**Payment Gateway Integration**

**using ‘razorpay**

**Step1.>> Add Depandancy in pom. Xml file-------**

<**dependency**>  
 <**groupId**>com.razorpay</**groupId**>  
 <**artifactId**>razorpay-java</**artifactId**>  
 <**version**>1.4.3</**version**>  
</**dependency**>

**Step2.>> Configure in application. properties files….**

**spring.datasource.url = jdbc:mysql://localhost:3306/paymentsys?useSSL=false&serverTimezone=UTC  
spring.datasource.username = root  
spring.datasource.password = test  
  
*# hibernate properties*spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.MySQL5InnoDBDialect  
  
*# Hibernate ddl auto*spring.jpa.hibernate.ddl-auto = update  
  
razorpay.key.id=rzp\_test\_sSqYVZEoJPeWV3  
razorpay.key.secret=M3mlLra4oVi1NGR6kefxBBX8**

**Step3.>> Create Databases schema in DB-------**

**Paymentsys**

**Step4.>> Create Entity class in Entity Package---------**

package com.PaymentSystem.entity;  
  
public class PaymentRequest {  
 private double amount;  
 private String currency;  
  
  
 public double getAmount() {  
 return amount;  
 }  
  
 public void setAmount(double amount) {  
 this.amount = amount;  
 }  
  
 public String getCurrency() {  
 return currency;  
 }  
  
 public void setCurrency(String currency) {  
 this.currency = currency;  
 }  
  
}

**Step 5.>> Create Controller layer in PaymentController—**

**package com.PaymentSystem.Controller;  
  
import com.PaymentSystem.Service.PaymentService;  
import com.razorpay.RazorpayException;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
  
@RestController  
@RequestMapping("/payments")  
public class PaymentController {  
 private final PaymentService paymentService;  
 @Autowired  
 public PaymentController(PaymentService paymentService) {  
 this.paymentService = paymentService;  
 }  
 @PostMapping("/create-order")  
 public String createOrder(@RequestParam("amount") int amount, @RequestParam("currency") String currency) throws RazorpayException {  
 String orderId = paymentService.createOrder(amount, currency);  
 return "Order created with ID: " + orderId;  
 }  
 @PostMapping("/verify-signature")  
 public String verifyPaymentSignature(@RequestParam String orderId, @RequestParam String paymentId, @RequestParam String signature) throws RazorpayException {  
 return paymentService.verifyPaymentSignature(orderId, paymentId, signature);  
 } *// Other controller methods for payment callbacks, etc.***  
**}**  
  
 **Step6.>> Create Service Layer on class PaymentService----**

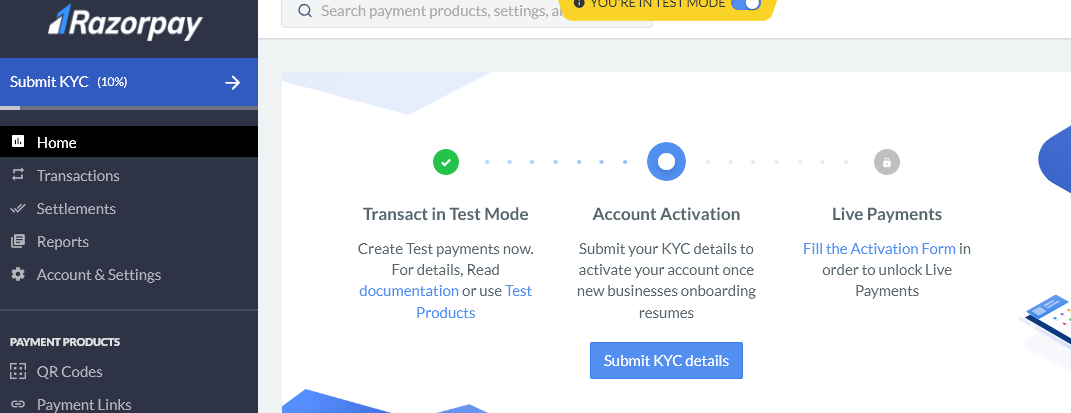
**package com.PaymentSystem.Service;  
  
  
import com.PaymentSystem.Config.RazorpayConfig;  
import com.razorpay.\*;  
import org.json.JSONObject;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service  
public class PaymentService {  
  
 private final RazorpayConfig razorpayConfig;  
  
 @Autowired  
 public PaymentService(RazorpayConfig razorpayConfig) {  
 this.razorpayConfig = razorpayConfig;  
 }  
  
 public String createOrder(int amount, String currency) throws RazorpayException {  
 RazorpayClient client = new RazorpayClient(razorpayConfig.getRazorpayKeyId(), razorpayConfig.getRazorpayKeySecret());  
 JSONObject options = new JSONObject();  
 options.put("amount", amount \* 100); *// Amount in paise or cents* options.put("currency", currency);  
 Order order = client.orders.create(options);  
 return order.get("id");  
 }  
  
 public String verifyPaymentSignature(String orderId, String paymentId, String signature) throws RazorpayException {  
 RazorpayClient client = new RazorpayClient(razorpayConfig.getRazorpayKeyId(), razorpayConfig.getRazorpayKeySecret());  
 JSONObject attributes = new JSONObject(client);  
 attributes.put("razorpay\_order\_id", orderId);  
 attributes.put("razorpay\_payment\_id", paymentId);  
 attributes.put("razorpay\_signature", signature);  
 Utils.*verifyPaymentSignature*(attributes, signature);  
 return "Payment signature verified successfully";  
 }**

**Step7.>> Create RozorPayConfig Class in Config package—**

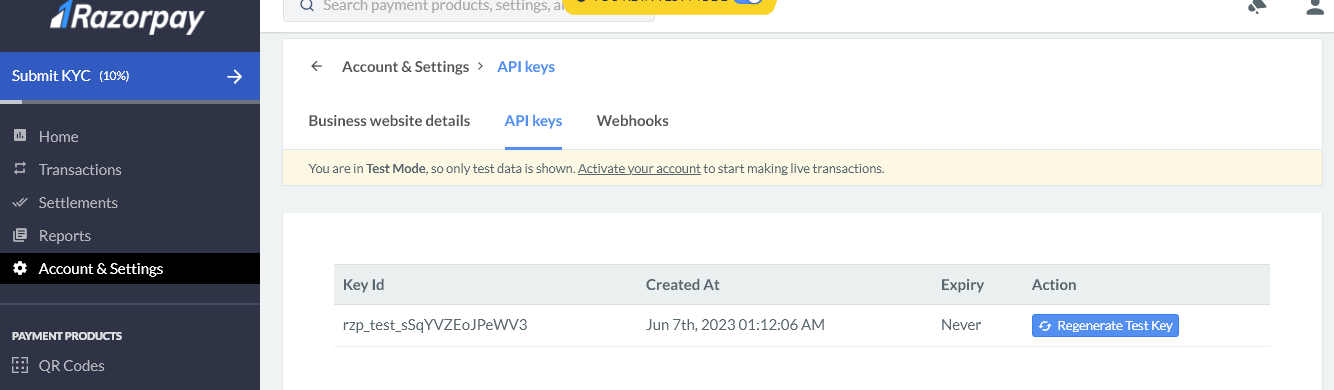
package com.PaymentSystem.Config;  
import org.springframework.beans.factory.annotation.Value;  
import org.springframework.context.annotation.Configuration;  
  
@Configuration  
public class RazorpayConfig {  
  
 @Value("${razorpay.key.id}")  
 private String razorpayKeyId;  
  
 @Value("${razorpay.key.secret}")  
 private String razorpayKeySecret;  
  
 public String getRazorpayKeyId() {  
 return razorpayKeyId;  
 }  
  
 public String getRazorpayKeySecret() {  
 return razorpayKeySecret;  
 }  
}

**Step8.>> Create ‘razorpay Account -------------**

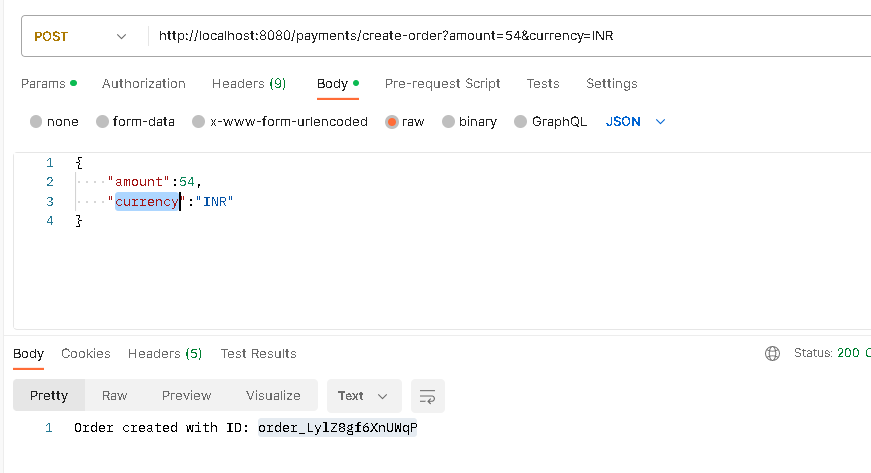
**Go to chrome and Type ‘razorpay once click and create accounts.**

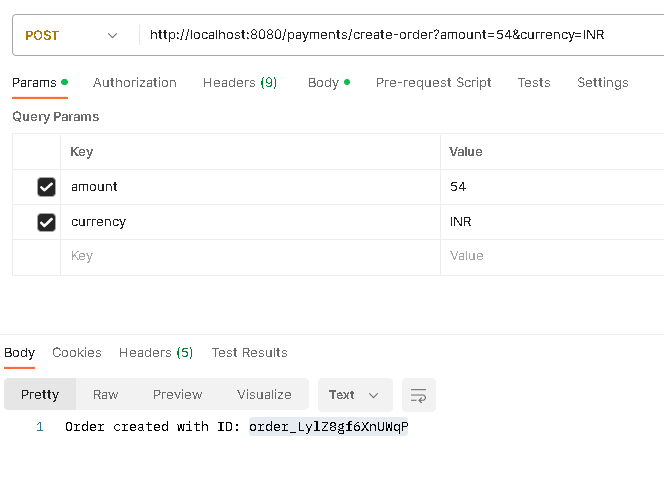
**1.** 

**2. Once the create the Go to Account -> API key here the available key and Secret copy and paste Application .Properties file.**

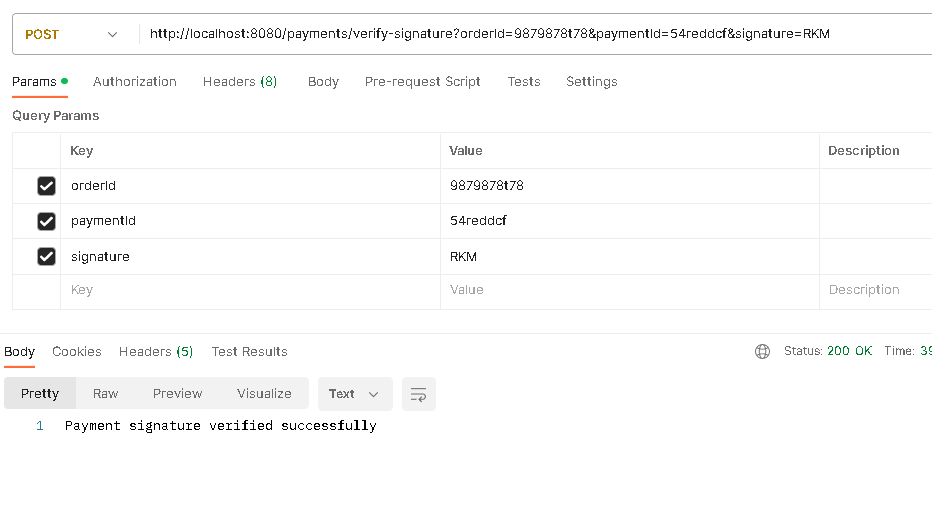


**3. Now Run our project Successfully then go to Postman :---------**





**4. After run we get Payment Successfully.**

**5. Then we go to Rozor Account and see in Order section here show Payment Details.-----**

