Microservices Architecture and Programming

**Lab Practical and date** – Practical 4, 29th September 2020

**Name and Roll Number**- Het Shah, 17BIT103

**Practical Objective-** Message queuing system based Micro-service application

**Steps Involved-**

We explored message queuing using the ActiveMQ and JMS. Maven was used as the project handler.

**Background**

Apache ActiveMQ is an open source message broker written in Java together with a full Java Message Service client. It provides "Enterprise Features" which in this case means fostering the communication from more than one client or server.

**Installing ActiveMQ**

* Download activemq <https://activemq.apache.org/components/classic/download/>
* Unzip the folder and keep in any directory you want.(Say in -> C:\Program Files\apache-activemq-5.16.0 )
* Set the PATH Environment variable till the bin directory. In above example (C:\Program Files\apache-activemq-5.16.0\bin)
* You are done with installation.

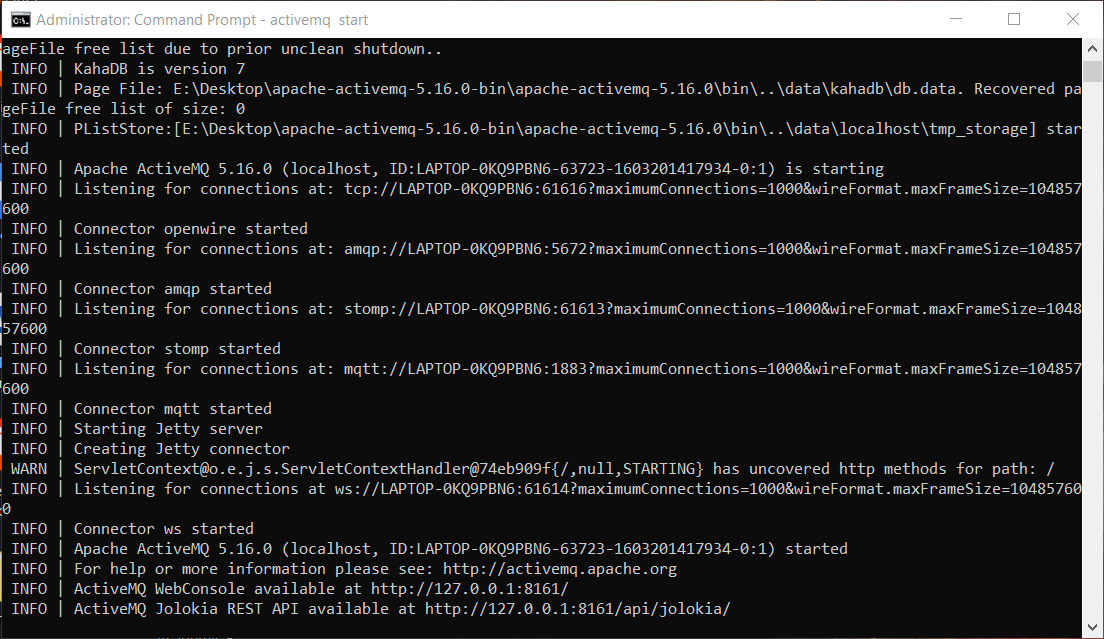
**Starting ActiveMQ web console**

* Run cmd as Administrator
* Type -> activemq start
* Wait for some time till it completely starts
* In your browser go to <http://127.0.0.1:8161/>
* Now go to Admin page from the opened web page. (Username : admin, Password: admin)

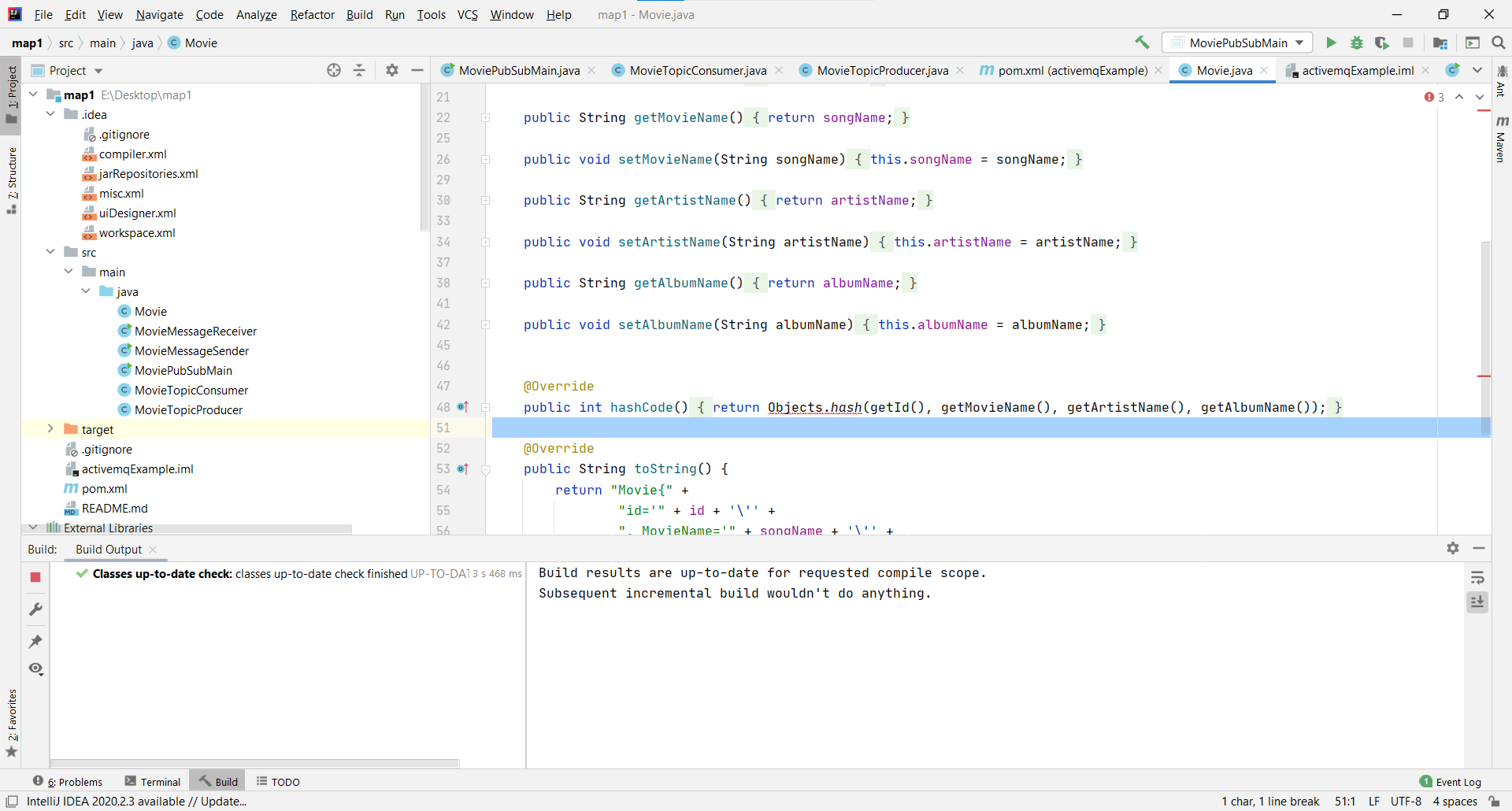
**Running this project**

* Complete all above installation
* Open Intellij-idea and open the project
* Run MoviePubSubMain.java
* from the activemq web console go to TOPICS and "Mov TOPIC" can be seen

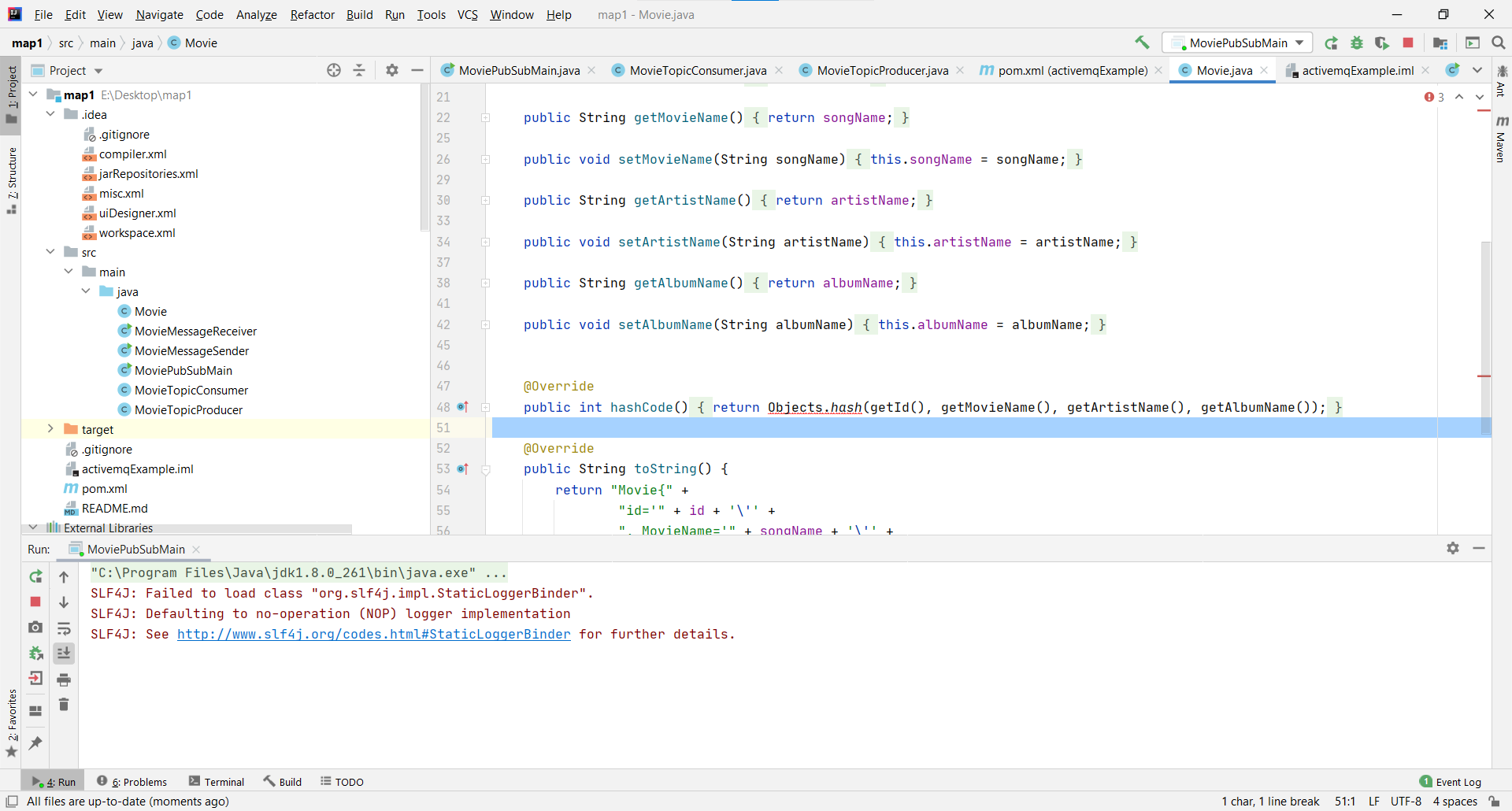
Output Screenshots



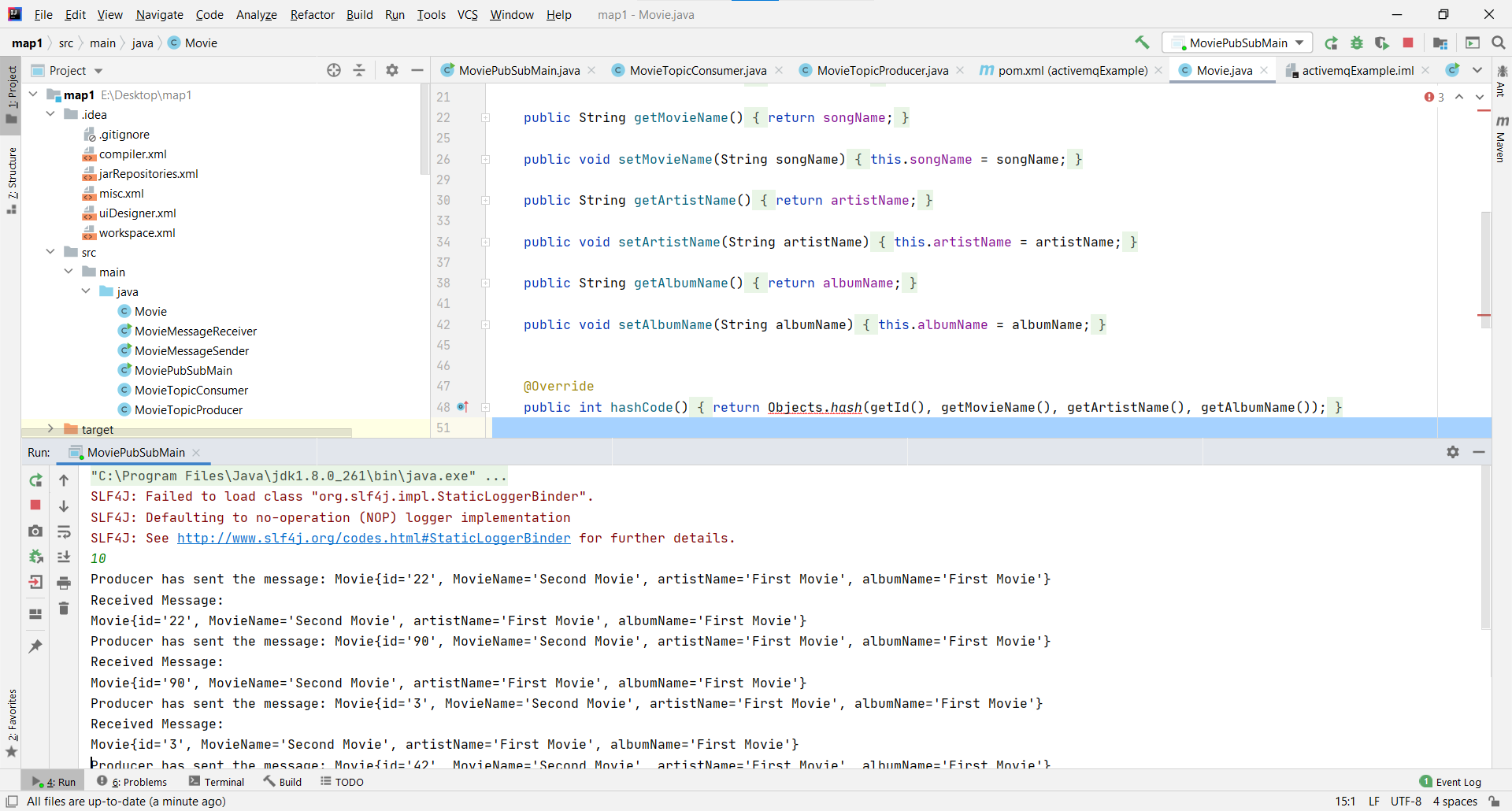
ActiveMQ server started



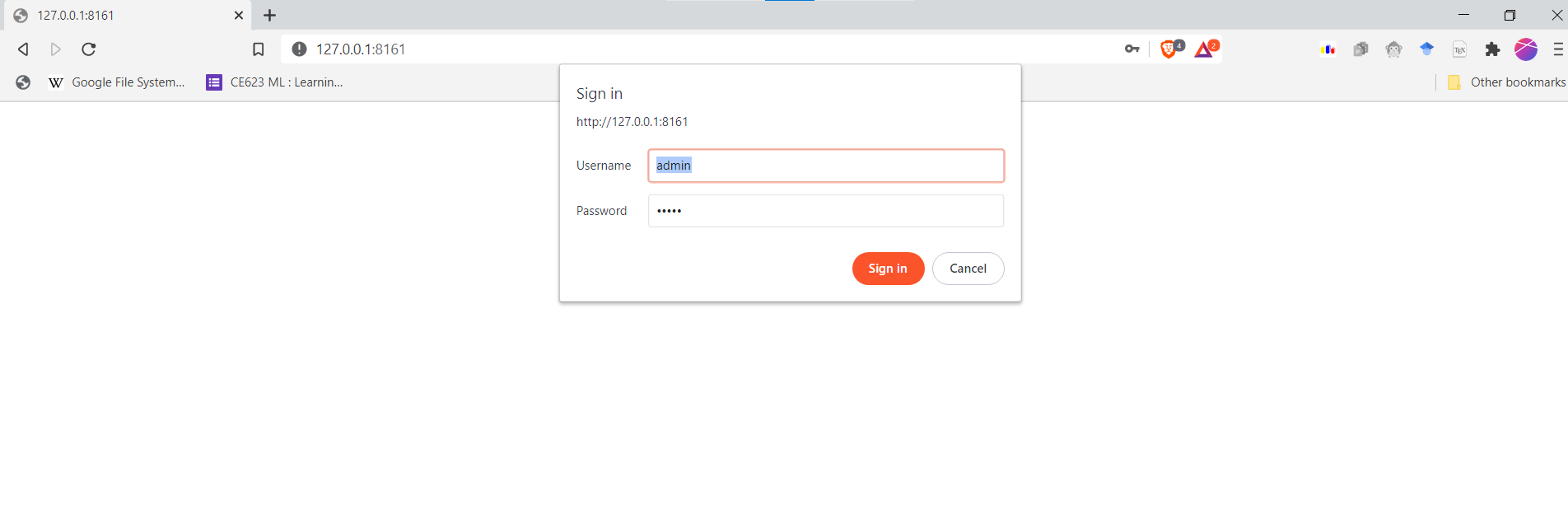
Building the project using maven



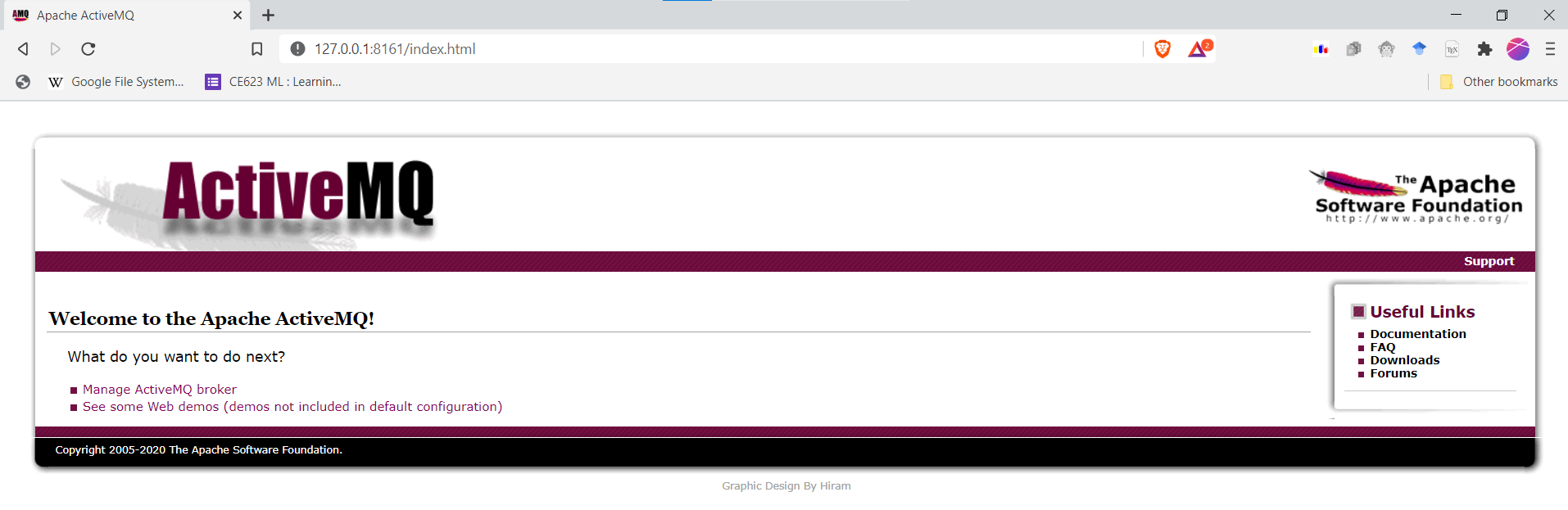
Running the project



The program sends 10 messages randomly at an interval of 1 second with the id of each movie being a random number between 1 and 100

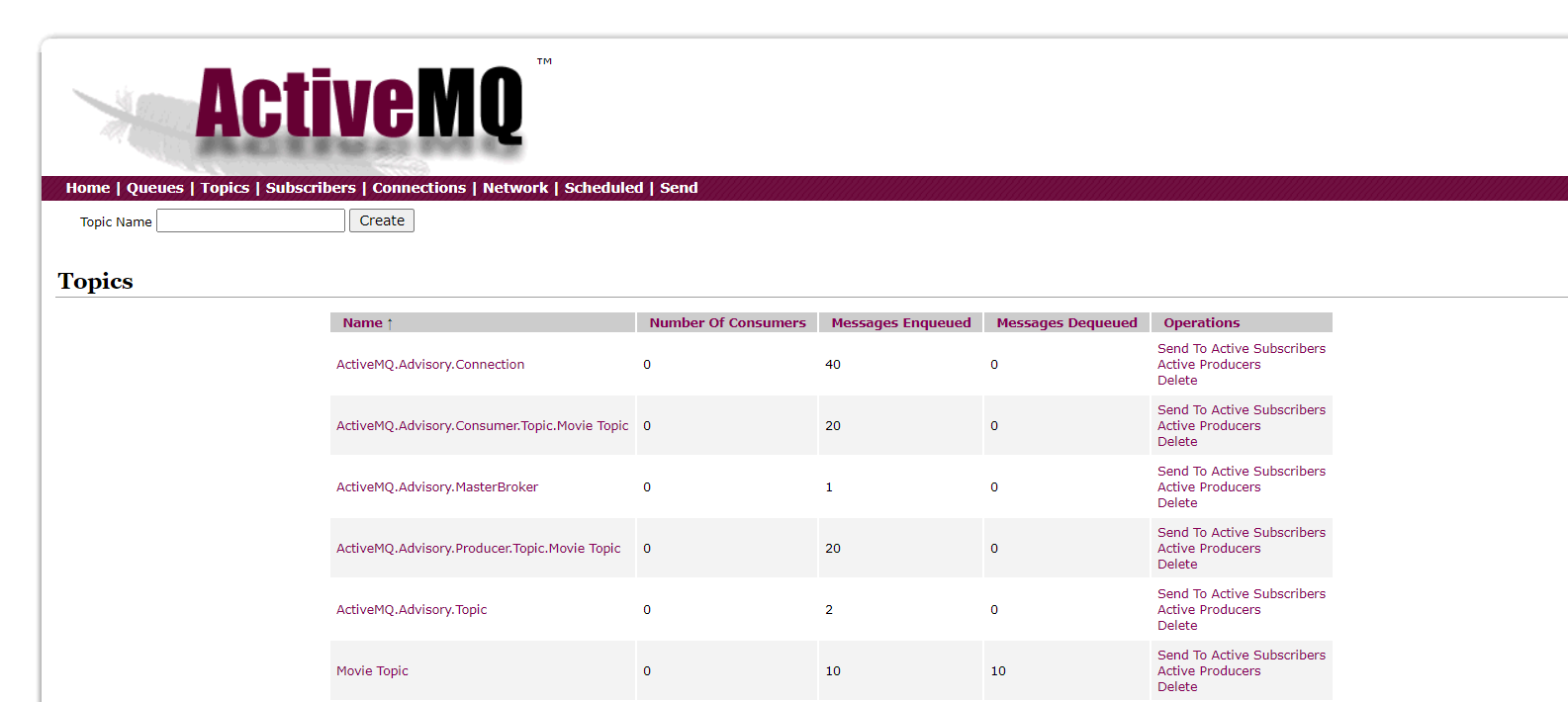


ActiveMQ login in the brower at 127.0.0.1

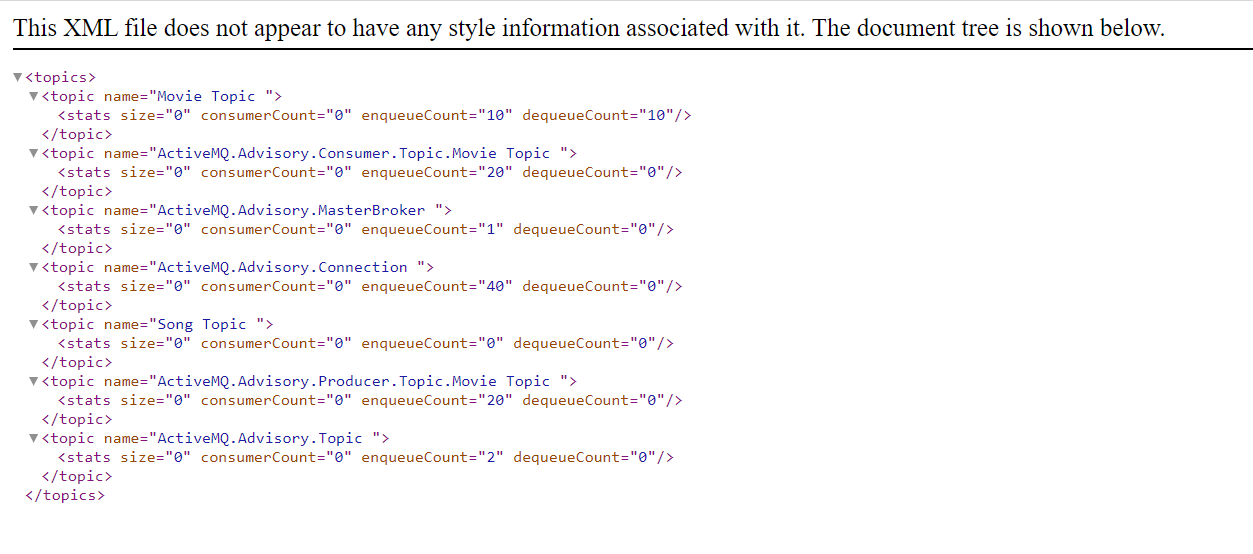
****

ActiveMQ homepage

Opened in the TOPIC section, we can see that 10 message have been received







The XML file shows the the number of messages enqueued and the number of messages enqueued

Conclusion

In this practical we learned about messasing system using JMS using Apache ActiveMQ as a broker. The project was build using Maven and the user was given an option of selecting number of messages it wants to send.