**版本历史**

**变更记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 日期 | 作者 | 版本 | **指导人** | 变更内容 |
| **2017年5月23日** | 杨晓风 | V1.0 |  | 开发规范 |
| **2017年5月30日** | 杨晓风 | V2.0 |  | [添加IDE开发环境规范](#_IDE) |
| **2017年5月31日** | 杨晓风 | V3.0 |  | [添加开发框架引入资源库地址](#_目前框架引入资源库) |
| **2017年6月4日** | 杨晓风 | V4.0 |  | [添加组件封装规范](#_组件封装规范) |
| **2017年6月5日** | 杨晓风 | V5.0 |  | [修改封装组件的命名](#_组件封装规范) |

# IDE

## @angular/cli nodejs



## Typescript



## 开发环境

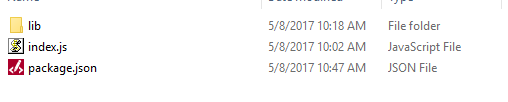
两种选择：Visual Studio Code Webstrom

# 组件封装规范

前提：此规范是针对于负责组件封装人员来说的，否则可忽略此规范

## 目录结构



1. lib文件：此文件存放要封装的所有组件文件
2. index.js：索引文件
3. package.json:发布组件的相关配置

PS：（1）发布包的名称命名为：组件名（烤肉串命名法+小驼峰）

C:\Users\linda\Documents\Jingoal\杨晓风@3625057\Image\EC7116E5-2E63-4A95-9287-FAF4872C8CB7.jpeg

（2）在发布组件的时候，用cnpm安装，用npm发布

（3）地址链接：<http://www.jianshu.com/p/36d3e0e00157>

# 背景

为使项目代码整洁，便于查找，可读性和可维护性好，遵循良好的编码习惯和编码规范是写出令人赏心悦目代码的前提。

# 具体要求

## 单一职责

### 单一文件

1.在开发中，在一个文件中，尽量只定义一个组件、服务或指令等。

2.通常一个文件的代码量不超过400行。

### 简单函数

即定义的函数尽量功能简单、目标明确、职责单一。

通常一个函数的代码不要超过75行

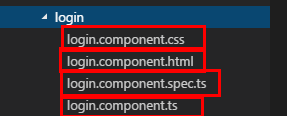
## 命名约定

### 命名风格统一

即保持命名方式和命名模式一致

在Angular中通常采用的命名模式为：feature.type.ts（特性+类型）

Eg:

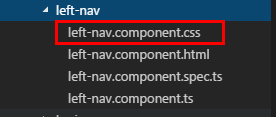


### 名称格式保持一致

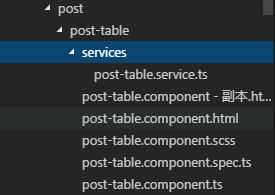
1.文件名称：通常使用“.”点号来分隔特性和类型

特性名称：对于多个英文单词组成的情况，使用“-”作为分隔符

Eg:



2.使用完整的类型名称，不要使用简写

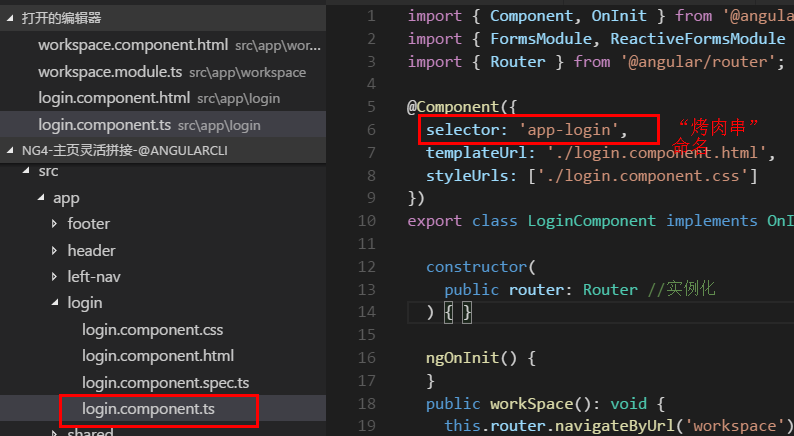


上图中，“.service”作为服务的类型描述，完整使用“service”,不要使用简写“srv、svc、serv”等形式

### 选择器命名约定

#### 组件选择器（selector）

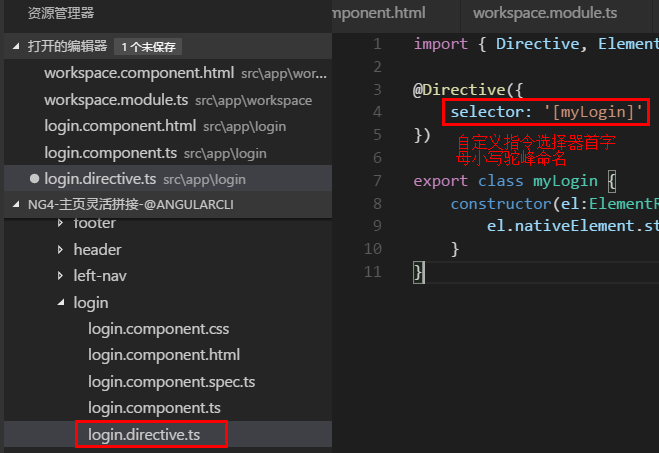
通常采用“烤肉串命名形式”



#### 自定义指令的选择器

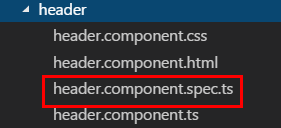
通常采用首字母小写的驼峰命名形式，与组件选择器区分开

Eg:

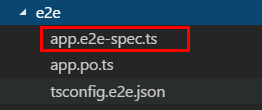


### 测试相关的文件命名约定

单元测试文件命名：与被测试的组件或者服务的名称保持一致，并且添加“.spec”后缀



端对端测试文件命名：与被测试的组件或者服务的名称保持一致，并且添加“.e2e-spec”，区别于单元测试



### 命名的前后缀

#### 组件

根据需要区分组件，依据组件的使用范围适当添加命名前缀来区分

Eg：

普通用户：

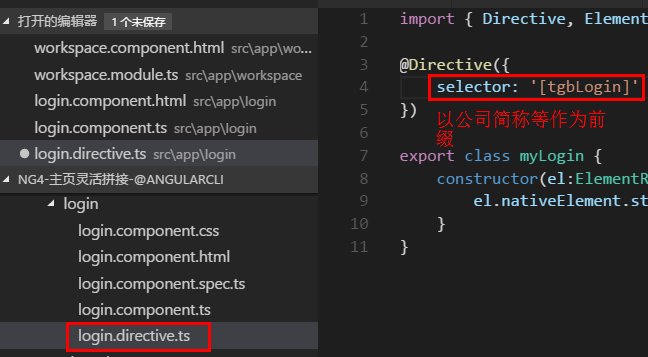
Users.component.ts

管理员：

admin-users.component.ts

#### 自定义指令

若一个组件为通用组件，可被其他模块或者项目复用， 名称常见，此时要用一个有意义的前缀来区分。

Eg：

### 启动文件的命名约定

即程序的入口文件：采用“main.ts”作为文件名称

启动文件应只包含和平台相关的启动代码，逻辑代码放到组件和服务中。

## 编码约定

即具体代码中一些约定俗成的命名或格式

### 类

采用首字母大写的驼峰命名方式



PS:类名切勿与文件命名或者选择器的命名混淆

### 常量

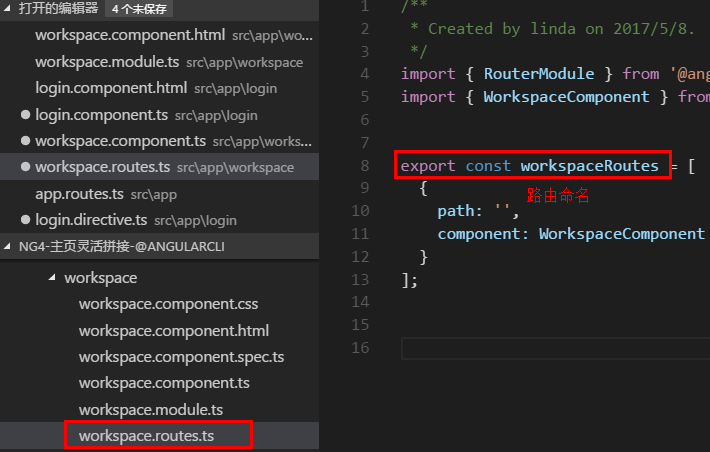
即在TrypeScript中，在整个应用的生命周期中都不会改变的变量，则使用const关键字修饰为常量

命名规则为：全字母大写+下划线分隔单词的方式

Eg：

Export const ARR\_CODE = [‘000300’, ‘000001’] //常量数组

export const URL = ‘angular.io’; //常量字符串



### 接口

采用首字母大写的驼峰命名

### 属性和方法

采用小写字母开头的驼峰命名

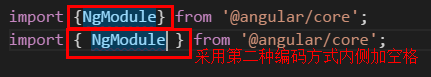


### Import语句

#### 加空格

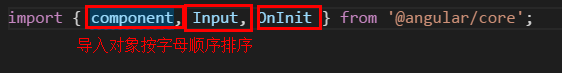
使用import导入一组对象时，在括号内侧加上一个空格，便于阅读

Eg：

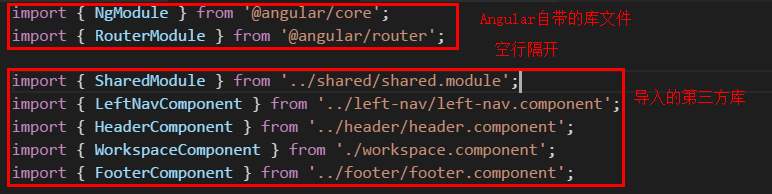


#### 按字母顺序排序

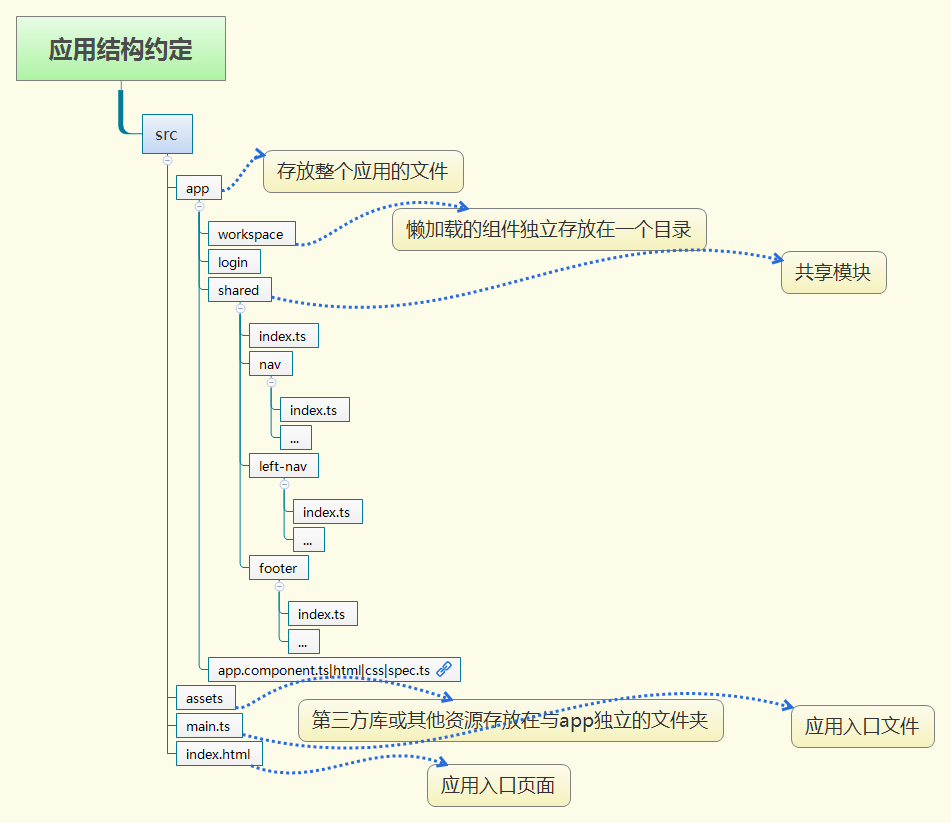
使用import语句导入对象时，导入对象最好按照字母顺序排序



#### 加空行



## 应用结构约定



1.所有的项目代码存放在一个名为app的文件夹中

2.将第三方代码存放在app目录之外的独立文件夹中

3.以功能特性创建并命名文件夹

4.将一个模块内的所有共享文件放在一个名为shared的文件夹中

5.将用于全局布局的组件放在shared目录中，如导航（nav）、页脚（footer）、左侧栏（left-nav）组件

6.使用index.ts文件作为一系列相关模块的统一导出文件

7.将需要懒加载特性的内容放在一个独立的文件夹中

8.使用路由实现组件的懒加载

## 组件相关约定

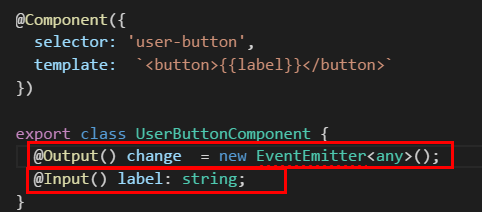
### 组件名称

采用“烤肉串”形式

### 模板与样式文件独立

当模板与样式文件内容超过3行应抽离成独立文件

### 使用@Input和@Output装饰器



### 模板逻辑数据定义在组件类中

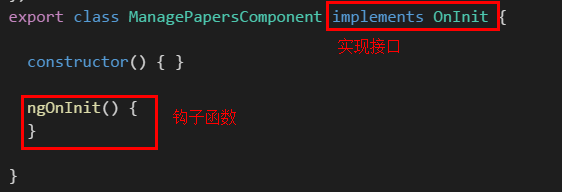
### 逻辑服务代码定义在服务中

### 输出属性不要添加on前缀

事件名称不要添加on前缀，on前缀在事件的处理方法上添加

### 实现接口来添加生命周期钩子

组件初始化时，添加钩子函数

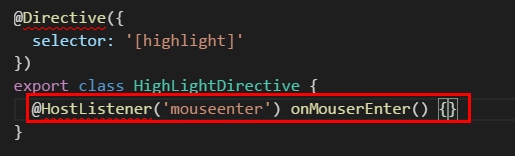


## 指令相关约定

### 使用指令增强已有元素的功能

### 使用Host装饰器替代host属性

使用@HostListener和@HostBinding装饰器代替@Component和@Directive装饰器的元数据中的host属性



## 服务相关约定

1. 在同一个注入器中的服务是单例的，通常用来共享数据或者共用函数
2. 服务遵循单一职责原则
3. 在服务呗共享的组件的最顶层组件注入器中注入服务
4. 使用@Injectable类装饰器

Eg:



1. 独立数据服务

## 开发小技巧

### 用dos命令直接创建组件、指令、管道、服务、类等

Scaffold 
Component 
Directive 
Pipe 
Service 
Class 
Interface 
Enum 
Module 
usage 
ng g component my-new-component 
ng g directive my-new-directive 
ng g pipe my-new-pipe 
ng g service my-new-service 
ng g class my-new-class 
ng g interface my-new-interface 
ng g enum my-new-enum 
ng g module my-module 

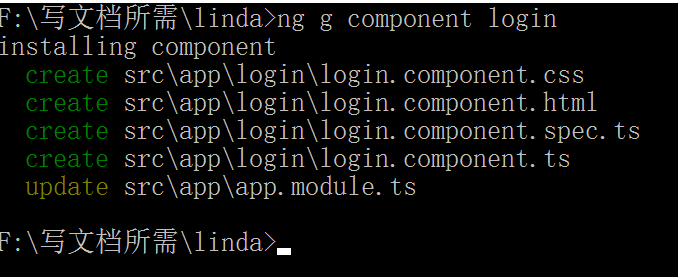
PS：路由文件需要自己手动创建，没有命令直接创建。

已创建组件为例：

#### 创建组件

##### 定位目录

##### 用命令创建



PS：用命令创建组件时，会创建出组件类的完整代码，包括Css\html\spec.ts\ts

需要注意的是，用命令创建时，会自动将新创建的组件添加到根组件的模块里，如果新创建的组件不需要首页加载时呈现，就需要手动将根组件模块里的引用删掉。

### 安装依赖包

在具体业务实现时，如需要安装依赖包，需要在安装命令加入“—save-dev”或者“--save”,这样安装的依赖包名称会自动加载在package.json文件中，也可方便查看，你所安装的依赖包与其他人是否有冲突，避免依赖包的冗余安装

### 目前框架引入资源库

#### Bootstrap

地址链接：<http://valor-software.github.io/ngx-bootstrap/>

#### Font-awesome

地址链接：<http://fontawesome.io/icons/>

### 使用TypeDoc帮助生成文档

### 使用Snippets代码片段工具

在Visual Studio Code中，输入ng2-component在提示项中选择Angular2 component snippet并点击回车，编辑器就自动输入了一个完整组件类的代码