

Step 1 : Need to create an account with Stripe.

Step 2 : Create a Spring boot project and add public and secret keys in the application.properties file.

Stripe Dashboard :

API keys				Learn more about API authentication →
Standard keys				
Create a key that unlocks full API access, enabling extensive interaction with your account. Learn more				
NAME	TOKEN	LAST USED	CREATED	
Publishable key	pk_test_51Lc4xSSAs5DsDSgQpOFFmoaa56cdPt3VIQbA1jLZRx2vn0Pt0jyijNmDAv43xtzVcn11jEUJIQZxgJaZKIIc0xNr00Ub18xbbk	27 Feb	29 Aug 2022	...
Secret key	sk_test_51Lc4xSSAs5DsDSgQpOFFmoaa56cdPt3VIQbA1jLZRx2vn0Pt0jyijNmDAv43xtzVcn11jEUJIQZxgJaZKIIc0xNr00Ub18xbbk	27 Feb	29 Aug 2022	...

Application.properties file :

```
stripe.secretKey = sk_test_51Lc4xSSAs5DsDSgQpOFFmoaa56cdPt3VIQbA1jLZRx2vn0Pt0jyijNmDAv43xtzVcn11jEUJIQZxgJaZKIIc0xNr00Ub18xbbk
stripe.apikey = pk_test_51Lc4xSSAs5DsDSgQZDwuQXQcocduHOoueQ8TNOAXKLN0nXPfvzX4gFRsN8ia3VOGaOtqDgUPygzzaOeJQYzy6xK00ZsOBsxOY
```

Step 3 : Add the Stripe Dependency into the pom.xml file.

```
<dependency>
    <groupId>com.stripe</groupId>
    <artifactId>stripe-java</artifactId>
    <version>22.13.0</version>
</dependency>
```

Step 4 : Create a Customer by using the Stripe API with some required parameters like name,email and phone number.

Stripe API Documentation :- <https://docs.stripe.com/api>

Required Parameters :

```
{  
    "name" : "Akshay",  
    "email" : "120.ap4@gmail.com",  
    "countryCode" : "+91",  
    "mobileNo": "9512463262"  
}
```

Code :

```
CustomerCreateParams customerParam = CustomerCreateParams.builder()  
    .setName(request.getName())  
    .setEmail(request.getEmail())  
    .setPhone(request.getCountryCode() + request.getMobileNo())  
    .build();  
  
return Customer.create(customerParam);
```

- Once you create a customer you can see those customers in their Stripe Dashboard.

Customers

Name	Email	Default payment method	Created
Akshay	120.ap4@gmail.com		27 Feb, 12:10

1 result

Akshay

120.ap4@gmail.com

Overview **Events and logs** **Create payment** **Create invoice** **...**

Payments **Create**

No payments

Insights

Spent	Since	Lost disputes
₹0.00	Feb 2024	₹0.00

Payment methods **Add**

No payment methods

Details **Edit**

Customer ID: cus_PdbXUffH0hwGpRt

Invoice credit balance **Adjust balance**

Customer does not currently have a balance.

- Also once you create a customer then stripe provides one unique customer id for further transactions.
- Also *stripe provides the logs details* with all the information.

Logs req_3PIjkPgZKdX5ps

POST /v1/customers 200 OK

Time	27/02/2024, 12:10:05	Source	Stripe/v1 JavaBindings/20.128.0
IP address	115.246.27.186	Related	customer - cus_PdbXUfHOhwGpRE
API Version	2020-08-27	Idempotency	Key - e9e93c8b-cdb3-4157-a885-4d9e22d23281

Request POST body <pre> 1 { 2 "email": "120.ap4@gmail.com", 3 "phone": "+919512463262", 4 "name": "Akshay" 5 }</pre>	Response body <pre> 1 { 2 "id": "cus_PdbXUfHOhwGpRE", 3 "object": "customer", 4 "address": null, 5 "balance": 0, 6 "created": 1709016005, 7 "currency": null, 8 "default_currency": null, 9 "default_source": null, 10 "delinquent": false, 11 "description": null, 12 "discount": null.</pre>
---	--

Request query parameters <p>No query parameters</p>	
---	--

- Also Stripe provides API For Update Customer,Get All customer, Delete Customer and other.

Step 5 : Create a Customer payment method by using a card. *Stripe test mode provides some test card details* which was listed in below link by using anyone you can test the things.

Stripe Testing Payment details document : <https://docs.stripe.com/testing>

Note :- As of now in the test mode we can't send a test card details. We need to pass card token and all the tokens mention in the above stripe document.

BRAND	TOKEN
Visa	<code>tok_visa</code>
Visa (debit)	<code>tok_visa_debit</code>
Mastercard	<code>tok_mastercard</code>
Mastercard (debit)	<code>tok_mastercard_debit</code>
Mastercard (prepaid)	<code>tok_mastercard_prepaid</code>
American Express	<code>tok_amex</code>
Discover	<code>tok_discover</code>

Code :-

```
// Retrieve the token using the test token
Token token = Token.retrieve(testingToken); //tok_visa
// Print the token ID
System.out.println("Token ID: " + token.getId());

        String newCardFingerPrint = token.getCard().getFingerprint();

        // getting all cards of customer
        PaymentMethodListParams paymentMethodListparams =
PaymentMethodListParams
        .builder()
        .setCustomer(customerId)
        .setType(com.stripe.param.PaymentMethodListParams.Type.CARD)
        .build();
        PaymentMethodCollection paymentMethods =
PaymentMethod.list(paymentMethodListparams);

        // matching card details
        Boolean isDuplicateCard = paymentMethods.getData().stream().anyMatch(method->
            method.getCard().getFingerprint().equals(newCardFingerPrint));

        // Duplicate Card
        if(isDuplicateCard) {
            throw new CardException("Duplicate Card Found Exception ",
                null, null, null, null, null, null, null);
        }
        .build();

        // create params for the payment method creation
        Map<String, Object> paymentMethodParams = new HashMap<>();
```

```

paymentMethodParams.put("type", "card");
paymentMethodParams.put("card", Map.of("token", token.getId()));
// Create PaymentMethod
PaymentMethod paymentMethod = PaymentMethod.create(paymentMethodParams);
    .build();

        PaymentMethodAttachParams attachParams =
PaymentMethodAttachParams.builder()
    .setCustomer(customerId)
    .build();

    paymentMethod.attach(attachParams);

    // for save data into the database
PaymentDetails paymentDetails = new PaymentDetails();
paymentDetails.setCustomerId(customerId);
paymentDetails.setPaymentMethodId(paymentMethod.getId());
paymentDetails.setTokenId(token.getId());

if(paymentDetailsRepository.findByCustomerIdAndIsDefaultPaymentMethodTrue(customerId) ==
null)
{
    // if any payment method is not available then set by default true
    paymentDetails.setDefaultPaymentMethod(true);

}

return paymentDetailsRepository.save(paymentDetails);

```

- After creation of the payment method you can see the method in the stripe dashboard.

Payment methods		Add
▼	Visa 4242	Expires Feb 2025 ...
Name	—	
Number 4242	
Fingerprint	kk0ZXRFkwJFwQKyK	
Expires	2 / 2025	
Type	Visa credit card	
Issuer	Stripe Payments UK Limited	
ID	pm_10oleFSAs5DsDSgQxW3zGSdJ	
Billing address	No address	
Phone	—	
Email	—	
Origin	United States	
CVC check	Passed	

Also stripe provides a default or primary paymethod where you can set your default payment method.

▼	Visa 4242	Default	Expires Feb 2025	...
Name	—			
Number 4242			
Fingerprint	kk0ZXRFkwJFwQKyK			
Expires	2 / 2025			
Type	Visa credit card			
Issuer	Stripe Payments UK Limited			
ID	pm_10p2c6SA5DsDSgQUNZwnwFi			
Billing address	No address			
Phone	—			
Email	—			
Origin	United States			
CVC check	Passed			

Step 6 :- Create a payment intent for the payments

```
// for find the primary payment method from the database
PaymentDetails method = paymentDetailsRepository.
    findByCustomerIdAndIsDefaultPaymentMethodTrue(customerId);

PaymentIntentCreateParams params =
    PaymentIntentCreateParams
        .builder()
        .setCustomer(customerId) // customer id
        .addPaymentMethodType("card") // payment method card
        //
.setSetupFutureUsage(PaymentIntentCreateParams.SetupFutureUsage.OFF_SESSION)
        .setAmount(amount * 100) // amount in paisa
```

```

.setCurrency("inr") // currency
.setPaymentMethod(method.getPaymentMethodId()) // payment
method id

.setCaptureMethod(CaptureMethod.MANUAL) // capture manual
.setConfirm(true) // for confirmation after creation automatically
confirm the payment intent

.setOffSession(true) // session off true means user can capture
payment without any session if false then provide the authentication 3D secure
.build();

PaymentIntent paymentIntent = PaymentIntent.create(params);

// for saving the data into the database
PaymentIntentDetailEntity entity = new PaymentIntentDetailEntity();

entity.setPaymentIntentId(paymentIntent.getId());
entity.setAmount(paymentIntent.getAmount()/100);
entity.setCustomerId(customerId);
entity.setCreatedAt(LocalDateTime.now());
entity.setStatus("CREATED");

return intentDetailRepository.save(entity);

```

- After successful payment intent creation you can see the payment intent under payment section.

PAYMENT

₹500.00 INR Uncaptured

Last update 28 Feb, 18:34 Customer Akshay Payment method VISA **** 4242 Risk evaluation 18 Normal

Timeline

+ Add note

- Payment authorised, but not yet captured
28 Feb 2024, 18:34
- Payment started
28 Feb 2024, 18:34

- Also stripe provides a Update, Cancel and other Payment Intent.

Cancelled payment Intent

PAYMENT

₹200.00 INR Cancelled

Last update 29 Feb, 11:27	Customer Aarya	Payment method 4242	Risk evaluation 32 Normal
------------------------------	-----------------------------------	------------------------	--

Timeline

- ⌚ Payment cancelled. It may take a few days for the money to reach the customer's bank account.
[View details](#)
29 Feb 2024, 11:27
- 🕒 Payment authorised, but not yet captured
29 Feb 2024, 11:27
- ⌚ Payment started
29 Feb 2024, 11:27

Step 7 :- Create a capture method to capture the payment. Capture means debit from the payment method.

Code :-

```
// Retrieve the paymentIntent
PaymentIntent resource = PaymentIntent.retrieve(paymentIntentId);

        // capture the amount
        resource.capture();

        // update the data into the database
        PaymentIntentDetailEntity detailEntity =
intentDetailRepository.findByPaymentIntentId(paymentIntentId);
        detailEntity.setCapturedAmount(amount);
        detailEntity.setStatus("Success");
        return intentDetailRepository.save(detailEntity);
```

- After completing the payment capture you can see payment is captured and payment is succeed.

PAYMENT

₹500.00 INR Succeeded ✓

Last update 28 Feb, 18:39	Customer Akshay	Payment method  4242	Risk evaluation 18 Normal
------------------------------	------------------------------------	---	---

Timeline

- 🕒 ₹500.00 captured
28 Feb 2024, 18:39
- 🕒 Payment authorised
28 Feb 2024, 18:34
- 🕒 Payment started
28 Feb 2024, 18:34

Step 8 : Create Payment Refund If you need Stripe provides a refund related functionality.

Code :

```
RefundCreateParams params =  
RefundCreateParams.builder().setPaymentIntent(paymentIntentId).build();  
    Refund.create(params); // Create Refund  
  
PaymentIntentDetailEntity paymentIntentDetailEntity = intentDetailRepository  
    .findByPaymentIntentId(paymentIntentId);  
paymentIntentDetailEntity.setStatus("REFUNDED"); // update the status in database  
  
return intentDetailRepository.save(paymentIntentDetailEntity);
```

- After the Payment Refund we can see the details on the stripe Dashboard As mentioned below

 PAYMENT

₹200.00 INR Refunded 

Last update
29 Feb, 11:40

Customer
[Aarya](#)

Payment method
 **** 4242

Risk evaluation
 10 Normal

Timeline

-  Payment refunded. It may take a few days for the money to reach the customer's bank account.
[View details](#)
29 Feb 2024, 11:40
-  ₹200.00 captured
29 Feb 2024, 11:39
-  Payment authorised
29 Feb 2024, 11:38
-  Payment started
29 Feb 2024, 11:38

Source Code :- <https://github.com/axayp29/Stripe-Payment-Gateway>

What's done in the Source Code :-

Customer :-

1. Add Customer
2. Update Customer
3. Get All Customer
4. Delete Customer

Payment Method :-

1. Create a Payment Method of Customers.(Type : Card)
2. Get All Payment Methods of Customers.
3. Set Default Payment Method of Customer.

Payment Intent :-

1. Create Payment Intent
2. Update Payment Intent
3. Cancel Payment Intent

4. Capture Payment Intent

Refund :-

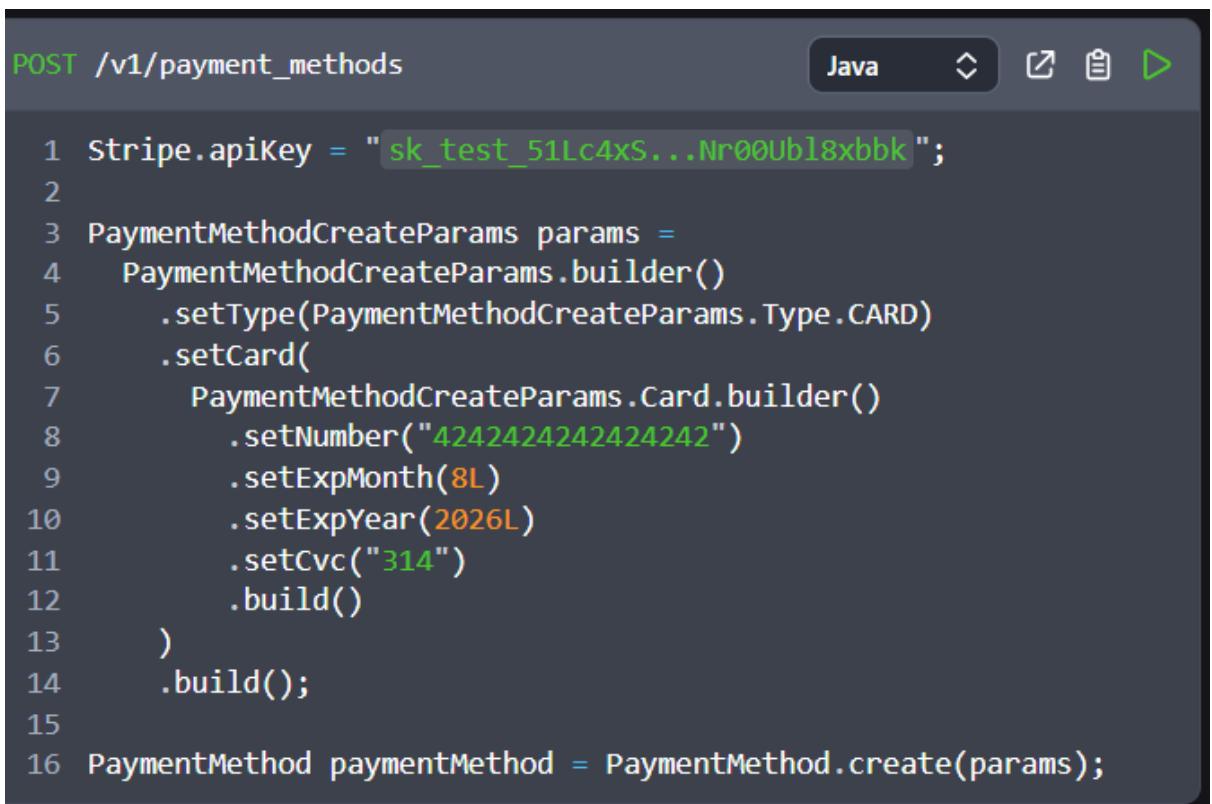
1. Create Refund

All the necessary data store into database.

What was the major issue I faced while implementation ?

1. Create a payment method using the card details because in the API documentation written for the creating a payment method using the card you need to send the card details(test card details if you are in test mode then) but while i send the request with card details it's gives me error like `Sending credit card number directly not the stripe api is generally unsafe. We suggest you use test tokens map to the test card you are using` so R&D for the same and find the solution.

- In documentation



The screenshot shows a code editor with a dark theme. At the top, there is a header bar with the text "POST /v1/payment_methods" on the left and "Java" on the right, along with some icons. Below the header, the Java code for creating a payment method is displayed:

```
1 Stripe.apiKey = "sk_test_51Lc4xS...Nr00Ubl8xbbk";
2
3 PaymentMethodCreateParams params =
4     PaymentMethodCreateParams.builder()
5         .setType(PaymentMethodCreateParams.Type.CARD)
6         .setCard(
7             PaymentMethodCreateParams.Card.builder()
8                 .setNumber("4242424242424242")
9                 .setExpMonth(8L)
10                .setExpYear(2026L)
11                .setCvc("314")
12                .build()
13            )
14        .build();
15
16 PaymentMethod paymentMethod = PaymentMethod.create(params);
```

Solution :-

```
// Retrieve the token using the test token
Token token = Token.retrieve(testingToken);
// Print the token ID
System.out.println("Token ID: " + token.getId());

Map<String, Object> paymentMethodParams = new HashMap<>();
paymentMethodParams.put("type", "card");
paymentMethodParams.put("card", Map.of("token", token.getId()));
// Create PaymentMethod
PaymentMethod paymentMethod = PaymentMethod.create(paymentMethodParams);
```

Here Testing token value is tok_visa as mentioned above. So by using the token create a payment method (in test mode).

2. After Creating the payment Intent I am not able to capture the amount stripe says you need to authenticate customer before capture the amount in stripe dashboard **3D Secure Attempt Incomplete**

The screenshot shows a Stripe dashboard page. At the top, there's a navigation bar with tabs like 'Payments', 'Charges', 'Refunds', etc. Below the navigation, there's a table with columns: 'Last update', 'Customer', and 'Payment method'. The 'Last update' column shows '28 Feb, 14:57'. The 'Customer' column shows 'Akshay'. The 'Payment method' column shows a VISA card icon followed by '.... 4242'. Below this table, there's a section titled 'Timeline' which lists two events: '3D Secure attempt incomplete' and 'Payment started'. Both events occurred on '28 Feb 2024, 14:57'. At the bottom, there's a section titled 'Payment details' showing an 'Amount' of '₹220.00'.

Solution :-

While creating the payment intent i need to set the to things
setConfirm(true) // for by default confirm the payment intent
setOffSession(true) // for not required any session of customer for the capture the payment

Payments

[Create](#)

AMOUNT	DESCRIPTION	DATE	...
₹500.00	Incomplete ⓘ pi_30oPs3SAs5DsDSgQ01J03z5B	27 Feb, 18:04	...
₹500.00	Incomplete ⓘ pi_30oPkisAs5DsDSgQ1Hlp5ahk	27 Feb, 17:57	...
₹5.00	Incomplete ⓘ pi_30oPhlSAs5DsDSgQ13nXQHdr	27 Feb, 17:54	...
₹50.00	Incomplete ⓘ Payment for Invoice	27 Feb, 17:16	...

