

同济大学计算机系软件工程课程

易场地 ExCourt 软件部署配置说明



小组名称

小组成员

学院（系）

专业

任课教师

沪爷冲击开发团队

侯青山

仇伟烨

赵嘉禾

朱雨欢

吴天宇

赵唯旭

计算机科学与技术学院

计算机科学与技术

赵君峤

Contents

1	代码结构	1
2	前端部署	2
2.1	部署环境	2
2.2	部署流程	2
3	后端部署	5
3.1	部署环境	5
3.2	所需变动	6
3.3	运行	7

1 代码结构

项目的整体代码结构如下所示：

```
1  ExCourt_front          # 前端项目目录
2  .eslintrc.js          # ESLint 配置文件
3  README.md             # 项目说明文档
4  app.js / app.json     # 小程序全局配置
5  app.wxss              # 全局样式文件
6  images/               # 图片资源目录
7      logo.png          # 应用Logo
8      城市.png / 景点.png # 示例图片资源
9      ...               # 其他图片资源
10 pages/                # 小程序页面目录
11     1-court-view/      # 场地查看页面
12         court-view.js
13         court-view.wxml
14         court-view.wxss
15     3-applyCourt/      # 申请场地页面
16     3-publishCourt/    # 发布场地页面
17     4-index/           # 失物招领首页页面
18     6-Me/              # 个人中心页面
19     6-login/           # 登录页面
20     6-register/        # 注册页面
21     ExCourtTeam/       # 小组展示页面
22     ...               # 其他功能页面
23     project.config.json # 小程序项目配置
24     sitemap.json       # 页面结构配置
25     utils/util.js      # 工具函数
26     ...               # 其他配置文件
27 SchoolAPI              # 学校API服务端
28     SchoolAPI.sql      # 学校数据库结构
29     app.py             # API 启动文件
30     config.py          # 配置文件
31 system                 # 系统服务端
32     Excourt.sql        # 系统数据库结构
33     app/               # 后端业务逻辑模块
34         routes/        # 路由模块
35             chat.py     # 聊天功能
36             exchangecourt.py # 场地交换功能
37             lost_and_found.py # 失物招领功能
38             offercourt.py # 送场功能
39             teamup.py   # 组队功能
40             upload.py   # 上传功能
41             sync_courtinfo.py # 场地信息同步脚本
42     config.py          # 系统服务端配置
43     run.py             # 服务端启动文件
```

- **ExCourt_front**：前端小程序代码，包含页面、配置和资源文件。
- **SchoolAPI**：模拟学校数据库 API 服务，主要处理学生验证及场地信息查询等功能。
- **system**：系统后端服务。

本项目前后端模块清晰，包含上述三个文件夹分别完成前端和两个后端服务。前端基于微信小程序框架实现，后端基于 Flask 提供 API 服务，数据通过 MySQL 数据库管理和存储。

2 前端部署

2.1 部署环境

本项目的前端基于微信小程序框架开发，部署需要使用 **微信开发者工具**。微信开发者工具的下载和使用指南请参考以下链接：[微信开发者工具官方文档](#)。

- **操作系统支持**：Windows / macOS
- **Node.js 环境（可选，本项目无需使用）**：安装 Node.js v14+ 版本
- **微信开发者工具版本**：最新稳定版或夜 ly 版

2.2 部署流程

前端的部署流程如下：

1. 下载并安装微信开发者工具

从官方链接下载安装微信开发者工具，具体下载页面如下图所示：

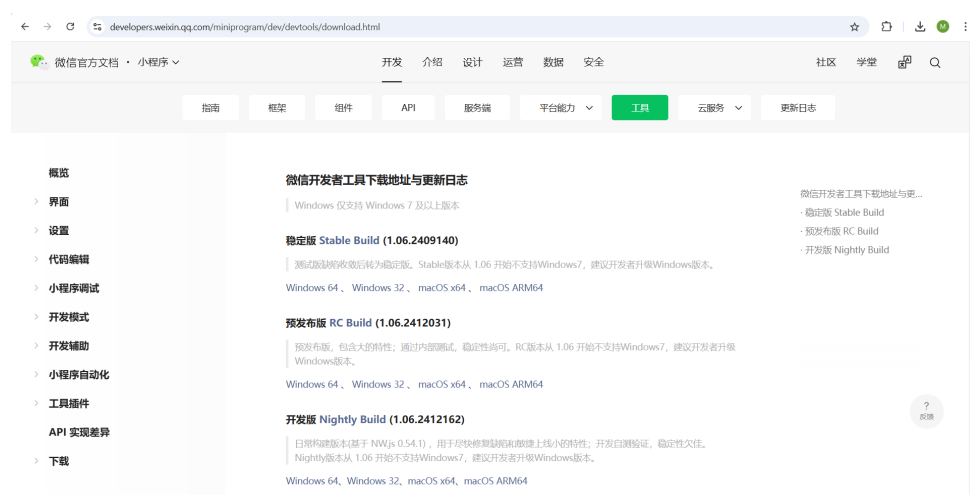


Figure 1: 微信开发者工具下载页面

2. 导入项目

打开微信开发者工具，选择“导入项目”，并指定项目路径为 **ExCourt_front** 文件夹：



Figure 2: 选择导入项目

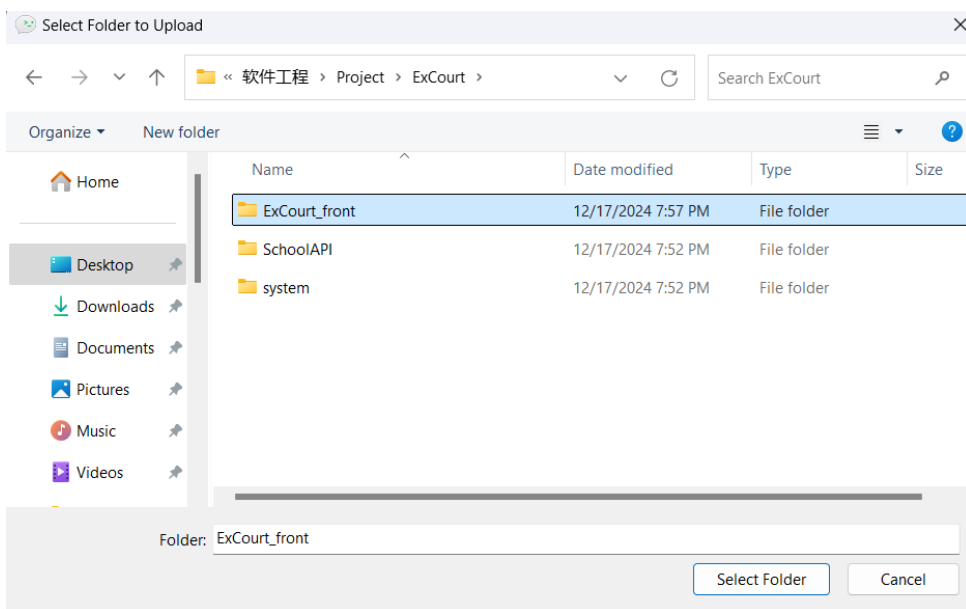


Figure 3: 导入项目文件夹

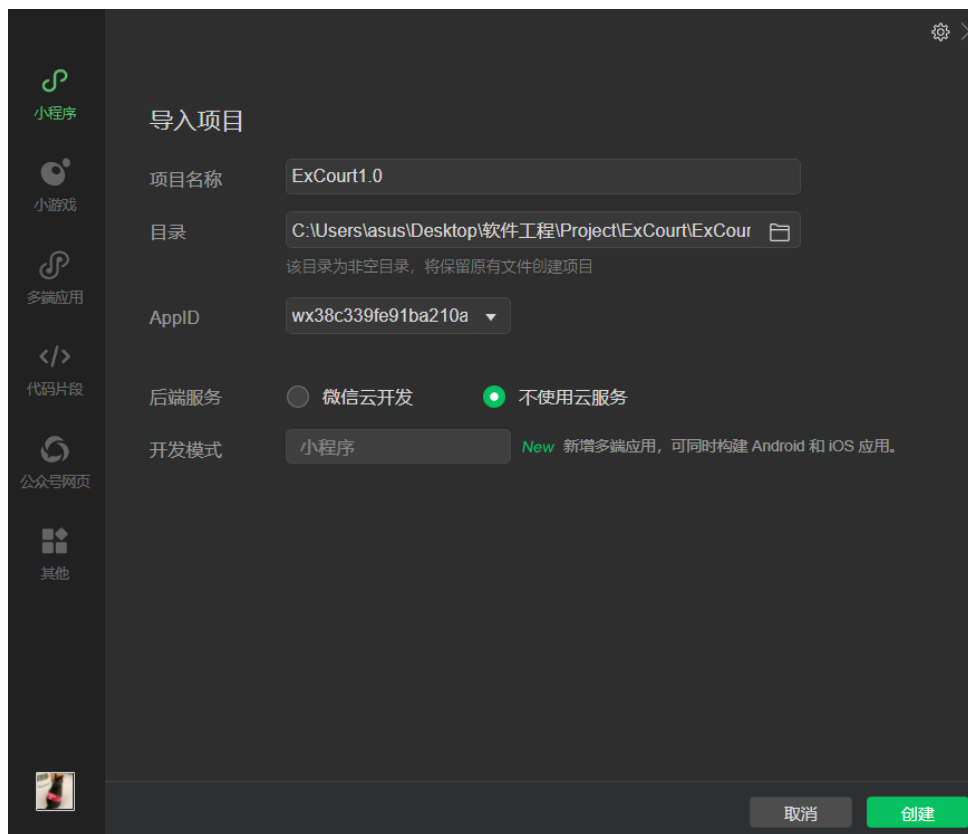


Figure 4: 导入项目配置

在这里，我们使用的 AppID 是在微信官方公众平台中申请的，读者可以自行申请一个自己的小程序 ID 进行替换，也可以使用测试 ID。

3. 检查项目配置

打开项目配置文件 `project.config.json`，确认配置是否正确，同时检查右侧的本地配置项，一定注意勾选“不校验合法域名”：

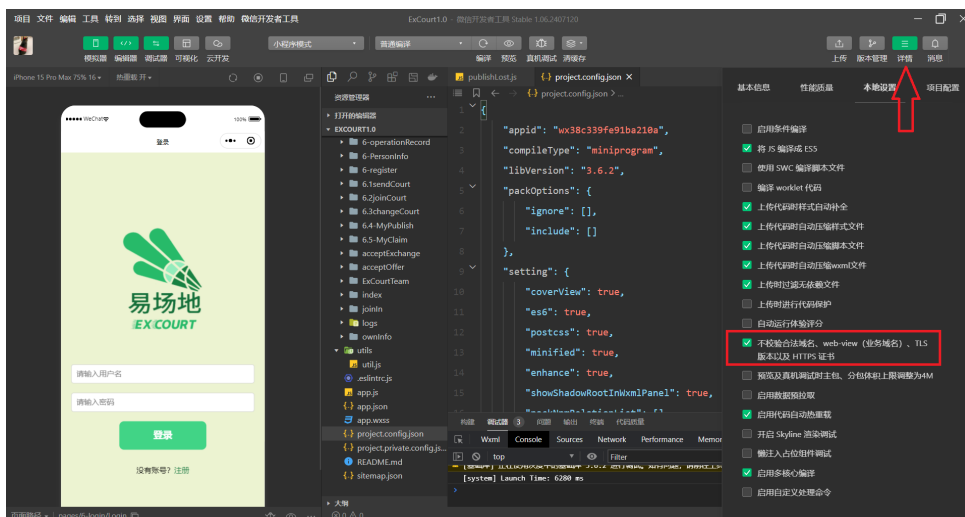


Figure 5: 检查项目配置

在这里需要注意小程序 ID 等内容的一致性。

4. 编译项目

在微信开发者工具中点击“编译”按钮，编译小程序：

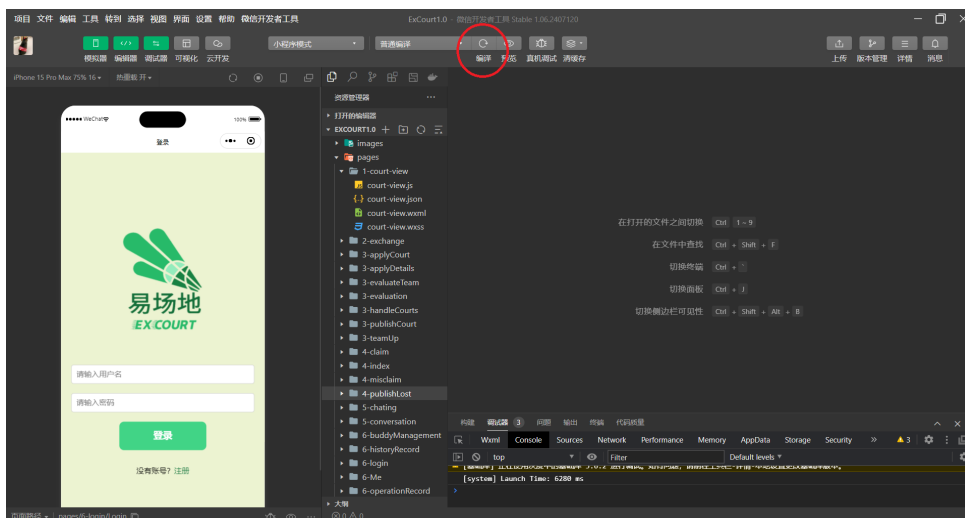


Figure 6: 编译小程序代码

如果编译正常，本项目的前端就部署完毕了，读者可以在后续点击预览，发布等功能进行探索。

3 后端部署

3.1 部署环境

需要再服务器上安装 python3, 本小组使用的版本是 python3.10.

需要用到的库有如下这些:

```
1 root@ecs-b859:~# pip list
2 Package                               Version
3 -----
4 bidict                                0.23.1
5 blinker                              1.9.0
6 click                                8.1.7
7 cloud-init                            19.1
8 Flask                                 2.3.3
9 Flask-Cors                            5.0.0
10 Flask-MySQLdb                        1.0.1
11 Flask-SocketIO                       5.4.1
12 h11                                   0.14.0
13 itsdangerous                         2.2.0
14 Jinja2                               3.1.4
15 MarkupSafe                           3.0.2
16 mysql-connector-python               8.0.23
17 mysqlclient                          2.1.1
18 pip                                  24.3.1
19 protobuf                             5.29.1
20 PyMySQL                              1.1.1
21 python-engineio                      4.10.1
22 python-socketio                      5.11.4
23 setuptools                           59.6.0
24 simple-websocket                     1.1.0
25 uWSGI                                2.0.28
26 Werkzeug                             3.1.3
27 wheel                                0.37.1
28 wsproto                              1.2.0
```

3.2 所需变动

system 表示系统数据库, 端口号 8000.SchoolAPI 表示学校数据库模拟, 端口号 5000.

对 SchoolAPI 后台, 需要修改根目录下面的 config.py:

```
1 class Config:
2     MYSQL_HOST = '123.60.86.239'
3     MYSQL_PORT = 3306
4     MYSQL_USER = 'root'
5     MYSQL_PASSWORD = 'QWer1234'
6     MYSQL_DB = 'school'
```

为自己使用的学校数据库的相关属性, 而 schoolAPI.sql 即为数据库的表样式。

对 system 后台, 需要修改根目录下面的 config.py:

```
1 class Config:
```



```

2   MYSQL_HOST = '123.60.86.239'
3   MYSQL_PORT =      3306
4   MYSQL_USER = 'root'
5   MYSQL_PASSWORD = 'QWer1234'
6   MYSQL_DB = 'system'

```

为自己使用的数据库主机，端口，用户，密码，数据库名，excourt.sql 即数据库的表列表。

同时需要修改 app/sync_courtInfo.py,

```

1   school_db_config = {
2   'host': '123.60.86.239',
3   'port': 3306,
4   'user': 'root',
5   'password': 'QWer1234',
6   'database': 'school'
7 }
8
9 local_db_config = {
10  'host': '123.60.86.239',
11  'port': 3306,
12  'user': 'root',
13  'password': 'QWer1234',
14  'database': 'system'
15 }

```

类似前面的操作即可

3.3 运行

数据库部署比较随意，只需要后端与之对应即可，注意上文提到的数据库中的表都是已经给出的，就可以很简单的部署数据库了。如果需要添加测试数据，请先向 schoolAPI 数据库中的 Student 表中添加学生数据，这样用户才能成功注册。（相当于学校统一身份验证）如需要添加场地测试数据可在 school.Courtinfo 表中添加，脚本会将场地信息同步到 system 数据库中。

打开终端，在 system 文件夹下，输入

```

1   python run.py # 可能是python3，根据系统而定

```

在另一个终端，在 schoolAPI 文件夹下，同样：

```

1   python app.py # 可能是python3，根据系统而定

```

即可运行本项目的后端