Synchronous process
(task)

executing step by step

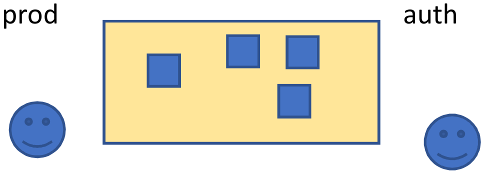
caller calling other method

MessageBroker
RabbitMq

Synchronous resource
variable/object

not	10
pass	3
pass	4
pass	5

prod

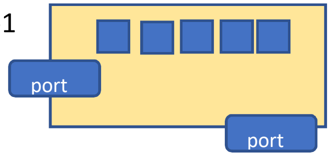


auth

amqp

hi,hello,how r u

app1

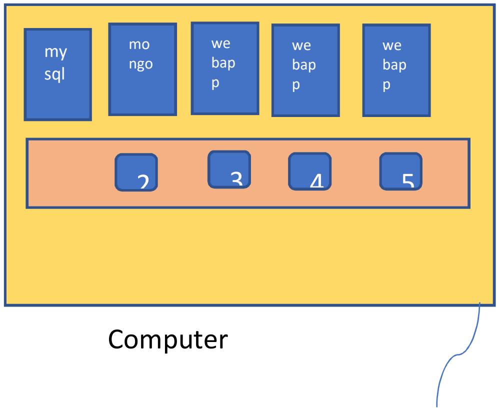


app2 (down)

RabbitMq webinterface

port no.

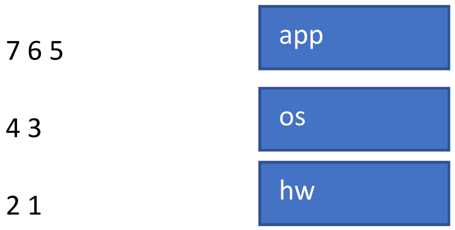
8888
9999
5555
65100



Computer

x.x.x.x
nw

http://x.x.x.x:1



7 6 5

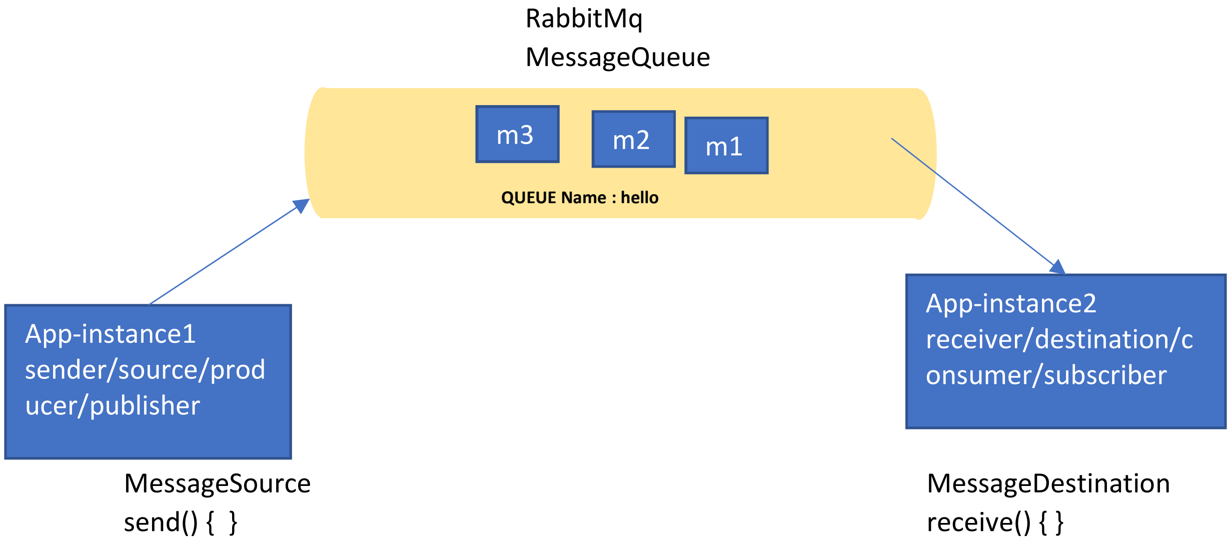
4 3

2 1

app

os

hw



RabbitMq
MessageQueue

m3

m2

m1

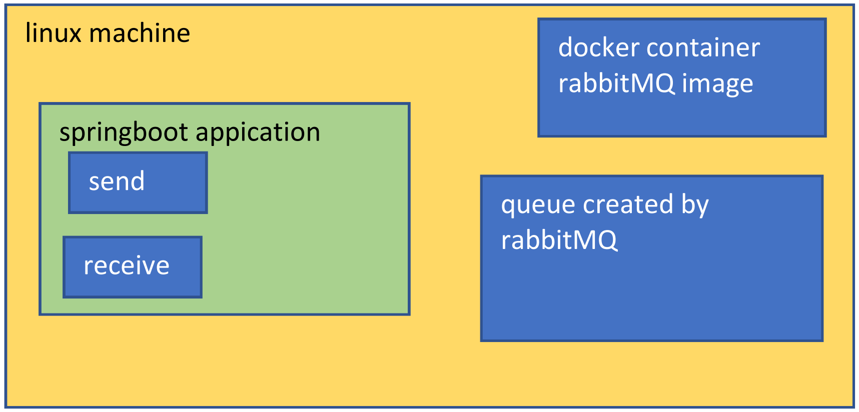
QUEUE Name : hello

App-instance1
sender/source/producer/publisher

MessageSource
send() { }

App-instance2
receiver/destination/consumer/subscriber

MessageDestination
receive() { }



linux machine

docker container
rabbitMQ image

springboot appication

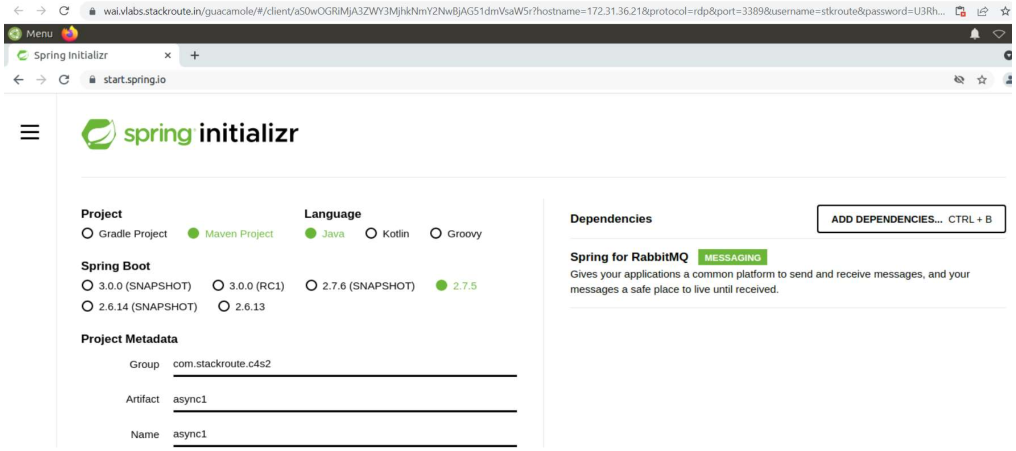
send

receive

queue created by
rabbitMQ

Step 1 Create springboot application by adding required depdendencies

RebbitMQ (messaging)


wai.vlabs.stackroute.in/guacamole/#/client/aS0wOGRiMjA3ZWY3MjhkNmY2NwBjAG51dmVsaW5r?hostname=172.31.36.21&protocol=rdp&port=3389&username=stkroute&password=U3Rh...
Menu
Spring Initializr
start.spring.io
spring initializr
ADD DEPENDENCIES... CTRL + B

Project
○ Gradle Project ● Maven Project

Language
● Java ○ Kotlin ○ Groovy

Spring Boot
○ 3.0.0 (SNAPSHOT) ○ 3.0.0 (RC1) ○ 2.7.6 (SNAPSHOT) ● 2.7.5
○ 2.6.14 (SNAPSHOT) ○ 2.6.13

Project Metadata

Group com.stackroute.c4s2

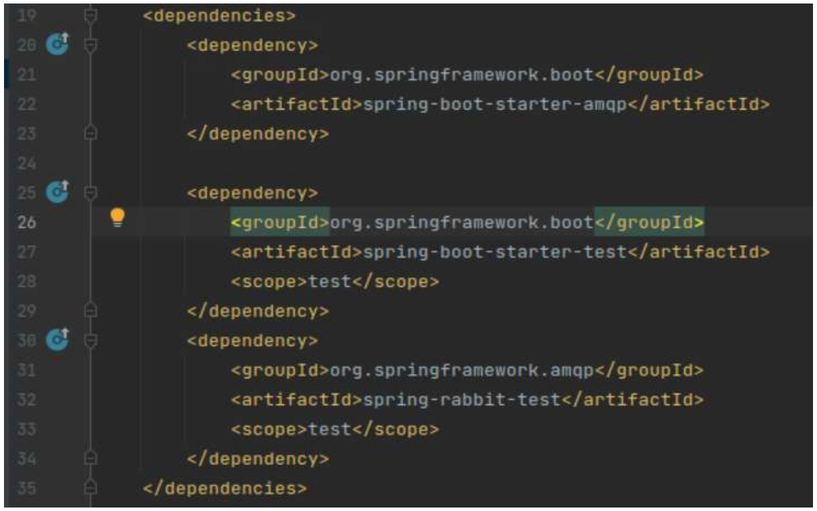
Artifact async1

Name async1

Dependencies

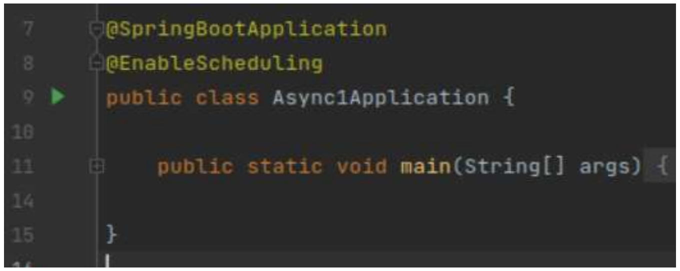
Spring for RabbitMQ MESSAGING
Gives your applications a common platform to send and receive messages, and your messages a safe place to live until received.

download project, extract
and open in intellij



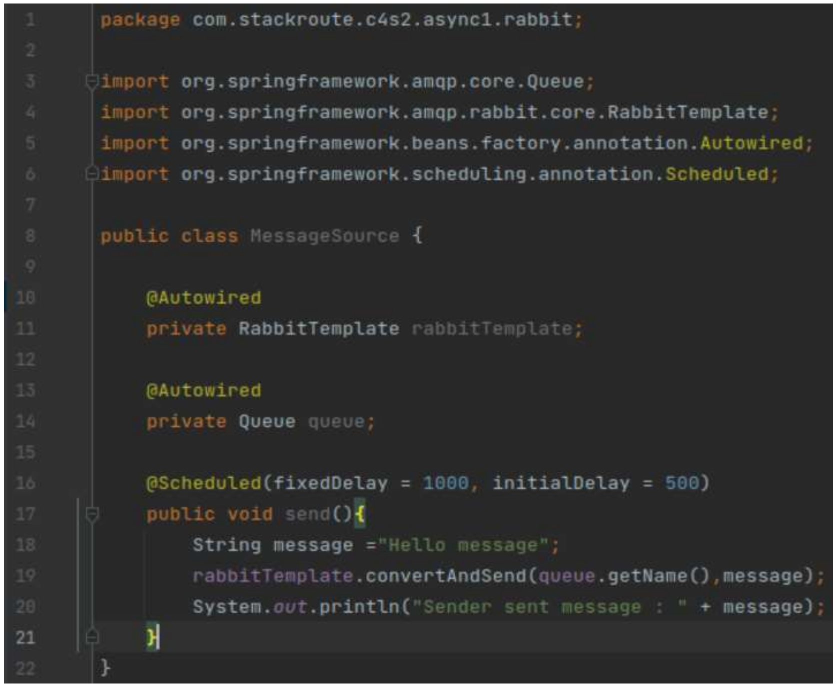
```xml
<dependencies>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-amqp</artifactId>
    </dependency>

    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>org.springframework.amqp</groupId>
        <artifactId>spring-rabbit-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>
```

Step 2 Enable scheduling on main



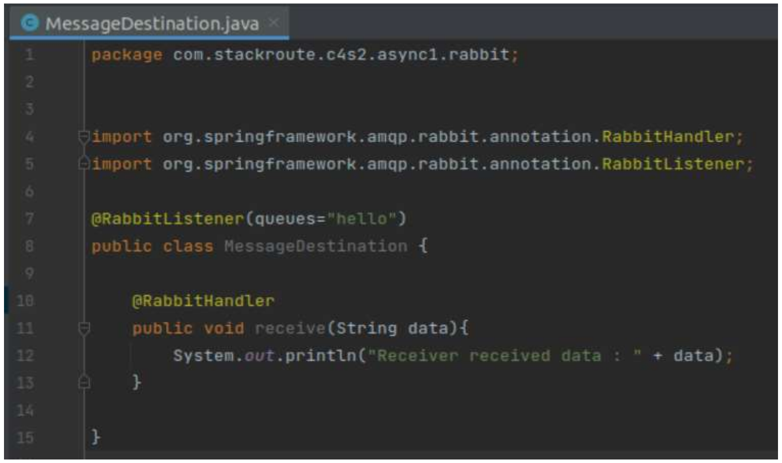
```
@SpringBootApplication
@EnableScheduling
public class Async1Application {

    public static void main(String[] args) {

}
```

Step 3 Define sender / producer / publisher / source



```
package com.stackroute.c4s2.async1.rabbit;

import org.springframework.amqp.core.Queue;
import org.springframework.amqp.rabbit.core.RabbitTemplate;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.scheduling.annotation.Scheduled;

public class MessageSource {

    @Autowired
    private RabbitTemplate rabbitTemplate;

    @Autowired
    private Queue queue;

    @Scheduled(fixedDelay = 1000, initialDelay = 500)
    public void send(){
        String message ="Hello message";
        rabbitTemplate.convertAndSend(queue.getName(),message);
        System.out.println("Sender sent message : " + message);
    }
}
```

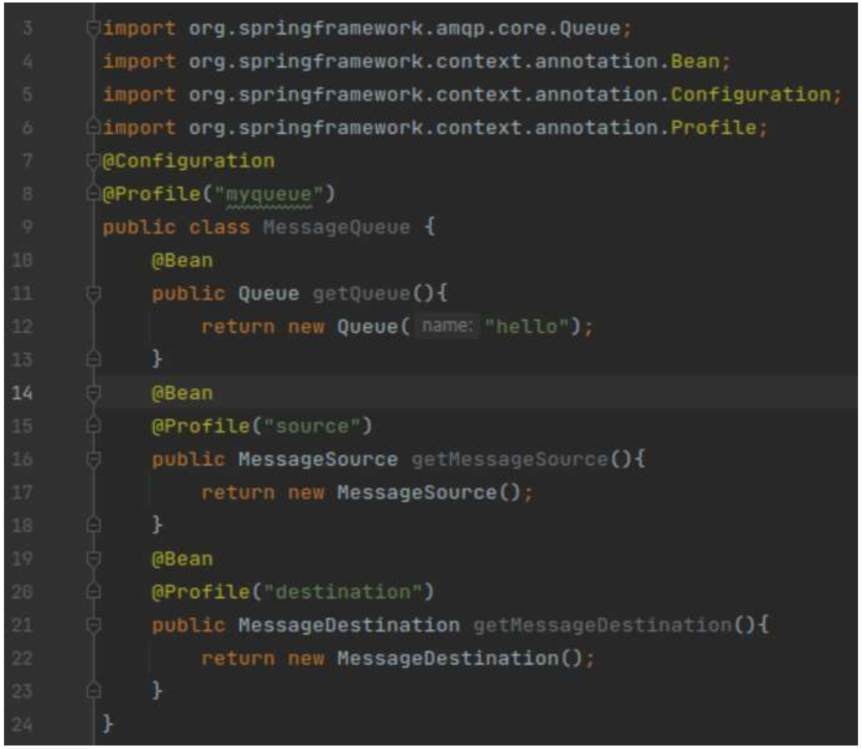
Step 4 Define receiver/destination/consumer/subscriber



MessageDestination.java

```java
package com.stackroute.c4s2.async1.rabbit;


import org.springframework.amqp.rabbit.annotation.RabbitHandler;
import org.springframework.amqp.rabbit.annotation.RabbitListener;

@RabbitListener(queues="hello")
public class MessageDestination {

    @RabbitHandler
    public void receive(String data){
        System.out.println("Receiver received data : " + data);
    }

}
```

Step 5 Define Queue



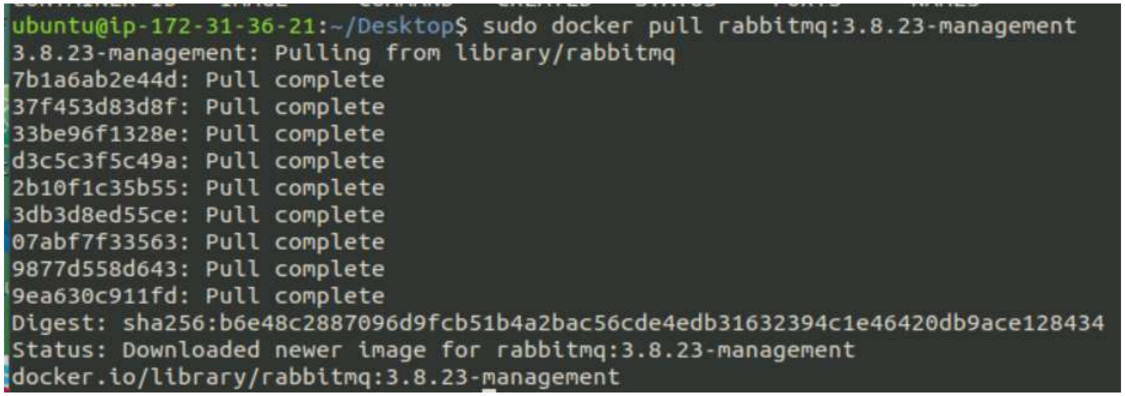
```java
import org.springframework.amqp.core.Queue;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.context.annotation.Profile;
@Configuration
@Profile("myqueue")
public class MessageQueue {
    @Bean
    public Queue getQueue(){
        return new Queue( name: "hello");
    }
    @Bean
    @Profile("source")
    public MessageSource getMessageSource(){
        return new MessageSource();
    }
    @Bean
    @Profile("destination")
    public MessageDestination getMessageDestination(){
        return new MessageDestination();
    }
}
```

Step 6 Start rabbitMq service if installed

or

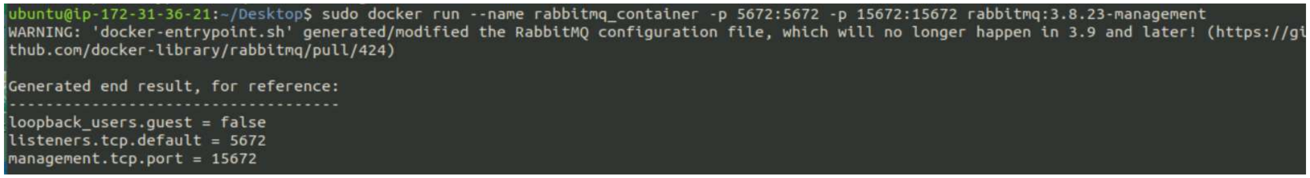
Pull rabbitMq image from docker hub and start container

sudo docker pull rabbitmq:3.8.23-management



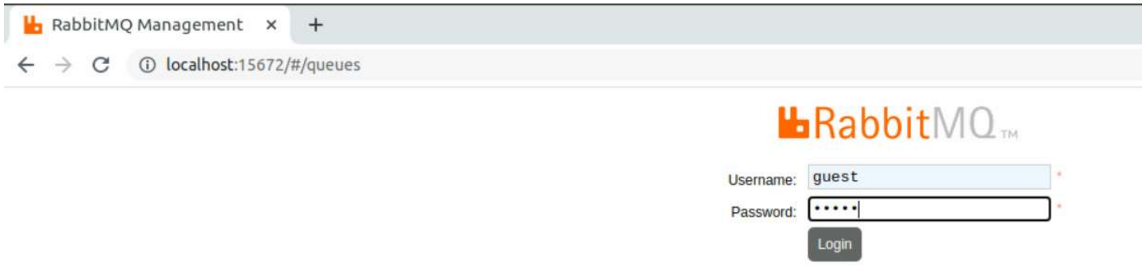
```
ubuntu@ip-172-31-36-21:~/Desktop$ sudo docker pull rabbitmq:3.8.23-management
3.8.23-management: Pulling from library/rabbitmq
7b1a6ab2e44d: Pull complete
37f453d83d8f: Pull complete
33be96f1328e: Pull complete
d3c5c3f5c49a: Pull complete
2b10f1c35b55: Pull complete
3db3d8ed55ce: Pull complete
07abf7f33563: Pull complete
9877d558d643: Pull complete
9ea630c911fd: Pull complete
Digest: sha256:b6e48c2887096d9fcb51b4a2bac56cde4edb31632394c1e46420db9ace128434
Status: Downloaded newer image for rabbitmq:3.8.23-management
docker.io/library/rabbitmq:3.8.23-management
```

sudo docker run --name rabbitmq_container -p 5672:5672 -p 15672:15672 rabbitmq:3.8.23-management



```
ubuntu@ip-172-31-36-21:~/Desktop$ sudo docker run --name rabbitmq_container -p 5672:5672 -p 15672:15672 rabbitmq:3.8.23-management
WARNING: 'docker-entrypoint.sh' generated/modified the RabbitMQ configuration file, which will no longer happen in 3.9 and later! (https://gi
thub.com/docker-library/rabbitmq/pull/424)

Generated end result, for reference:
------------------------------------
loopback_users.guest = false
listeners.tcp.default = 5672
management.tcp.port = 15672
```

http://localhost:15672 guest guest



Username: guest

Password: •••••

in 1st isntance

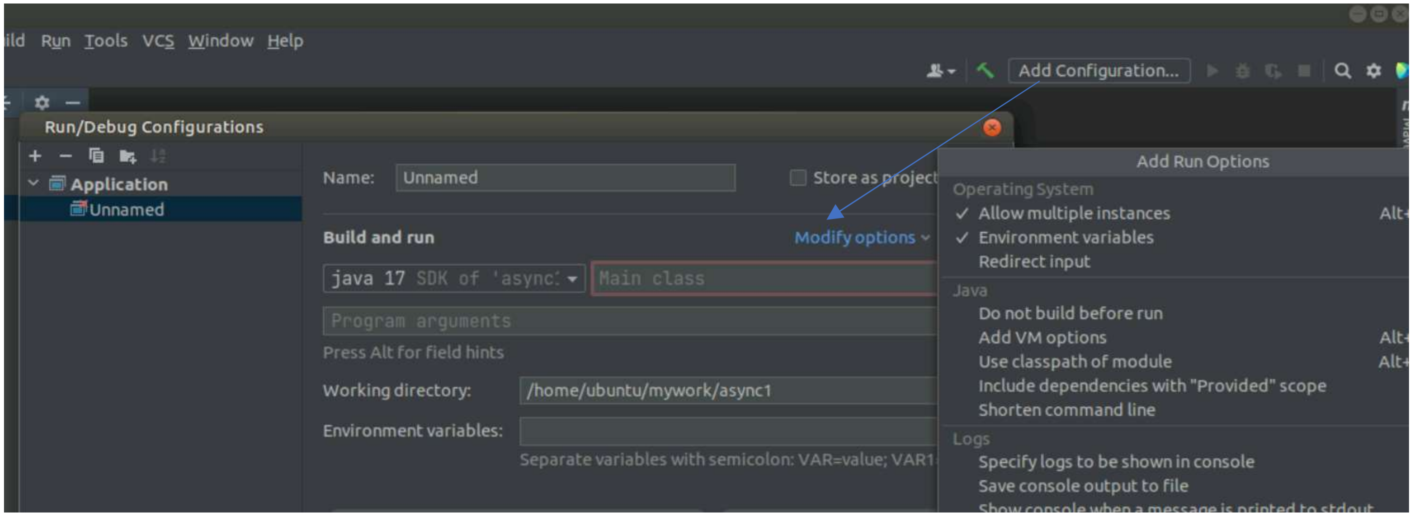
queue+sender

8888

in 2nd isntance

queue+receiver

enable multi instance execution



Run/Debug Configurations

Name: Unnamed

□ Store as project

Build and run

java 17 SDK of 'async:

Working directory: /home/ubuntu/mywork/async1

Environment variables:

Add Run Options

Operating System

✓ Allow multiple instances

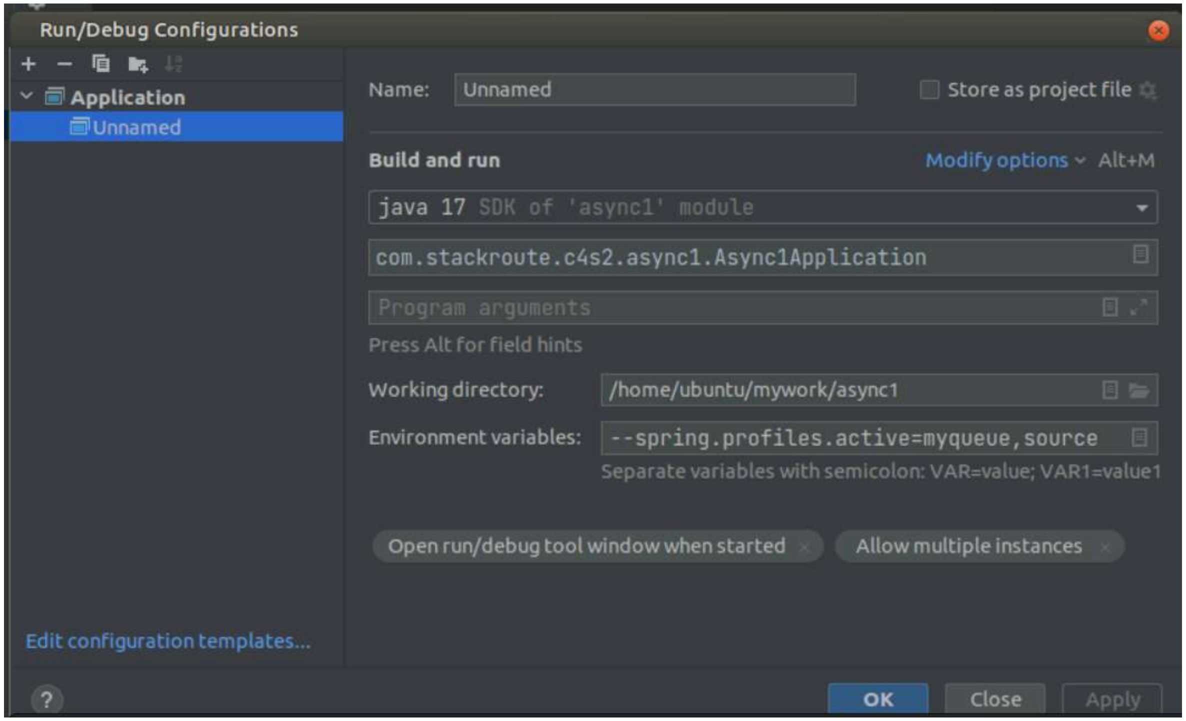
✓ Environment variables

ild Run Tools VCS Window Help
Add Configuration...
Application
Unnamed
Modify options
Main class
Program arguments
Press Alt for field hints
Separate variables with semicolon: VAR=value; VAR1
Redirect input
Java
Do not build before run
Add VM options
Use classpath of module
Include dependencies with "Provided" scope
Shorten command line
Logs
Specify logs to be shown in console
Save console output to file
Show console when a message is printed to stdout
Alt+
Alt+
Alt+

run instance 1

queue + source

8888 8888 is configured in application.properties



Run/Debug Configurations
Application
Unnamed
Store as project file
Modify options Alt+M
Press Alt for field hints
Separate variables with semicolon: VAR=value; VAR1=value1
Edit configuration templates...
OK
Close
Apply

Name: Unnamed

Build and run

java 17 SDK of 'async1' module

com.stackroute.c4s2.async1.Async1Application

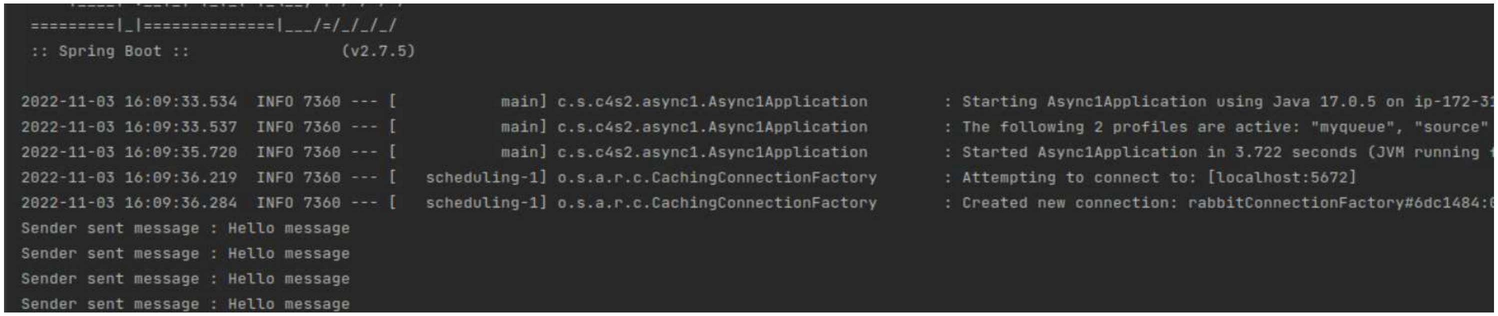
Program arguments

Working directory: /home/ubuntu/mywork/async1

Environment variables: --spring.profiles.active=myqueue,source

Open run/debug tool window when started

Allow multiple instances

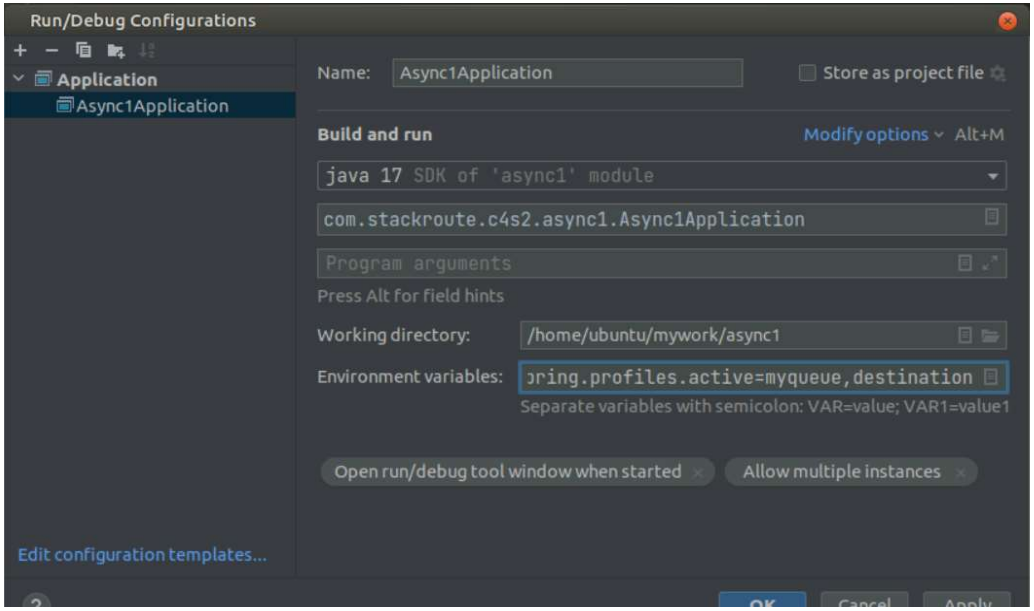


```
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.7.5)

2022-11-03 16:09:33.534  INFO 7360 --- [           main] c.s.c4s2.async1.Async1Application        : Starting Async1Application using Java 17.0.5 on ip-172-3
2022-11-03 16:09:33.537  INFO 7360 --- [           main] c.s.c4s2.async1.Async1Application        : The following 2 profiles are active: "myqueue", "source"
2022-11-03 16:09:35.720  INFO 7360 --- [           main] c.s.c4s2.async1.Async1Application        : Started Async1Application in 3.722 seconds (JVM running
2022-11-03 16:09:36.219  INFO 7360 --- [   scheduling-1] o.s.a.r.c.CachingConnectionFactory       : Attempting to connect to: [localhost:5672]
2022-11-03 16:09:36.284  INFO 7360 --- [   scheduling-1] o.s.a.r.c.CachingConnectionFactory       : Created new connection: rabbitConnectionFactory#6dc1484:
Sender sent message : Hello message
Sender sent message : Hello message
Sender sent message : Hello message
Sender sent message : Hello message
```

Sender started, sending messages to queue run instance 2

queue + distanation

9999 port number to be changed in application.properties file



Run/Debug Configurations
Application
Async1Application
Store as project file
Modify options
Alt+M
Press Alt for field hints
Separate variables with semicolon: VAR=value; VAR1=value1
Open run/debug tool window when started
Allow multiple instances
Edit configuration templates...
OK
Cancel
Apply

Name: Async1Application

Build and run

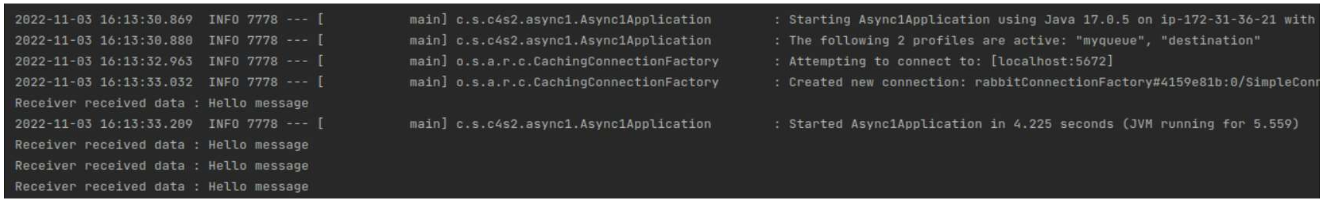
java 17 SDK of 'async1' module

com.stackroute.c4s2.async1.Async1Application

Program arguments

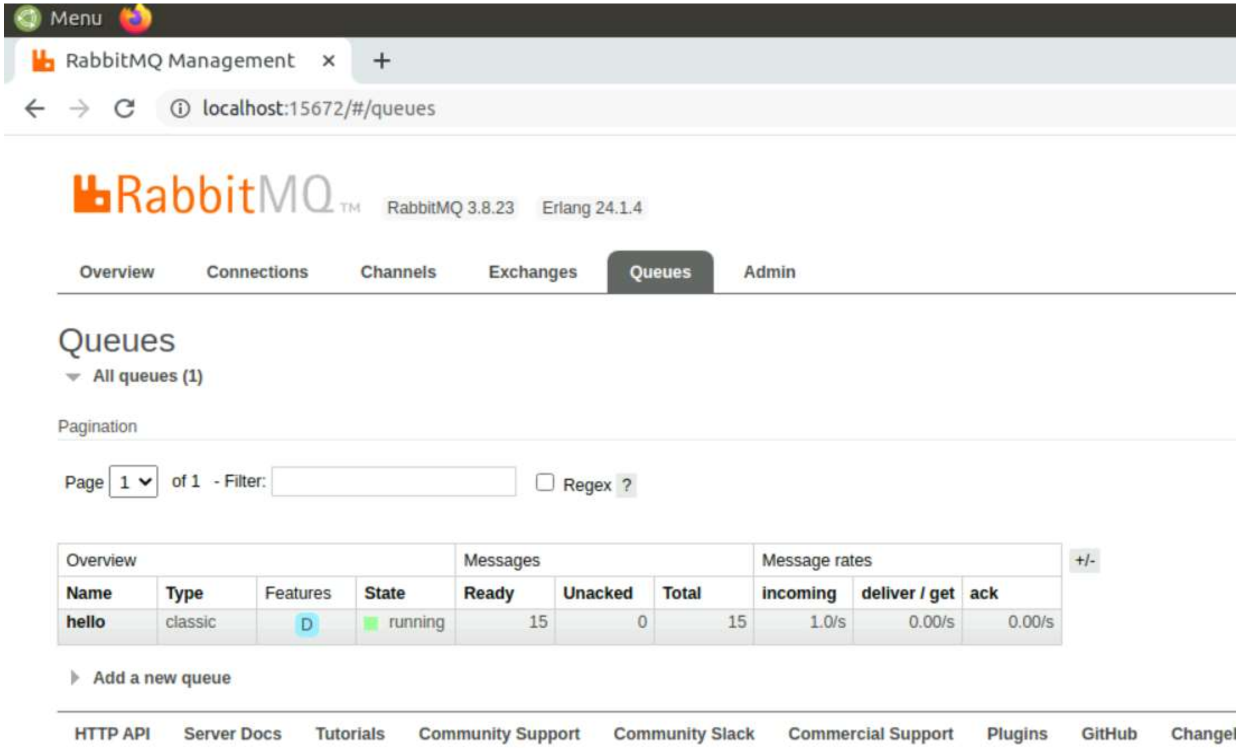
Working directory: /home/ubuntu/mywork/async1

Environment variables: pring.profiles.active=myqueue,destination



```
2022-11-03 16:13:30.869  INFO 7778 --- [           main] c.s.c4s2.async1.Async1Application        : Starting Async1Application using Java 17.0.5 on ip-172-31-36-21 with
2022-11-03 16:13:30.880  INFO 7778 --- [           main] c.s.c4s2.async1.Async1Application        : The following 2 profiles are active: "myqueue", "destination"
2022-11-03 16:13:32.963  INFO 7778 --- [           main] o.s.a.r.c.CachingConnectionFactory       : Attempting to connect to: [localhost:5672]
2022-11-03 16:13:33.032  INFO 7778 --- [           main] o.s.a.r.c.CachingConnectionFactory       : Created new connection: rabbitConnectionFactory#4159e81b:0/SimpleConn
Receiver received data : Hello message
2022-11-03 16:13:33.209  INFO 7778 --- [           main] c.s.c4s2.async1.Async1Application        : Started Async1Application in 4.225 seconds (JVM running for 5.559)
Receiver received data : Hello message
Receiver received data : Hello message
Receiver received data : Hello message
```

observe queue in dashboard before starting receiver



Queues

All queues (1)

Pagination

Page 1 of 1 - Filter: □ Regex ?

Overview				Messages			Message rates		
Name	Type	Features	State	Ready	Unacked	Total	incoming	deliver / get	ack
hello	classic	D	🟩 running	15	0	15	1.0/s	0.00/s	0.00/s

+/-

Add a new queue