


零基础教程：如何为你的个人网站开通在线支付功能（无需营业执照）

 本期视频我们会使用到zpay作为支付平台，如果你想给你的网站接入个人国内支付，可以申请

网站地址：

<https://payment-template.botbio.site>

（部署在海外可能访问会比较慢）



个人中心

加入学习社区

小白0基础入门

AI编程

借助AI编程工具，即使没有编程基础，也能快速开发出你的第一个AI应用和网站，本课程专为零基础学员设计。

加入学习社区

10+
基础内容

20+
实战项目

461+
学员社区



课程内容精选

全部内容 27+

基础知识 9+

编程进阶 8+

项目实战 10+

AI工具箱 5+

提醒

为了避免出错，教程里提供的是现成的已经设计好的模板，可以拿来直接用直接配置，跑通之后，再去问AI怎么接入你设计的网站，怎么改相关界面外观设计就比较简单了。

1. 起步



1.1 初始化项目

- 首先下载起始项目源代码：



- 修改环境变量

打开`.env.exmaple`文件，你需要将`.env-exmplate`修改成为`.env.local`，并填入对应的环境变量
去`supabase`新建一个项目获取对应的环境变量
去`zpay`平台获取到PID和KEY（可以使用测试的）

代码块

```
1  # Supabase 配置
2  NEXT_PUBLIC_SUPABASE_URL=your-supabase-url
3  NEXT_PUBLIC_SUPABASE_ANON_KEY=your-supabase-anon-key
4  SUPABASE_SERVICE_ROLE_KEY=your-supabase-service-role-key # 服务角色密钥，具有绕过 RLS的权限
5
6  # 应用基础配置
7  NEXT_PUBLIC_APP_URL=http://localhost:3000
8
9  # Z-Pay 支付网关配置
10 ZPAY_PID=your-zpay-pid # Z-Pay商户ID
11 ZPAY_KEY=your-zpay-key # Z-Pay密钥
```

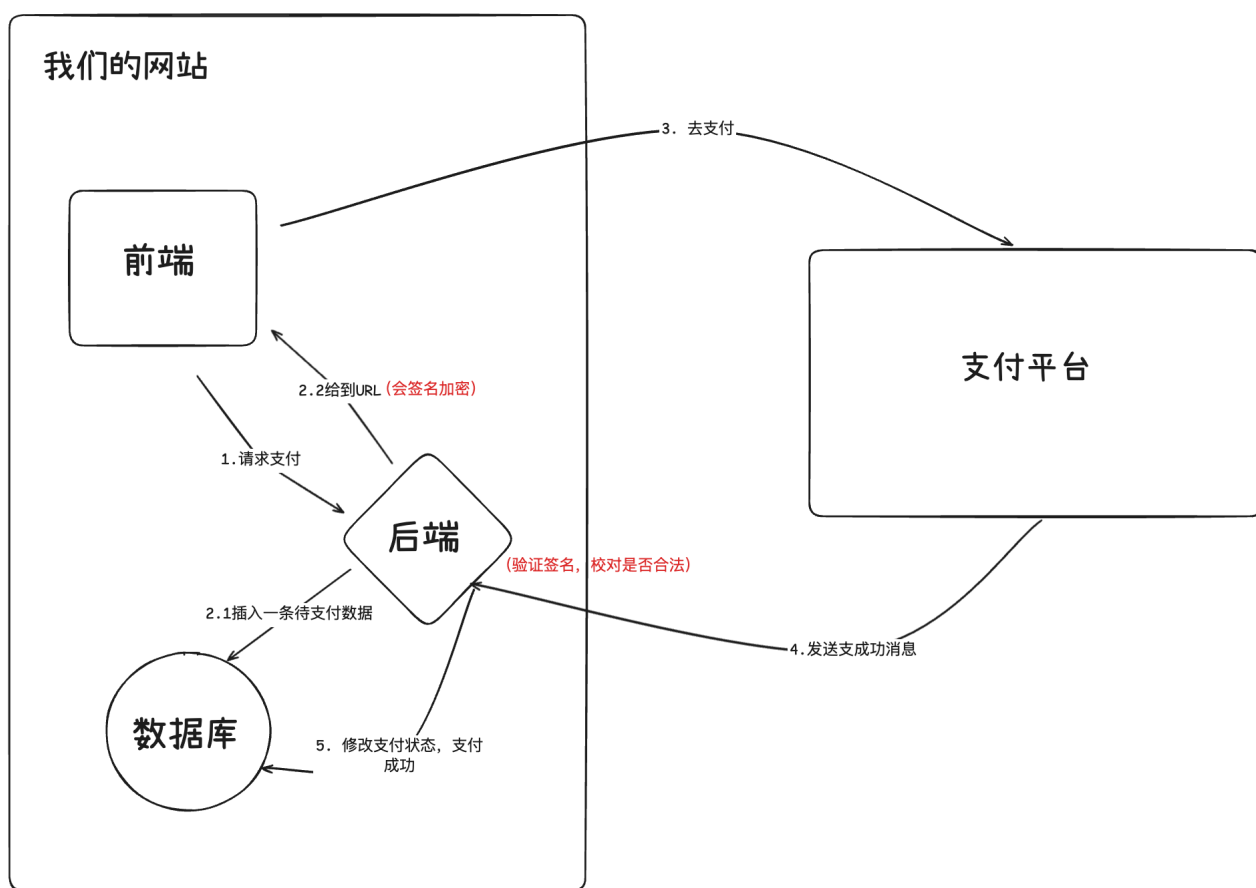
💡 这里zpay作为支付平台，如果你暂时不想申请，可以使用平台的测试支付逻辑。然后将测试金额调整为0.1，减少你的测试成本。本人和平台无利益关系，只是我个人使用这个比较多，体验好很安全比较推荐。

你也可以使用易支付等同类产品。

执行`npm install`，安装对应的依赖

执行`npm run dev`，运行起来整个项目

1.2 支付流程说明



核心需要开发2个接口：

接口1: `/api/checkout/providers/zpay/url`，用于前端获取支付链接

接口2: `/api/checkout/providers/zpay/webhook`，用户支付平台通知我们的后端服务

2. 接入支付逻辑

2.1 开发支付

2.1.1 写提示词

你可以点击进入[zpay官方文档](#)，查看文档。我们核心关心的是两个接口：

1.页面跳转接口

页面跳转支付

请求URL

https://zpayz.cn/submit.php

请求方法

POST 或 GET（推荐POST，不容易被劫持或屏蔽）

此接口可用于用户前台直接发起支付，使用form表单跳转或拼接成url跳转。

请求参数

参数	名称	类型	是否必填	描述	范例
name	商品名称	String	是	商品名称不超过100字	iphonexs max
money	订单金额	String	是	最多保留两位小数	5.67
type	支付方式	String	是	支付宝：alipay 微信支付：wxpay	alipay
out_trade_no	订单编号	Num	是	每个商品不可重复	201911914837526544601
notify_url	异步通知页面	String	是	交易信息回调页面，不支持带参数	http://www.aaa.com/bbb.php
pid	商户唯一标识	String	是	一串字母数字组合	201901151314084206659771

2.支付结果通知

支付结果通知

请求URL

服务器异步通知（notify_url）、页面跳转通知（return_url）

请求方法

GET

请求参数

参数	名称	类型	描述	范例
pid	商户ID	String		201901151314084206659771
name	商品名称	String	商品名称不超过100字	iphone
money	订单金额	String	最多保留两位小数	5.67
out_trade_no	商户订单号	String	商户系统内部的订单号	201901191324552185692680
trade_no	易支付订单号	String	易支付订单号	2019011922001418111011411195
param	业务扩展参数	String	会通过notify_url原样返回	金色 256G
trade_status	支付状态	String	只有TRADE_SUCCESS是成功	TRADE_SUCCESS

给到对应的提示词：

代码块

```
1 Ma现在请你帮我开发两个支付接口：
2 - 接口1：`/api/checkout/providers/zpay/url`，用于前端获取支付链接
3 - 接口2：`/api/checkout/providers/zpay/webhook`，用于支付平台通知我们的后端服务
4
5 我使用的是zpay支付实现支付
6
7 1. 关于前端获取支付链接的接口`/api/checkout/providers/zpay/url`，需要拼接成为的支付链
  接如下文档：
8 ```
9 页面跳转支付
10 请求URL
11 https://zpayz.cn/submit.php
12 请求方法
13 POST 或 GET（推荐POST，不容易被劫持或屏蔽）
14 此接口可用于用户前台直接发起支付，使用form表单跳转或拼接成url跳转。
15 请求参数
16 参数      名称      类型      是否必填      描述      范例
```

17	name	商品名称	String	是	商品名称不超过100字	
	iphonexs max					
18	money	订单金额	String	是	最多保留两位小数	5.67
19	type	支付方式	String	是	支付宝：alipay 微信支付：wxpay	
	alipay					
20	out_trade_no	订单编号	Num	是	每个商品不可重复	
	201911914837526544601					
21	notify_url	异步通知页面	String	是	交易信息回调页面，不支持带参数	
	http://www.aaa.com/bbb.php					
22	pid	商户唯一标识	String	是	一串字母数字组合	
	201901151314084206659771					
23	cid	支付渠道ID	String	否	如果不填则随机使用某一支付渠道	
	1234					
24	param	附加内容	String	否	会通过notify_url原样返回	
	金色 256G					
25	return_url	跳转页面	String	是	交易完成后浏览器跳转，不支持带参数	
	http://www.aaa.com/ccc.php					
26	sign	签名（参考本页签名算法）	String	是	用于验证信息正确性，采用md5加密	
	28f9583617d9caf66834292b6ab1cc89					
27	sign_type	签名方法	String	是	默认为MD5	MD5
28	用法举例					
29	https://zpayz.cn/submit.php?name=iphone xs Max 一台 &money=0.03&out_trade_no=201911914837526544601¬ify_url=http://www.aaa.com/notify_url.php&pid=201901151314084206659771¶m=金色 256G&return_url=http://www.baidu.com&sign=28f9583617d9caf66834292b6ab1cc89&sign_type=MD5&type=alipay					
30						
31	成功返回					
32	直接跳转到付款页面					
33	说明：该页面为收银台，直接访问这个url即可进行付款					
34	失败返回					
35	{ "code": "error", "msg": "具体的错误信息" }					
36	...					
37						
38	你可以参考的签名算法：					
39	...					
40	const utility=require("utility"); //导入md5第三方库					
41						
42	let data={					
43	pid:"你的pid",					
44	money:"金额",					
45	name:"商品名称",					
46	notify_url:"http://xxxxx", //异步通知地址					
47	out_trade_no:"2019050823435494926", //订单号,自己生成。我是当前时间					
	YYYYMMDDHHmmss再加上随机三位数					
48	return_url:"http://xxxx", //跳转通知地址					
49	sitename:"网站名称",					

```

50         type:"alipay",//支付方式:alipay:支付宝,wxpay:微信支付,qqpay:QQ钱
        包,tenpay:财付通,
51     }
52
53     //参数进行排序拼接字符串(非常重要)
54     function getVerifyParams(params) {
55         var sPara = [];
56         if(!params) return null;
57         for(var key in params) {
58             if((!params[key]) || key == "sign" || key == "sign_type") {
59                 continue;
60             };
61             sPara.push([key, params[key]]);
62         }
63         sPara = sPara.sort();
64         var prestr = '';
65         for(var i2 = 0; i2 < sPara.length; i2++) {
66             var obj = sPara[i2];
67             if(i2 == sPara.length - 1) {
68                 prestr = prestr + obj[0] + '=' + obj[1] + '';
69             } else {
70                 prestr = prestr + obj[0] + '=' + obj[1] + '&';
71             }
72         }
73         return prestr;
74     }
75
76
77
78     //对参数进行排序,生成待签名字符串--(具体看支付宝)
79     let str=getVerifyParams(data);
80
81     let key="你的key";//密钥,易支付注册会提供pid和密钥
82
83     //MD5加密--进行签名
84     let sign=utility.md5(str+key);//注意支付宝规定签名时:待签名字符串后要加key
85
86     最后要将参数返回给前端,前端访问url发起支付
87     let result = `https://z-pay.cn/submit.php?${str}&sign=${sign}&sign_type=MD5`;
88
89     ``
90
91     2. 关于/api/checkout/providers/zpay/webhook`,用于支付平台通知我们的后端服务的
    webhook,接口文档如下:
92     ``
93     支付结果通知
94     请求URL

```

95 服务器异步通知 (notify_url) 、页面跳转通知 (return_url)

96 请求方法

97 GET

98 请求参数

99 参数	名称	类型	描述	范例
100 pid	商户ID	String		201901151314084206659771
101 name	商品名称	String	商品名称不超过100字	iphone
102 money	订单金额	String	最多保留两位小数	5.67
103 out_trade_no	商户订单号	String	商户系统内部的订单号	201901191324552185692680
104 trade_no	易支付订单号	String	易支付订单号	2019011922001418111011411195
105 param	业务扩展参数	String	会通过notify_url原样返回	金色 256G
106 trade_status	支付状态	String	只有TRADE_SUCCESS是成功	TRADE_SUCCESS
107 type	支付方式	String	包括支付宝、微信	alipay
108 sign	签名 (参考本页签名算法)	String	用于验证接受信息的正确性	ef6e3c5c6ff45018e8c82fd66fb056dc
109 sign_type	签名类型	String	默认为MD5	MD5

110 如何验证

111 请根据签名算法，验证自己生成的签名与参数中传入的签名是否一致，如果一致则说明是由官方向您发送的真实信息

112 注意事项

113 1.收到回调信息后请返回“success”，否则程序将判定您的回调地址未正确通知到。

114

115 2.同样的通知可能会多次发送给商户系统。商户系统必须能够正确处理重复的通知。

116

117 3.推荐的做法是，当收到通知进行处理时，首先检查对应业务数据的状态，判断该通知是否已经处理过，如果没有处理过再进行处理，如果处理过直接返回结果成功。在对业务数据进行状态检查和处理之前，要采用数据锁进行并发控制，以避免函数重入造成的数据混乱。

118

119 4.特别提醒：商户系统对于支付结果通知的内容一定要做签名验证,并校验返回的订单金额是否与商户侧的订单金额一致，防止数据泄漏导致出现“假通知”，造成资金损失。

120

121 5.对后台通知交互时，如果平台收到商户的应答不是纯字符串success或超过5秒后返回时，平台认为通知失败，平台会通过一定的策略（通知频率为0/15/15/30/180/1800/1800/1800/1800/3600，单位：秒）间接性重新发起通知，尽可能提高通知的成功率，但不保证通知最终能成功。

122

123 MD5签名算法

124 1、将发送或接收到的所有参数按照参数名ASCII码从小到大排序（a-z），sign、sign_type、和空值不参与签名！

125

126 2、将排序后的参数拼接成URL键值对的格式，例如 a=b&c=d&e=f，参数值不要进行url编码。

127

128 3、再将拼接好的字符串与商户密钥KEY进行MD5加密得出sign签名参数，sign = md5（a=b&c=d&e=f + KEY）（注意：+ 为各语言的拼接符，不是字符！），md5结果为小写。

129

130 4、具体签名与发起支付的示例代码可下载SDK查看。

131 ```

132

133 请你帮我实现这两个接口，要求：

134 1. 服务端操作数据库的相关操作，请使用createServerAdminClient实现

135 2. 在webhook中需要检查对应业务的状态，避免重复调用数据库或者插入数据

136 3. 支付结果通知的内容一定要做签名验证，并校验返回的订单金额是否与商户侧的订单金额一致，防止数据泄漏导致出现“假通知”，造成资金损失。

137 4. 请你给到我supabase需要创建的表sql，并且保存在项目中，表名叫做`zpay_transactions`

138 5. 我们的环境变量中字段包括NEXT_PUBLIC_SUPABASE_URL、

NEXT_PUBLIC_SUPABASE_ANON_KEY、SUPABASE_SERVICE_ROLE_KEY、NEXT_PUBLIC_BASE_URL、

ZPAY_PID、ZPAY_KEY

139

140

141 同时请你帮我实现 @pricing.tsx 接入请求获取购买链接的接口，包括一次性购买逻辑 + 订阅模式，要求：

142 1. 订阅模式需要设置过期时间等信息，而且订阅需要考虑重复订阅的情况，例如用户在2025-03-15订阅了一个月，那么过期时间是2025-04-15，但是用户在2025-04-01又订阅了一个一个月，此时用户的开始订阅时间需要从2025-04-15开始计算，过期时间是2025-05-15，这一定要注意

143 2. 用户需要登录后才能请求获取购买链接，因为要获取用的uid，如果用户没有登陆点击支付，跳转登录页/signinrkdown

2.1.2 初始化数据库

在supabase的SQL Editor中运行Cursor给到的数据库，点击Run，创建表。



2.2 调试 + 修改bug



注意：你的bug可能和我不一样，因为我们在复杂提示词下，无法保证Cursor生成的结果100%一致。只能说遇到问题，解决问题。

2.2.1 解决bug：没有使用已有的依赖

目前发现supabase的使用没有使用已有的依赖实现，现在我们去解决它

提示词

```
1 @@route.ts @route.ts 错了，请你使用相关已有的依赖实现
```

2.2.2 解决bug: 获取价目表错误

目前能够正确的弹出支付了，但是价目表和我们的不一样，我们需要让他参考已有的价目表实现修改

提示词

```
1 @route.ts @pricing.tsx 价格请你参考这两个文件，并且以id的形式发送到后端，后端去获取价目表信息。
```

2.2.3 本地回调方法 + 解决回调签名错误



因为zpay无法请求本地，所以你需要手动调用，把回调地址的url替换成为：

<http://localhost:3000/api/checkout/providers/zpay/webhook>。记得保留参数，也就是？后面的内容都需要保留！

此时遇到校验参数签名报错，修复它，提示词如下：

提示词

```
1 请你检查为什么有Invaild singature，这是一个正确的签名
```

2.4 开发个人页面

提示词

1. 请你在 `@layout.tsx` 中获取本人的订阅信息，并在邮箱下面显示
2. 请你在中封装一个购买历史，并在 `/dashboard` 这个页面显示，要求显示产品名称、购买日期、价格、状态和操作按钮。如果是待支付状态，那么点击后用户去跳转支付；如果支付成功，点击后alert相关的购买订单的详细信息

2.5. 解决订阅逻辑错误

提示词

- 1 `@layout.tsx` 订阅逻辑查询有问题，请你结合 `@zpay_transactions.sql` 数据库的字段，修改。目前有一条最新的订阅记录，但是你没有显示成功

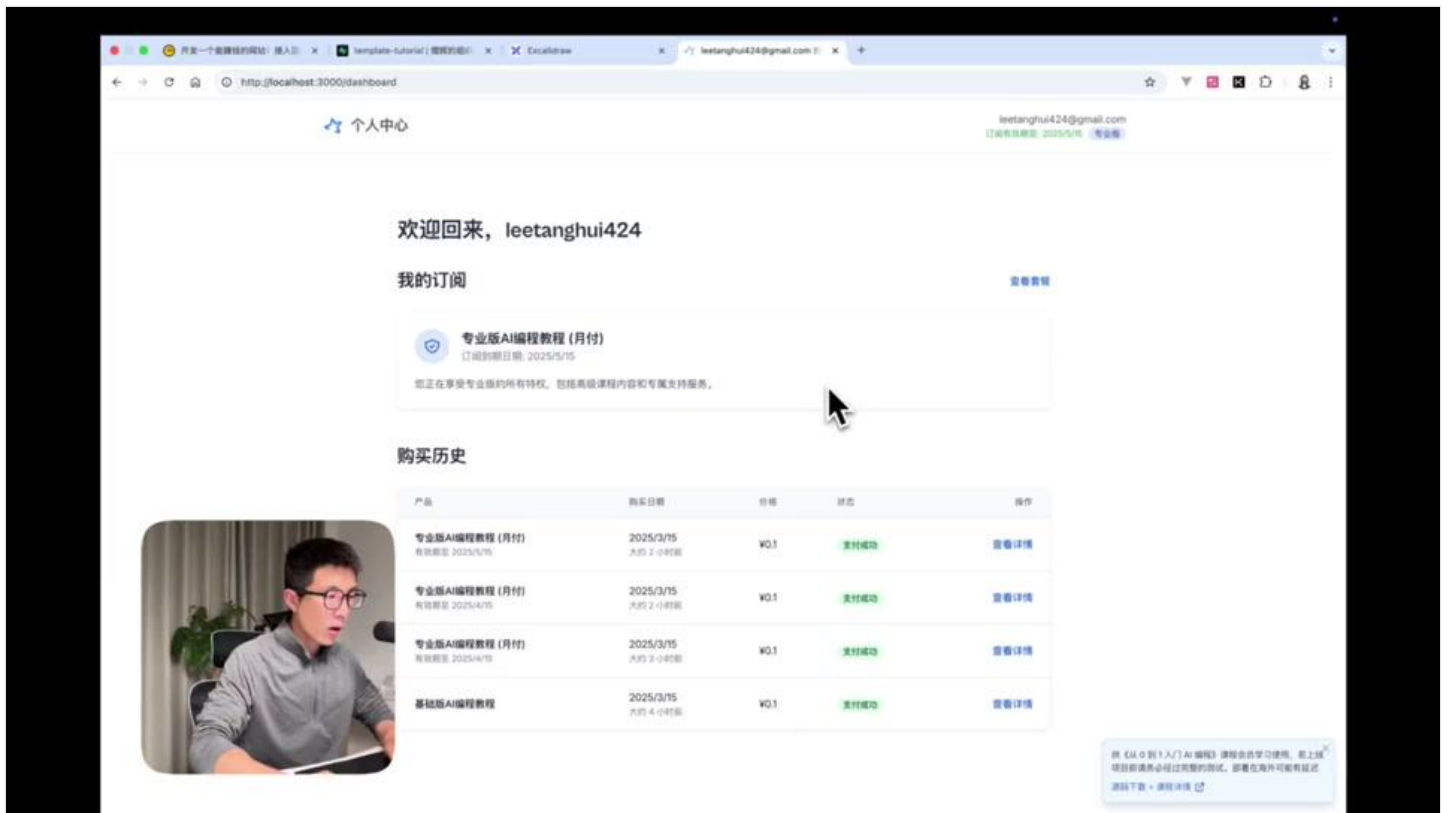
2.6 解决bug: 订阅逻辑错误(难)

目前遇到订阅逻辑错误，如果多次订阅，没有将最远的一次订阅作为开始时间，让我们来修复这个bug：

提示词

- 1 修复bug：
- 2 1. 你不应该在创建订单的时候添加上 `subscription_start_date`、`subscription_end_date`，而是应该在用户支付成功之后再添加，因为你这样时间是不正确的
- 3 2. 在 `@route.ts` 中计算 `subscription_start_date`、`subscription_end_date`，请务必结合已有的订阅记录。如果用户是在订阅期内，那么需要找到最新的一个订阅结束期，作为订阅的开始时间
- 4 3. `@zpay_transactions.sql` 表的数据，请你好好的参考该sql

3. Vercel部署上线



步骤讲解：

1.本地编译一次，查看bug，有遇到问题我们解决问题

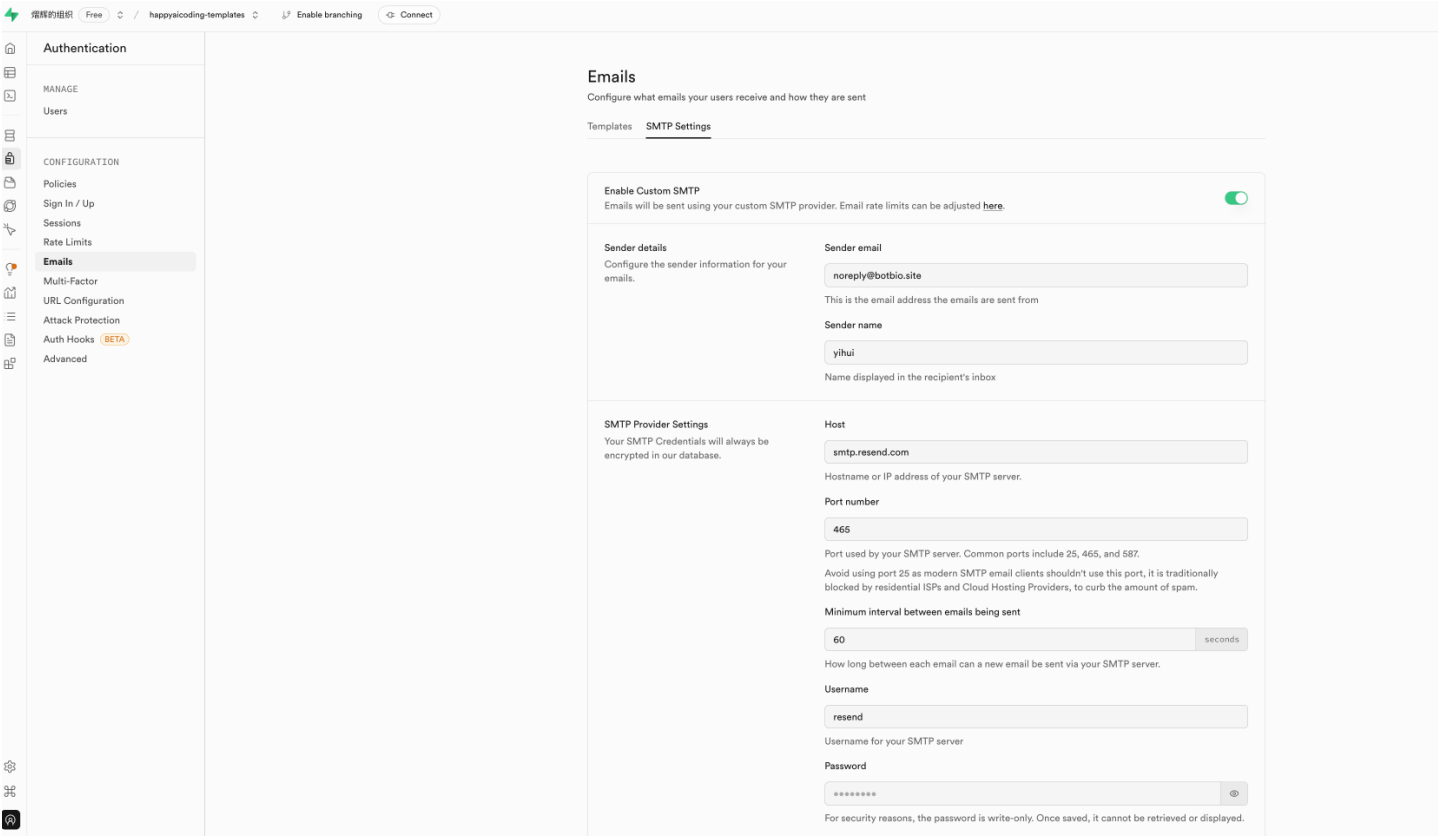
2.使用Vercel部署上线

- 新建项目，并连接Github
- 部署上线
- 修改自定义域名（如果需要的话）

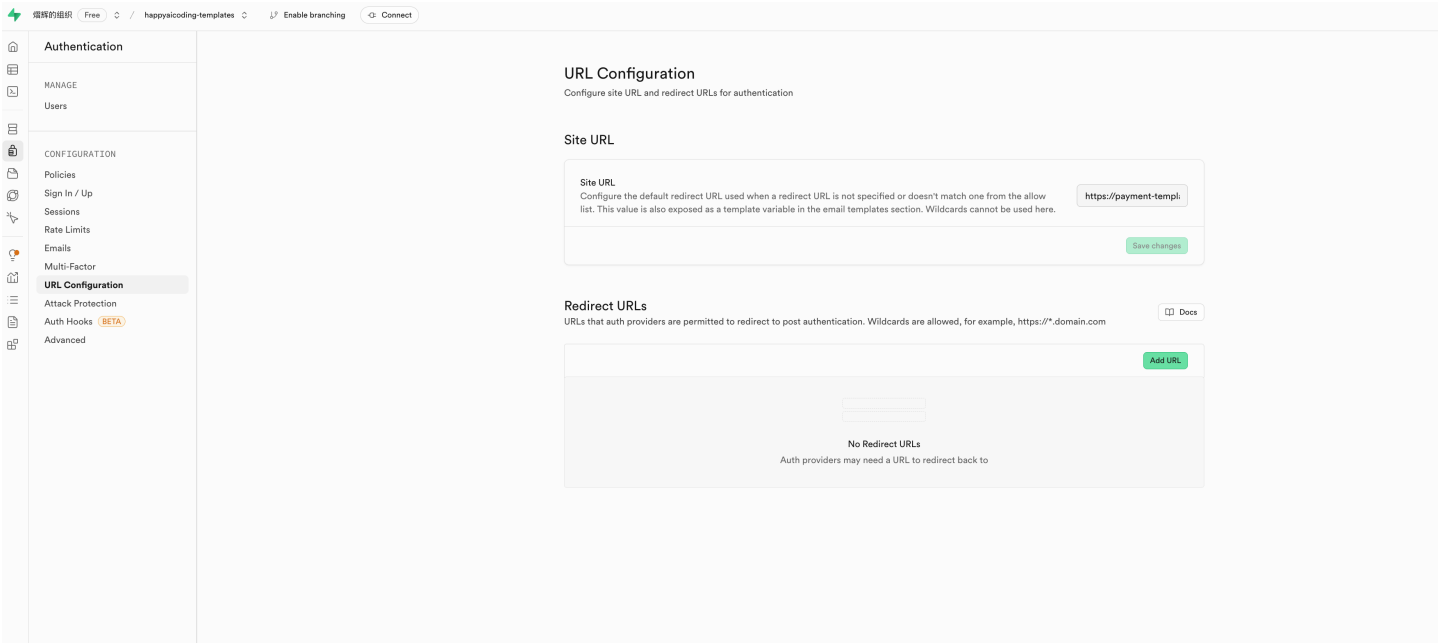
3.最终上线后supabase调整（这部分视频里面没有讲到，真实上线前你需要自行调整）

4.4.轻松接入邮件登陆注册功能(Auth)

- supabase需要替换SMTP，避免邮件发送速率限制



- 调整supabase的URL Configuration为你的域名（重要）



目前只是接入了支付逻辑，支付逻辑包含一次性购买、订阅模式。我们可以组合不同的支付逻辑来完成我们的业务，例如：

- 1.只使用一次性模式，来售卖一份付费的百度网盘链接？
- 2.只使用订阅模式，只有订阅期的用户能够使用我们的AI对话功能
- 3.同时使用订阅 + 一次性模式，用户订阅期内可以使用我们的AI对话功能1000次，使用完成后，可以通过一次性模式购买叠加包！

来尝试挑战一下其中的1个任务把，使用Cursor对话的形式来完成。如果完不成也不用担心，接下来我们会基于这套模板来开发一些有趣的实战项目。让大家知道能盈利、具有生产级别的网站应用是如何开发的!

4. 源码

4.1 1初始脚手架



happyaicoding-template-starter.zip

9.97MB



4.2 最终项目



happyaicoding-template-completed.zip

9.99MB



这期视频主要是为了让大家了解支付流程，方便以后扩展使用。目前该最终项目会有一些小bug，不建议用做真实项目使用。

4.3 网站版本



happyaicdong-template-main.zip

9.95MB



<https://payment-template.botbio.site/>

这是对应的示例网站版本代码，和课程略有不同