

GABRIELE ROMANATO

Menu

Usare le API di Stripe in Java

Stripe è una delle piattaforme di pagamento più popolari per l'elaborazione di transazioni online. Grazie alla sua API ben documentata, gli sviluppatori possono integrarla facilmente nelle proprie applicazioni. In questa guida, vedremo come utilizzare le API di Stripe con Java.

Prerequisiti

Prima di iniziare, assicurati di avere:

- Un account Stripe attivo.
- Java Development Kit (JDK) installato.
- Un progetto Java con Maven o Gradle.

Installazione della libreria Stripe

Per utilizzare l'SDK Java di Stripe, aggiungi la dipendenza nel tuo `pom.xml` se usi Maven:

```
<dependency>
  <groupId>com.stripe</groupId>
  <artifactId>stripe-java</artifactId>
  <version>22.9.0</version>
</dependency>
```

Se usi Gradle, aggiungi questa riga al file `build.gradle`:

```
dependencies {
    implementation 'com.stripe:stripe-java:22.9.0'
}
```

Configurazione dell'API Key

Per interagire con l'API di Stripe, devi impostare la tua chiave segreta:

```
import com.stripe.Stripe;

public class StripeConfig {
    public static void main(String[] args) {
        Stripe.apiKey = "sk_test_your_secret_key";
    }
}
```

Creazione di un pagamento

Per elaborare un pagamento con Stripe, puoi usare il seguente codice:

```
import com.stripe.exception.*;
import com.stripe.model.PaymentIntent;
import java.util.*;

public class StripePayment {
    public static void main(String[] args) {
        Stripe.apiKey = "sk_test_your_secret_key";

        Map params = new HashMap<>();
        params.put("amount", 5000); // Importo in centesimi
        params.put("currency", "eur");
        params.put("payment_method_types",
Arrays.asList("card"));

        try {
            PaymentIntent paymentIntent =
PaymentIntent.create(params);
            System.out.println("Payment Intent ID: " +
paymentIntent.getId());
        } catch (StripeException e) {
            e.printStackTrace();
        }
    }
}
```

```
}  
}
```

Gestione dei webhook

I webhook sono utili per ricevere notifiche sugli eventi di pagamento. Un esempio di gestione di un webhook con Java:

```
import spark.Spark;  
import com.stripe.model.Event;  
import com.stripe.net.Webhook;  
  
public class WebhookHandler {  
    public static void main(String[] args) {  
        Spark.post("/webhook", (request, response) -> {  
            String payload = request.body();  
            String sigHeader = request.headers("Stripe-  
Signature");  
            String endpointSecret = "your_webhook_secret";  
  
            try {  
                Event event = Webhook.constructEvent(payload,  
sigHeader, endpointSecret);  
                System.out.println("Evento ricevuto: " +  
event.getType());  
            } catch (Exception e) {  
                response.status(400);  
                return "Errore nella validazione del  
webhook";  
            }  
  
            response.status(200);  
            return "Webhook ricevuto";  
        });  
    }  
}
```

Conclusione

In questa guida abbiamo visto come integrare Stripe in un'applicazione Java, dalla configurazione iniziale alla gestione dei pagamenti e dei webhook.