

如何使用

1、阅读之前请先仔细阅读 GitHub 上的 README.md

项目概述

本质是基于 Spring Boot 开发的 Web 项目。

采用多模块 Maven 构建。

版本号

2019-12-07: v1.0-Beta

建议先跟着下面的步骤走一遍，等熟悉了工程是什么样的，再发挥各自的创造力，我相信你们的代码水平比我高!!!

使用---微信“Native 模式二”支付

步骤一

在 MySQL 5.7 以上的数据库中运行数据库脚本 (<https://github.com/yysCyber/union-pay-platform/blob/master/SQL/example.sql>) 上的语句，因为涉及到使用 `CURRENT_TIMESTAMP` 所以对 MySQL 的版本有要求，详细情况上网查阅资料。

使用 IDEA 导入工程。

将 web 模块的下的 Spring Boot 的配置文件 `application.yml` 有关数据库的配置进行修改，修改成你的配置。

另外，在 `application.yml` 中配置了 `server.port`、`server.servlet.context-path`，要注意。当然，如果后面使用内网穿透进行调试的话，也要注意。

步骤二

打开 Service 模块，对 `com.yscopyber.myspringboot.projectd.service.pay.wechat` 下的 `weChatPayConstant.java` 进行修改，配置成你自己的常量。

常量名	是否必须	说明
EXAMPLE_APP_ID	是	应用的 ID，需要到微信官方申请获得，详细参考微信官方的开发文档
EXAMPLE_MERCHANT_ID	是	商户的 ID，需要到微信官方申请获得，详细参考微信官方的开发文档
EXAMPLE_MERCHANT_KEY	是	商户的密钥，需要到微信官方申请获得，详细参考微信官方的开发文档
EXAMPLE_NOTIFY_URL	是	“支付结果通知”要传递给的 URL 这一机制详见微信官方文档 要求是来自外部的网络可以正确的访问到 建议在开发的时候使用内网穿透工具，将 127.0.0.1:8080 映射到外网 因为在 Controller 中提供了处理“支付结果通知”的映射，为 <code>/union-pay/wechat/native-second/async-notify</code> ，所以最终该常量值的示例为： <code>http://xxx.xxx.cc/union-pay/wechat/native-second/async-notify</code> ， <code>http://xxx.xxx.cc</code> 是 127.0.0.1:8080 映射到外网的结果
EXAMPLE_ORDER_NAME_PREFIX	否	订单名称前缀，这个是要不要都行的，如果你觉得不需要，在熟悉整个工程之前， 建议为空串
EXAMPLE_ORDER_PAYEE	是	收款方名称，最好有，这个主要是为了渲染“支付”页面使用的。同样，如果你觉得不需要，在熟悉整个工程之前， 建议为空串

步骤三

有关微信支付的 Controller 在 web 模块的 `com.yscyber.myspringboot.projecttd.controller.pay.weChatPayController` 中。请熟悉：

注：下面的 URL 忽略 `application.yml` 中的 context 配置以及内网穿透等

URL	请求方式	说明
/wechat/native-second	GET/POST	传入参数，跳转到“支付”页面，同时将“生成二维码图片的链接”等相关信息传递给“支付”页面
/wechat/native-second/async-notify	POST	之前希望你配置的 EXAMPLE_NOTIFY_URL 就来源于此，用来接收处理“支付结果通知”的映射
/wechat/native-second/success-tip	GET	跳转到“支付成功”提示页面
/wechat/native-second/is-success	GET	因为微信支付官方并没有涉及到 <code>return_url</code> （“支付成功”后跳转到哪一个页面）这一说，所以，需要在“支付”页面以一定的时间间隔向后台发起请求（轮询），后台会根据数据库的查询得到的结果反馈给前端，如果前端得到的是“订单已经支付成功”的消息，利用 JavaScript 跳转至“支付成功”提示页面。

步骤四

开启内网穿透。

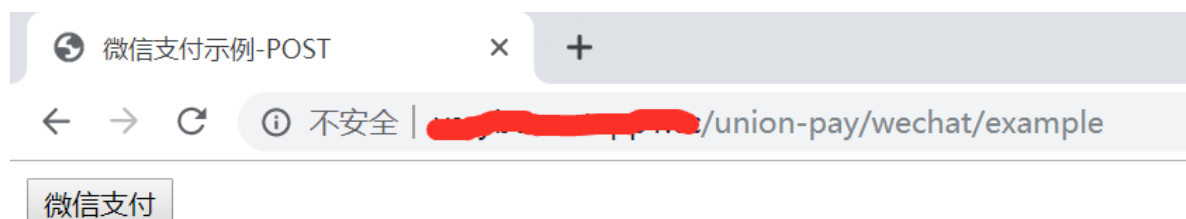
在该工程中，也给出了用来测试的，在 web 模块的

`com/yscyber/myspringboot/projectd/controller/example/ExampleController.java` 以及 `web\src\main\resources\templates\example\wechat.html`。

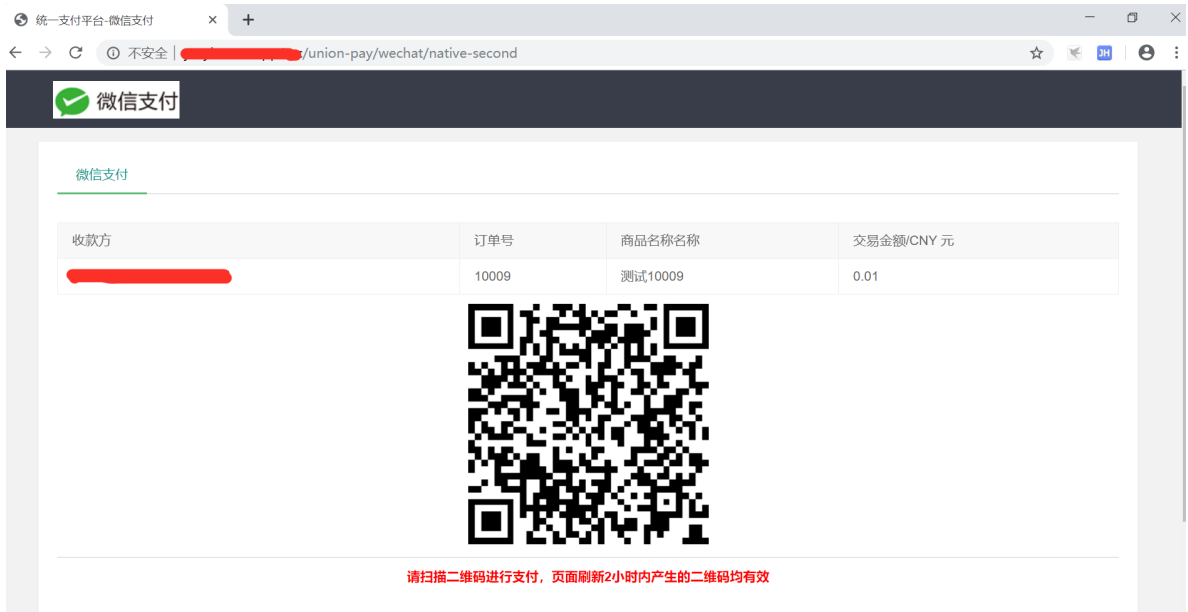
根据你的需要修改 `wechat.html` 中 `<form>` 下的前3个 `<input>` 的 `value`，分别对应“订单号”、“订单名称”、“支付金额”。

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>微信支付示例-POST</title>
    <script type="text/javascript" th:src="@{/js/vendor/jquery.min.js}">
</script>
</head>
<body>
    <form method="post" th:action="@{/wechat/native-second}">
        <input type="hidden" name="id" value="10009"/>
        <input type="hidden" name="name" value="测试10009"/>
        <input type="hidden" name="amount" value="0.01"/>
        <input type="button" id="pay" value="微信支付"/>
    </form>
    <script type="text/javascript">
        $("#pay").on('click', function () {
            $("form").submit();
        });
    </script>
</body>
</html>
```

启动项目，根据 `ExampleController` 所给出的映射，访问至 `wechat.html` 这一页面，如下：



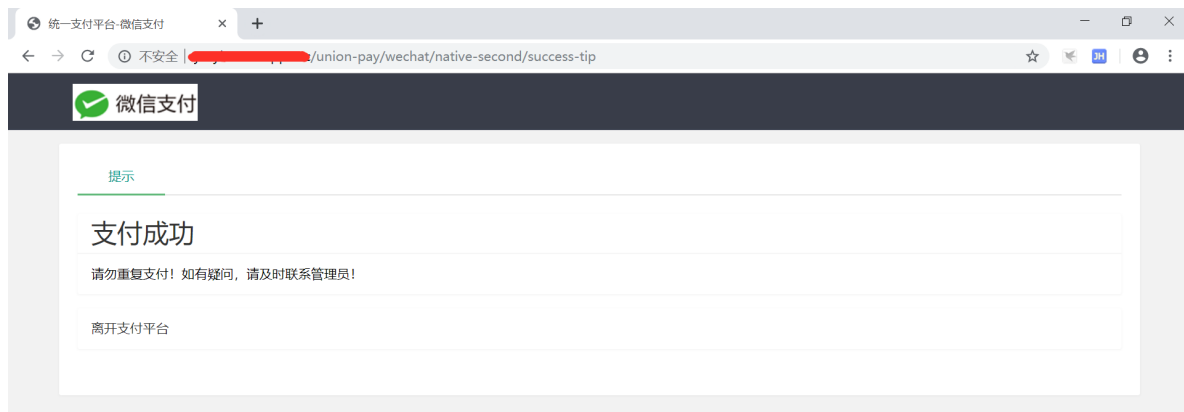
点击“微信支付”，跳转至“支付”页面，如下：



使用手机微信扫码，进行支付，如下：



支付完成后，跳转至“支付成功”提示页面，如下：



注意：渲染该提示页面的一些常量在 `com.yscyber.myspringboot.projectd.service.pay.wechat.WeChatPayTipConstant` 中，同样的，在熟悉项目后，可以根据需要进行调整。

这个过程中，也请密切注意数据库中数据的变化。支付成功后，可以到微信中的支付凭证的详情页和数据库中的数据进行对比。



步骤五

可以考虑将其部署在服务器上进行调试，当然，由于当前版本不一定完善，可能会出现一定的问题。

局限性

因为实现的核心是 `best-pay-sdk`，所以说，最终能够做成什么样取决于对 SDK 的理解以及 SDK 的开发人员。在 `best-pay-sdk` 中对于支付过程中产生的一些错误，以抛出异常的处理方式偏多，加上微信“支付结果通知”机制并不是想象的那么简单，可能出现错误后“给用户友好提示”的方面实现起来较为困难，所以在后面的版本中尝试解决。

敬请期待

.....

声明

- 1、如有疑问，请发送邮件至y2161624113@163.com，邮件主题为：union-pay-platform
- 2、可以根据实际，去编写属于自己的“支付平台”，一旦“支付平台”能够运行到线上环境，将商户系统的订单信息以 POST 方式请求到“支付接口”即可。