

fromdevelopertodeveloper

# Spring Cloud Stream

## Content



What is Spring Cloud Stream



Our Project overview



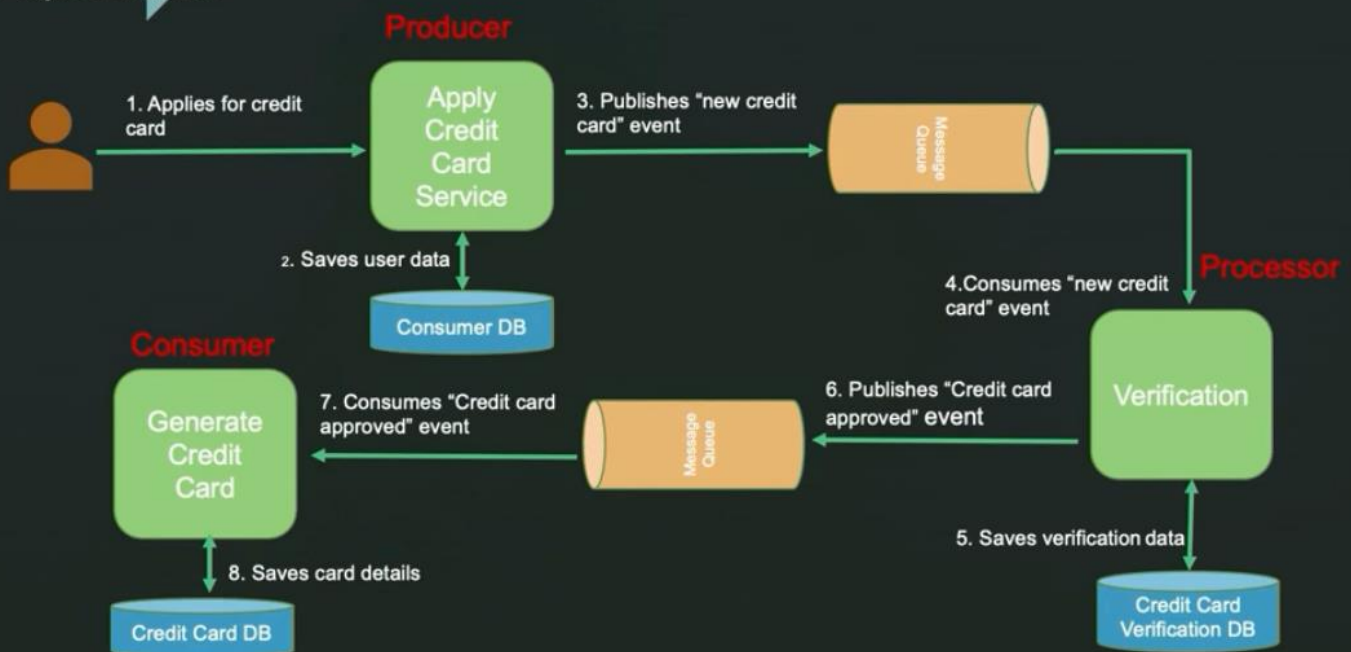
What is Spring cloud stream?



Spring Cloud Stream is a framework to build scalable event-driven microservices connected with shared messaging systems.

It provides a flexible programming model built on already established Spring Framework

## Project overview



# What we need

IDE

RabbitMQ

MySql

We shall use Docker image for Mysql and Rabbitmq. If you already have docker installed in your computer, you can run "RabbitMQ" and "MySql" using the command :

```
docker run -d --hostname rabbitmq --name rabbitmq -p 15672:15672 -p 5672:5672 rabbitmq:3.9.1-management
```

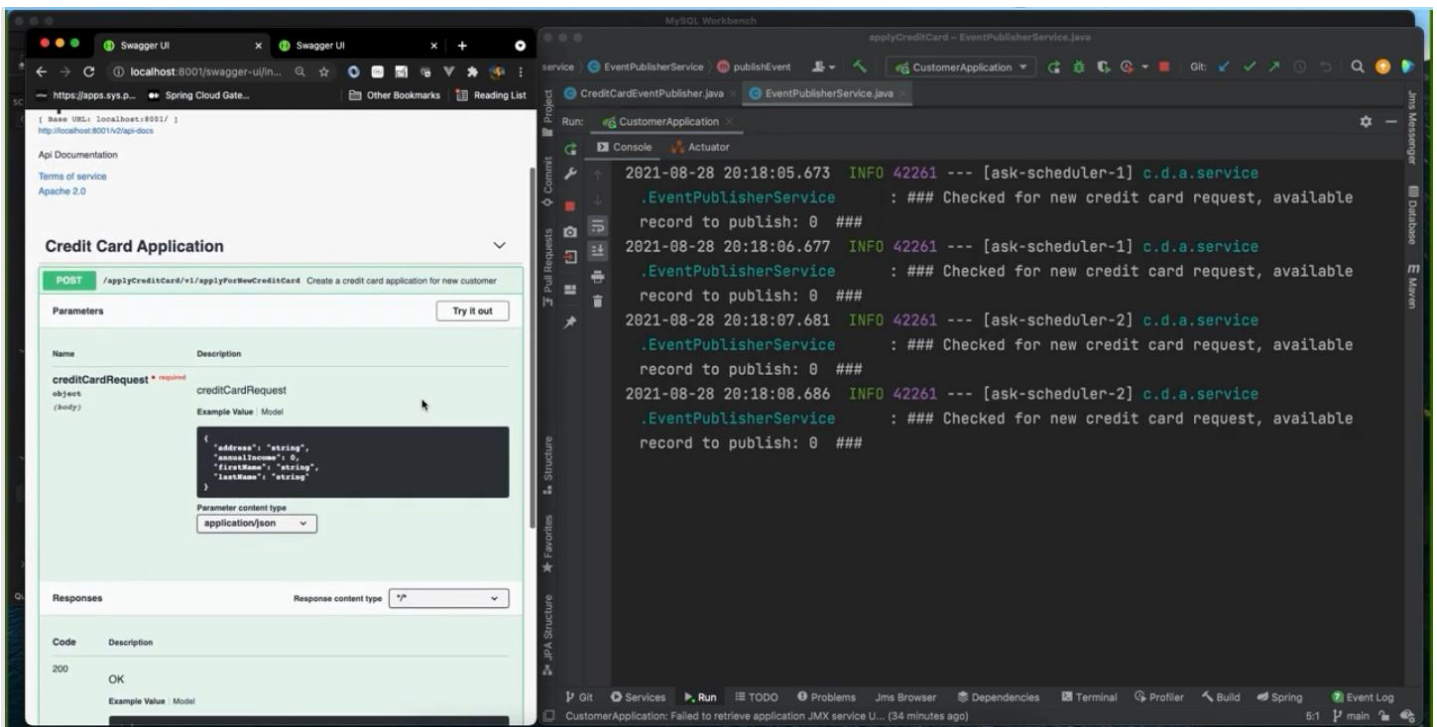
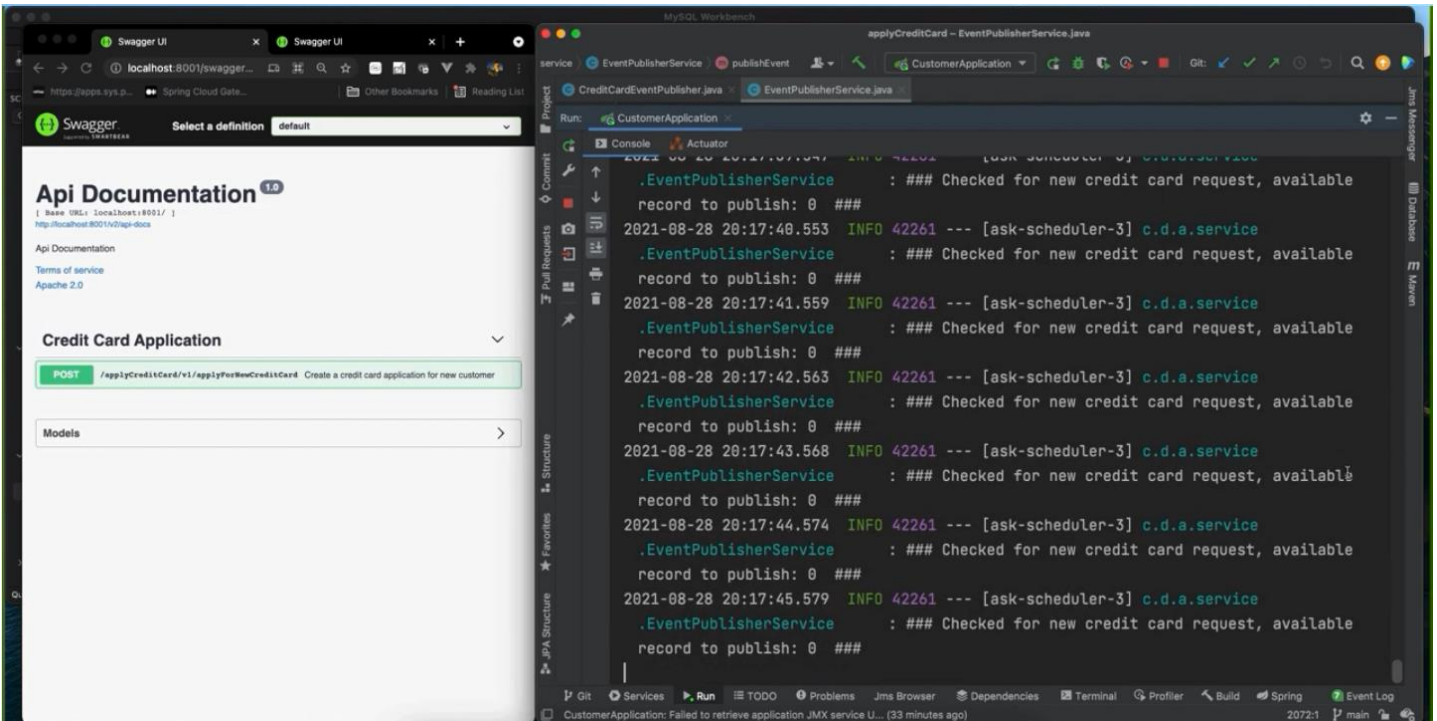
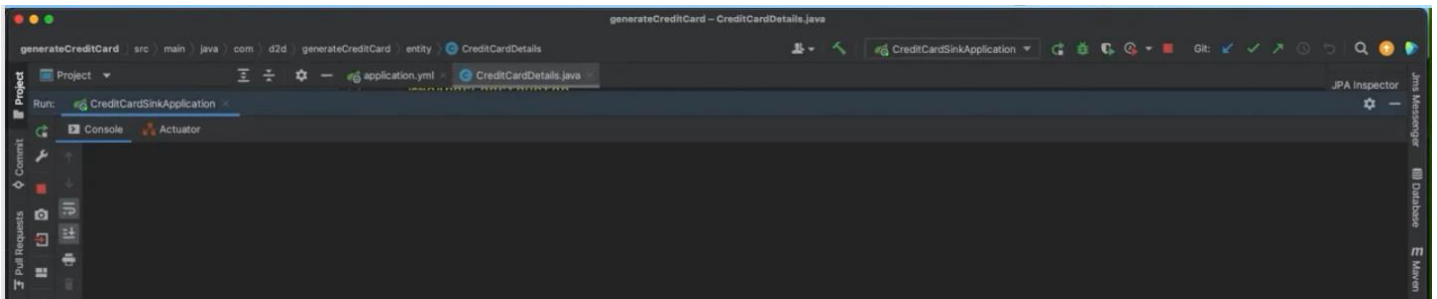
```
docker run --name mysql8 -p 3306:3306 -e MYSQL_ROOT_PASSWORD=root -e MYSQL_DATABASE=testdb -e MYSQL_USER=admin -e MYSQL_PASSWORD=root -d mysql
```

Alternatively , you can also download and install from the below link. Incase you face any issue please share with us in comment section.

Link: <https://lostechies.com/derekgreer/2012/03/05/rabbitmq-for-windows-introduction/> ,  
<https://www.rabbitmq.com/install-homebrew.html>  
<https://dev.mysql.com/doc/refman/8.0/en/windows-installation.html>, <https://flaviocopes.com/mysql-how-to-install/>

```
Run: CustomerApplication
2021-08-28 20:09:49.008 INFO 42261 --- [ask-scheduler-4] c.d.a.service.EventPublisherService : ### Checked for new credit card request,
available record to publish: 0 ###
2021-08-28 20:09:50.013 INFO 42261 --- [ask-scheduler-4] c.d.a.service.EventPublisherService : ### Checked for new credit card request,
available record to publish: 0 ###
2021-08-28 20:09:51.019 INFO 42261 --- [ask-scheduler-4] c.d.a.service.EventPublisherService : ### Checked for new credit card request,
available record to publish: 0 ###
2021-08-28 20:09:52.025 INFO 42261 --- [ask-scheduler-4] c.d.a.service.EventPublisherService : ### Checked for new credit card request,
available record to publish: 0 ###
2021-08-28 20:09:53.029 INFO 42261 --- [ask-scheduler-4] c.d.a.service.EventPublisherService : ### Checked for new credit card request,
available record to publish: 0 ###
2021-08-28 20:09:54.035 INFO 42261 --- [ask-scheduler-4] c.d.a.service.EventPublisherService : ### Checked for new credit card request,
available record to publish: 0 ###
2021-08-28 20:09:55.040 INFO 42261 --- [ask-scheduler-4] c.d.a.service.EventPublisherService : ### Checked for new credit card request,
available record to publish: 0 ###
2021-08-28 20:09:56.044 INFO 42261 --- [ask-scheduler-4] c.d.a.service.EventPublisherService : ### Checked for new credit card request,
available record to publish: 0 ###
2021-08-28 20:09:57.048 INFO 42261 --- [ask-scheduler-4] c.d.a.service.EventPublisherService : ### Checked for new credit card request,
available record to publish: 0 ###
2021-08-28 20:09:58.053 INFO 42261 --- [ask-scheduler-4] c.d.a.service.EventPublisherService : ### Checked for new credit card request,
available record to publish: 0 ###
```

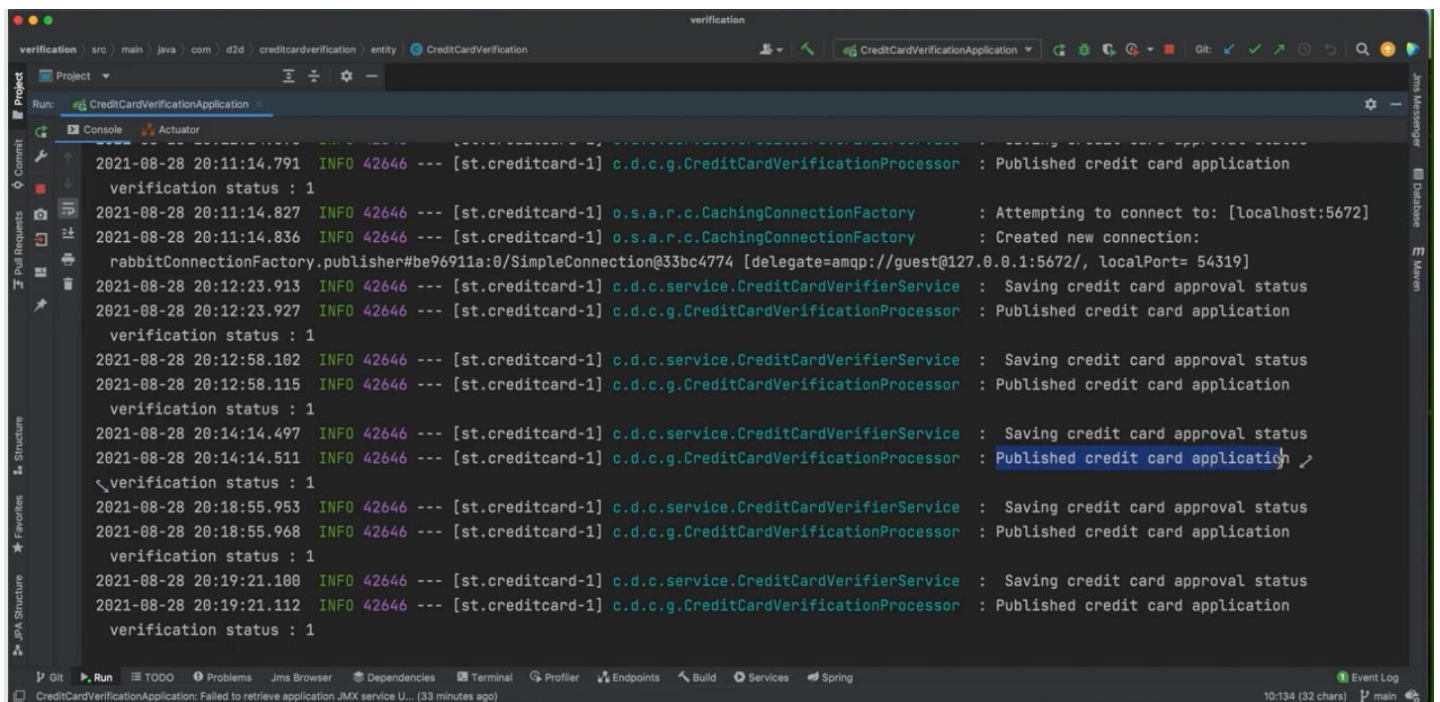
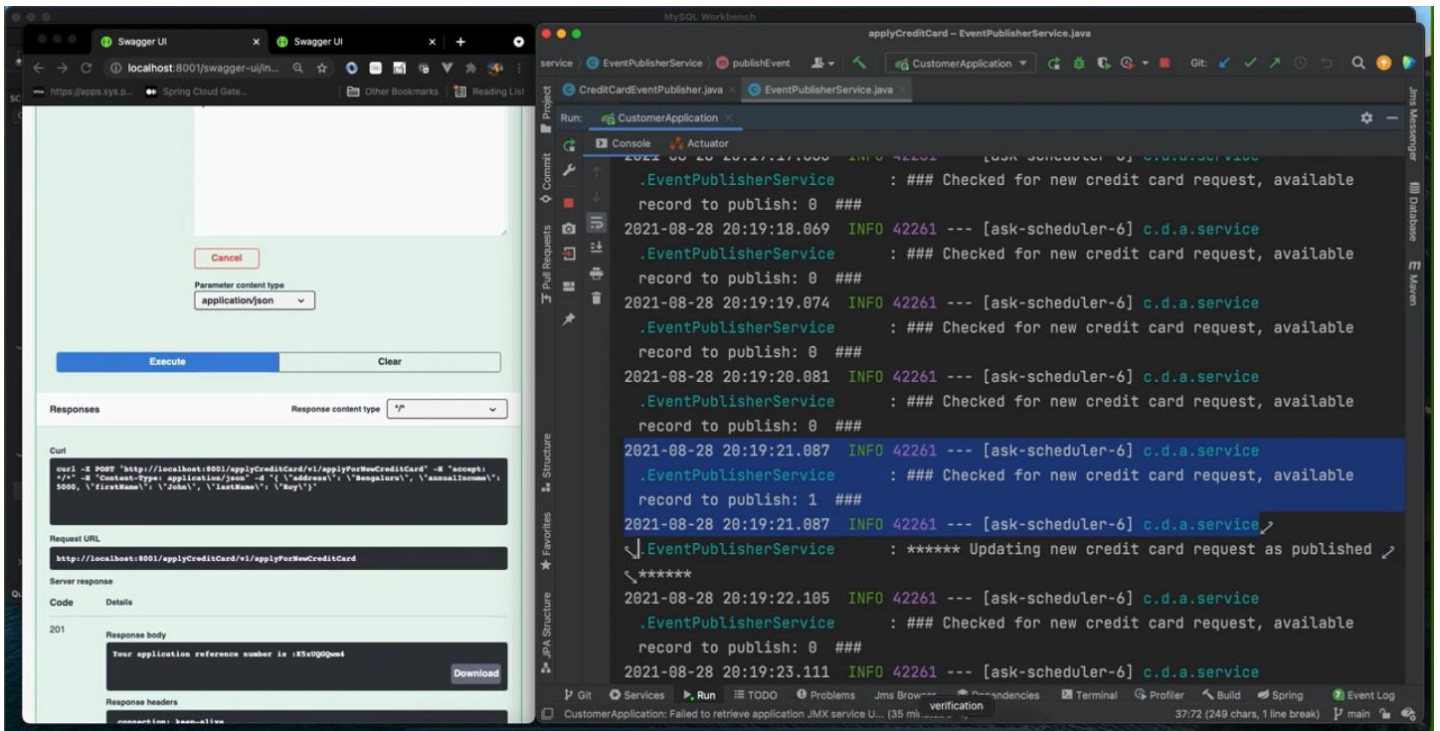
```
Run: CreditCardVerificationApplication
```





The top-left window shows the Swagger UI for a 'Credit Card Application' endpoint. The request is a POST to `/applyCreditCard/v1/applyForNewCreditCard` with a JSON body: `{ "address": "Bangalore", "annualIncome": 5000, "firstName": "John", "lastName": "M" }`. The top-right window shows the IDE console with logs from `EventPublisherService` indicating that the request is being processed and a record is available to be published.

The bottom-left window shows the Swagger UI after the request has been executed. The response is a 201 status code with the message: `Your application reference number is :85a5vd0m8`. The bottom-right window shows the IDE console with logs indicating that the record is now published (record to publish: 1) and the service is updating the new credit card request as published.





generateCreditCard - CreditCardDetails.java

Project: generateCreditCard | src: main | java: com | d2d: generateCreditCard | entity: CreditCardDetails

Run: CreditCardSinkApplication

Console:

```
2021-08-28 20:12:58.118 INFO 43405 --- [rd.creditcard-1] c.d.g.gateway.CreditCardConsumer : Received credit card application 1 :
2021-08-28 20:12:58.118 INFO 43405 --- [rd.creditcard-1] c.d.g.service.CreditCardService : ***** Generate credit card *****
2021-08-28 20:12:58.118 INFO 43405 --- [rd.creditcard-1] c.d.g.service.CreditCardService : ***** Generate credit card *****
2021-08-28 20:12:58.119 INFO 43405 --- [rd.creditcard-1] c.d.g.service.CreditCardService : **** Saving credit card ***
2021-08-28 20:12:58.133 INFO 43405 --- [rd.creditcard-1] c.d.g.service.CreditCardService : **** Credit card saved ***
2021-08-28 20:14:14.514 INFO 43405 --- [rd.creditcard-1] c.d.g.gateway.CreditCardConsumer : Received credit card application 1 :
2021-08-28 20:14:14.514 INFO 43405 --- [rd.creditcard-1] c.d.g.service.CreditCardService : ***** Generate credit card *****
2021-08-28 20:14:14.515 INFO 43405 --- [rd.creditcard-1] c.d.g.service.CreditCardService : ***** Generate credit card *****
2021-08-28 20:14:14.515 INFO 43405 --- [rd.creditcard-1] c.d.g.service.CreditCardService : **** Saving credit card ***
2021-08-28 20:14:14.528 INFO 43405 --- [rd.creditcard-1] c.d.g.service.CreditCardService : **** Credit card saved ***
2021-08-28 20:18:55.971 INFO 43405 --- [rd.creditcard-1] c.d.g.gateway.CreditCardConsumer : Received credit card application 1 :
2021-08-28 20:18:55.971 INFO 43405 --- [rd.creditcard-1] c.d.g.service.CreditCardService : ***** Generate credit card *****
2021-08-28 20:18:55.971 INFO 43405 --- [rd.creditcard-1] c.d.g.service.CreditCardService : ***** Generate credit card *****
2021-08-28 20:18:55.972 INFO 43405 --- [rd.creditcard-1] c.d.g.service.CreditCardService : **** Saving credit card ***
2021-08-28 20:18:55.985 INFO 43405 --- [rd.creditcard-1] c.d.g.service.CreditCardService : **** Credit card saved ***
2021-08-28 20:19:21.114 INFO 43405 --- [rd.creditcard-1] c.d.g.gateway.CreditCardConsumer : Received credit card application 1 :
2021-08-28 20:19:21.114 INFO 43405 --- [rd.creditcard-1] c.d.g.service.CreditCardService : ***** Generate credit card *****
2021-08-28 20:19:21.115 INFO 43405 --- [rd.creditcard-1] c.d.g.service.CreditCardService : ***** Generate credit card *****
2021-08-28 20:19:21.115 INFO 43405 --- [rd.creditcard-1] c.d.g.service.CreditCardService : **** Saving credit card ***
2021-08-28 20:19:21.126 INFO 43405 --- [rd.creditcard-1] c.d.g.service.CreditCardService : **** Credit card saved ***
```

Run: CreditCardSinkApplication: Failed to retrieve application JMX service U... (30 minutes ago)

MySQL Workbench

Local

Schemas

Query 1

```
1. select * from cc_application.t_credit_card_application;
2. select * from cc_verification.t_credit_card_verification;
3. select * from cc_details.t_credit_card_details;
4. commit
```

Result Grid:

id	credit_card_number	cvv	first_name	last_name	reference_id	approval_status
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						

MySQL Workbench

Local

Schemas

Query 1

```
1. select * from cc_application.t_credit_card_application;
2. select * from cc_verification.t_credit_card_verification;
3. select * from cc_details.t_credit_card_details;
4. commit
```

Result Grid:

id	address	annual_income	first_name	last_name	is_published	reference_id
1	Bengaluru	5000	John	Roy	1	REoSVadWb8
2	Bengaluru	5000	John	Roy	1	K5xUQGQwm4
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						

MySQL Workbench

Local

Administration Schemas Query 1

Limit to 1000 rows

```
1 select * from cc_application.t_credit_card_application;
2 select * from cc_verification.t_credit_card_verification;
3 select * from cc_details.t_credit_card_details;
4 commit
```

100% 1:2

Result Grid

id	address	annual_income	first_name	last_name	reference_id	application_status
1	Bengaluru	5000	John	Roy	REoSvSdWb8	APPROVED
2	Bengaluru	5000	John	Roy	K5xUQGQwm4	APPROVED

credit\_card\_details 2 t\_credit\_card\_application 10 t\_credit\_card\_details 13 t\_credit\_card\_details 14 t\_credit\_card\_application 16 t\_credit\_card\_verification 25 Apply Reset

MySQL Workbench

Local

Administration Schemas Query 1

Limit to 1000 rows

```
1 select * from cc_application.t_credit_card_application;
2 select * from cc_verification.t_credit_card_verification;
3 select * from cc_details.t_credit_card_details;
4 commit
```

100% 1:3

Result Grid

id	credit_card_number	cvv	first_name	last_name	reference_id	approval_status
1	843236719474	919	John	Roy	REoSvSdWb8	APPROVED
2	917582777160	782	John	Roy	K5xUQGQwm4	APPROVED

credit\_card\_details 2 t\_credit\_card\_application 10 t\_credit\_card\_details 13 t\_credit\_card\_details 14 t\_credit\_card\_application 16 t\_credit\_card\_details 26 Apply Reset

Swagger UI

localhost:8003/swagger-ui/index.html

https://apps.sys.p... Spring Cloud Gate...

Swagger

Select a definition default

# Api Documentation 1.0

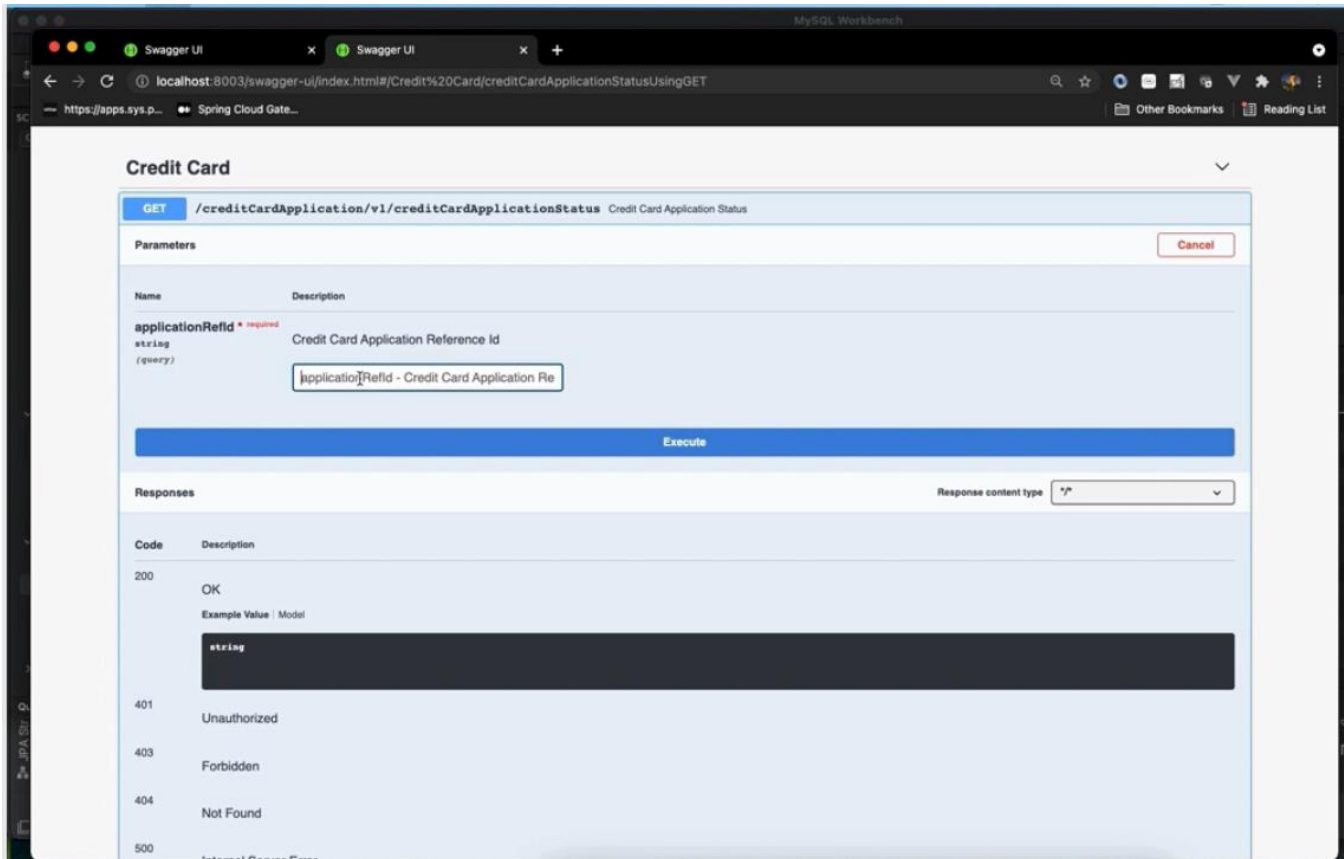
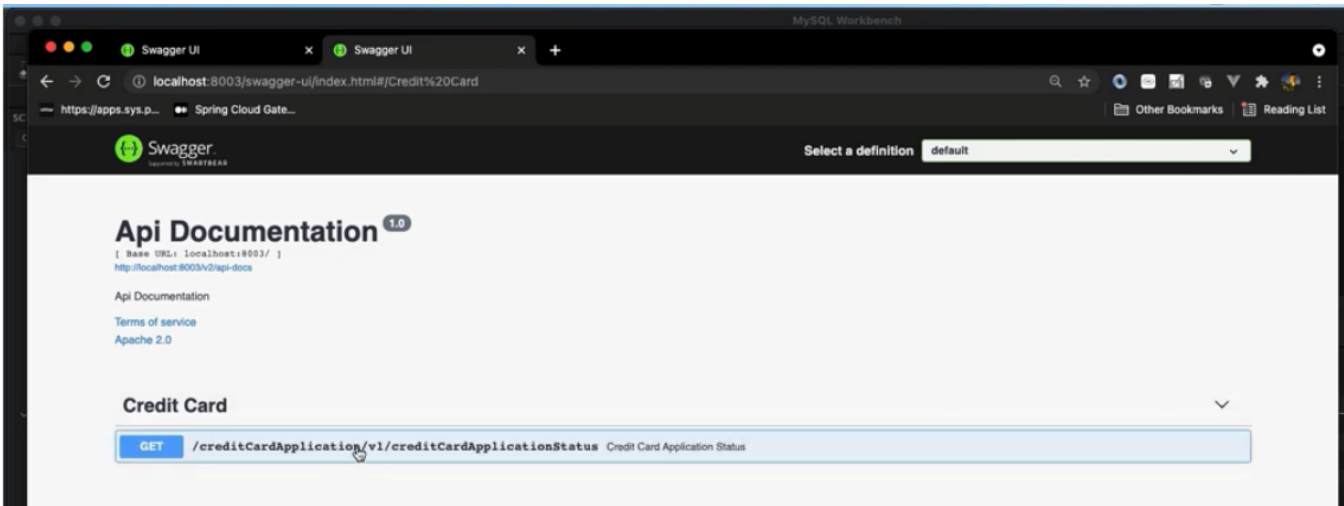
[ Base URL: localhost:8003/ ]  
http://localhost:8003/v2/api-docs

Api Documentation

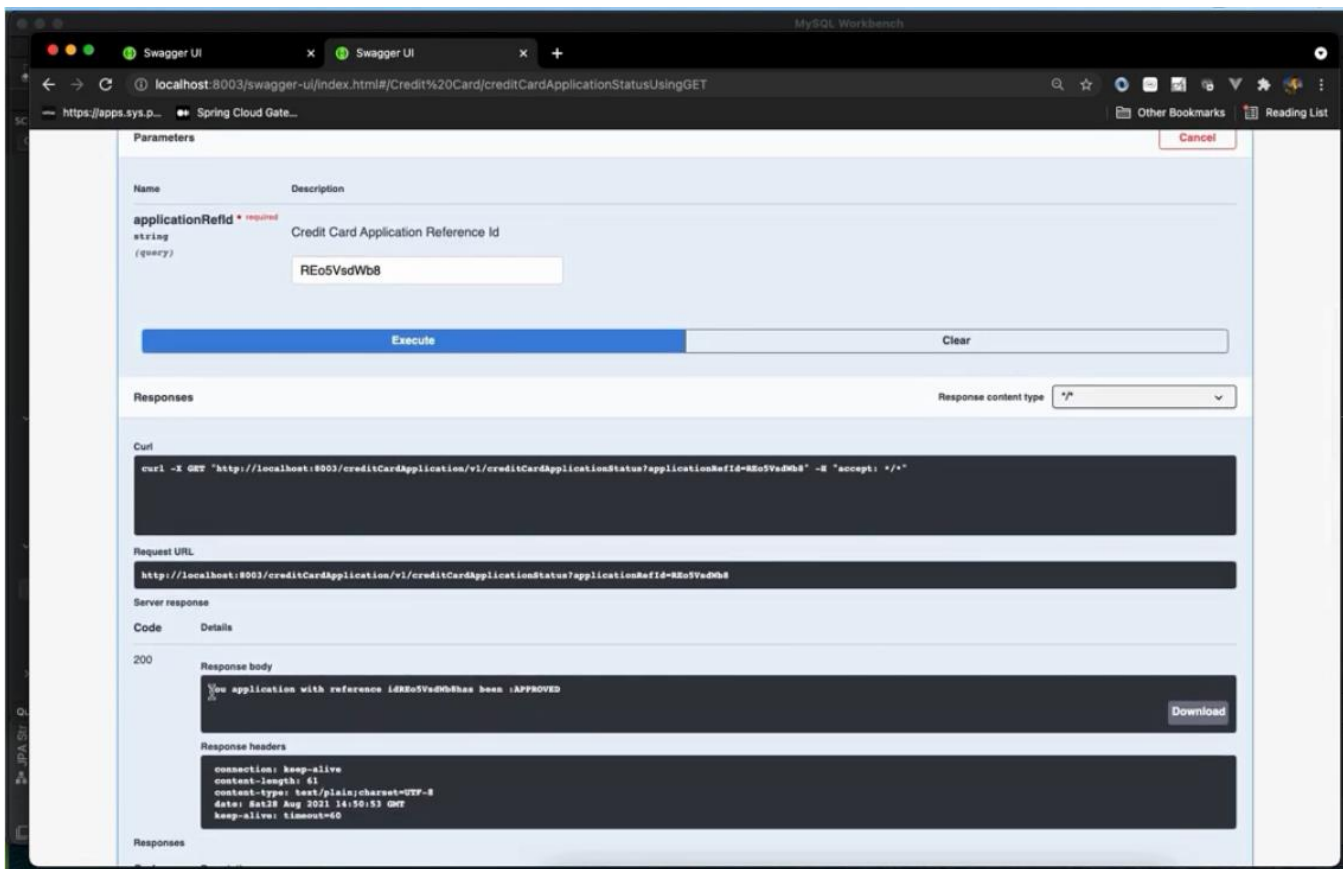
[Terms of service](#)

[Apache 2.0](#)

## Credit Card

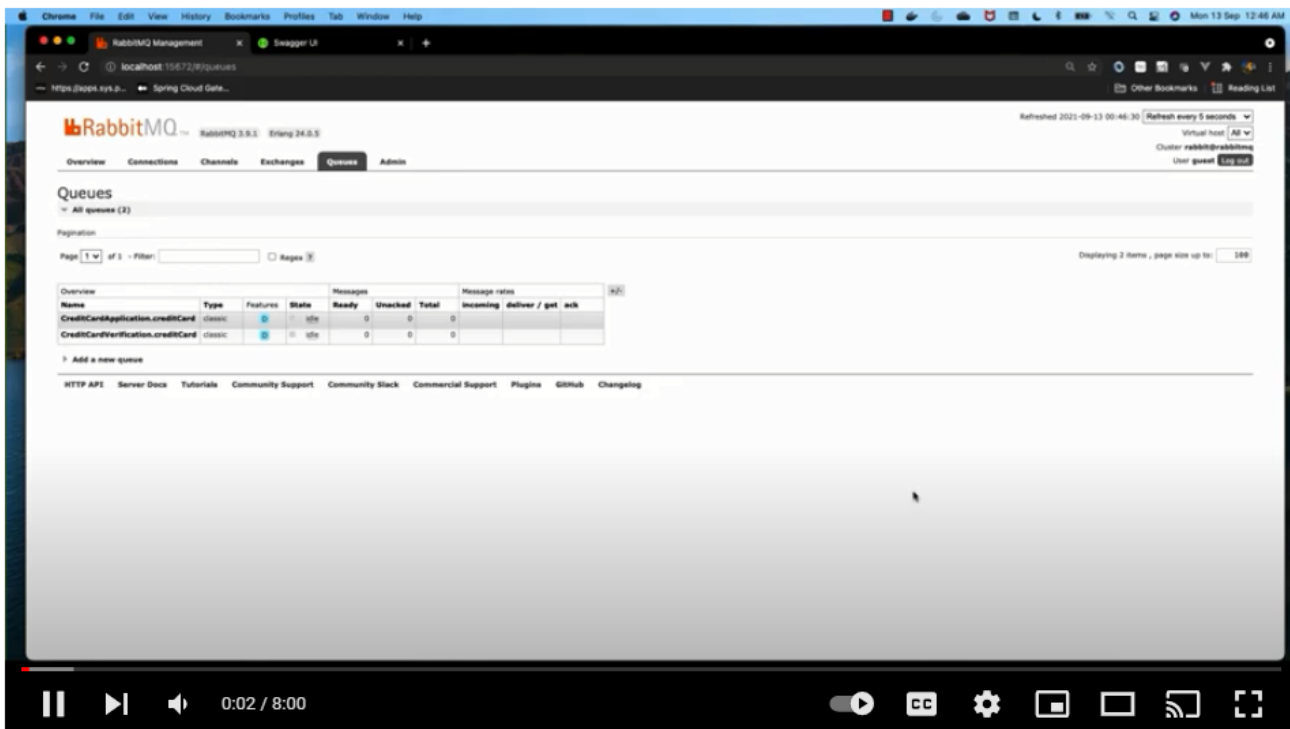






YouTube

Search



## Spring Cloud Stream Ep 07:- How to handle exception in Spring Cloud Stream using Dead Letter Queue



Code Techies  
328 subscribers

Subscribe

7

Share

Download

...

How to handle exception in Spring cloud stream with deal letter queue? How to deal with invalid message format in Spring Cloud Stream?

Sample Data:

```
{
  "event": "VERIFY_Credit_Card",
  "creditCardVerificationStatus": [
    {
      "firstName": "John",
      "lastName": "Roy",
      "annualIncome": 9999,
      "address": "Bang",
      "refId": "OphNsblPHB",
      "status": "APPROVED"
    }
  ]
}
```

RabbitMQ Management interface showing the 'Queues' tab for a queue named 'CreditCardVerification.creditCard'. The queue is currently idle. The 'Publish message' section is active, showing a form to publish a message. The payload field is empty, and the 'Publish message' button is visible.

RabbitMQ Management interface showing the 'Queues' tab for a queue named 'CreditCardVerification.creditCard'. The queue is currently idle. The 'Publish message' section is active, showing a form to publish a message. The payload field contains a JSON object representing the sample data provided in the first block. The 'Publish message' button is visible.

MySQL Workbench

Local

Administration Schemas Query 1

Limit to 1000 rows

1 • select \* from cc\_application.t\_credit\_card\_application;  
2 • select \* from cc\_verification.t\_credit\_card\_verification;  
3 • select \* from cc\_details.t\_credit\_card\_details;  
4 • commit

Result Grid

id	address	annual_income	first_name	last_name	is_published	reference_id
NULL	NULL	NULL	NULL	NULL	NULL	NULL

t\_credit\_card\_application 47

Query Completed

MySQL Workbench

Local

Administration Schemas Query 1

Limit to 1000 rows

1 • select \* from cc\_application.t\_credit\_card\_application;  
2 • select \* from cc\_verification.t\_credit\_card\_verification;  
3 • select \* from cc\_details.t\_credit\_card\_details;  
4 • commit

Result Grid

id	address	annual_income	first_name	last_name	reference_id	application_verification_status
NULL	NULL	NULL	NULL	NULL	NULL	NULL

t\_credit\_card\_verification 48

Query Completed



