# Angular 实战（下）

# 服务器搭建

## 通过node搭建后端服务

# 改造原有项目

# 添加搜索功能

共享数据流对象的情况下，我们有两种方式：

1、eventEmitter。（@output 发射器）

# 最佳实践（下半节课）

进入到我们Angular课程学习的最后的部分了，细心的同学会发现，无论是代码文件的目录结构、代码组织还是文件或类的命名，我们都遵循了一定的规范，这使得项目代码看上去很整洁，也便于查找，可读性和可维护性非常好，所以良好的编码习惯和编码规范是写出让人赏心悦目的代码的前提。接下来我会根据项目的开发经验并结合Angular官方提供的风格指南，总结了一些Angular技术开发的最佳实践。

Angular官方风格指南： <https://www.angular.cn/guide/styleguide>

## 单一职责

Angular是基于Web Components思想设计而来，所有核心概念包括组件、服务、指令等都是遵循单一职责原则的，这是最佳实践中很重要的一条，它有助于代码保持良好的可读性和可维护性，而且极大的方便单元测试。

1. 单一文件： 指在一个文件中，应该只定义一个组件、服务或指令等。一个文件的代码量不应该超过400行。
2. 简单函数：指定义的函数尽量功能简单、目的明确、职责单一。通常代码不超过75行。

## 命名约定

关于命名约定是一个老生常谈的问题，但是良好的命名规范确实能起到见名知意的效果，能使维护代码的开发者仅仅在只通览代码文件结构或名称的情况下就能大致了解整个项目。

1. 命名风格统一：指保持命名方式和命名模式一致。 在Angular中通常采用feature.type.ts的命名模式，即先描述特性，再描述类型。
2. 名称格式保持一致：对于文件的名称，通常使用“.” 号来分割特性和类型，而对于特性名称由多个英文单词组成的情况，是用“–” 短横杆来做分隔符的。
3. 选择器命名约定：对于组件通常采用是“烤肉串”命名形式。app-component其他指令等命名“[myUnless]”首字母小写的驼峰命名形式。
4. 测试命名约定。.spec.ts
5. 命名的前缀。 User.component.ts，admin-user.conponent.ts
6. 启动文件的命名约定。 通常以main.ts做为整个项目的入口文件。让代码阅读者一眼便能找到程序的入口。（JAVA的启动方法public static void main()）

## 编码约定

编码约定是指具体代码中一些约定俗成的命名或格式。

1. 类。 类名采用首字符大写的驼峰命名。
2. 常量。 全字母大写并用下划线分隔单词。Typescript用 const定义。
3. 接口。 接口名采用首字母大写的驼峰命名。传统面向对象的语言中会在接口名前面加上大写字母I前缀，但是typescript不建议。
4. 属性和方法。 属性名称和方法名称都是小写开头的驼峰命名。 私有变量或者方法并不需要添加 “\_”下划线前缀。Typescript中有private关键字。
5. Import语句。 使用import导入遇见一组对象时，在括号内侧加上空格，方便阅读。用空行分隔项目内置的依赖和第三方依赖导入。Import 和项目的代码之间用空行风格。

## 应用结构约定

前面提到的最佳实践都是立足于单个文件或单个类，任何项目都是从很小的项目开始，但是随着功能的不断迭代，代码会变得越来越庞大，如果没有对代码结构进行良好地组织，项目很快就会变得难以维护，甚至到了最后不得不进行重构。因此，在项目初始阶段就规划好应用的整体结构是非常有必要的。

1. 所有的项目代码应该存放在app文件夹下。
2. 第三方的代码存放在app目录外的独立文件夹下。
3. 以功能特性创建并命名文件夹。（例如新建组件的时候，组件所在的文件夹）。
4. 将一个模块内的所有共享文件放在一个名为shared（tools）的文件夹中。将公共的组件代码（footer）放在shared-component文件夹下。
5. 将需要懒加载的内容放在一个独立的文件夹中（参见路由）

## 其他

1. 在组件的模板中（html）不要写过多逻辑。
2. 在组件的控制文件中（ts）只写跟模板视图相关的代码，把业务逻辑放到服务中去。
3. tsLint来检测代码质量。