

如果要将Symfony项目发布到服务器上,我们可以搜索`symfony deploy`,查看symfony官方文档。我们往下看,文档中有发布项目的流程,首先我们要在项目中检查运行环境。我们可以复制这个composer命令行。

```
composer require symfony/requirements-checker
```

回到项目,打开控制台粘贴,检查一下当前的环境。当安装完这个包之后,会自动的检查环境,现在我们这里显示OK,运行环境是没有问题的。如果有问题的话,你需要调整你的运行环境。

回到文档继续往下看,然后就是配置我们系统的环境变量。回到项目,我们当前的环境变量是保存在`.env`文件中的。我们可以使用命令行将`.env`文件中的环境变量,保存到一个PHP文件中来提升性能。

```
composer dump-env prod
```

查看文档,如果你要部署到生产环境上,你就直接复制这个命令行,否则的话你需要设置运行环境。复制,打开控制台粘贴。我这里仍然部署到开发环境了,我们看到在项目根目录下新增加了一个文件`.env.local.php`文件。

在文件中就将我们`.env`文件中配置的环境变量直接进行了返回,省去了读取`.env`文件的过程。查看文档,继续往下看,另外就是我们需要安装项目所依赖的包。如果你是在生产环境下部署,你需要添加这个选项`--no-dev`。

这样的话在`composer.json`文件中,在`require-dev`下配置的所有包就不会进行安装。回到浏览器,然后使用这个选项会重新生成`autoloader`文件来优化性能。

复制命令行,打开控制台粘贴,我这里是部署到开发环境的,就删掉这个配置。点击回车,我这里已经安装过了,就不会再次安装。

再次查看文档,然后就是清除缓存,前面这个设置就是设置当前的运行环境,我这里就直接使用后面的命令行清除缓存了。粘贴,现在开发环境下的缓存就已经清除完成了,关闭控制台。你可能需要将项目连接到你自己的数据库上,那么你就需要修改`.env`文件了,每次修改完之后,你都要使用命令行将环境变量保存到PHP文件中。

再次打开控制台,我们当前的资源开发服务器仍然在运行状态,我们可以把webpack资源打包输出一下。打开`package.json`文件,如果是生产环境,我们可以使用`yarn encore production --progress`来将资源文件进行打包输出。

我这里要部署到开发环境上,就使用了`encore dev`了,暂停开发服务器,输入`yarn encore dev`。打包完成后,我们的资源文件就会保存到`public/build`目录中。

然后就是对我们的服务器进行配置了,我的电脑上装的是Apache,你也可以使用nginx。回到浏览器,我们搜索`symfony server`,我们往下看,这里有个配置web服务器的文档。我们点击。

我的本机装的是Apache,你也可以使用nginx,Apache下有几种配置方式,我这里将使用第一种方式,为了节省时间,我这里已经配置好了。

```
<VirtualHost *:80>
    ServerName teebblog.cc

    ServerAdmin admin@teebblog.cc
    DocumentRoot /Users/quanweiwei/Repository/teebblog/public

    <Directory /Users/quanweiwei/Repository/teebblog/public>
        AllowOverride All
        Require all granted

        FallbackResource /index.php
    </Directory>

    ErrorLog /private/var/log/apache2/teebblog-cc.error.log
    CustomLog /private/var/log/apache2/teebblog-cc.access.log common
</VirtualHost>
```

在Apache的配置文件中，配置项里面DocumentRoot和Directory两个设置要设置到项目的public目录中。因为我们的index.php文件在public目录中，如果没有意外的话，你就可以访问你的项目了。

现在我们来总结一下Symfony项目的开发关键，在项目中我们所编写的类大部分都是服务类。在src下，我们编写的服务类。默认情况下，Symfony会自动的配置到容器中。你可以在服务类的构造方法中或者在controller方法中注入服务类对象。

然后直接使用这些对象，然后我们还学习了事件系统，通过事件系统我们可以最大限度的解耦合代码。还可以在处理请求的过程中，使用事件系统来对请求进行不同的处理。

另外Symfony还提供了很多组件，这些组件都是通过在Symfony容器中提供服务类，然后实现不同的功能。如果你感兴趣的话，你可以查看源码，只要了解它们服务类的功能，然后在需要使用的地方注入它们，就可以完成项目的开发。

这就是我对Symfony的一个个人总结，希望能够帮助到你。现在我们《Symfony5全面开发》视频教程，就结束了。如果在学习课程的过程中出现了问题，欢迎与我交流。

我们在下套课程再见，拜拜！