

## 运行Teebblog项目

本节课我们把Teebblog项目重新部署一下，Teebblog项目在开发时使用到了PHP8.0版本提供的注解，所以我们要保证PHP的运行环境在8.0版本以上。我这里使用的是8.1.0版本，打开浏览器，打开我们Teebblog项目的仓库。

Teebblog项目仓库地址：  
<https://github.com/teebbstudios/teebblog>

复制仓库地址，在我磁盘的CODE卷上，我们新增一个目录。目录名称我们叫做bundle-dev，我们把Teebblog项目部署到这个目录中。打开控制台，使用`git clone`命令把Teebblog项目下载下来。下载完成后我们使用PhpStorm打开项目目录。首先，我们使用composer来安装一下项目的依赖包。

这里出错了，项目的依赖包对PHP的版本有要求。大于7.4或者8.0.0版本，我们本机装的是8.1.0版本。我们使用`composer update`命令来更新一下依赖包。这得选择yes, yes。我们看到在清除缓存时出错了，这里我们需要修改一下源代码，添加方法的返回值。

```
//src/Controller/Admin/PostCrudController.php

@@ -163,7 +163,7 @@ private function generatePageUrl(string
$crudControllerName, string $actionName)
    ->generateUrl();
    }

-   public static function getSubscribedServices()
+   public static function getSubscribedServices(): array
    {
        return array_merge(parent::getSubscribedServices(), [
            'workflow.blog_publishing' => '?' . WorkflowInterface::class
        ]);
    }
}
```

我们查看一下项目的源代码，这里我们需要设置方法的返回值。然后使用`symfony console cache:clear`重新清除一下缓存，现在项目就没有问题了。

我们来修改一下项目的环境变量。Mysql的连接，我使用的是本机Docker容器提供的环境，用户名是root、密码是123456、端口是本机的3306端口。现在我们需要初始化数据库，我们先来创建数据库。输入`symfony console d:d:c`，现在Teebblog数据库就已经创建好了。我们可以使用命令行的简写。下面我们使用migration文件来创建数据库的表，输入`symfony console d:m:m`。输入yes，然后我们加载一些初始数据fixtures。输入`symfony console d:f:l`，输入yes。

下面我们启动symfony提供的服务器，`symfony serve`。我们打开浏览器访问这个地址<http://127.0.0.1:8000>。这里提示没有找到entrypoints.json文件，这是我们前端的资源文件。我们需要使用`encore`来编译前端的资源文件。

打开新的控制台。首先,我们使用`yarn install`命令,来安装前端依赖的资源包。然后使用`yarn dev`来对前端的资源进行打包。打包完成后 我们刷新页面,现在我们的前台项目就可以访问了。

文章的图片出错了,我们把课程的临时图片放置到public目录中。复制一下图片地址`uploads/images`,在public目录中新增`uploads/images`目录,把0-9图片复制过来,再次刷新项目首页,现在就可以了。

查看项目的源代码,我们为bundle项目的代码创建另外一个分支,`git checkout -b bundle`,现在我们的分支就在bundle分支。关于bundle开发的所有代码将会发布到bundle分支里,如果你需要学习的话 请查看bundle分支的代码。现在我们提交一次代码的修改,修复bug,现在我们把代码提交到bundle分支,现在我们的Teebblog项目就已经部署完成了。

在下节课,我们来创建一个单独的文件夹,用来保存我们bundle项目的代码。

## 在Windows系统上部署时的一些说明

因为录制问题:把windows上课程的部分取消了,下边是php配置文件的设置。

把Teebblog项目在Windows系统上跑起来,需要安装amqp扩展,学习安装php扩展。

安装composer依赖包,如果没有安装php\_amqp扩展会出错,这里教下大家在windows下怎么安装扩展,另外安装一下xdebug、opcache扩展,这两个是Zend模块,需要指定`zend_extension="path/to/file"`全路径。

```
[opcache]
zend_extension="D:\Quan\php-8.0.12\ext\php_opcache.dll"

opcache.enable=1
opcache.enable_cli=1
opcache.memory_consumption=256
opcache.interned_strings_buffer=8
opcache.max_accelerated_files=60000
; jit配置
opcache.jit=function
opcache.jit_buffer_size=256M

; xdebug开启后如果phpstorm不开启监听会影响php速度,需要调试的时候再开启xdebug
[xdebug]
zend_extension="D:\Quan\php-8.0.12\ext\php_xdebug.dll"

xdebug.mode = debug
xdebug.idekey = PHPSTORM
xdebug.cli_color = 0
xdebug.start_with_request = yes
xdebug.log_level = 0
```