

CLI 对外开放指南

Shoplazza App 命令使用指南（新增文章目录）

概述

`shoplazza app` 是 Shoplazza CLI 中用于开发和管理 Shoplazza 应用的核心命令组。它提供了完整的应用开发生命周期管理，从项目初始化到部署发布的全流程支持。

前置条件

在开始使用 `shoplazza app` 命令之前，请确保：

- 已安装 Node.js (推荐版本: 20.18.0)
- 已安装 Git
- 已登录 Shoplazza Partner (`shoplazza login partner`) (外链到shoplazza login)
- 有稳定的网络连接
- 已安装最新版本 Shoplazza CLI

Code block

```
1 npm i -g shoplazza-cli
```

命令概览（每个命令进行外链）

命令	功能描述	使用场景
<code>app init</code>	初始化新的应用项目	创建全新的应用项目
<code>app generate extension</code>	生成扩展模板	为现有应用添加新扩展
<code>app dev</code>	启动本地开发服务器	本地开发和调试

app deploy	部署应用到生产环境	发布应用到生产环境
app config link	链接应用配置文件	连接现有应用到本地项目
app config use	切换应用配置文件	在多个应用配置间切换
app versions list	查看应用版本列表	管理应用版本历史

详细命令说明

1. shoplazza app init - 初始化应用项目

功能: 创建一个全新的 Shoplazza app，包括项目模板和配置文件，或者链接一个已存在的app 项目

使用场景:

- 开始开发新的 Shoplazza 应用，从零开始创建应用项目
- 链接已存在的app项目

命令格式:

Code block

```
1 shoplazza app init [选项]
```

选项:

- `--path <path>` : 指定项目创建路径 (默认: 当前目录)
- `--client-id <client-id>` : 指定要连接的现有应用客户端ID
- `--name <name>` : 指定应用名称

使用示例:

Code block

```
1 在当前目录创建新应用
2 shoplazza app init
3 在指定目录创建新应用
```

```
4 shoplazza app init --path ./my-new-app
5 创建指定名称的应用
6 shoplazza app init --name "your_app_name"
7 链接到现有应用
8 shoplazza app init --client-id your-client-id
```

2. `shoplazza app generate extension` - 生成扩展模板

功能: 为现有应用生成新的扩展模板，支持多种扩展类型。

使用场景:

- 为现有应用添加新功能
- 创建主题扩展、结账扩展或功能扩展

命令格式:

Code block

```
1
2 shoplazza app generate extension
```

选项:

- `--path <path>`: 指定项目路径 (默认: 当前目录)

支持的扩展类型:

- **Theme Extension:** 主题扩展

Theme Extension 提供一种扩展和定制主题页面的方式

- **Checkout Extension:** 结账扩展

Checkout UI Extension 允许开发人员对结账页进行定制，包括隐藏部分模块，修改商品信息、地址信息、物流信息、支付信息和订单结果信息，以及插入自定义的内容。

- **Function Extension:** 功能扩展

cart-transform function 允许开发者自定义 购物车价格计算 逻辑。当 cart-transform 绑定到 购物车改价点位 后，Shoplazza 会在 购物车或checkout阶段 自动执行绑定的 function 代码，通过更新操作，实现动态控制价格调整策略。

使用示例:

Code block

```
1 在当前项目中生成新扩展
```

```
2 shoplazza app generate extension
3
4 在指定项目中生成新扩展
5 shoplazza app generate extension --path ./my-app
```

3. `shoplazza app dev` - 本地开发

功能: 启动本地开发服务器

使用场景:

- 本地开发和调试应用
- 实时预览应用效果
- 测试扩展功能

命令格式:

Code block

```
1 shoplazza app dev
```

选项:

- `--path <path>`: 指定项目路径 (默认: 当前目录)

使用示例:

Code block

```
1 启动开发服务器
2 shoplazza app dev
3
4 在指定项目中启动开发服务器
5 shoplazza app dev --path ./my-app
```

4. `shoplazza app deploy` - 部署应用

功能: 将应用部署到生产环境，使应用在商店中生效。

使用场景:

- 将开发完成的应用发布到生产环境
- 更新现有应用的版本

命令格式:

Code block

```
1 shoplazza app deploy [选项]
```

选项:

- `--path <path>`: 指定项目路径 (默认: 当前目录)

使用示例:

Code block

```
1 部署应用
2 shoplazza app deploy
3
4 部署指定项目
5 shoplazza app deploy --path ./my-app
```

5. `shoplazza app config link` - 链接应用配置

功能: 将现有的 Shoplazza 应用链接到本地项目，创建配置文件。

使用场景:

- 将现有应用连接到本地开发环境
- 为现有应用创建本地配置文件
- 在本地项目中管理多个应用配置

命令格式:

Code block

```
1 shoplazza app config link [options]
```

选项:

- `--path <path>` : 指定项目路径 (默认: 当前目录)
- `--client-id <client-id>` : 指定要链接的应用客户端ID

使用示例:

Code block

```
1  链接应用配置
2  shoplazza app config link
3
4  链接指定应用
5  shoplazza app config link --client-id your-client-id
6
7  在指定目录中链接
8  shoplazza app config link --path ./my-project
```

6. `shoplazza app config use` - 切换应用配置

功能: 在项目的多个应用配置文件之间切换，激活指定的配置。

使用场景:

- 项目中有多个应用配置
- 需要在不同应用间切换开发
- 管理多环境配置

命令格式:

Code block

```
1  shoplazza app config use [选项]
```

选项:

- `--path <path>` : 指定项目路径 (默认: 当前目录)

使用示例:

Code block

- 1 切换应用配置
- 2 shoplazza app config use
- 3
- 4 在指定项目中切换配置
- 5 shoplazza app config use --path ./my-project

7. `shoplazza app versions list` - 查看应用版本

功能: 查看应用的所有已部署版本，支持分页浏览和详细信息展示。

使用场景:

- 查看应用版本历史
- 管理应用的不同版本
- 了解版本部署状态

命令格式:

Code block

- 1 shoplazza app versions list [选项]

选项:

- `--path <path>`: 指定项目路径 (默认: 当前目录)
- `--page-size <number>`: 每页显示的版本数量 (默认: 10)

使用示例:

Code block

- 1 查看版本列表
- 2 shoplazza app versions list
- 3
- 4 指定每页显示数量
- 5 shoplazza app versions list --page-size 50
- 6
- 7 查看指定项目的版本
- 8 shoplazza app versions list --path ./my-app

配置文件说明

shoplazza.app.toml app 配置文件

这是应用的核心配置文件，包含应用的基本信息：

Code block

```
1
2  # 应用客户端ID (必需)
3  client_id = "your-client-id"
4
5  # 应用权限范围
6  scopes = "read_customer write_cart_transform"
```

shoplazza.extension.toml 扩展配置文件

每个扩展都有自己的配置文件：

Code block

```
1  # 扩展ID (由系统生成)
2  id = "extension-id"
3
4  # 扩展名称
5  name = "My Extension"
6
7  # 扩展类型 (theme/checkout/function)
8  type = "theme"
```

常见问题解答

Q: 如何创建多个应用配置？

A: 使用 `shoplazza app config link` 命令，系统会自动生成不同名称的配置文件，如 `shoplazza.app.production.toml`、`shoplazza.app.staging.toml` 等。

Q: 如何在不同应用间切换?

A: 使用 `shoplazza app config use` 命令，选择要激活的配置文件即可。

Q: 如何查看应用的详细状态?

A: 使用 `shoplazza app versions list` 命令查看版本历史，或使用 `shoplazza app info` 命令查看应用基本信息。

Q: 权限报错

A: 可能是缓存问题，优先尝试加上`--reset`看是否可以解决

最佳实践（新增文章目录）

1. 下载或者更新 shoplazza cli

Code block

```
1 npm i -g shoplazza-cli
```

Then press any key to open the login page on your browser

2. 在网页上选择或创建 Partner 账户

3. 初始化 app 项目

Code block

```
1 shoplazza app init
```

- 选择对应的 organization 账户
- 选择创建新app，或者链接已存在app
 - 选择新app后，输入app 名称（例:your-project），生成目录和基础文件结构，`shoplazza.app.toml` 配置文件

```

1 my-app/
2   ├── shoplazza.app.toml      # 主配置文件
3   ├── shoplazza.app.prod.toml # 生产环境配置
4   ├── shoplazza.app.staging.toml # 测试环境配置
5   ├── extensions/            # 扩展目录
6   │   ├── theme-extension/    # 主题扩展
7   │   ├── checkout-extension/ # 结账扩展
8   │   └── function-extension/ # 功能扩展
9   ├── package.json
10  └── README.md

```

4. 创建 app 扩展

Code block

```

1 # 进入 app 项目目录
2 cd your-project
3
4 # 执行创建 extension 的命令
5 shoplazza app generate extension

```

- 选择扩展类型 (Theme/Checkout/Function)
 - 选择theme extension 后会选择theme extension的类型： embed 或者 basic (外链)
- 输入扩展名称
- 生成扩展的基础模板文件

Code block

```

1
2 extensions/
3   └── my-extension/
4       ├── shoplazza.extension.toml # 扩展配置文件
5       ├── src/                     # 源代码目录
6       │   ├── index.js             # 主入口文件
7       │   ├── index.html           # HTML 模板
8       │   ├── style.css             # 样式文件
9       │   └── script.js             # JavaScript 逻辑
10      └── README.md                 # 扩展说明

```

5. 调试

Code block

1 shoplazza app dev

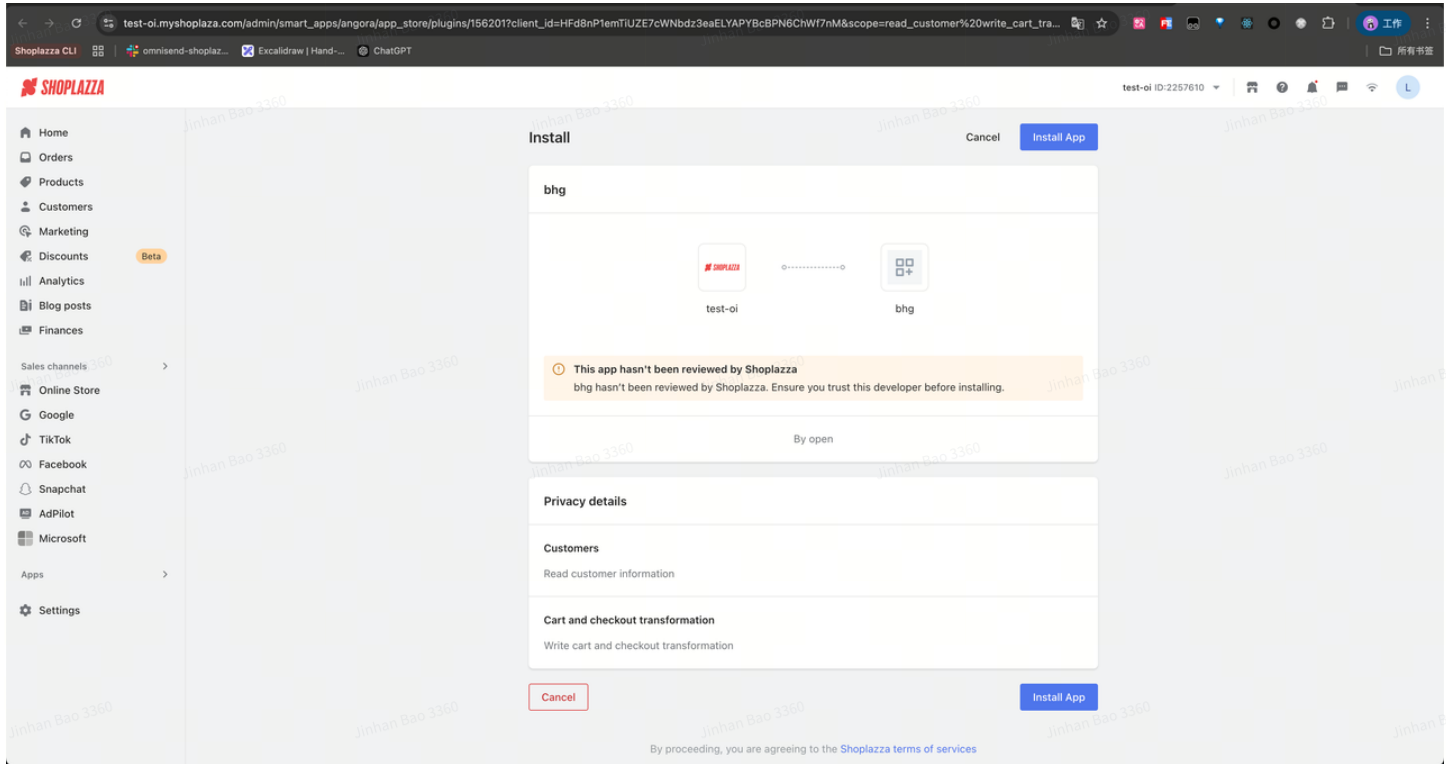
上述命令执行成功后：

1. 根据终端提示，输入测试用商店domain
2. 终端返回App URL、Redirect URL，Install URL 如图中所示
3. 前往 [Partner Admin](#)，将返回的 App URL、Redirect URL 填写到 App 设置中

```
Dev server start success 🎉🎉🎉
Your App URL: https://technology-seek-multimedia-loved.trycloudflare.com/auth
Your Redirect URL: https://technology-seek-multimedia-loved.trycloudflare.com/auth/callback
Your Install URL: https://technology-seek-multimedia-loved.trycloudflare.com/auth?hmac=04bf1bf587e5533136d95c30160c7f45193dbe1220e047a2bd37ab2dcd5d8196&install_from=cli
&shop=test-01.myshoplaza.com&store_id=2257610
```

The screenshot shows the 'App setup' page in the Shoplazza Partner Admin. The page is divided into two main sections: 'App information' and 'URLs'. The 'App information' section includes fields for 'App icon', 'App name', 'App contact email', and 'Urgent contact method'. The 'URLs' section includes fields for 'App URL' and 'Redirect URL'. The 'App URL' and 'Redirect URL' fields are highlighted with a red box, indicating they are the focus of the current step. The 'App URL' field contains the value 'https://technology-seek-multimedia-loved.trycloudflare.com/auth' and the 'Redirect URL' field contains the value 'https://technology-seek-multimedia-loved.trycloudflare.com/auth/callback'.

4. 访问 Install URL, 进行安装测试

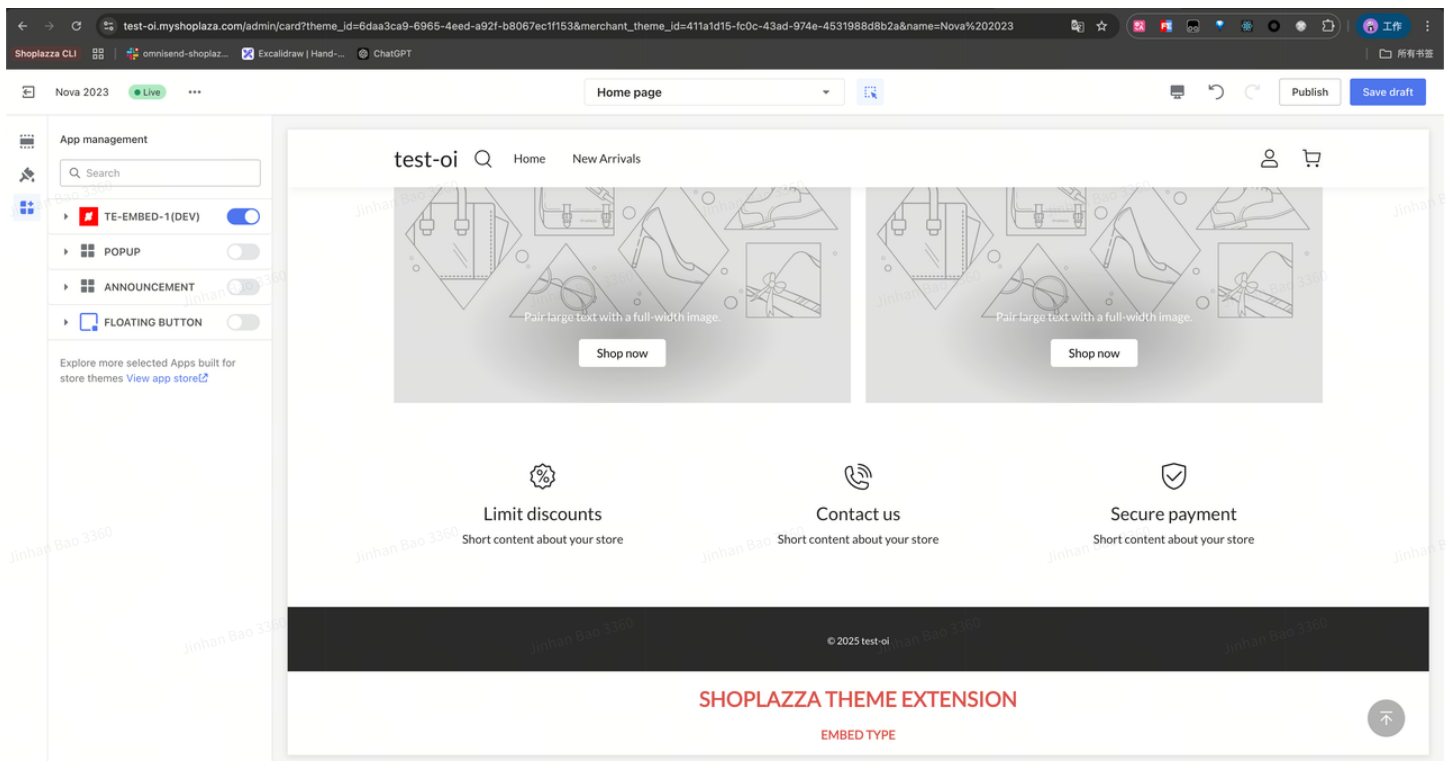


5. 查看效果

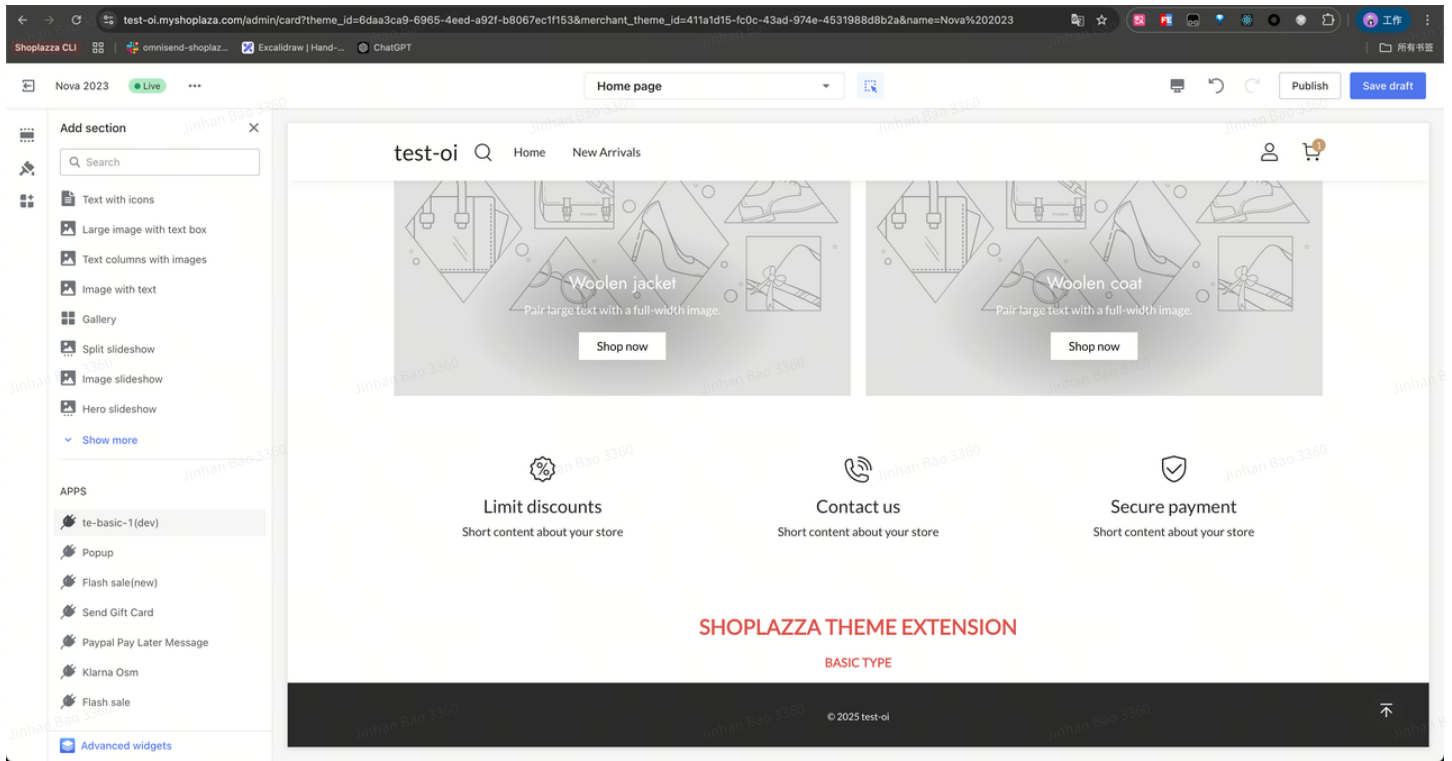
Theme:

1. 商店admin, 进入在线商店, 主题装修, 选择对应主题进行点击编辑, 进入预览页面
2. 打开应用管理, 进行主题预览

Embed theme extension:

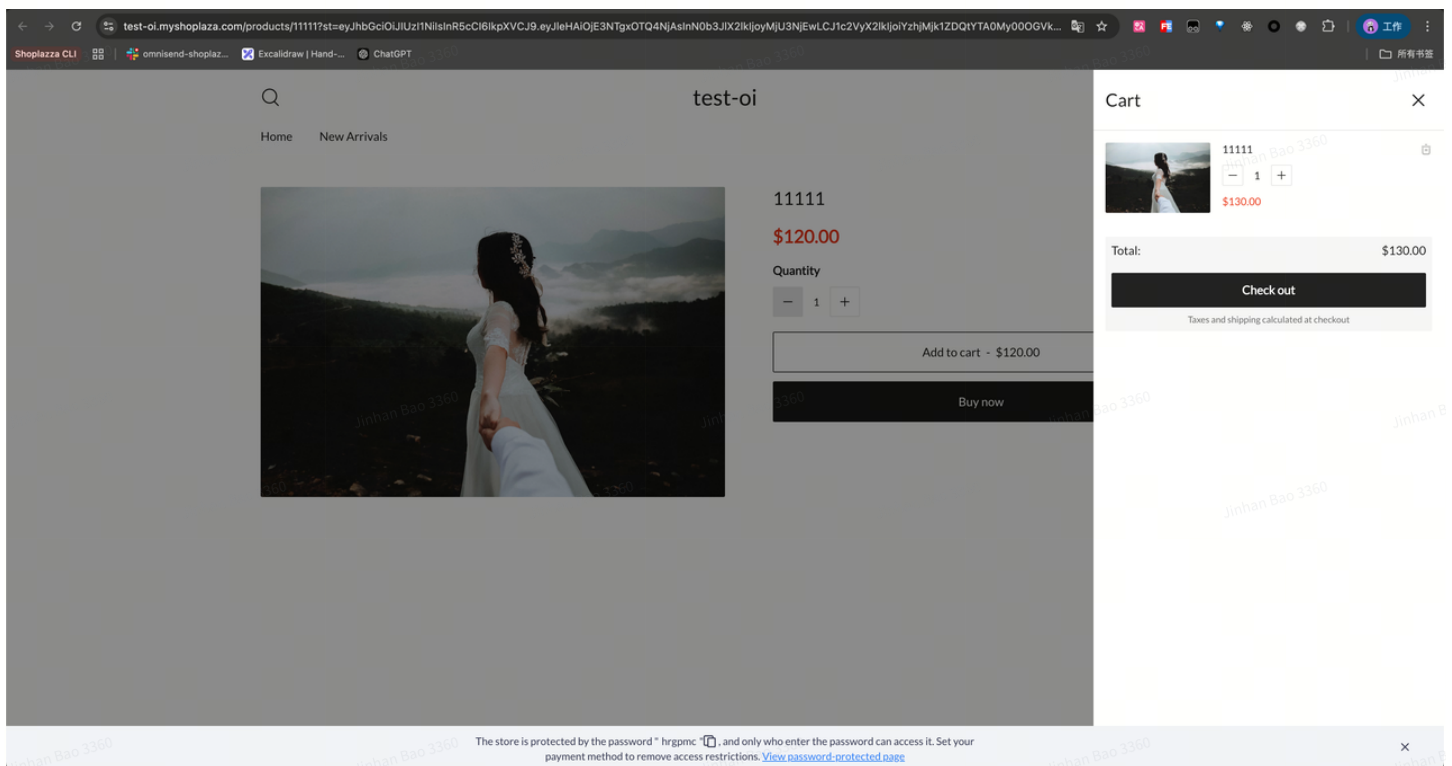


Basic theme extension:



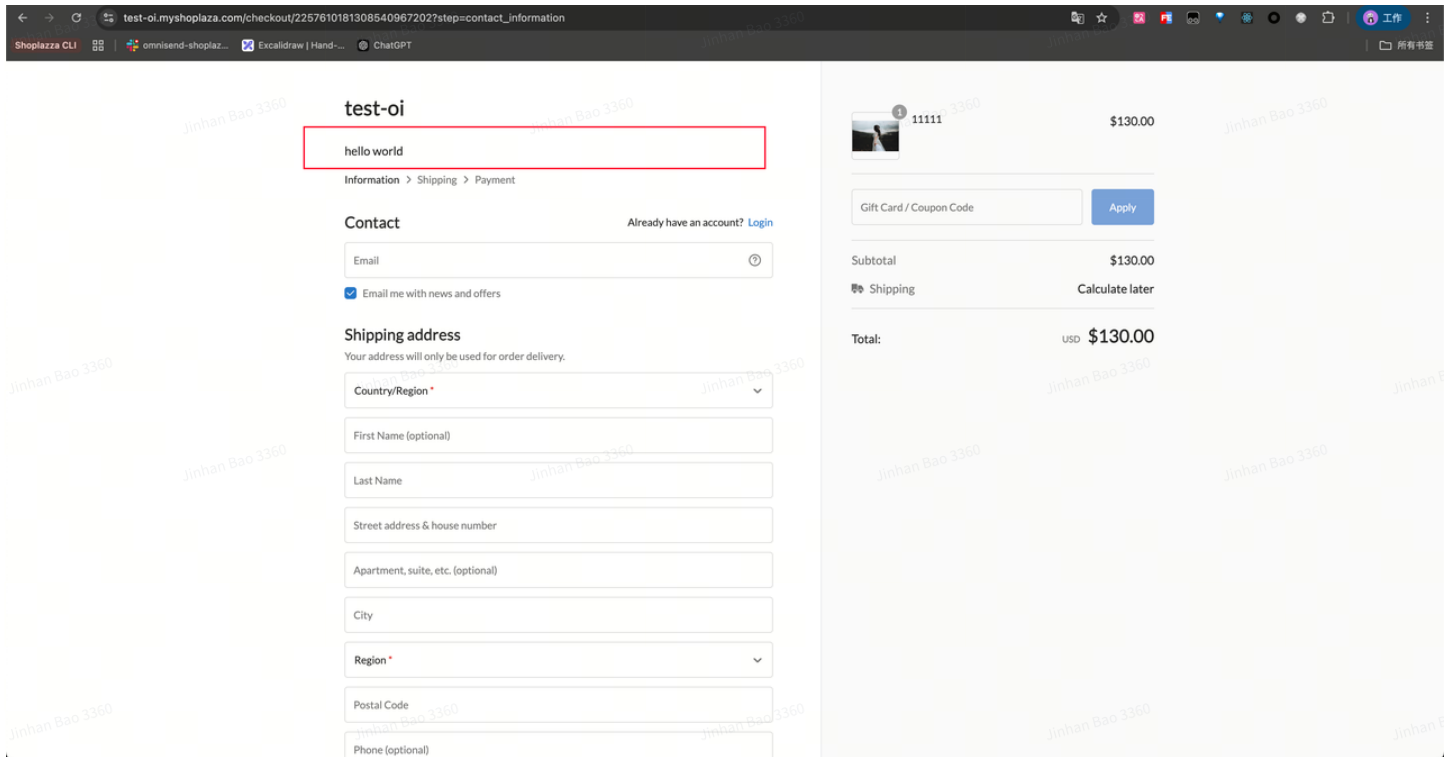
Function :

1. 进入在线商店，进行加购动作，观察改价脚本是否执行



Checkout:

1. 进入在线商店，进入结账页面，观察脚本是否注入



6. 部署发布

Code block

```
1 shoplazza app deploy
```

部署发布: shoplazza app deploy

1. 检查项目配置和扩展， 构建所有扩展文件，上传资源到 OSS，创建应用版本
2. 部署到生产环境， 提供部署结果和预览链接

其他技术支持

3. 版本管理

- 使用语义化版本号 (如 1.0.0, 1.1.0, 2.0.0)
- 定期查看版本历史: shoplazza app versions list

4. 配置管理

- 为不同环境创建不同的配置文件
- 使用 `shoplazza app config use` 在环境间切换
- 定期备份配置文件

`shoplazza app config` [link](#)

如果在使用过程中遇到问题，可以：

2. 查看帮助信息: `shoplazza app --help`

3. 查看具体命令帮助: `shoplazza app <command> --help`

最后更新: 2025-09