

# 抗战胜利80周年项目部署文档

## 项目简介

本项目是基于 Yii2 Advanced 模板开发的抗战胜利80周年数字档案馆系统，包含前台门户和后台管理两个应用。

- 前台门户：<http://localhost:8080>
- 后台管理：<http://localhost:8081>
- 默认账号：VignaChu
- 默认密码：114514

## 一、环境要求

### 必需环境

| 软件           | 最低版本     | 说明              |
|--------------|----------|-----------------|
| PHP          | >= 5.6.0 | 建议使用 PHP 7.4+   |
| MySQL        | >= 5.7   | 或 MariaDB 10.2+ |
| Composer     | 最新版      | PHP 依赖管理工具      |
| Apache/Nginx | -        | 或使用 PHP 内置服务器   |

### PHP 扩展要求

- `pdo\_mysql` - MySQL 数据库驱动
- `mbstring` - 多字节字符串处理
- `openssl` - 加密功能
- `curl` - HTTP 请求
- `zip` - 压缩文件处理

### 检查环境

```
代码块
1  #检查 PHP 版本
```

```
2  php -v
3
4  #检查 MySQL 版本
5  mysql --version
6
7  #检查 Composer
8  composer --version
9
10 #检查 PHP 扩展
11 php -m | findstr pdo_mysql
12 php -m | findstr mbstring
```

## 二、自动部署（推荐）

### 2.1 部署步骤

#### 1. 确保环境已安装

- PHP、MySQL、Composer 已正确安装并配置到系统 PATH

#### 2. 运行部署脚本

代码块

```
1  # 双击运行或在命令行执行
2  deploy(1).bat
```

#### 3. 按提示输入信息

- 输入 MySQL root 密码（若无密码直接回车）

#### 4. 等待部署完成

- 脚本会自动完成以下操作：
  - 初始化 Yii2 环境
  - 配置数据库连接
  - 创建数据库
  - 导入数据表结构及初始数据

#### 5. 启动服务

代码块

```
1  # 双击运行
```

```
2 start_all.bat
```

## 2.2 自动部署脚本说明

`deploy(1).bat` 脚本执行以下操作：

### 1. 初始化 Yii2 环境

代码块

```
1 cd advanced
2 php init --overwrite=All --env=Development
```

### 2. 生成数据库配置文件

- 自动创建 `advanced/common/config/main-local.php`
- 配置数据库连接信息

### 3. 创建数据库

代码块

```
1 CREATE DATABASE IF NOT EXISTS yii CHARACTER SET utf8mb4 COLLATE
  utf8mb4_unicode_ci;
```

### 4. 导入数据

- 执行 `advanced/data/install.sql` 导入表结构和初始数据

## 三、手动部署

### 3.1 安装依赖

代码块

```
1 #进入项目目录
2 cd advanced
3
4 #安装 Composer 依赖 (如果 vendor 目录不存在)
5 composer install
```

### 3.2 初始化 Yii2 环境

代码块

```
1 #在 advanced 目录下执行
2 php init
3
4 #选择环境类型: 0 (Development)
5 #选择是否覆盖: All
```

### 3.3 配置数据库

编辑 `advanced/common/config/main-local.php` :

代码块

```
1 <?php
2 return [
3     'components' => [
4         'db' => [
5             'class' => 'yii\db\Connection',
6             'dsn' => 'mysql:host=localhost;port=3306;dbname=yii',
7             'username' => 'root',
8             'password' => '你的MySQL密码',
9             'charset' => 'utf8mb4',
10        ],
11        'mailer' => [
12            'class' => 'yii\swiftmailer\Mailer',
13            'viewPath' => '@common/mail',
14            'useFileTransport' => true,
15        ],
16    ],
17 ];
```

### 3.4 创建数据库

代码块

```
1 方式1: 使用命令行
2 mysql -u root -p -e "CREATE DATABASE IF NOT EXISTS yii CHARACTER SET utf8mb4
  COLLATE utf8mb4_unicode_ci;"
3
4 方式2: 使用 MySQL 客户端
5 登录 MySQL 后执行:
6 CREATE DATABASE IF NOT EXISTS `yii` CHARACTER SET utf8mb4 COLLATE
  utf8mb4_unicode_ci;
```

### 3.5 导入数据

代码块

```
1  #方式1: 使用命令行
2  mysql -u root -p yii < advanced/data/install.sql
3
4  #方式2: 使用 MySQL 客户端
5  #选择数据库后执行 SQL 文件内容
```

## 3.6 设置目录权限 (Linux/Mac)

代码块

```
1  #设置运行时目录权限
2  chmod -R 777 advanced/common/runtime
3  chmod -R 777 advanced/backend/runtime
4  chmod -R 777 advanced/frontend/runtime
5  chmod -R 777 advanced/console/runtime
6
7  #设置 Web 资源目录权限
8  chmod -R 777 advanced/backend/web/assets
9  chmod -R 777 advanced/frontend/web/assets
```

Windows 系统无需设置权限

## 四、启动服务

### 4.1 使用启动脚本 (推荐)

代码块

```
1  #双击运行
2  start_all.bat
```

脚本会自动:

- 启动前台服务 (端口 8080)
- 启动后台服务 (端口 8081)
- 自动打开浏览器访问

## 4.2 手动启动

### 方式1：使用 PHP 内置服务器

代码块

```
1  #启动前台（新开终端窗口）
2  cd advanced
3  php yii serve --docroot=frontend/web --port=8080
4
5  #启动后台（新开终端窗口）
6  cd advanced
7  php yii serve --docroot=backend/web --port=8081
```

### 方式2：配置 Web 服务器

Apache 配置示例（`httpd-vhosts.conf`）：

代码块

```
1  <VirtualHost *:80>
2      ServerName frontend.local
3      DocumentRoot "E:/document/数据库开发/大作业2/大作业/advanced/frontend/web"
4      <Directory "E:/document/数据库开发/大作业2/大作业/advanced/frontend/web">
5          AllowOverride All
6          Require all granted
7      </Directory>
8  </VirtualHost>
9
10 <VirtualHost *:81>
11     ServerName backend.local
12     DocumentRoot "E:/document/数据库开发/大作业2/大作业/advanced/backend/web"
13     <Directory "E:/document/数据库开发/大作业2/大作业/advanced/backend/web">
14         AllowOverride All
15         Require all granted
16     </Directory>
17 </VirtualHost>
```

Nginx 配置示例：

代码块

```
1  server {
2      listen 8080;
3      server_name localhost;
4      root E:/document/数据库开发/大作业2/大作业/advanced/frontend/web;
5      index index.php;
6
7      location / {
8          try_files $uri $uri/ /index.php?$args;
9      }
10
11     location ~ /\.php$ {
12         fastcgi_pass 127.0.0.1:9000;
13         fastcgi_index index.php;
14         include fastcgi_params;
15         fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
16     }
17 }
```

## 五、访问地址

部署成功后，可通过以下地址访问：

- 前台门户：<http://localhost:8080>
- 后台管理：<http://localhost:8081>
- Gii 代码生成器（开发环境）：
  - 前台：<http://localhost:8080/gii>
  - 后台：<http://localhost:8081/gii>
- Debug 工具栏（开发环境）：自动显示在页面底部

## 六、常见问题

### 6.1 数据库连接失败

问题：`SQLSTATE[HY000] [2002] No connection could be made`

解决方案：

1. 检查 MySQL 服务是否启动
2. 确认数据库用户名、密码、端口配置正确
3. 检查 `main-local.php` 配置文件是否存在

## 6.2 Composer 安装失败

问题：`composer install` 执行失败

解决方案：

代码块

```
1  #使用国内镜像
2  composer config -g repo.packagist composer https://mirrors.aliyun.com/composer/
3
4  #清除缓存后重试
5  composer clear-cache
6  composer install
```

## 6.3 权限错误 (Linux/Mac)

问题：`Runtime directory is not writable`

解决方案：

代码块

```
1  chmod -R 777 advanced/*/runtime
2  chmod -R 777 advanced/*/web/assets
```

## 6.4 端口被占用

问题：`Address already in use`

解决方案：

代码块

```
1  Windows 查看端口占用
2  netstat -ano | findstr :8080
3
4  修改启动脚本中的端口号，或关闭占用端口的程序
```

## 6.5 中文乱码

问题：页面显示乱码

解决方案：

1. 确保数据库字符集为 `utf8mb4`



2. 检查 PHP 文件编码为 UTF-8（无 BOM）

3. 确保 Web 服务器响应头包含 `Content-Type: text/html; charset=utf-8`

## 七、项目结构

代码块

```
1  大作业/
2  |—— advanced/           # Yii2 项目主目录
3  |   |—— backend/        # 后台管理应用
4  |   |—— frontend/       # 前台门户应用
5  |   |—— common/         # 公共代码
6  |   |—— console/        # 控制台应用
7  |   |—— data/           # 数据文件
8  |   |   |—— install.sql  # 数据库初始化脚本
9  |   |   |—— vendor/      # Composer 依赖
10 |—— deploy(1).bat        # 自动部署脚本
11 |—— start_all.bat        # 启动服务脚本
```

## 八、技术支持

如遇到问题，请检查：

1. PHP 错误日志：`advanced/\*/runtime/logs/app.log`

2. Web 服务器错误日志

3. MySQL 错误日志