

我这里准备了博客系统前台的模板，我们查看一下模板的首页，首页的顶部是站点的logo和菜单，下边左边是文章的列表，右边是边栏，在底部是版权信息。然后我们再来查看一下文章的详情页，详情页的页面结构和首页一样，左边是文章的详情，在文章详情下边是评论框和评论列表，我已经将模板文件夹放置在项目的 `course` 目录下了。你可以在 `course` 目录下看到完整的模板文件。

在将模板集成到我们项目之前，先来了解一下webpack，打开浏览器搜索webpack介绍。webpack是一个前端资源加载和打包工具，可以将用到的一些静态资源，比如javascript、CSS或者图片等资源进行统一的管理以及打包发布，现在有很多前端的开发者都使用webpack来对前端资源进行打包。当然如果你对前端有兴趣的话，我建议你学习一下webpack。

Symfony也提供了一个包，可以让我们的Symfony项目快速的使用webpack，课程过程中我们不会详细讲解webpack，但是我们使用Symfony提供的一个包，对webpack进行了最简单的配置，我们打开 `symfony.sh` 这个网站，搜索 `encore`，`webpack-encore-bundle` 是对webpack的一个封装 我们可以在项目中使用 `encore` 提供的一些方法，对webpack进行配置。

复制别名回到项目，我们通过 `webpack-encore` 别名来安装这个包，安装完成后，我们查看一下flex组件添加的一些配置，首先它在根目录中添加了 `package.json` 文件，然后是 `webpack.config.js` 文件。

在 `webpack.config.js` 文件中首先引入了 `encore` 库，然后使用Encore的一些方法对webpack进行了一些配置，比如它设置public目录下的build目录是资源文件最终的输出目录。然后使用 `addEntry()` 方法添加了一个入口配置，我们再看项目目录，在目录中添加一个assets目录，在assets目录下我们就看到了app.js这个文件，打开app.js，我们可以通过import关键字，可以将所有的资源引入到项目中，最终项目的资源文件会统一打包到public下的build目录中。

我们打开控制台，看看刚刚命令行的提示。它提示我们运行 `yarn install` 来安装包，flex组件在项目的根目录下创建了package.json文件 使用 `yarn install` 命令，将会安装package.json文件中的所有依赖项，然后让我们取消 `base.html.twig` 中的注释。再使用命令 `yarn encore dev-server` 来启动开发服务器，开发服务器是用来自动的，对项目的资源文件进行打包的，我们打开项目的course目录 course目录中就是我们的模板文件目录。

模板文件目录中有个package.json，我们打开package.json，我们的模板文件使用了bootstrap和jquery，我们复制依赖项，拷贝粘贴到我们项目根目录下的package.json文件中。回到控制台，我们使用 `yarn install` 命令来安装项目的依赖。

当然package.json文件中的依赖项，你也可以使用 `npm` 来安装，但是你不能使用 `npm` 和 `yarn` 来回切换着安装，这样会丢失包，现在包就已经安装完成了。我们打开 `base.html.twig` 模板文件，在模板文件中，第9行和第13行，这里有两个encore方法，encore方法分别引入了encore入口 `app` 的样式文件和JS文件。

我们取消代码的注释，回到浏览器，打开我们项目的主页，啊，报错了。我们还没有运行开发服务器，复制 `yarn encore dev-server` 这个命令行，粘贴，现在Webpack Encore，就对我们的资源进行了打包输出，回到浏览器再次刷新。现在我们界面的样式就自动的修改了。

再次回到项目，我们打开app.js文件，在第9行它引入了styles目录下的app.css文件。我们不想使用CSS文件，我们使用SCSS文件来编写样式代码，同样的修改文件名，现在控制台提示我们出错了，因为webpack没有合适的加载器来加载SCSS文件。

我们需要在encore中启用 `saas-loader`，然后安装 `sass-loader` 库，打开 `webpack.config.js` 文件，下拉。我们启用 `SassLoader`，然后复制这段命令行，打开新的控制台，在控制台中粘贴命令行，现在仍然出

错。我们已经启用了SassLoader，也安装了sass-loader库，现在仍然出错，我们需要重启一下资源服务器。

在新的控制台中，我们输入`yarn encore dev-server`，现在资源文件就已经构建成功了，在每次修改webpack.config.js文件后，我们都要启动一下资源开发服务器，然后回到浏览器刷新检查源代码，SCSS文件最终编译成了CSS文件。

再次回到项目，我们在项目的package.json文件中引入了bootstrap库和jquery库，但是app.js文件中并没有使用这两个库，我们需要在app.js文件中引入这两个库，使用`import`关键字，我们引入bootstrap库后，编译服务器报错了。bootstrap库需要使用到popper.js这个库，我们安装一下这个库，打开新的控制台输入`yarn add`，复制库的名称，现在库就安好了，编译成功了。

我们回到浏览器刷新，没有什么问题。在下节课，我们把模板的首页和详情页添加到项目的模板目录中。