

部署文档

一、概述

经过与任课老师的讨论并得到了其许可，本小组的项目使用了 `Vue3` + `Yii2` 的前后端分离模式

1. 前端技术栈：Vue.js

- Vue.js是一个灵活且易于学习的JavaScript框架，用于构建用户界面。
- 它采用组件化的开发方式，使开发者能够轻松地构建复杂的交互式应用程序。
- Vue提供了响应式数据绑定、虚拟DOM、路由和状态管理等核心功能，以及大量的生态系统插件和工具。

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。
- 它提供了一套强大的工具和组件，用于处理请求、路由、数据库访问、身份验证等常见的后端任务。
- Yii2采用了现代化的设计模式和最佳实践，使开发者能够高效地构建可维护和健壮的应用程序。

3. 前后端通信：RESTful API

- Vue前端通过使用Axios等HTTP库与Yii2后端进行通信。
- 通常使用RESTful API来定义前后端之间的数据交互规范。
- 前端发送HTTP请求到后端的API接口，后端处理请求并返回JSON格式的数据给前端。

4. 数据库：Mysql

- Yii2提供了对多种数据库的支持，包括MySQL、PostgreSQL、SQLite等。
- 我们选择使用Mysql作为数据存储和管理数据。

二、部署说明

前期准备

请你参照南开大学计网学院大二下的必修课《数据库系统》，完成Mysql数据库的安装，为了更好地适配后端，可以提前将数据库密码修改为'root'

前端部署

1. 首先确保你已经安装了Node.js和npm，如果没有可以去<https://nodejs.org/>下载，相信聪明的你不会在这一步遇到问题
2. 进入到 `frontend` 文件夹，这一步需要你安装前端项目所需要的依赖项，请你依次运行以下命令(下列命令可以在requirements.txt中找到)

```
1  npm install
2  npm run format
3  npm install element-plus
4  npm install plyr
5  npm install comment-core-library
6  npm install gsap
7  npm install axios
8  npm install ip
```

3. 如果你运行完上述指令并且没有出现问题，就可以在当前目录运行 `npm run dev`，启动开发服务器，以便在本地进行开发和测试。

到了这一步，已经**足够完成**本次作业的要求，如果你需要将本项目部署到Web服务器，可以按照下面的流程进行。

4. 运行构建命令 `npm run build`，这将在项目根目录下生成一个 `dist` 文件夹，其中包含了构建后的前端资源文件

5. 将 `dist` 文件夹中的所有文件上传到Web服务器上。你可以使用FTP工具、命令行工具（如rsync）或其他方式将文件复制到服务器上
6. 根据你使用的Web服务器类型（如Nginx或Apache等），进行相应的配置。确保你的服务器已正确设置用于访问静态文件的目录，并配置好相关的URL重写规则（如果需要）
7. 通过浏览器访问你的网站的URL，即可查看和使用部署后的Vue前端项目

后端部署

1. 首先需要安装Composer，如果你没有安装，可以去<https://getcomposer.org/download/>，并且按照教程进行安装
2. 进入到 `backend` 文件夹，运行 `composer install` 命令，安装本后端项目所需依赖，其中一些依赖可能需要你使用Github的token进行api的获取，相信常年与github打交道并且过了英语四级的一定能够根据控制台输出的提示完成这一步
3. 如果你运行完上述指令并且没有出现问题，就可以在当前目录运行 `php yii serve --port=8080` 来启动后端服务器
(如果你的8080端口被占用了，请你去前端项目中将所有的'localhost:8080'替换成'localhost:[你后端打开的端口]')
4. 至此，就可以使用浏览器通过 <http://localhost:8080/> 来访问后端界面了

一键部署功能

为了方便作业检查，本组编写了一个简单的一键部署脚本，在你安装了 `Node.js` 和 `Composer` 以及将.sql文件导入数据库中后就可以运行 `start.bat` 来帮你完成上述的命令行输入

一键部署脚本说明

start.bat只是另外两个部署脚本 `deploy_frontend.ps1` 和 `deploy_backend.ps1` 的启动项

```
1  @echo off
2  echo Please make sure you have imported the SQL file into the
   database. Press Enter to continue...
3  pause>nul
4  cd /d "%~dp0\frontend"
5  start powershell -NoProfile -ExecutionPolicy Bypass -File
   "%~dp0\frontend\deploy_frontend.ps1"
6  cd /d "%~dp0\backend"
7  start powershell -NoProfile -ExecutionPolicy Bypass -File
   "%~dp0\backend\deploy_backend.ps1"
```

deploy_frontend.ps1

```
1  # 进入项目目录
2  Write-Host "Entering project directory..."
3  cd ./
4
5  # 安装依赖库
6  Write-Host "Installing dependencies..."
7  npm install --silent 2>&1 | Out-Null
8
9  # 运行格式化命令
10 Write-Host "Running formatting command..."
11 npm run format 2>&1 | Out-Null
12
13 # 安装其他依赖库
14 Write-Host "Installing other dependencies..."
15
```

```
16 $packages = @("element-plus", "plyr", "comment-core-  
    library", "gsap", "axios", "ip")  
17 $totalPackages = $packages.Count  
18 for ($i = 0; $i -lt $totalPackages; $i++) {  
19     $package = $packages[$i]  
20     Write-Progress -Activity "Installing package '$package'"  
        -Status "Package $($i + 1) of $totalPackages" -  
        PercentComplete ((($i + 1) / $totalPackages * 100)  
21     npm install $package --silent --progress=false 2>&1 |  
        Out-Null  
22 }  
23  
24 # 构建前端项目  
25 Write-Host "Starting frontend project..."  
26 npm run dev  
27  
28 Write-Host "Script execution completed!"  
29
```

deploy_backend.ps1

```
1 # 获取用户输入的mysql密码  
2 $mysql_password = Read-Host "Please input the mysql  
    password"  
3  
4 # 替换db.php中的mysql密码  
5 $config_file_path = "./config/db.php"  
6 (Get-Content $config_file_path) | ForEach-Object { $_ -  
    replace "'password' => 'root'", "'password' =>  
        '$mysql_password'" } | Set-Content $config_file_path  
7  
8 # 返回脚本文件所在目录  
9 cd ..  
10  
11 # 运行composer install
```

```
12 composer install
13
14 # 运行php yii serve
15 php yii serve
16
```

后端部署脚本提供了一个修改db.php中数据库密码的功能