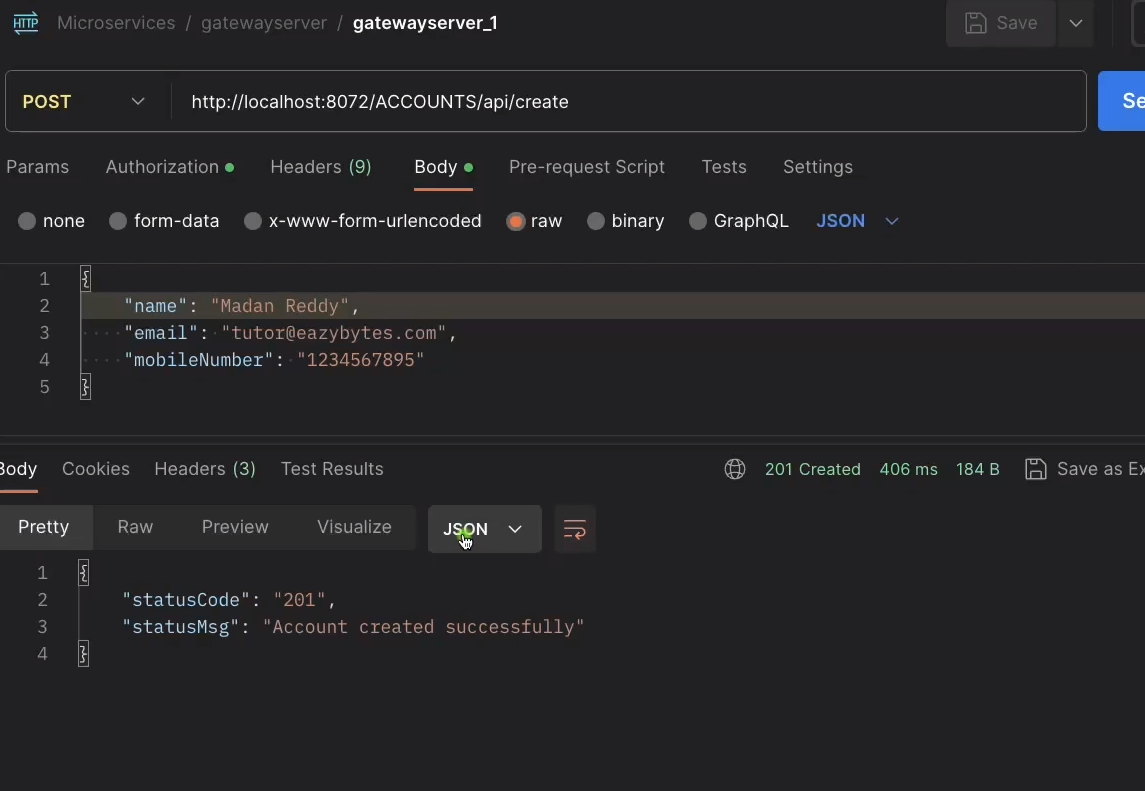
Then click

<http://localhost:8072/actuator/gateway/routes>

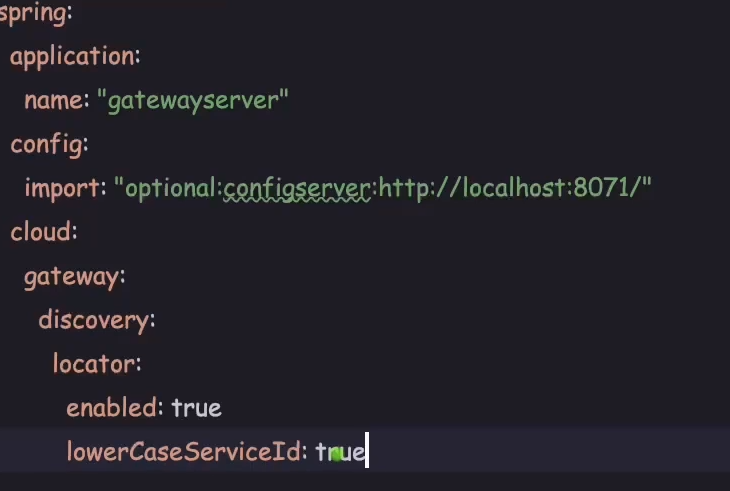
It shows all the services related to microservices and how the url mapping is done all url with ACCOUNTS/\*\* will be forwarded to accounts service



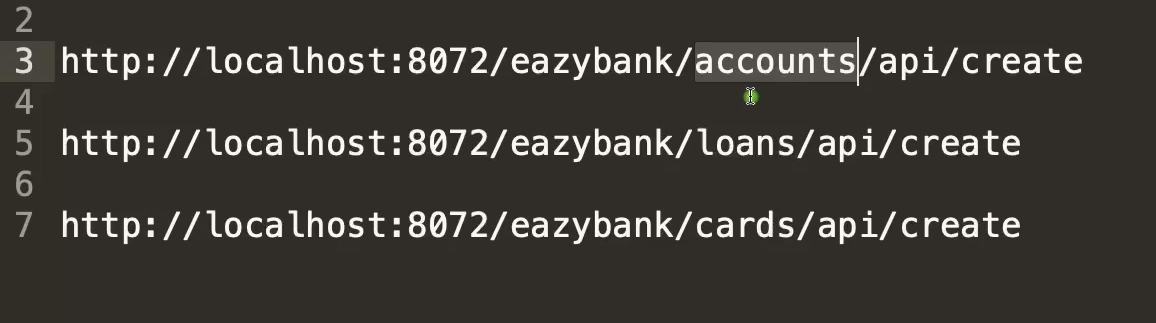
Now call the accounts microservice via gatewayserver



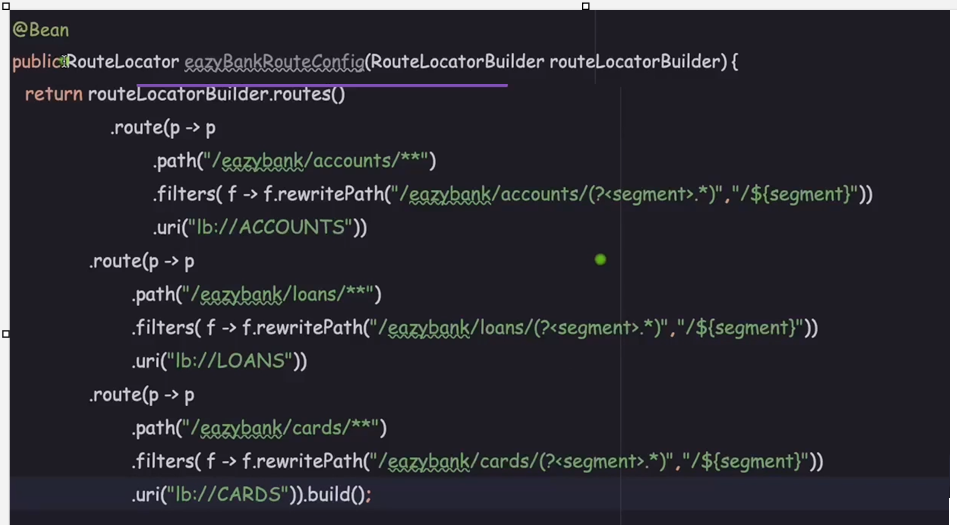
By default gateway server configures microservices name in capital we can change it to lowercase by adding lowerCaseServiceId in property file



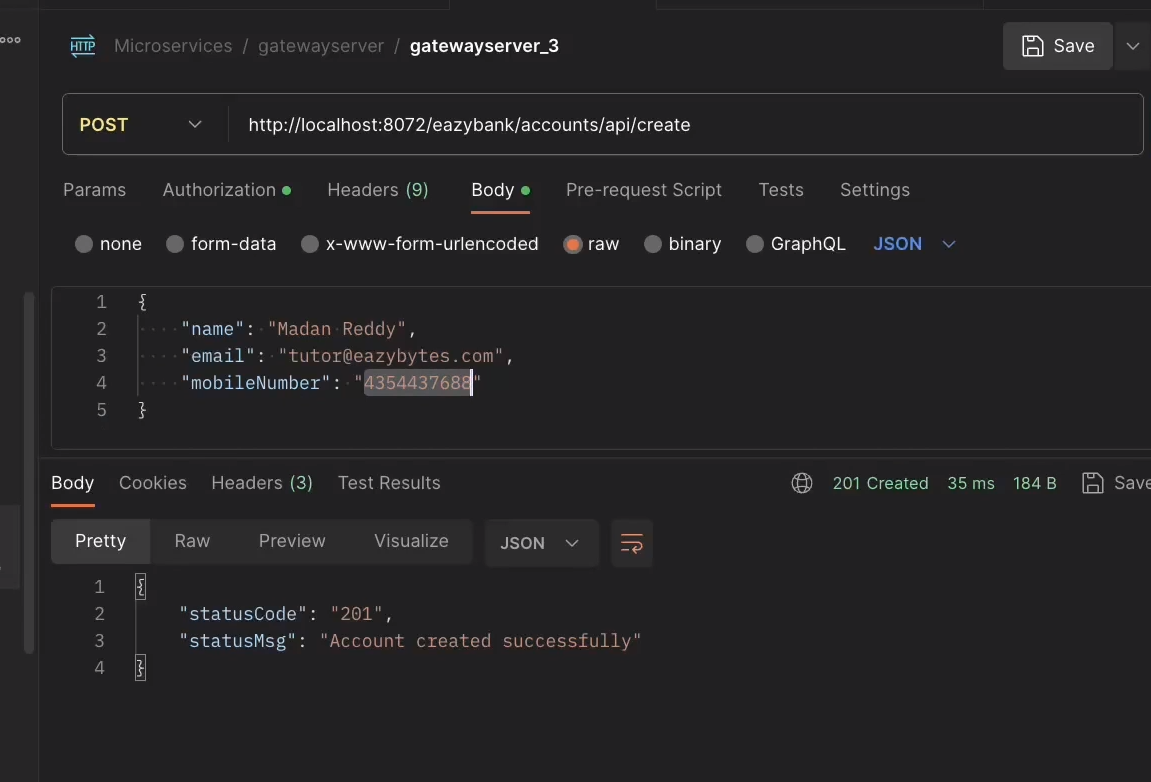
Implementing custom routing like url call for the microservices with appending application name



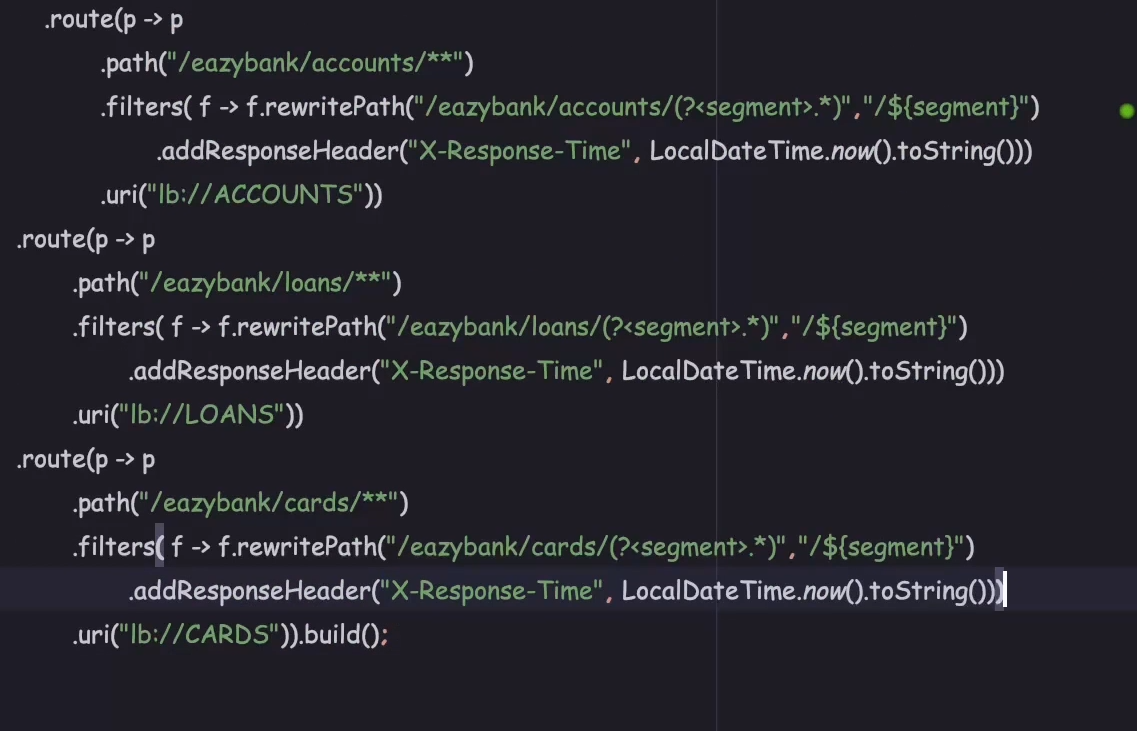
For this we need to configure in java application main file as below also please note the lb is appended with application name same as in eureka dashboard its in capital

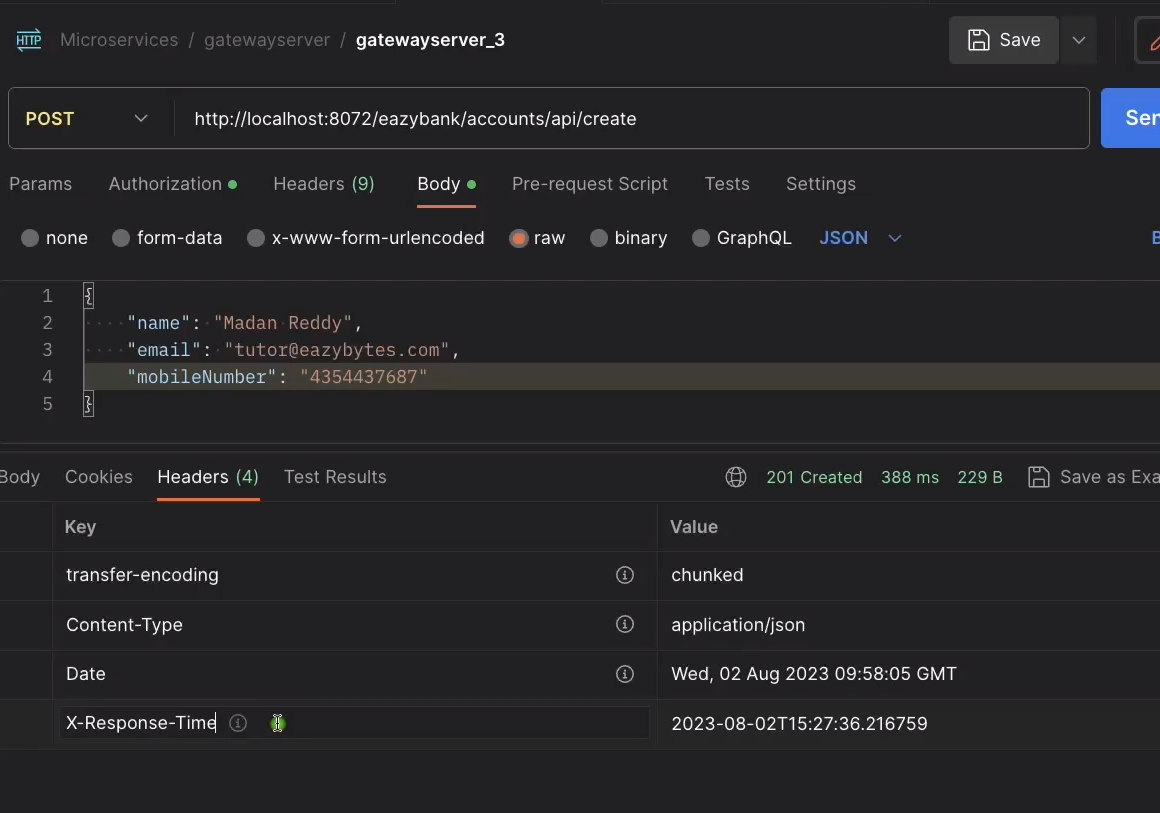


Now call with custom url you can see its working fine now



Add response header filter





Tracing and logging

Will add corelation id in headers and will forward it to microservice so that if any exception then we can chk for this id till where and which microservices it had travelled.

Check for video topic 127 bcoz code is not explained directly added