

打印机连接

SN: 922356303

KEY: ISyhBRHG

- **USER:** 3304559720@qq.com
- **UKEY:** Rlx7qlLZZIldKJZr

飞鹅打印机官方文档:

<http://help.feieyun.com/home/doc/zh;nav=1-1>

```
1  import org.springframework.web.bind.annotation.PostMapping;
2  import org.springframework.web.bind.annotation.RequestBody;
3  import org.springframework.web.bind.annotation.RestController;
4  import java.security.MessageDigest;
5  import java.security.NoSuchAlgorithmException;
6  import java.util.Base64;
7  import java.math.BigInteger;
8
9  import org.springframework.http.*;
10 import org.springframework.web.client.RestTemplate;
11
12
13
14 @RestController
15 public class OrderController {
16
17     @PostMapping("/print")
18     public String handlePaymentSuccess(@RequestBody Orders order) throws NoSuchAl
19         gorithmException {
20         // 假设你已经有了printer对象和business对象
21         Printer printer = printerService.selectByBusinessId(order.getBusinessId
22         ());
23         Business business = businessService.selectById(order.getBusinessId());
24
25         // 准备公共参数
26         String user = printer.getUser(); // 假设business实体中有用户信息
27         long stime = System.currentTimeMillis() // 1000;
28         String backurl = business.getBackurl(); // 回调URL
29         String expired = String.valueOf(stime + 86400); // 订单24小时后失效
30         String ukey = printer.getUkey();
31         String printerNumber = printer.getPrinterNumber();
32
33         // 生成签名
34         String baseString = user + ukey + stime;
35         String sig = generateSha1Signature(baseString);
36
37         // 准备私有参数
38         String content = getOrderContent(order); // 订单内容，需要自定义生成
39         int times = 1; // 打印次数
40
41         // 构建请求URL
42         String url = "https://api.feieyun.cn/Api/Open/";
43         String postData = String.format(
44             "user=%s&stime=%d&backurl=%s&expired=%s&sig=%s&apiname=Open_printMsg
45             &sn=%s&content=%s&times=%d",
46             user, stime, backurl, expired, sig, printerNumber, content, times
47         );
48
49         // 发送POST请求
50         boolean success = sendPostRequest(url, postData);
51
52         return success ? "打印成功" : "打印失败";
53     }
54 }
```

```

52
53     private String getOrderContent(Order order) {
54         StringBuilder contentBuilder = new StringBuilder();
55
56         // 添加订单编号和用户信息
57         contentBuilder.append("<CB></CB>").append(order.getOrderNo())
58             .append("<BR>").append(order.getUser())
59             .append("<BR>").append(order.getAddress())
60             .append("<BR>订单详情: <BR>");
61
62         // 遍历订单项, 添加商品信息
63         for (OrderItem item : order.getOrderItems()) {
64             contentBuilder.append(item.getGoodsName())
65                 .append(" x ")
66                 .append(item.getNum())
67                 .append(", 单价: ")
68                 .append(item.getPrice())
69                 .append("<BR>");
70         }
71
72         // 添加订单总价和实际支付金额
73         contentBuilder.append("<BR>订单总价: ")
74             .append(order.getAmount())
75             .append("<BR>实付款: ")
76             .append(order.getActual())
77             .append("<CUT>");
78
79         return contentBuilder.toString();
80     }
81
82     @Autowired
83     private RestTemplate restTemplate;
84
85     private boolean sendPostRequest(String url, String postData) {
86         HttpHeaders headers = new HttpHeaders();
87         headers.setContentType(MediaType.APPLICATION_FORM_URLENCODED);
88
89         // 将请求体作为表单数据发送
90         HttpEntity<String> entity = new HttpEntity<>(postData, headers);
91
92         // 执行POST请求
93         try {
94             ResponseEntity<String> response = restTemplate.postForEntity(url, en
95 tity, String.class);
96             // 检查响应状态码是否为200 (OK)
97             return HttpStatus.OK.equals(response.getStatusCode());
98         } catch (Exception e) {
99             e.printStackTrace();
100             return false;
101         }
102
103     private String generateSha1Signature(String baseString) throws NoSuchAlgorit
104 hmException {
105         MessageDigest md = MessageDigest.getInstance("SHA-1");
106         byte[] hash = md.digest(baseString.getBytes());

```

```
106     BigInteger number = new BigInteger(1, hash);
107     StringBuilder hexString = new StringBuilder(number.toString(16));
108
109     while (hexString.length() < 32) {
110         hexString.insert(0, '0');
111     }
112
113     return hexString.toString();
114 }
115
116 // 以下为辅助方法，用于获取Printer和Business对象
117 private Printer getPrinterByBusinessId(Integer businessId) {
118     // 实现获取Printer对象的逻辑
119     return new Printer();
120 }
121
122 private Business getBusinessById(Integer businessId) {
123     // 实现获取Business对象的逻辑
124     return new Business();
125 }
```