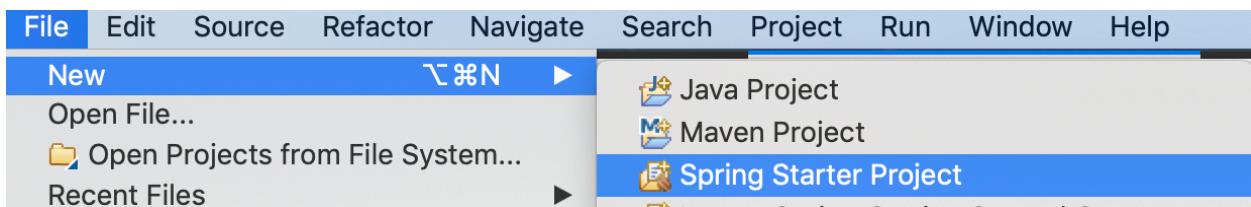
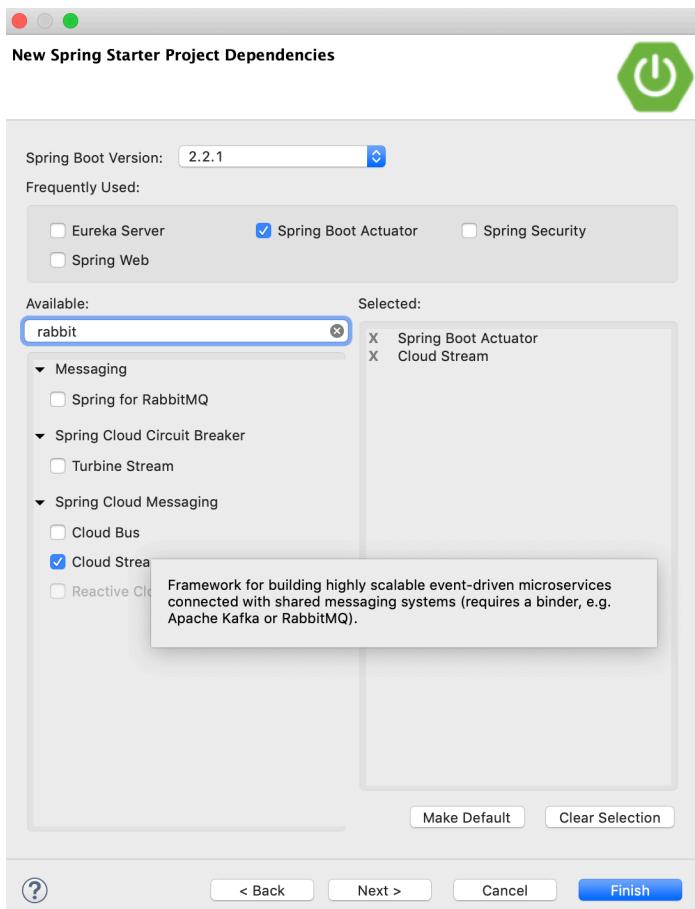


1. Create a new Spring Boot Project called

**demo-eureka-toll-intake**



2. Add modules Actuator & cloud stream, click on **Finish**



### 3. Add a dependency for RabbitMQ

**pom.xml**

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-stream-rabbit</artifactId>
    <version>1.3.5.RELEASE</version>
</dependency>
```

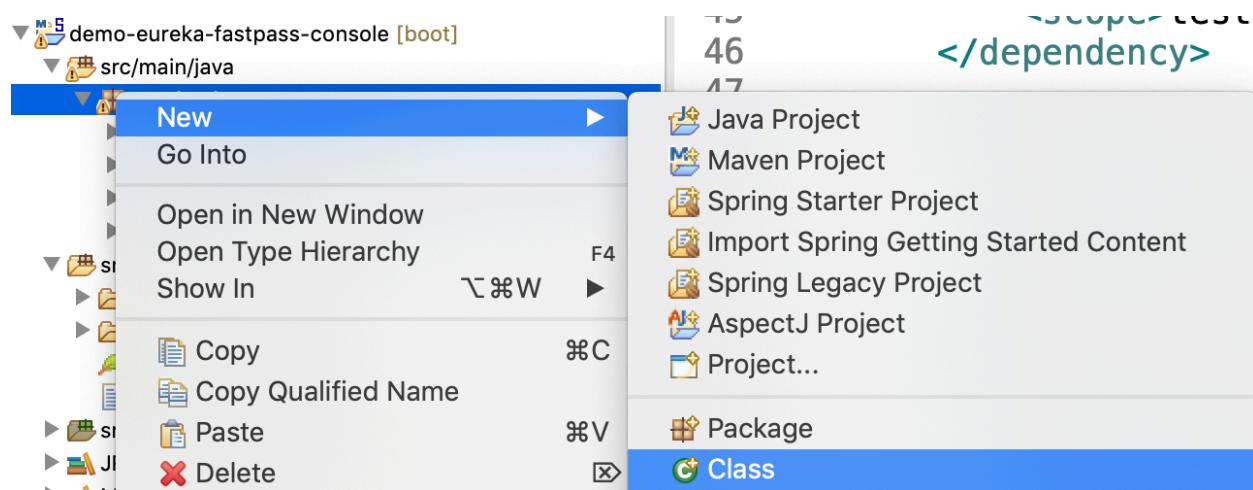
### 4. Open the **fastpass-console** application

### 5. Add a dependency for **spring-cloud-stream-rabbit**

**pom.xml**

```
<dependency>
    <groupId>org.springframework.cloud</groupId>
    <artifactId>spring-cloud-starter-stream-rabbit</artifactId>
    <version>1.3.5.RELEASE</version>
</dependency>
```

### 6. Create a new JAVA Class



## 6.1

### TollPublisher.java

```
package eureka.demo;

import org.springframework.cloud.stream.annotation.EnableBinding;
import org.springframework.cloud.stream.messaging.Source;
import
org.springframework.integration.annotation.InboundChannelAdapter;

@EnableBinding(Source.class)
public class TollPublisher {

    @InboundChannelAdapter(channel = Source.OUTPUT)
    public String sendTollCharge() {

        return "{station:\"20\", customerid:\"100\", timestamp:
\"2019-11-09-12T03:15:00\"}";
    }

}
```

## 7. Add properties to connect to RabbitMQ

### application.properties

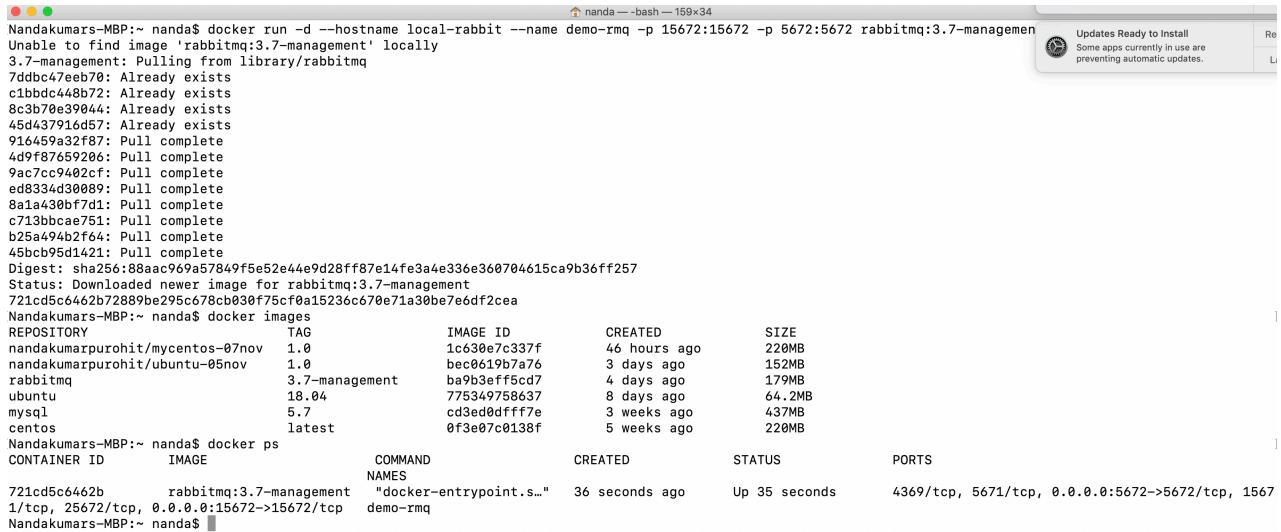
```
spring.rabbitmq.host=localhost
spring.rabbitmq.port=5672
spring.rabbitmq.username=guest
spring.rabbitmq.password=guest
```

## 8. Pull a Docker Image for RabbitMQ and start a Container

```
# docker run -d --hostname local-rabbit --name demo-rmq -p
15672:15672 -p 5672:5672 rabbitmq:3.7-management
```

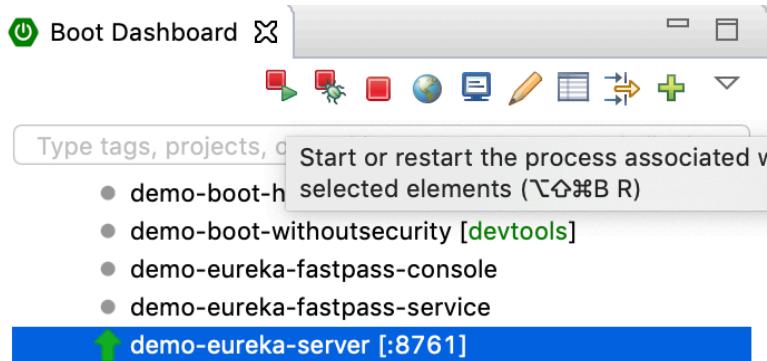
```
# docker images
```

```
# docker ps
```

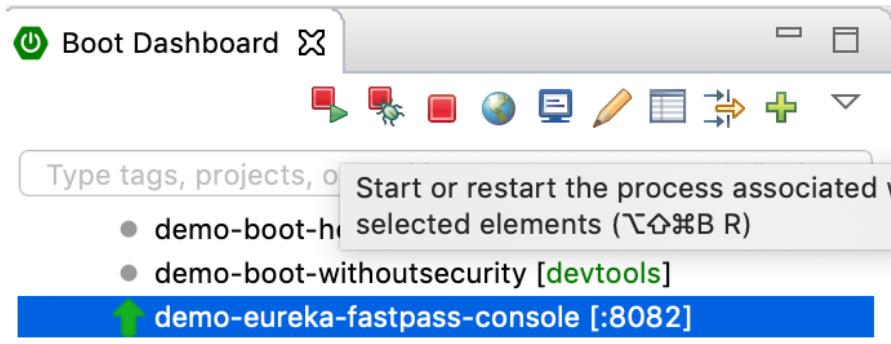


```
nanda$ docker run -d --hostname local-rabbit --name demo-rmq -p 15672:15672 -p 5672:5672 rabbitmq:3.7-management
Unable to find image 'rabbitmq:3.7-management' locally
3.7-management: Pulling from library/rabbitmq
7ddbc47eeb70: Already exists
c1bbdc448b72: Already exists
8c3b70e39044: Already exists
45d437916d57: Already exists
916459a32f87: Pull complete
4d9f87659286: Pull complete
9ac7cc94b2cf: Pull complete
ed8334d30089: Pull complete
8a1a430bf7d1: Pull complete
c713bbcae751: Pull complete
b25a494ab2f64: Pull complete
45ccb95d1421: Pull complete
Digest: sha256:88aac969a57849f5e52e44e9d28ff87e14fe3a4e336e360704615ca9b36ff257
Status: Downloaded newer image for rabbitmq:3.7-management
721cd5c6462b72889be295c678cb030f75cf0a15236c670e71a30be7e6df2cea
Nandakumars-MBP:~ nanda$ docker images
REPOSITORY          TAG        IMAGE ID      CREATED       SIZE
nandakumarpurohit/mycentos-07nov  1.0        1c630e7c337f    46 hours ago  220MB
nandakumarpurohit/ubuntu-05nov    1.0        bec0619b7a76    3 days ago   152MB
rabbitmq            3.7-management  ba9b3eff5cd7    4 days ago   179MB
ubuntu              18.04      775349758637    8 days ago   64.2MB
mysql               5.7        cd3ed0dff7e     3 weeks ago  437MB
centos              latest      0f3e07c0138f    5 weeks ago  220MB
Nandakumars-MBP:~ nanda$ docker ps
CONTAINER ID        IMAGE           COMMAND          CREATED          STATUS          PORTS
NAMES
721cd5c6462b        rabbitmq:3.7-management "docker-entrypoint.s..." 36 seconds ago Up 35 seconds  4369/tcp, 5671/tcp, 0.0.0.0:5672->5672/tcp, 15672/tcp, 25672/tcp, 0.0.0.0:15672->15672/tcp
demo-rmq
Nandakumars-MBP:~ nanda$
```

## 9. START demo-eureka-server



## 10. Start **demo-eureka-fastpass-console**



## 11. Visit <http://localhost:15672>

The screenshot shows the 'Overview' page of the RabbitMQ Management Interface. At the top, it displays system statistics: Erlang 22.1.5, Refreshed 2019-11-09 14:35:47, Virtual host: All, Cluster rabbit@local-rabbit, User guest, and Log out. Below this, the 'Overview' section includes a 'Totals' chart for queued messages and message rates. The 'Message rates' chart shows a fluctuating rate of approximately 1.0/s. To the right, various metrics are listed: Publish (1.0/s), Publisher confirm (0.00/s), Deliver (auto ack) (0.00/s), Consumer ack (0.00/s), Redelivered (0.00/s), Get (manual ack) (0.00/s), Get (auto ack) (0.00/s), Disk read (0.00/s), Disk write (0.00/s), and Return (0.00/s). At the bottom, global counts are shown: Connections: 2, Channels: 2, Exchanges: 8, Queues: 0, Consumers: 0. The 'Nodes' section lists the single node 'rabbit@local-rabbit' with its resource usage details.

Name	File descriptors	Socket descriptors	Erlang processes	Memory	Disk space	Uptime	Info	Reset stats
rabbit@local-rabbit	33 1048576 available	2 943626 available	419 1048576 available	99MB 799MB high watermark	53GB 48MB low watermark	2h 22m	basic disc rss	+/- This node All nodes

## Click on Connections

localhost:15672/#/connections

RabbitMQ 3.7.21 Erlang 22.1.5

Overview Connections Channels Exchanges Queues Admin

### Connections

All connections (2)

Pagination

Page 1 of 1 - Filter:   Regex ?

Overview		Details			Network		+/-
Name	User name	State	SSL / TLS	Protocol	Channels	From client	To client
172.17.0.1:35106 rabbitConnectionFactory#576a8b00:0	guest	green running	○	AMQP 0-9-1	1	2iB/s	0iB/s
172.17.0.1:35108 SpringAMQP#7a17c5b2:0	guest	green running	○	AMQP 0-9-1	1	342iB/s	0iB/s

HTTP API Server Docs Tutorials Community Support Community Slack Commercial Support Plugins GitHub Changelog

Click on Exchanges

The screenshot shows the RabbitMQ Management UI at the URL `localhost:15672/#/exchanges`. The top navigation bar includes icons for back, forward, and home, along with the URL. Below the URL is the RabbitMQ logo and version information (3.7.21 Erlang 22.1.5). The main navigation menu has tabs for Overview, Connections, Channels, Exchanges (which is highlighted), Queues, and Admin.

## Exchanges

▼ All exchanges (8)

Pagination: Page 1 of 1 - Filter:   Regex [?](#)

Name	Type	Features	Message rate in	Message rate out	+/-
(AMQP default)	direct	D			
amq.direct	direct	D			
amq.fanout	fanout	D			
amq.headers	headers	D			
amq.match	headers	D			
amq.rabbitmq.trace	topic	D I			
amq.topic	topic	D			
output	topic	D	0.00/s		

▶ Add a new exchange

---

[HTTP API](#) [Server Docs](#) [Tutorials](#) [Community Support](#) [Community Slack](#) [Community GitHub](#)

12. ST0P **demo-eureka-fastpass-console**

### 13. Go To **demo-eureka-toll-intake**

14. Add Annotation to the Main Class and add a method to consume the messages

```
import org.springframework.boot.SpringApplication;
import
org.springframework.boot.autoconfigure.SpringBootApplication;
import
org.springframework.cloud.stream.annotation.EnableBinding;
import
org.springframework.cloud.stream.annotation.StreamListener;
import org.springframework.cloud.stream.messaging.Sink;

@EnableBinding(Sink.class)
@SpringBootApplication
public class DemoEurekaTollIntakeApplication {

    public static void main(String[] args) {

SpringApplication.run(DemoEurekaTollIntakeApplication.class,
args);
}

    @StreamListener(Sink.INPUT)
    public void log(String msg) {
        System.out.println(msg);
    }
}
```

15. Add RabbitMQ Connection configuration to properties file

### **application.properties**

```
spring.rabbitmq.host=localhost
spring.rabbitmq.port=5672
spring.rabbitmq.username=guest
spring.rabbitmq.password=guest
```

16. Also add the INPUT destination from which messages to be consumed

### application.properties

```
spring.cloud.stream.bindings.input.destination = fastpasstoll
```

17. Go back to **demo-eureka-fastpass-console**

Add the OUTPUT destination to pump-in messages

### application.properties

```
spring.cloud.stream.bindings.output.destination = fastpasstoll
```

18. START **demo-eureka-toll-intake**

19. Visit <http://localhost:15672>

19.1 There is a new connection in RabbitMQ

The screenshot shows the RabbitMQ Management UI with the 'Connections' tab selected. The top navigation bar includes links for Overview, Connections (which is active), Channels, Exchanges, Queues, and Admin. Below the navigation, a section titled 'Connections' displays a single connection entry. The connection details are as follows:

Name	User name	State	SSL / TLS	Protocol	Channels	From client	To client
172.17.0.1:35872 rabbitConnectionFactory#4d847d32:0	guest	running	○	AMQP 0-9-1	1	0iB/s	0iB/s

At the bottom of the page, there are links for HTTP API, Server Docs, Tutorials, Community Support, Community Slack, Commercial Support, and Plugins.

## 19.2 There is a new Exchange Called **fastpasstoll**

The screenshot shows the RabbitMQ Management UI interface. At the top, there is a logo for RabbitMQ with the text "RabbitMQ™" and "3.7.21 Erlang 22.1.5". Below the logo, there is a navigation bar with tabs: Overview, Connections, Channels, Exchanges (which is highlighted), and Queues. The main content area is titled "Exchanges" and shows a list of exchanges with the heading "All exchanges (9)". Below the list, there is a "Pagination" section with a page number "1" and a dropdown arrow, followed by "of 1 - Filter:" and a search input field. A checkbox for "Regex" and a help icon "?" are also present. The table below lists the exchanges:

Name	Type	Features	Message rate in	Message rate out
(AMQP default)	direct	D		
amq.direct	direct	D		
amq.fanout	fanout	D		
amq.headers	headers	D		
amq.match	headers	D		
amq.rabbitmq.trace	topic	D I		
amq.topic	topic	D		
fastpasstoll	topic	D		
output	topic	D	0.00/s	

19.3 Nothing is shown in this new exchange as of now



Overview

Connections

Channels

Exchanges

## Exchange: fastpasstoll

### ▼ Overview

Message rates last minute ?

Currently idle

### Details

Type	topic
Features	durable: true
Policy	

### ► Bindings

### ► Publish message

### ▼ Delete this exchange

Delete

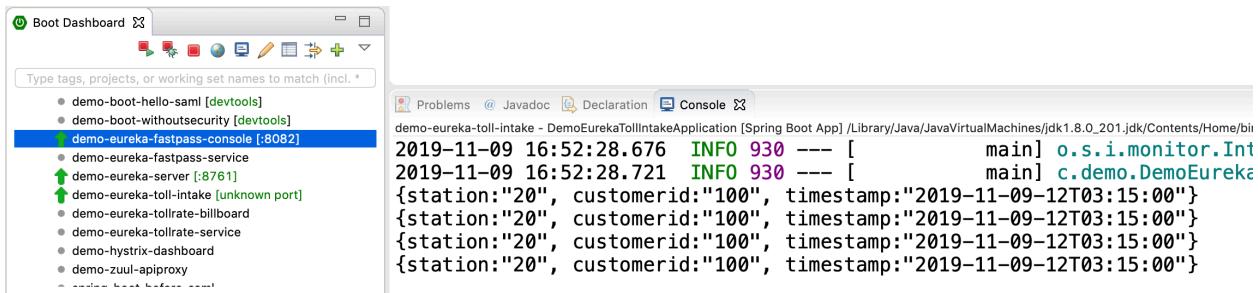
20. Select demo-eureka-toll-intake app in the TV icon and click on PIN icon to pin the console to watch logs as shown below



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. At the top, there are tabs for 'Problems', 'Javadoc', 'Declaration', and 'Console'. Below the tabs, the title bar displays 'demo-eureka-toll-intake 1 demo-eureka-server - DemoEurekaServerApplication [Spring Boot App...es/jdk1.8.0\_201.jdk/Contents/Home/bin/java (Nov 9, 2019, 2:30:28 PM)' and '2019-11-09 16:52:28.648 INFO 930 --- [main] o.s.c.stream.binder.BinderErrorChannel : Channel'. The main area shows log entries from the application, with the most recent one being '2019-11-09 16:52:28.648 INFO 930 --- [main] o.s.c.stream.binder.BinderErrorChannel : Channel'.

## 21. START **demo-eureka-fastpass-console**

22. Once the fastpass-console app starts streaming messages to the Exchange, we can observe that Toll-intake app starts fetching those every second as shown below



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. At the top, there are tabs for 'Problems', 'Javadoc', 'Declaration', and 'Console'. Below the tabs, the title bar displays 'demo-eureka-toll-intake - DemoEurekaTollIntakeApplication [Spring Boot App] /Library/Java/JavaVirtualMachines/jdk1.8.0\_201.jdk/Contents/Home/bi' and '2019-11-09 16:52:28.676 INFO 930 --- [main] o.s.i.monitor.Int'. The main area shows log entries from the application, with the most recent ones being '2019-11-09 16:52:28.721 INFO 930 --- [main] c.demo.DemoEureka' and '{station:"20", customerid:"100", timestamp:"2019-11-09-12T03:15:00"} {station:"20", customerid:"100", timestamp:"2019-11-09-12T03:15:00"} {station:"20", customerid:"100", timestamp:"2019-11-09-12T03:15:00"} {station:"20", customerid:"100", timestamp:"2019-11-09-12T03:15:00"}'. To the left, the 'Boot Dashboard' window is open, showing a list of projects: 'demo-boot-hello-saml [devtools]', 'demo-boot-withoutsecurity [devtools]', 'demo-eureka-fastpass-console [8082]' (which is highlighted in blue), 'demo-eureka-server [:8761]', 'demo-eureka-toll-intake [unknown port]', 'demo-eureka-tolrate-billboard', 'demo-eureka-tolrate-service', 'demo-hystrix-dashboard', and 'demo-zuul-apiproxy'. The 'demo-eureka-toll-intake' application is also pinned to the Eclipse interface.

