1. 配置翻墙网络，用于下载angular的cli

Npm config set proxy=””设置代理

Npm config set delete proxy删除代理

1. 创建项目，初始化angular2的项目

Npm install -g angular-cli下载命令行

（上面的命令不行 npm install -g @angular/cli）

Ng --version 看版本信息

Ng help 看有些什么命令

Ng new my-project创建新项目

Npm install 创建package依赖的所有包

（ng g c（就是指你想生成的） User 会自动生成一些模块）

Ng serve启动项目

（为了让包变得最小 ng serve --prod --aot）

Ng build 编译最终生成的东西

（ng build --prod --aot）

删除node\_modules文件夹的方式

npm **install** rimraf -g

rimraf node\_modules

# 正式angular2的内容

1. 建项目需要的第一个用于但也应用的内容

Router

1. 需要在index页面引入 <base href=”/”> 初始化路由
2. 建议外部引用路由文件，增加文件的可读性；写法

Const routes = [

{

path:””, /\*不写就用重定向\*/

redirectTo:”路由名字”，/\*默认页面初始化显示的路由\*/

pathMatch:”full”

},

{

Path:”路由名字”,

Component:””

},

{

Path:”/post/:id”/\*传入参数的方式\*/

}

]

1. 在app模块中引入，需要用到的路由模块并且RouterModule

Import {RouterModule} from ‘@angular/router’

在 imports中引入

RouterModule.*forRoot*(appRoutes)

1. 由于需要引入路由的模板字符串，所以需要一个引入口，在引入口用

<router-outlet></router-outlet>

1. 路由中的跳转，两种方式在html和js中的跳转（这里并且用要传参的方式）

Html:

<标签 [routerLink]=”[‘/post’,id]”></标签>

Js:

<标签 (click)=””></标签>

Public routeTiao(){

this.router.navigateByUrl(“post”+id);

}

1. 从远端获取数据的http，以及用于数据与逻辑分离所需要的service；
2. 对于http来说，由于它是拥有单独的模块，所以需要在app.module里引入到import中；在对应的service文件中也需要引用对应的内置服务Http，并且这个http返回的是一个rxjs对应的observable（观察者对象）；这时如果想把它转变成toPromise函数转变成promise对象，但是这里需要引入这个操作符，自带的没有；import 'rxjs/add/operator/toPromise';
3. 对于服务service来说，需要在创建页面引入Injectable从@angular/core中，需要用到这个装饰器函数去修饰元字符，这个里面可以添加其他的依赖；

在引用服务的页面，需要加载对应的文件，并且需要在对应组件的装饰器里，加入提供商providers，这里相当于对服务进行内置的实例化；

constructor(private postListService:PostListService) { } （这里就是把实例化到postListService）

1. 对于常见的表单的用法
2. 这个也是单独的模块，需要在module.ts中进行引入相应的模块，才能使用不然会报错；
3. 双向数据绑定的写法 [(ngModel)]=”这里是绑定的变量”
4. 对于在 input上运用 模板自变量的方式，以及在每个angular想运用指令的表单元素上，一定要记得加上对应的 name属性（这个是必须的）；

I)<input type=”text” [(ngModel)]=”user.username” name=”username” #usern/>（这种写法usern是指的是对应的dom对象）

Ii)<input type=”text” [(ngModel)]=”user.username” name=”username” #usern=”ngModel”/>（这里绑定模板变量为ngModel——指的是对应指令的exprotAs属性为ngModel，这个就是对应的angular的对象）

\*\*\*\*\*这里分析两者的区别，是因为在运用上的不同

主要是：运用对应表单元素上的class属性判断他的合法性、是否改变、是否触碰过这几个顺序来进行显示。

第一种：访问这个属性的方式是通过 usern.className.某个属性 来访问

第二种：访问属性是通过 usern.某个属性 来访问

1. 对于特别的装饰器@Input,@Output

@Input:作用是为了 从 父 向 子 ，两个不同的作用域之间传值，在子中获取传送的值，通过这个装饰器 来获取 元数据；

@Output:作用是识别 从 子 到 父，获取在子中发生的 事件 传送 给 父，在 父 中执行相应的操作；