

部署文档

一、概述

本项目采用前后端分离模式，使用 **Vue3** 和 **Yii2** 框架搭建，结合 **MySQL** 作为数据库。以下是本项目的技术栈概述：

1. 前端技术栈：Vue.js

- Vue.js 是一个灵活且易于学习的 JavaScript 框架，支持组件化开发，提供响应式数据绑定、虚拟 DOM、路由等核心功能。

2. 后端技术栈：Yii2 (PHP)

- Yii2 是一个高性能的 PHP 框架，专注于快速、安全和可扩展的 Web 开发，提供了完整的数据库、路由和用户身份验证支持。

3. 前后端通信：RESTful API

- 前端通过 Axios 与后端的 RESTful API 进行通信，数据交互采用 JSON 格式。

4. 数据库：MySQL

- 负责存储用户数据、歌单数据和评论等内容。

二、部署说明

1. 前期准备

- 确保已安装 **Node.js** 和 **npm**，如未安装请访问 [Node.js官网](#)。
- 确保已安装 **Composer**，如未安装请访问 [Composer官网](#)。
- 确保已安装 **MySQL** 数据库，并将数据库的 root 用户密码设置为 **123456**。

2. 前端部署

1. 安装依赖

```
cd project-frontend #前端项目文件夹
npm install
npm install element-plus plyr comment-core-library gsap axios ip
```

2. 本地开发

- 运行以下命令启动本地开发服务器：

```
npm run dev
```

- 默认会启动在 <http://localhost:8081>。

3. 生产环境构建

- 构建生产环境文件：

```
npm run build
```

- 构建完成后，会生成 [dist](#) 文件夹，将其中的内容上传到服务器。

4. 服务器部署

- 使用 FTP 工具或命令行（如 rsync）将 [dist](#) 文件夹上传到服务器。
- 配置服务器（如 Nginx 或 Apache），确保正确指向静态文件目录并配置 URL 重写规则（如果需要）。

3. 后端部署

1. 安装依赖

```
cd project-backend #后端项目文件夹
composer install
```

2. 本地开发

- 运行以下命令启动后端服务器：

```
php yii serve --port=8080
```

- 如果端口 [8080](#) 被占用，请使用其他端口，并修改前端中所有的 [localhost:8080](#) 为新的端口号。

3. 数据库初始化

- 导入项目提供的 [.sql](#) 文件到 MySQL 数据库中，创建相关的表结构和数据。

4. 后端访问

- 启动后端服务器后，通过浏览器访问 <http://localhost:8080> 进行测试。

三、常见问题

1. 依赖安装失败

- 确保网络正常，或检查 npm 和 Composer 源是否可用。

2. 端口冲突

- 如果端口被占用，可以在启动命令中指定其他端口，并修改前端配置。

3. 跨域问题

- 确保后端启用了跨域支持（CORS），以便前后端能够正常通信。

四、访问方式

1. 前端访问

- 部署成功后，通过 `http://<服务器地址>:8081` 访问前端页面。

2. 后端访问

- 部署成功后，通过 `http://<服务器地址>:8080` 测试后端接口。

本部署文档旨在简化本项目的安装和部署流程，如有其他问题，请参考项目中的相关文档。

这份文档涵盖了项目的技术栈、前后端部署流程、一键脚本使用说明及常见问题解答，可根据实际项目细节进一步修改和完善。