Description

1. Framework –
   1. serviceRouge – Dropwizard Java 11+
   2. Serviceundererred – Spring Boot Java 11+
2. The repo contains two microservices
   1. serviceRouge – this is a highly unreliable and unavailable server. To mimic the unavailability the modules is configured in such a way that once the API call is made it will return with 504 Gateway Timeout with 2 sec delay. And continue to do the same until the minute value on the server changes.

Eg. If first API call is made at 11:30am then all API calls made before 11:31 will return with 504 Gateway Timeout and after that the first API call on or after 11:31 will return with 202 Hello from server and for the next call again 504 until minute value changes.

* 1. Seviceundererred – a microservice to interact with serviceRouge and make exponential backoff and after every retry it updates the client.

The module also contains a web application with web socket connection which acts as a client and receives updates from the user.

The module uses exponential backoff with time delay with

timeDelay= defaultTime \* factor^(attemptCount) + Random(constant)

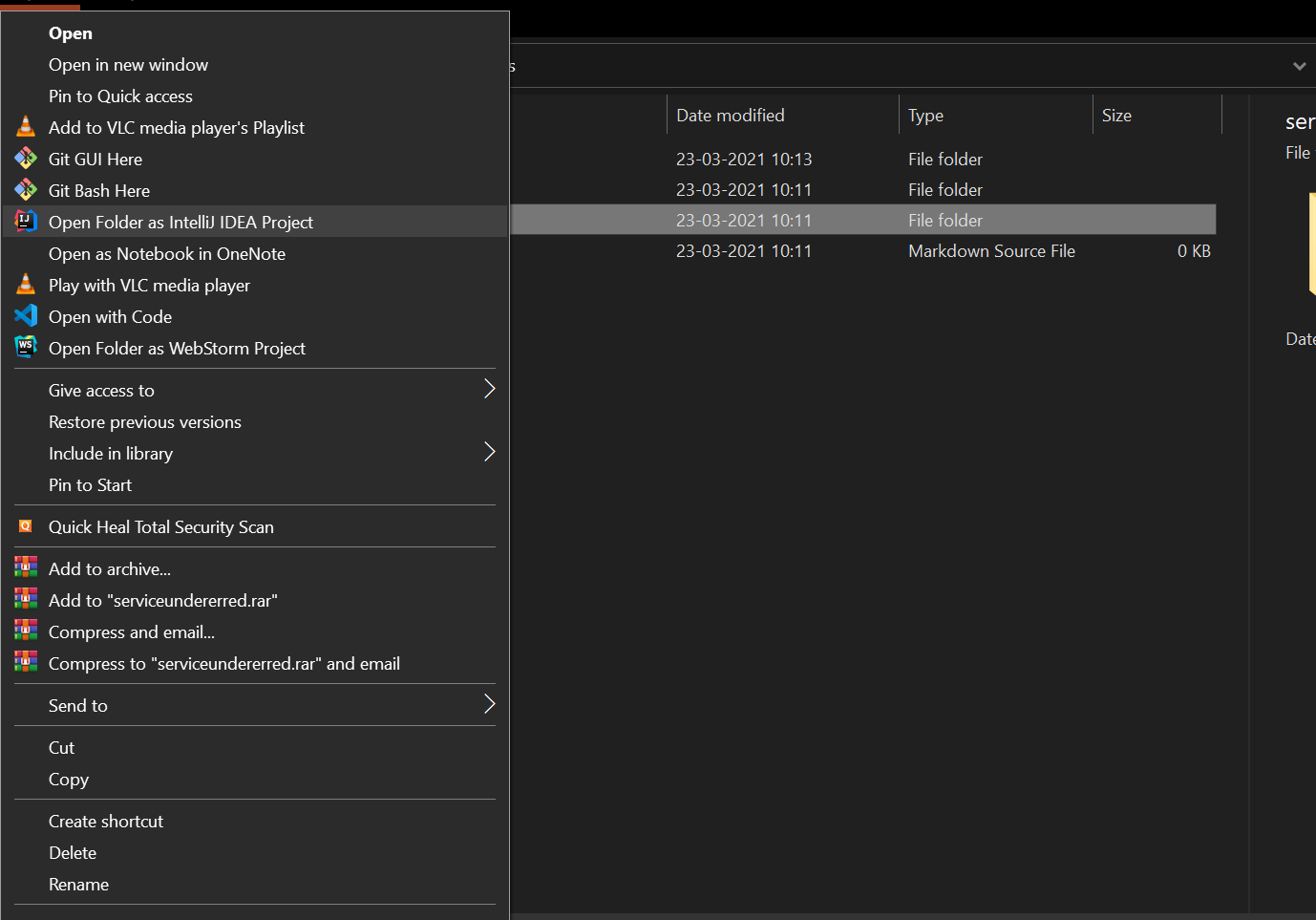
the system is configured with values as

1. DefaultTime=1000ms
2. Factor=2
3. AttemptCount= attempt count till 5 attempts at max
4. Constant=random constant = 1000ms

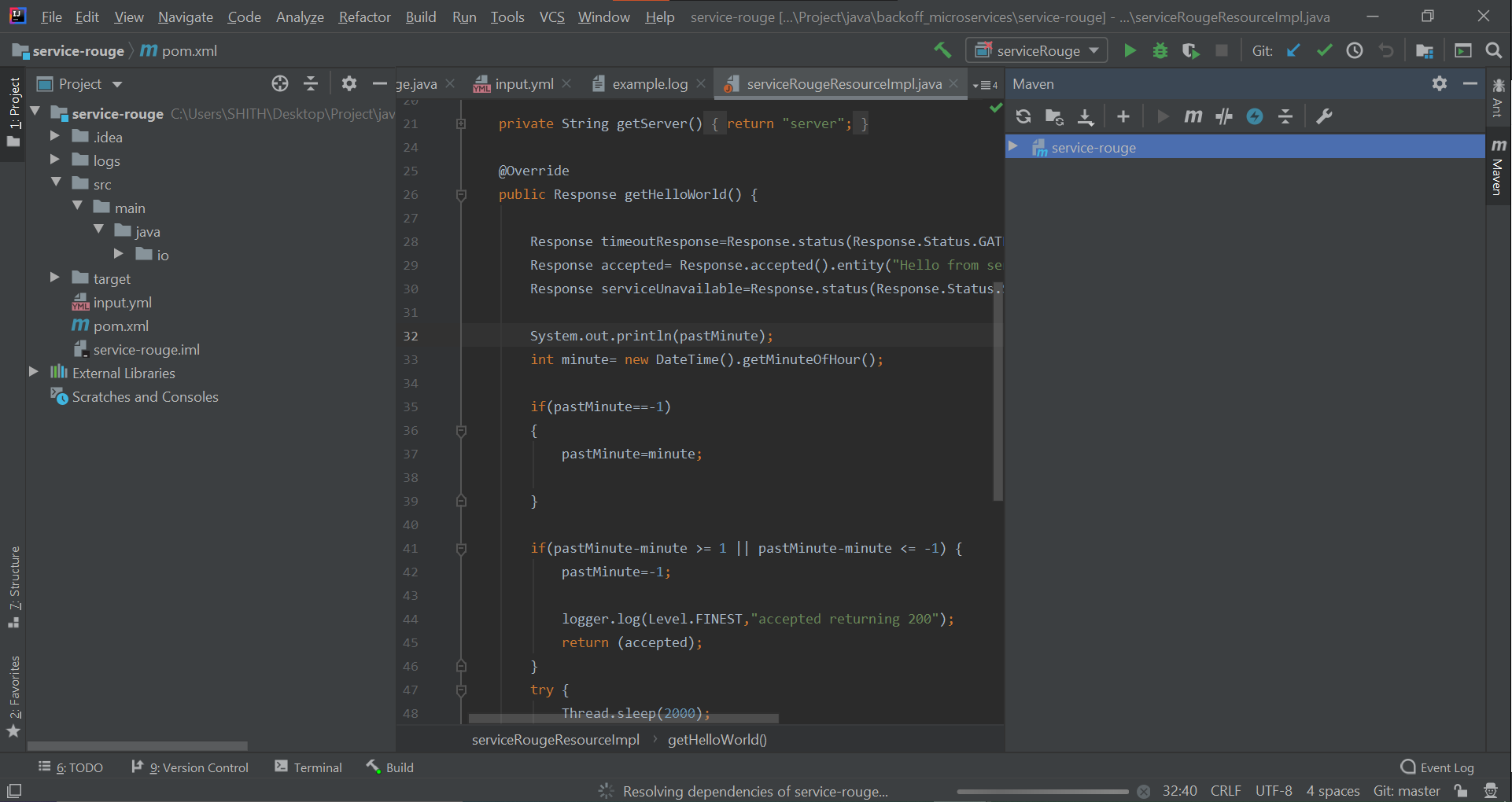
This allows system to make api call with exponential delay of 1s, 2s, 4s, 8s, 16s, 32s as we have configured serviceRouge to give success response atleast one in 60s so there will always a successful result. But if you reduce the max retrycount to 2 then after 2nd attempt the system will stop.

Steps to build and run

1. Clone the repository – git clone <https://github.com/SHITHANSHU/backoff_microservices.git>
2. Open each microservice in IDE like IntelliJ with Java 11 or above

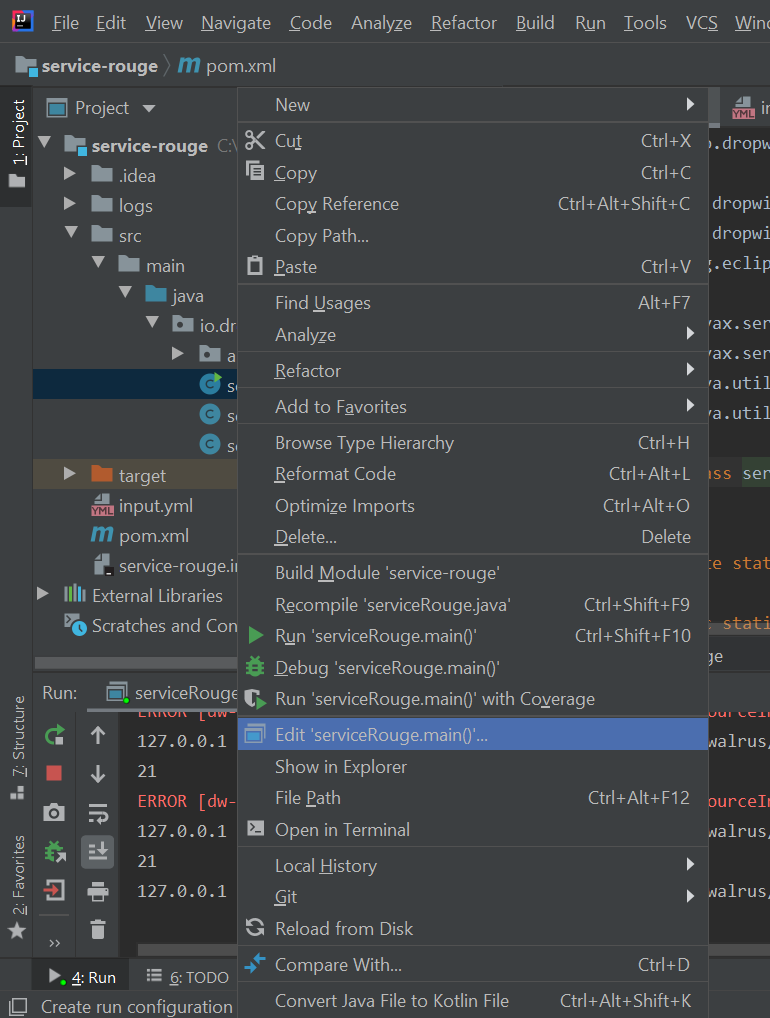


1. Allow some time to the IDE to download necessary jar files using maven

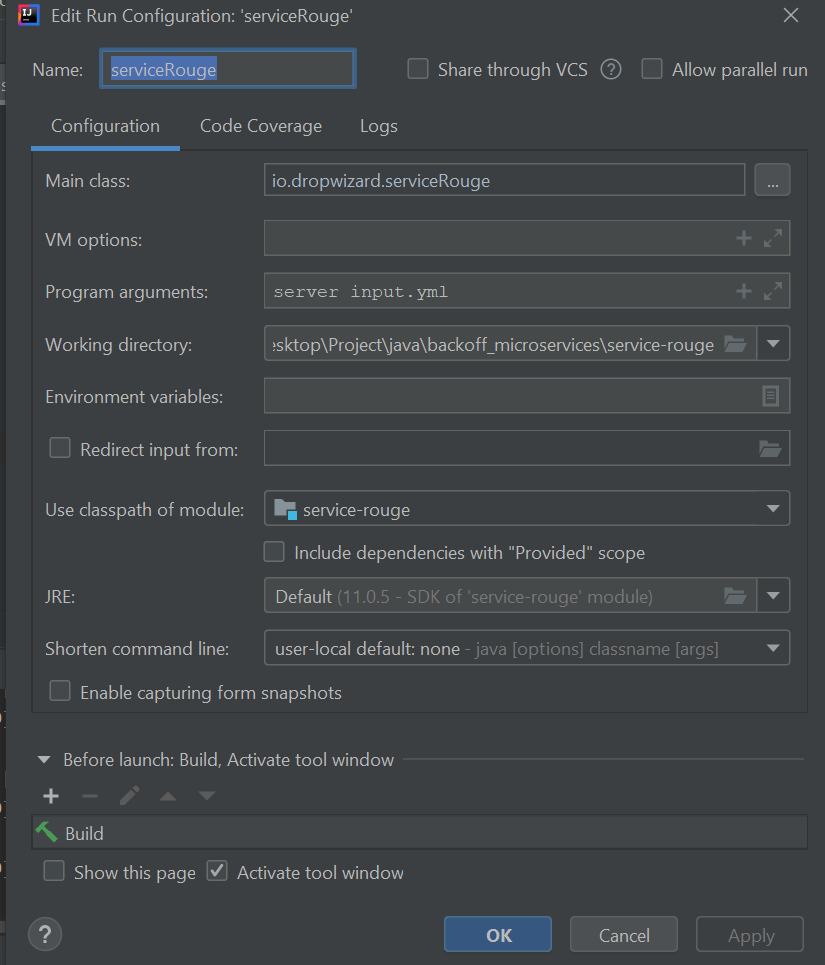


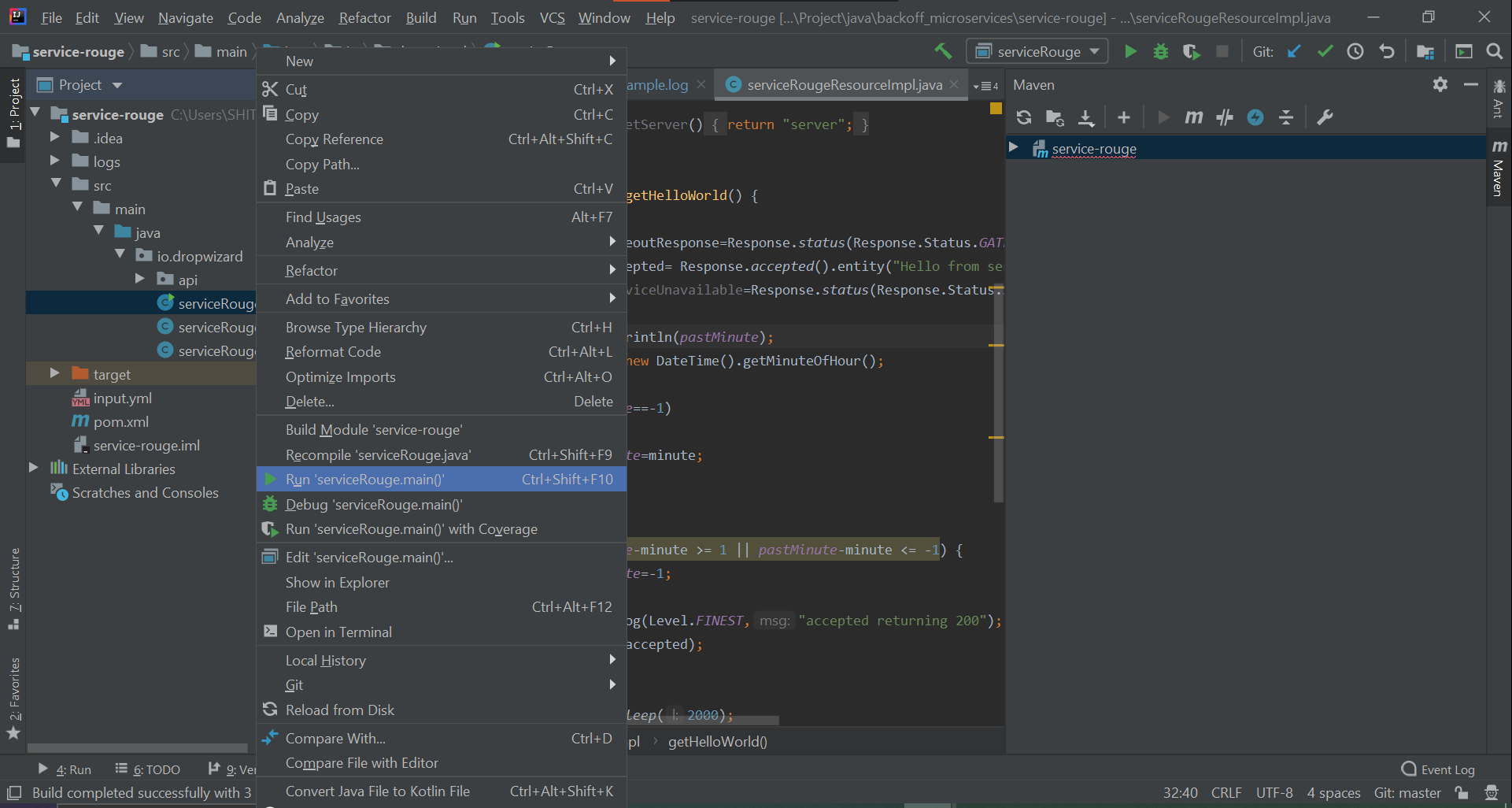
1. For ServiceRouge run the program in io.dropwizard serviceRouge

You might need to configure the program argument by rightclicking the main file and select edit parameters as

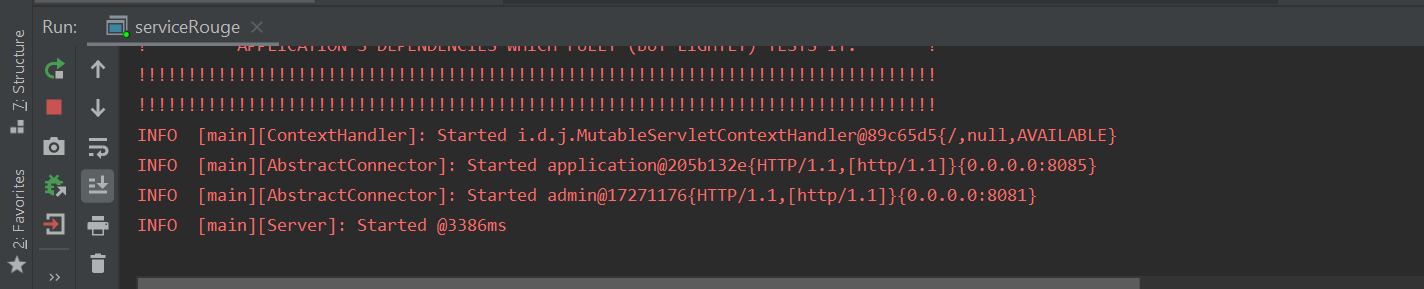


Set Program argument as server input.yml

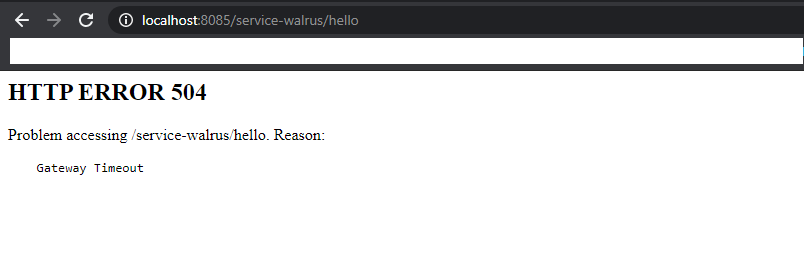




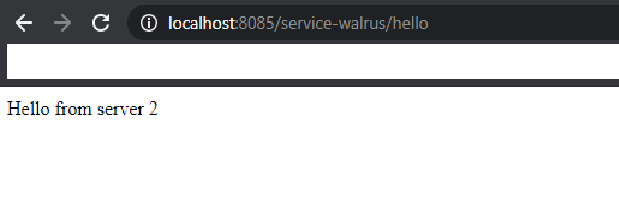
1. On getting this screen the ServiceRouge is ready to listen at port 8085



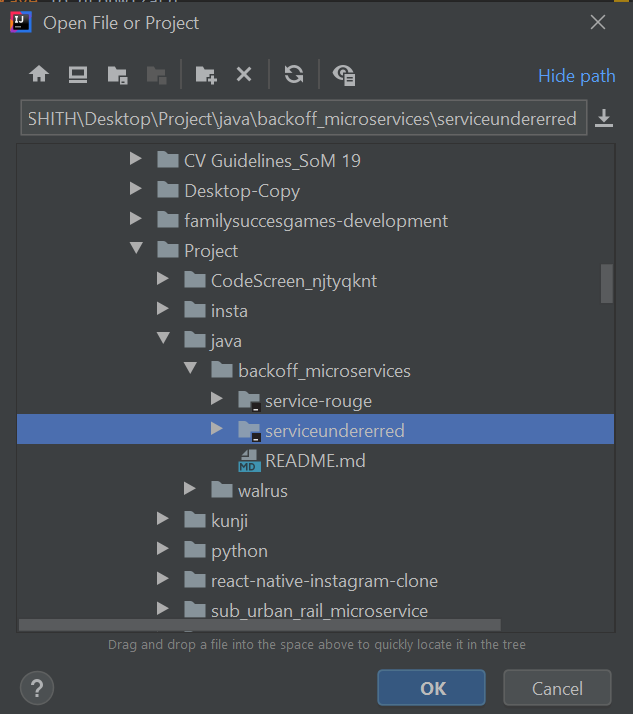
1. You can test this by hitting <http://localhost:8085/service-walrus/hello>



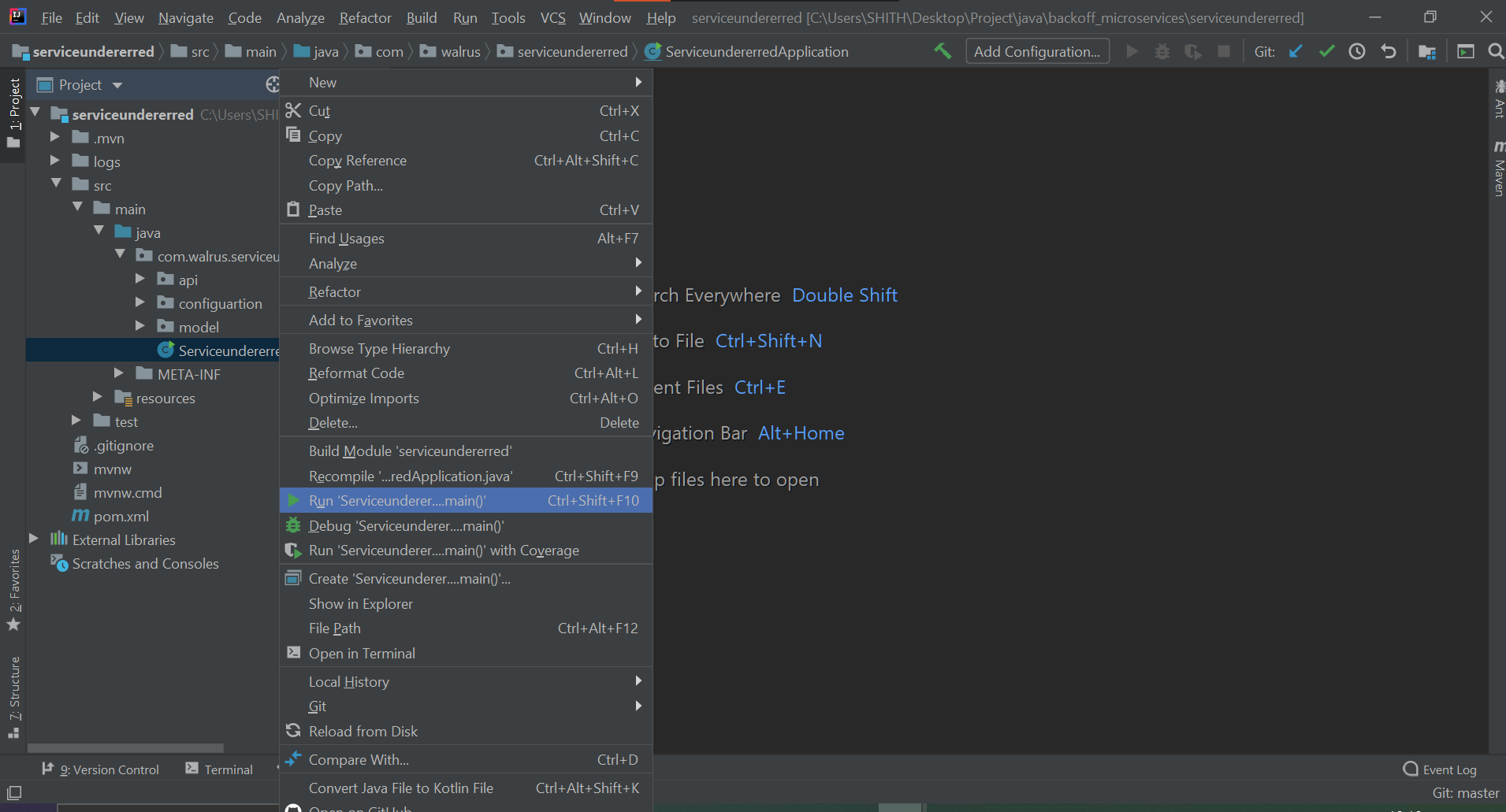
1. Retry after a min and it will respond with Hello from server 2



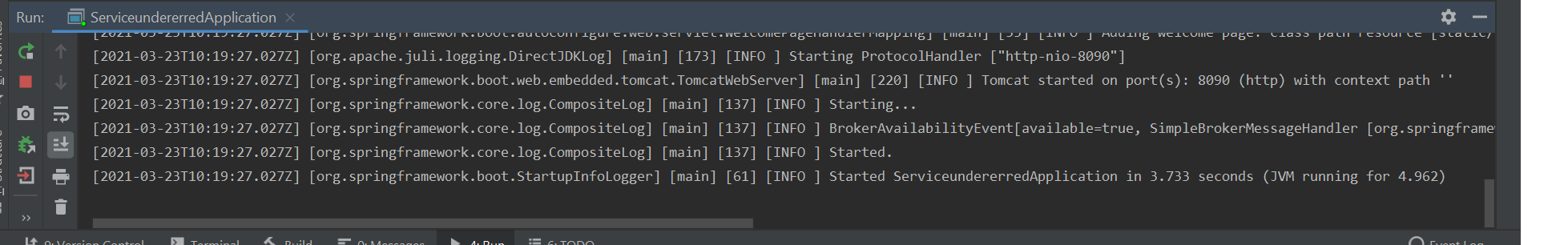
1. For seviceundererred we need to again open it in IDE like IntelliJ with Java 11+



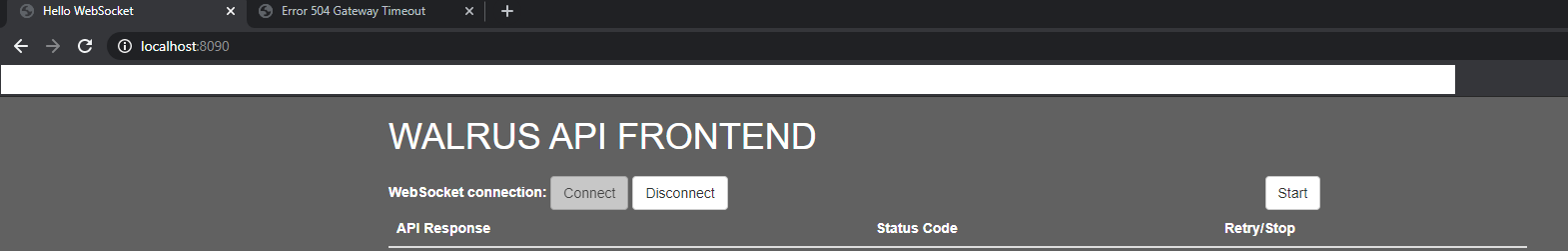
1. Wait for some time so that dependencies are loaded



1. Run the file Serviceundererred in com.walrus.serviceunderred
2. On getting this screen the Spring boot is ready with web application



1. <http://localhost:8090/>
2. Click on connect-> this will establish a websocket between webpage and app



1. After that click on Start -> this will trigger API call with exponential backoff

