



35分钟掌握Angular核心概念

大漠穷秋 2017-08-18

内容提纲

➤ **第一块内容：集成开发环境@angular/cli**

➤ **第二块内容：Angular三大核心概念**

Component,Module,Route

➤ **第三块内容：Angular架构特色**

依赖注入、数据绑定

➤ **第四块内容：UI库**

PrimeNG,NG-Zorro,Clarity,Angular-Material,Jigsaw,ionic

➤ **第五块内容：参考资源推荐**

第一块：集成开发环境@angular/cli

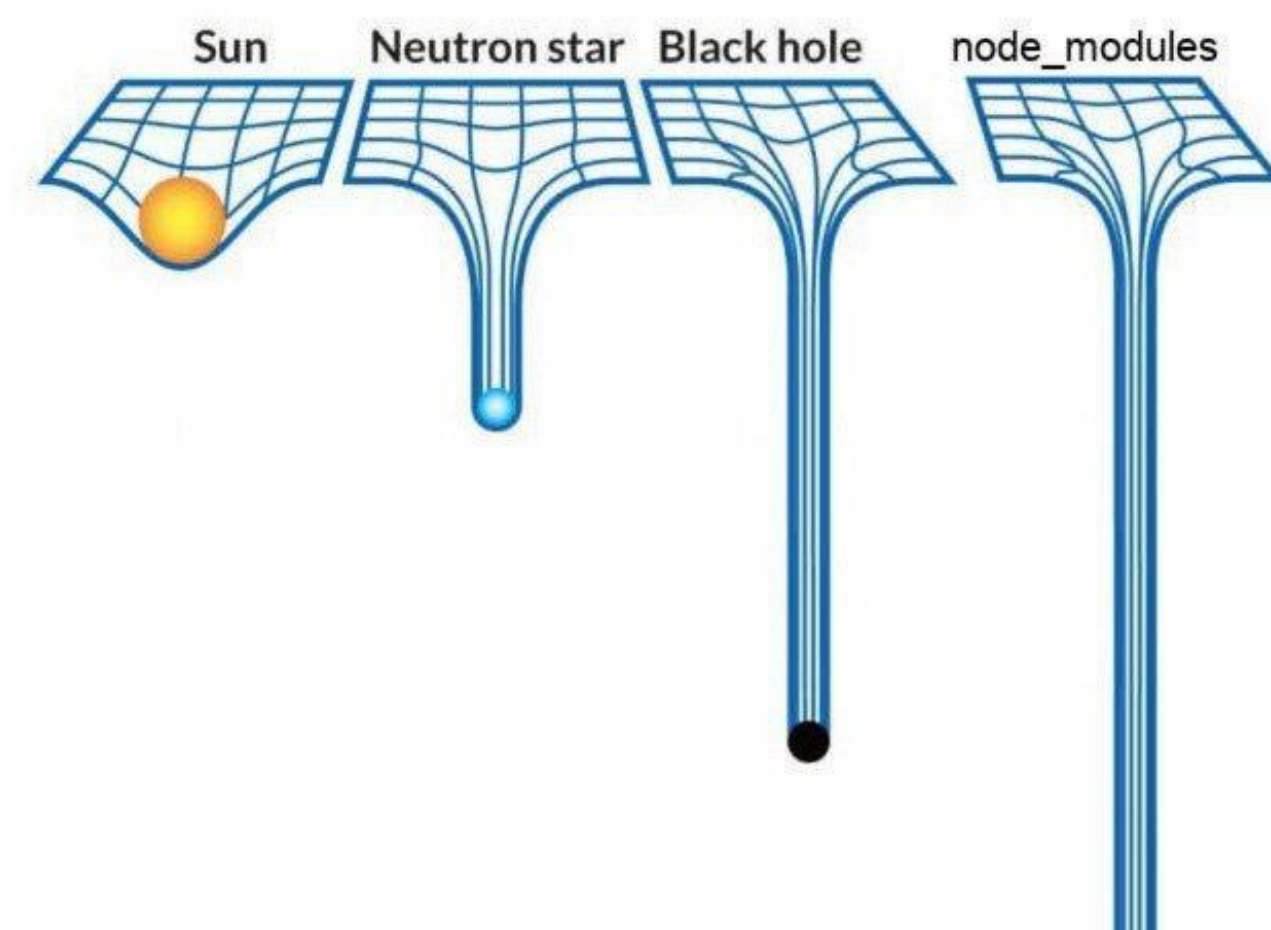


2009年，前端Big Bang!

Tool Chains based on NodeJS



npm : 世界上最糟糕的模块管理器

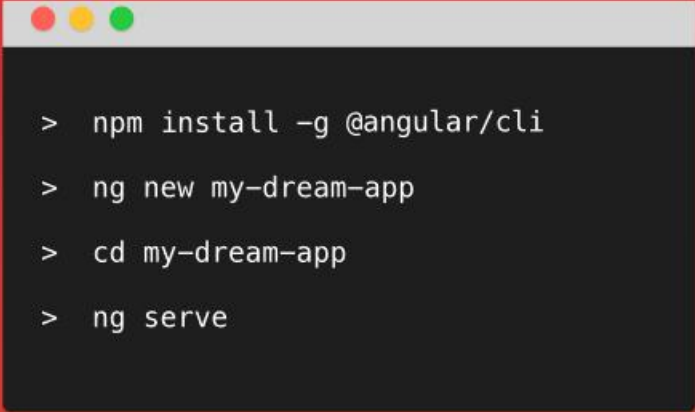


@angular/cli



把所有零散的Node工具都集成起来多好啊！

@angular/cli

A terminal window with a dark background and a light gray title bar containing three colored window control buttons (red, yellow, green). The terminal displays four commands, each preceded by a prompt character '>'.

```
> npm install -g @angular/cli  
> ng new my-dream-app  
> cd my-dream-app  
> ng serve
```

Angular CLI

A command line interface for Angular

GET STARTED

@angular/cli

> npm install -g @angular/cli

> ng --version

> ng help

> ng new my-project1

> ng serve

@angular/cli

> ng generate

cl:class

c:component

d:directive

e:enum

m:module

p:pipe

s:service

命令有简化版本 : ng g c User

@angular/cli

- 在Windows下面，node-gyp这个包依赖于Visual Studio
- node-sass这个node模块被墙掉了，原因不明
- 所以强烈推荐采用cnpm安装

@angular/cli

```
angular-cli
E:\github-my\NiceFish>ng serve --prod --aot
*****
This is a simple server for use in testing or debugging Angular applications locally.
It hasn't been reviewed for security issues.

DON'T USE IT FOR PRODUCTION USE!
*****
** NG Live Development Server is running on http://localhost:4200. **
Hash: 4a9786ce6bclfe4630d6
Time: 50871ms
chunk {0} 0.chunk.js, 0.bundle.map 1.45 MB {1} {3} [rendered]
chunk {1} 1.chunk.js, 1.bundle.map 66.9 kB {0} {3} [rendered]
chunk {2} scripts.bundle.js, scripts.bundle.map (scripts) 2.07 MB {6} [initial] [rendered]
chunk {3} main.bundle.js, main.bundle.map (main) 303 kB {5} [initial] [rendered]
chunk {4} styles.bundle.js, styles.bundle.map (styles) 10.3 kB {6} [initial] [rendered]
chunk {5} vendor.bundle.js, vendor.bundle.map (vendor) 2.01 MB [initial] [rendered]
chunk {6} inline.bundle.js, inline.bundle.map (inline) 0 bytes [entry] [rendered]
webpack: bundle is now VALID.
```

ng serve --prod

注意：最新版的CLI加上--prod参数就自动AOT了，官网上的那篇文档过时了！

@angular/cli

```
angular-cli
WARNING in ./~/@angular/core/src/linker/system_js_ng_module_factory_loader.js
45:15 Critical dependency: the request of a dependency is an expression

WARNING in ./~/@angular/core/src/linker/system_js_ng_module_factory_loader.js
57:15 Critical dependency: the request of a dependency is an expression

ERROR in [default] E:\github\ng2-bootstrap-demo\src\app\user\add-user\add-user.component.spec.ts:8:20
Supplied parameters do not match any signature of call target.
13 10 2016 10:58:21.772:INFO [karma]: Karma v1.2.0 server started at http://localhost:9876/
13 10 2016 10:58:21.773:INFO [launcher]: Launching browser Chrome with unlimited concurrency
13 10 2016 10:58:21.897:INFO [launcher]: Starting browser Chrome
13 10 2016 10:58:24.805:INFO [Chrome 53.0.2785 (Windows 10 0.0.0)]: Connected on socket /#i6GL146TuIVZ1kPKAAAA with id 79091848
Chrome 53.0.2785 (Windows 10 0.0.0) App: Angular2BootstrapSass should create the app FAILED
    'router-outlet' is not a known element:
    1. If 'router-outlet' is an Angular component, then verify that it is part of this module.
    2. If 'router-outlet' is a Web Component then add "CUSTOM_ELEMENTS_SCHEMA" to the '@NgModule.schema' of this component to suppress this message.
    </div>
    <div class="main-container">
      [ERROR ->]<router-outlet></router-outlet>
    </div>
    ): AppComponent@42:4
Error: Template parse errors:
  at TemplateParser.parse (http://localhost:9876/_karma_webpack_/0.bundle.js:7333:19)
  at RuntimeCompiler._compileTemplate (http://localhost:9876/_karma_webpack_/0.bundle.js:15632:51)
  at http://localhost:9876/_karma_webpack_/0.bundle.js:15555:83
  at Set.forEach (native)
  at compile (http://localhost:9876/_karma_webpack_/0.bundle.js:15555:47)
  at RuntimeCompiler._compileComponents (http://localhost:9876/_karma_webpack_/0.bundle.js:15557:13)
  at RuntimeCompiler._compileModuleAndAllComponents (http://localhost:9876/_karma_webpack_/0.bundle.js:15474:37)
  at RuntimeCompiler.compileModuleAndAllComponentsSync (http://localhost:9876/_karma_webpack_/0.bundle.js:15462:21)
  at TestingCompilerImpl.compileModuleAndAllComponentsSync (http://localhost:9876/_karma_webpack_/0.bundle.js:20504:35)
  at TestBed._initIfNeeded (webpack:///E:/github/ng2-bootstrap-demo/@angular/core/bundles/core-testing.umd.js:1059:0 <- src/test.ts:8814:40)
Chrome 53.0.2785 (Windows 10 0.0.0): Executed 1 of 10 (1 FAILED) (0 secs / 0.101 secs)
Chrome 53.0.2785 (Windows 10 0.0.0) App: Angular2BootstrapSass should create the app FAILED
    'router-outlet' is not a known element:
    1. If 'router-outlet' is an Angular component, then verify that it is part of this module.
    2. If 'router-outlet' is a Web Component then add "CUSTOM_ELEMENTS_SCHEMA" to the '@NgModule.schema' of this component to suppress this message.
    </div>
    <div class="main-container">
      [ERROR ->]<router-outlet></router-outlet>
    </div>
    ): AppComponent@42:4
Error: Template parse errors:
  at TemplateParser.parse (http://localhost:9876/_karma_webpack_/0.bundle.js:7333:19)
```

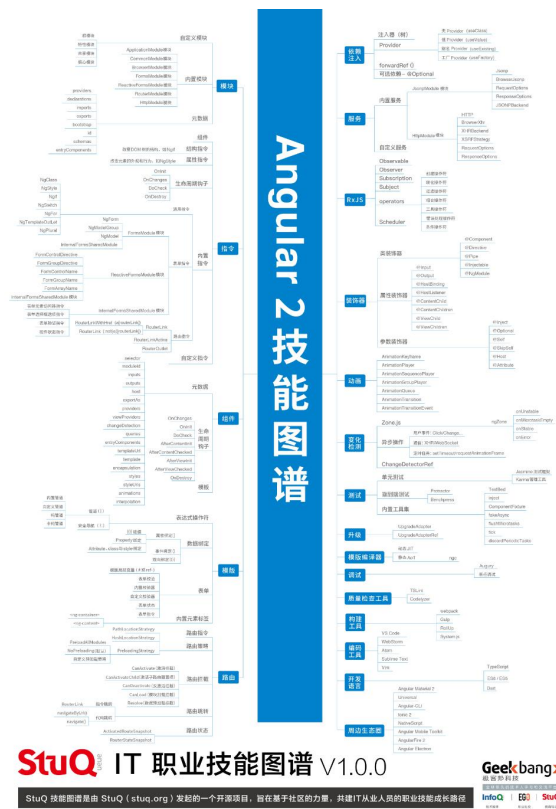
ng test

@angular/cli



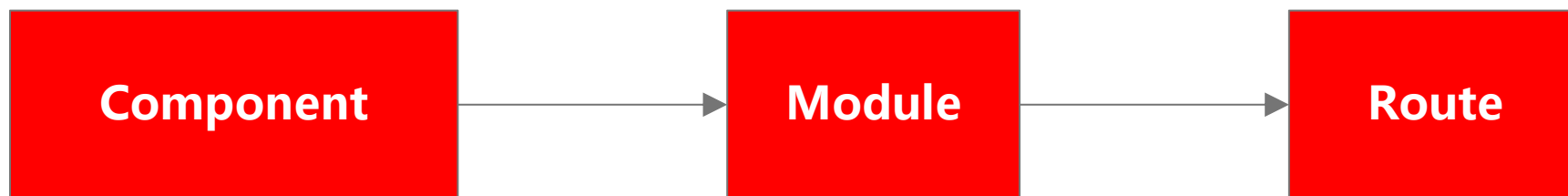
第二块：Angular中的3大核心概念

一张图覆盖Angular所有概念



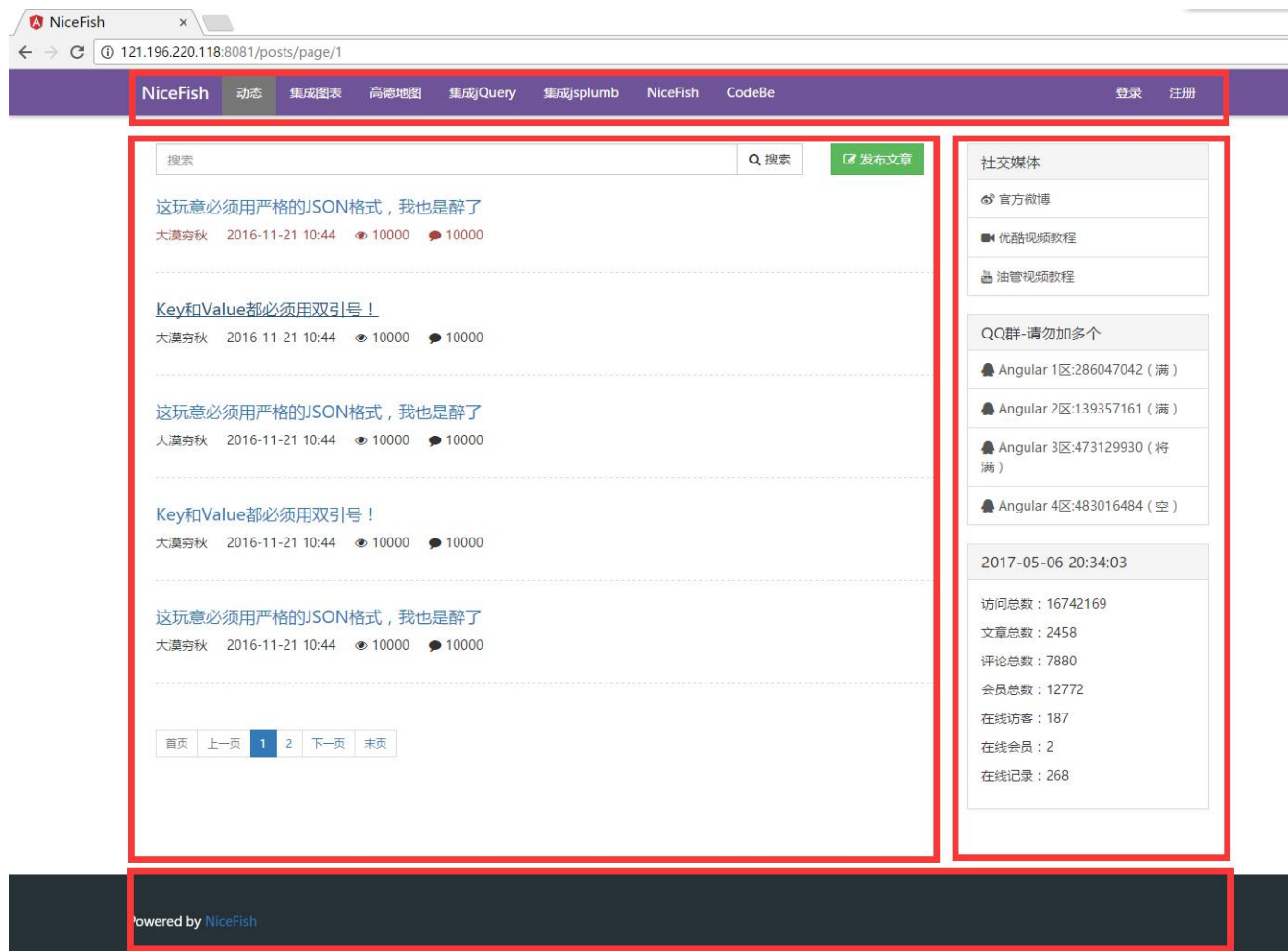
感谢《揭秘Angular2》的汤桂川老师提供如此详细的脑图！

Angular最核心的3个概念



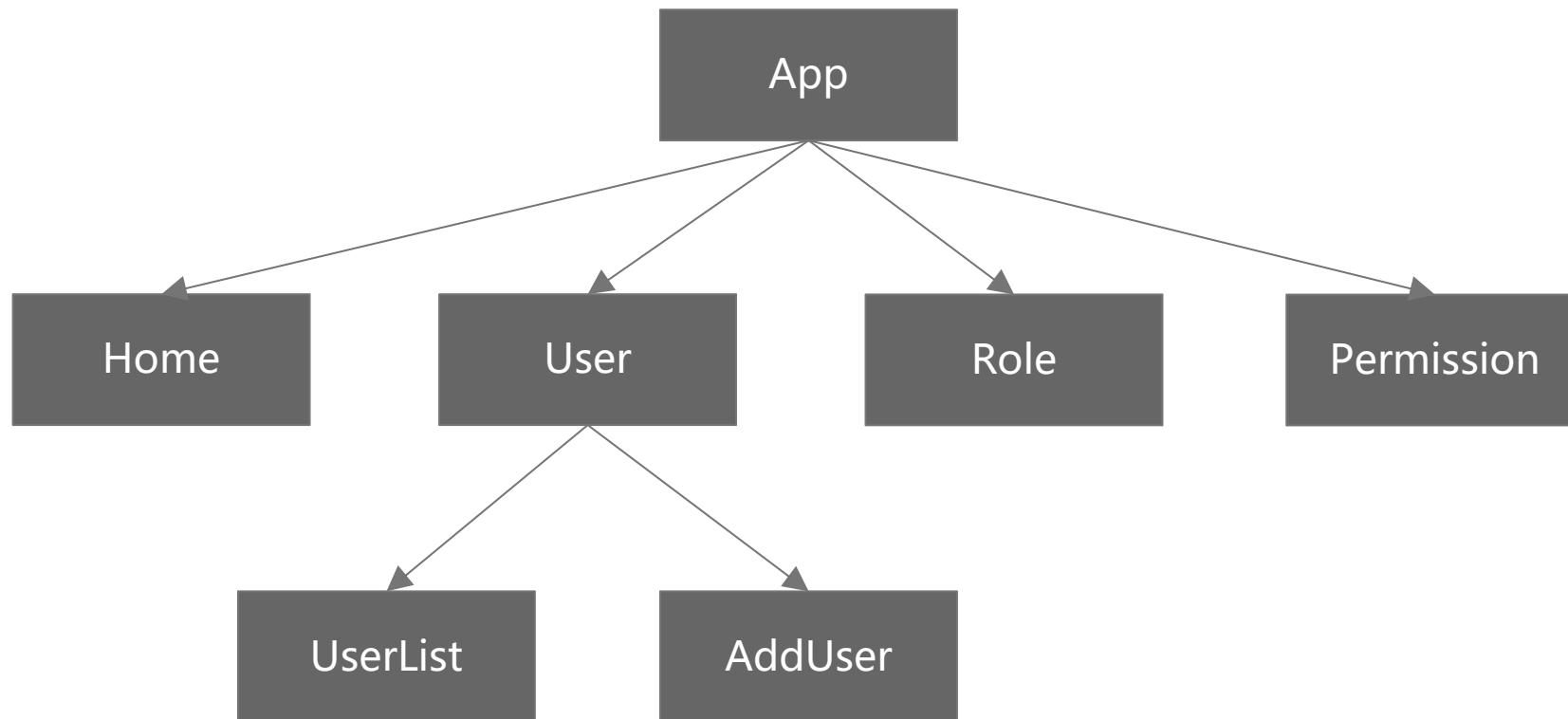
Angular最核心的概念是“组件化”

第一个核心概念：Component



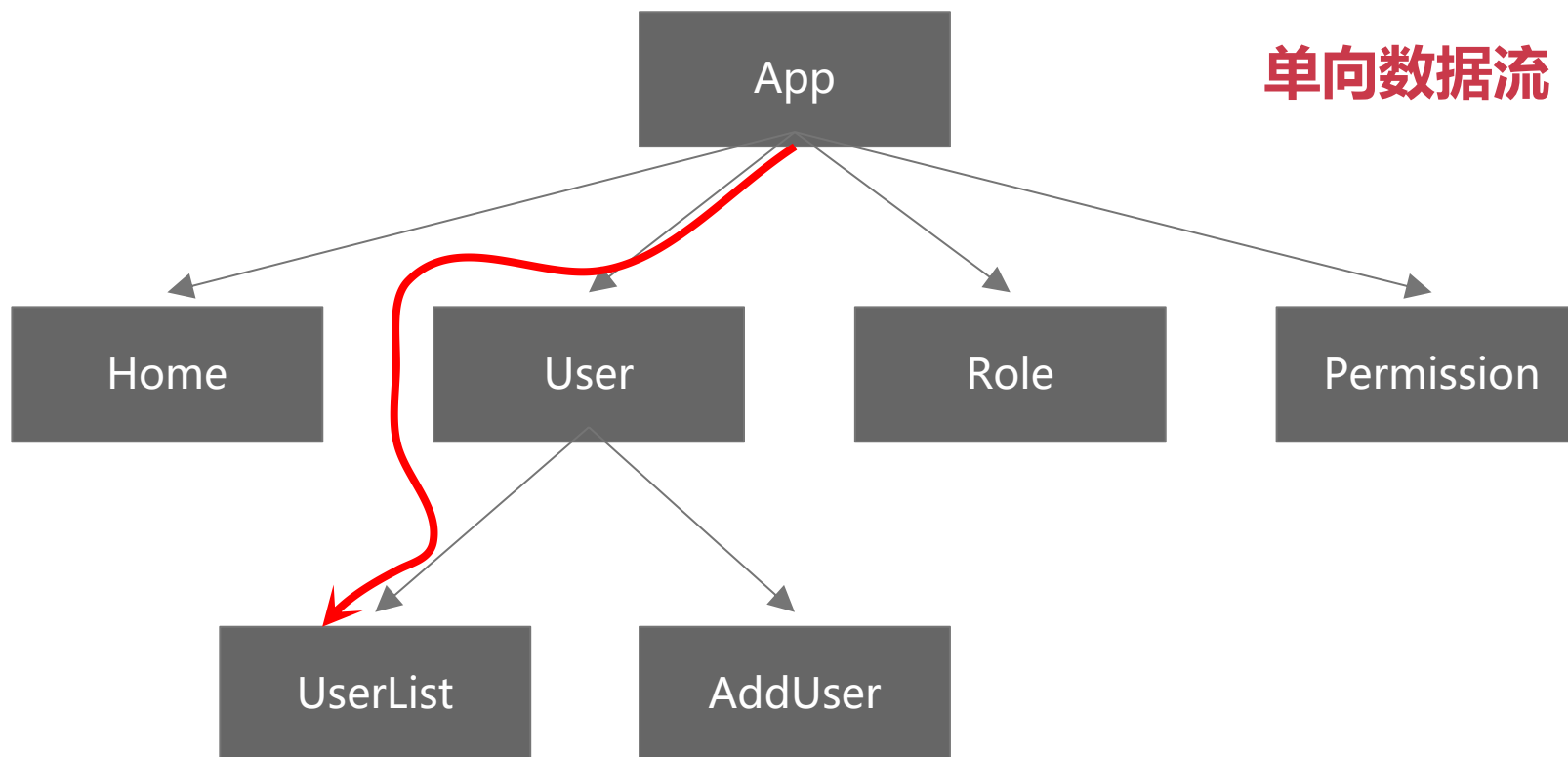
<http://git.oschina.net/mumu-osc/NiceFish>

Component Tree



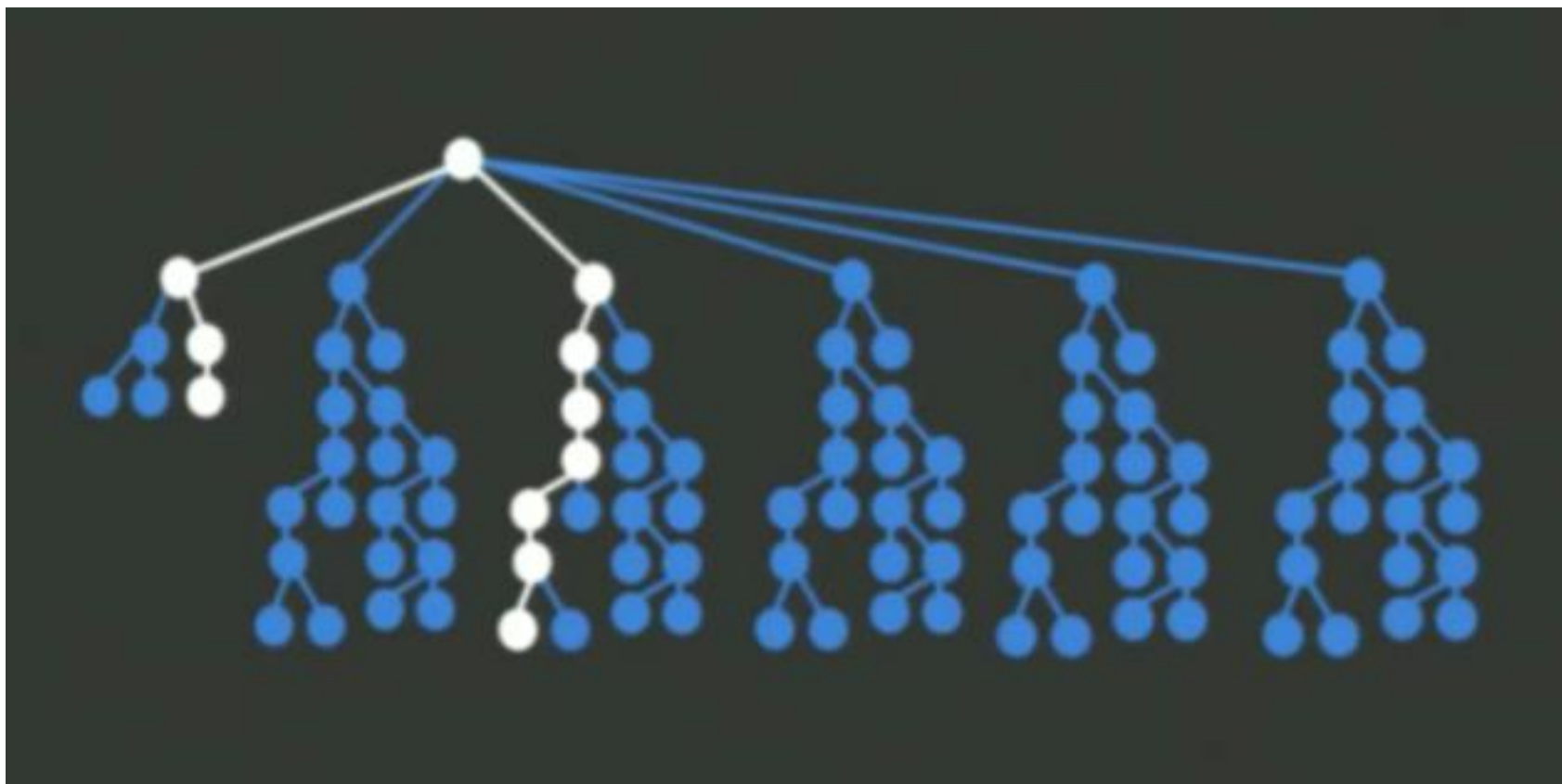
NiceFish实例项目的组件树结构

单向数据流



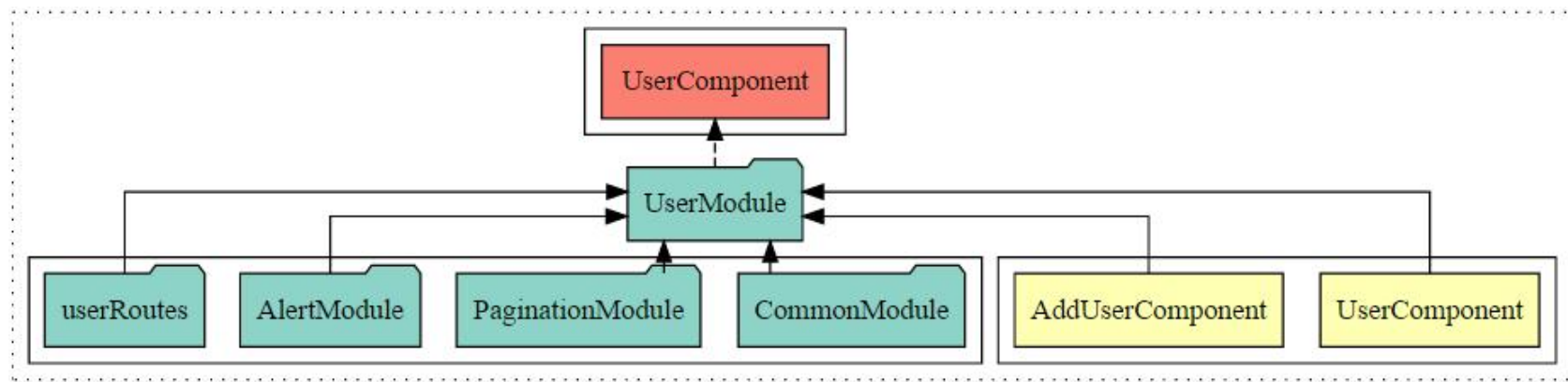
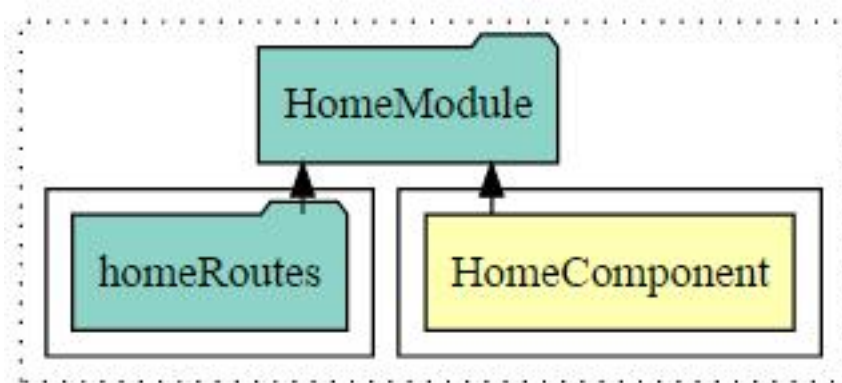
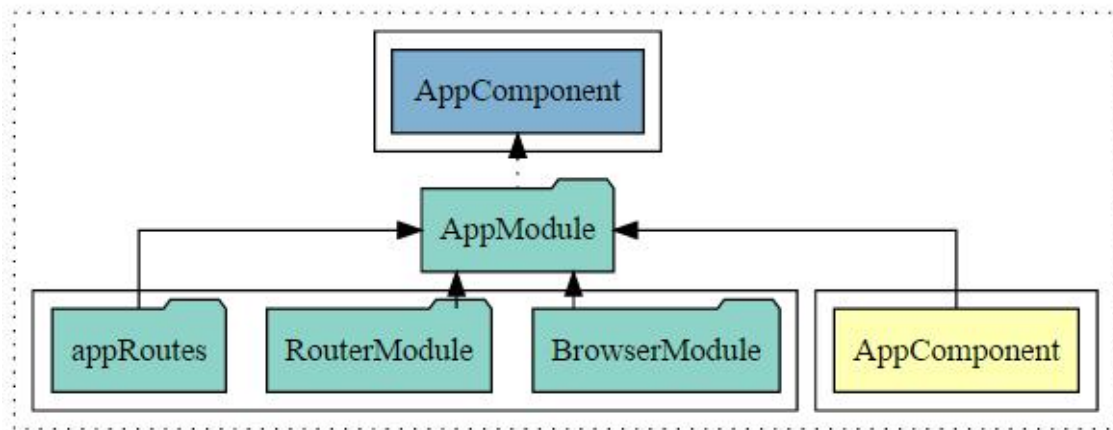
<https://github.com/modern-javascript/angular2-data-flow>

实际项目中的组件树



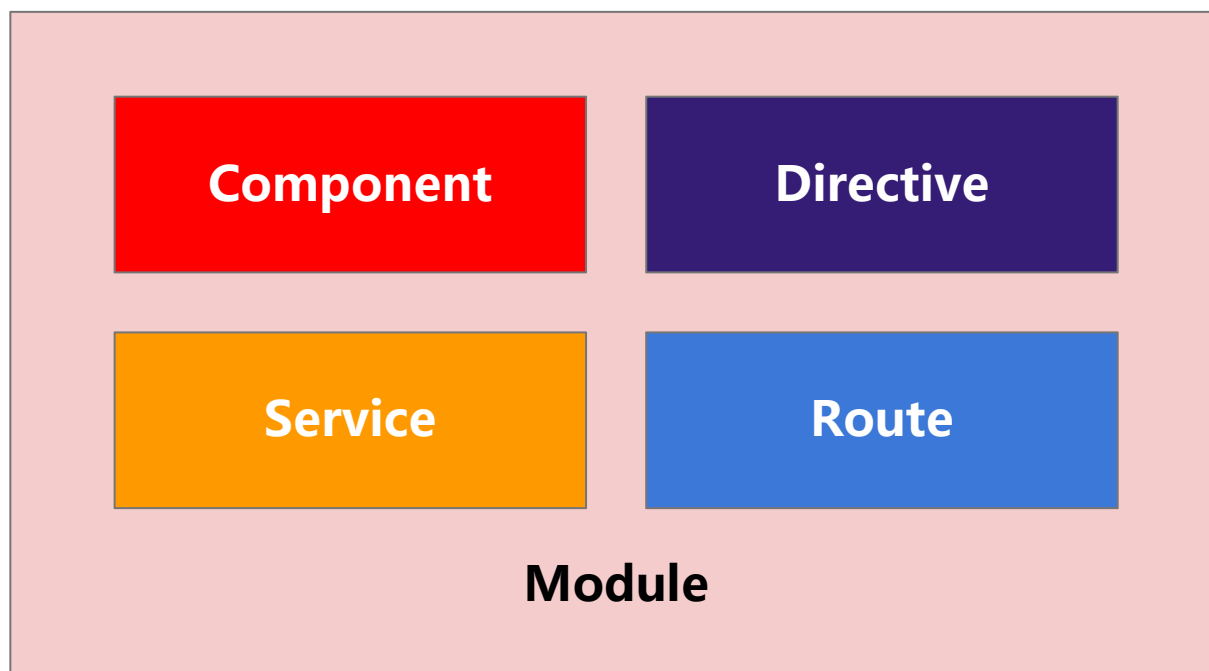
不可变数据类型

组件树生成器



<https://github.com/manekinekko/angular2-dependencies-graph>

第二个核心概念：NgModule



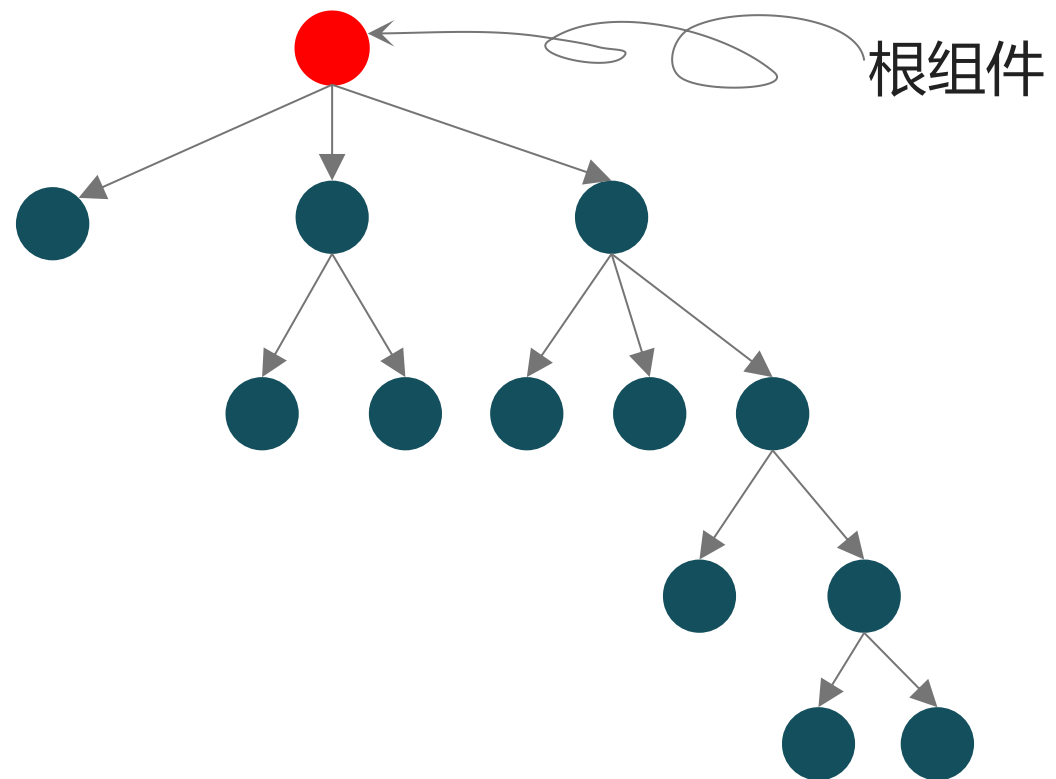
什么是模块？

第二个核心概念：NgModule

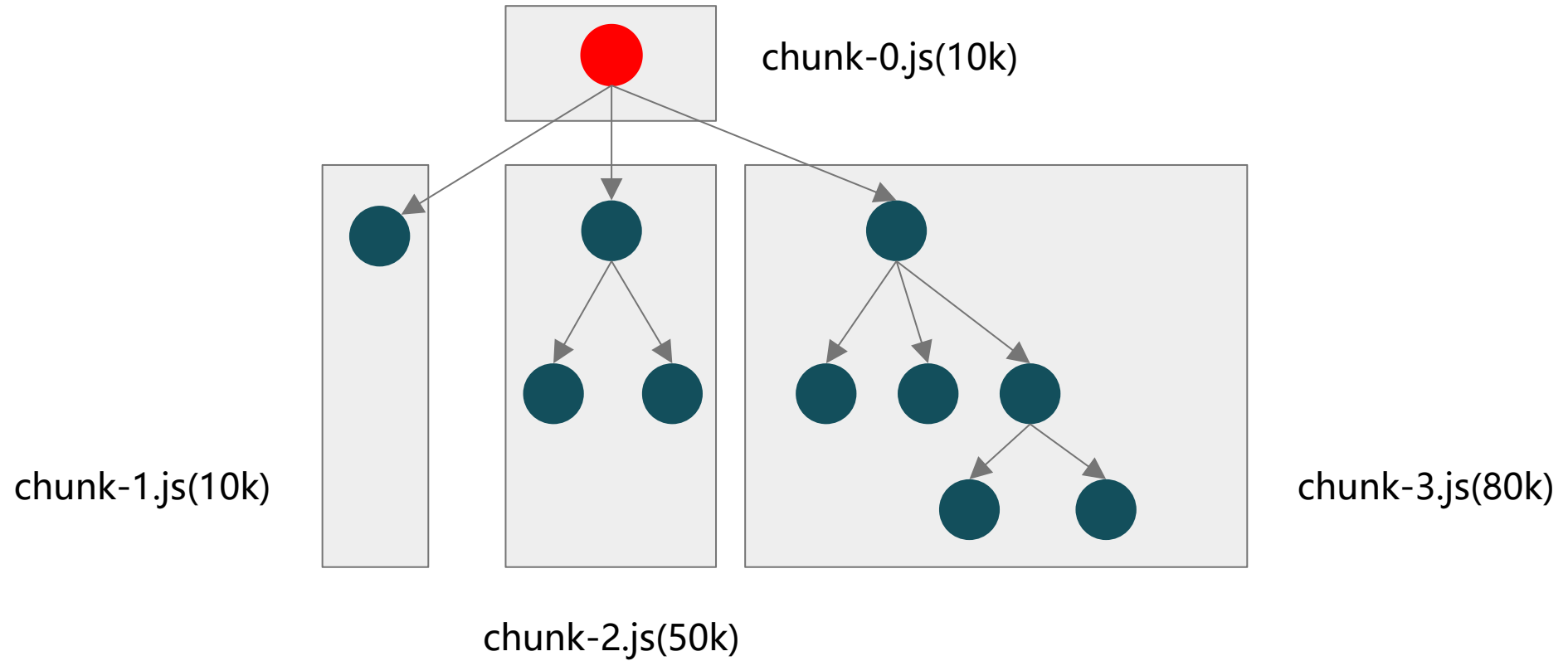
问题：为什么需要NgModule？

<https://angular.cn/docs/ts/latest/cookbook/ngmodule-faq.html>

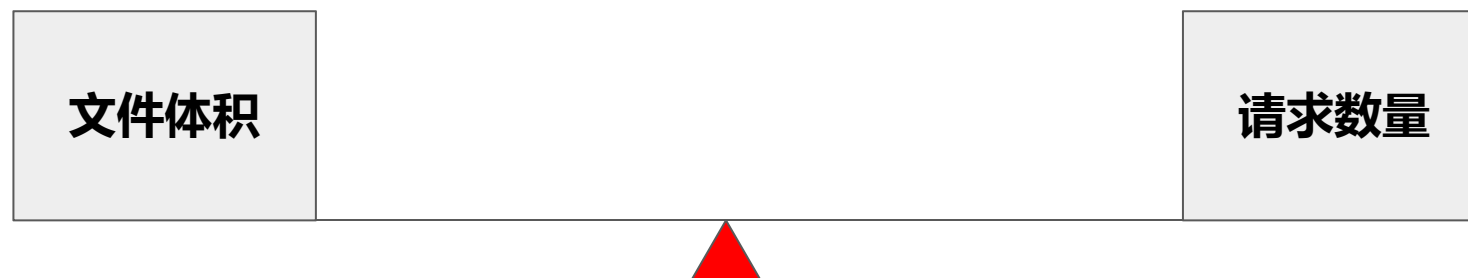
NgModule



NgModule



NgModule



这里需要取得一个平衡

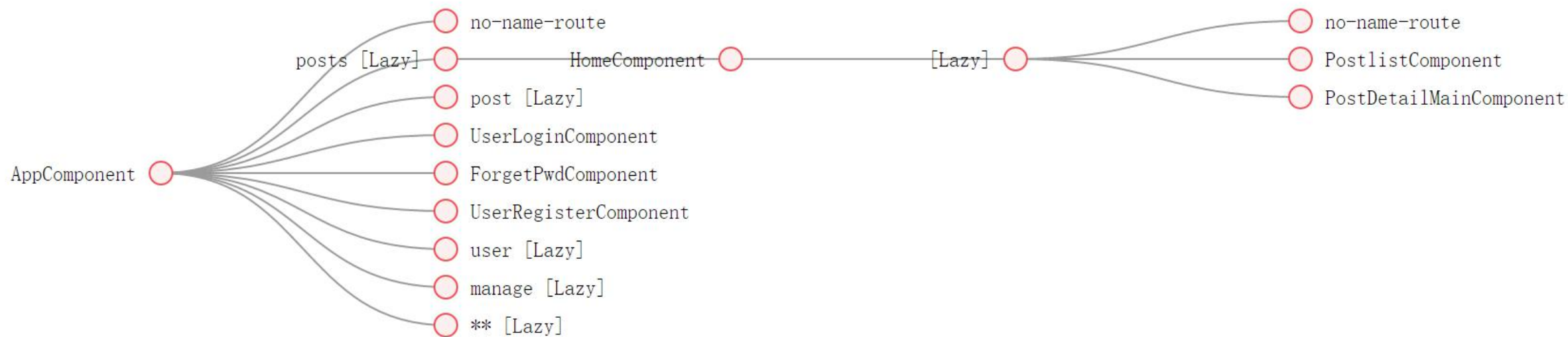
第三个核心概念：Router

问题：前端为什么需要Router？

前端为什么需要Router ?

- 如果没有Router，浏览器的前进后退按钮没法用
- 如果没有Router，你将无法把URL拷贝并分享给你的朋友

Angular应用中路由形成的Tree形结构



Angular路由用法：静态路由

```
home.routes.ts
1 import {RouterModule} from "@angular/router";
2 import {HomeComponent} from "../home.component";
3
4 const homeRoutes= [
5   {path: '', component: HomeComponent}
6 ];
7
8 export default RouterModule.forChild(homeRoutes);
```

Angular路由用法：异步路由

```
app.routes.ts
1 import { RouterModule } from '@angular/router';
2
3 const appRoutes=[
4   {
5     path: '',
6     loadChildren: 'app/home/home.module'
7   }, {
8     path: 'home',
9     loadChildren: 'app/home/home.module'
10  }, {
11    path: 'user',
12    loadChildren: 'app/user/user.module'
13  }, {
14    path: 'role',
15    loadChildren: 'app/role/role.module'
16  }, {
17    path: '**', //fallback router must in the last
18    loadChildren: 'app/home/home.module'
19  }
20 ];
21 export default RouterModule.forRoot(appRoutes);
```

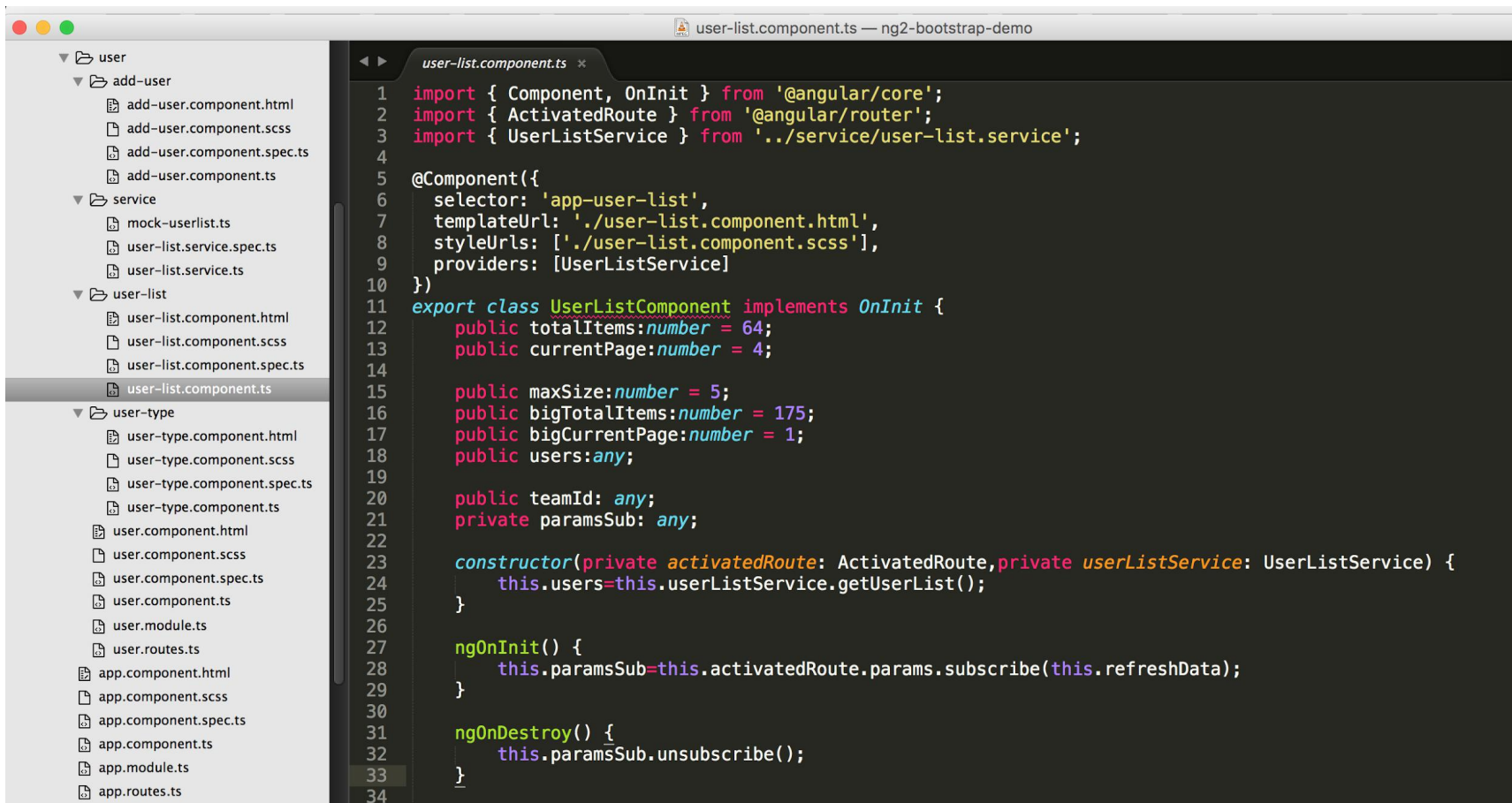
Angular路由用法：路由守卫

```
manage.routes.ts
1 import { ManageMainComponent } from './manage-main/manage-main.component';
2 import { PostTableComponent } from './post-table/post-table.component';
3 import { CommentTableComponent } from './comment-table/comment-table.component';
4 import { UserTableComponent } from './user-table/user-table.component';
5 import { UserProfileComponent } from './user/user-profile/user-profile.component';
6 import { SysParamComponent } from './sys-param/sys-param.component';
7 import { AuthGuard } from './auth-guard';
8
9 export const manageRoutes = [
10   {
11     path: '',
12     component: ManageMainComponent,
13     canActivate: [AuthGuard],
14     children: [
15       { path: '', redirectTo: 'posttable/page/1', pathMatch: 'full' },
16       { path: 'posttable/page/:page', component: PostTableComponent },
17       { path: 'commenttable/page/:page', component: CommentTableComponent },
18       { path: 'usertable/page/:page', component: UserTableComponent },
19       { path: 'usertable/edituser/:userId', component: UserProfileComponent },
20       { path: 'usertable/newuser', component: UserProfileComponent },
21       { path: 'sysparam', component: SysParamComponent },
22       { path: '**', redirectTo: 'posttable/page/1' }
23     ]
24   }
25 ];
```

防止未授权访问

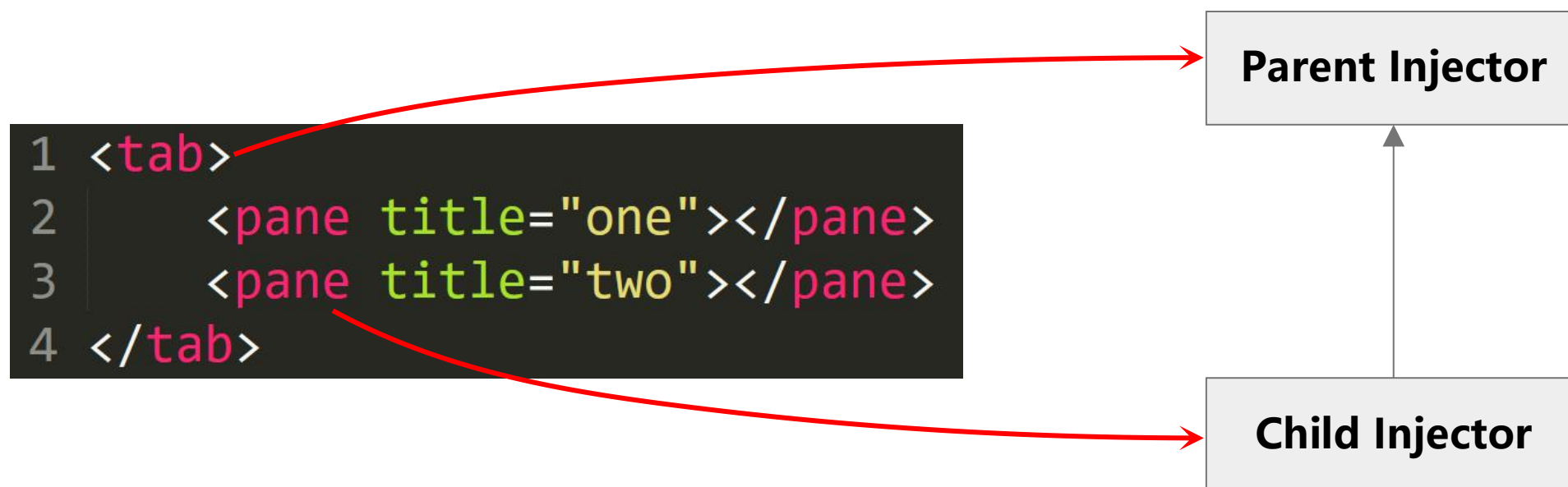
第三块：Angular架构特色

依赖注入



```
1 import { Component, OnInit } from '@angular/core';
2 import { ActivatedRoute } from '@angular/router';
3 import { UserListService } from '../service/user-list.service';
4
5 @Component({
6   selector: 'app-user-list',
7   templateUrl: './user-list.component.html',
8   styleUrls: ['./user-list.component.scss'],
9   providers: [UserListService]
10 })
11 export class UserListComponent implements OnInit {
12   public totalItems:number = 64;
13   public currentPage:number = 4;
14
15   public maxSize:number = 5;
16   public bigTotalItems:number = 175;
17   public bigCurrentPage:number = 1;
18   public users:any;
19
20   public teamId: any;
21   private paramsSub: any;
22
23   constructor(private activatedRoute: ActivatedRoute,private userListService: UserListService) {
24     this.users=this.userListService.getUserList();
25   }
26
27   ngOnInit() {
28     this.paramsSub=this.activatedRoute.params.subscribe(this.refreshData);
29   }
30
31   ngOnDestroy() {
32     this.paramsSub.unsubscribe();
33   }
34 }
```


依赖注入



注射器（Injector）也是一个树型结构

依赖注入

🚫 🔍 <top frame> ▼ ☐ Preserve log

▼ EXCEPTION: No provider for NgForm (ShowError -> NgForm) [browser_adapter.js:110](#)

STACKTRACE: [browser_adapter.js:106](#)

Error: DI Exception [browser_adapter.js:106](#)

- at NoBindingError.BaseException ([lang.js:163](#))
- at NoBindingError.AbstractBindingError ([exceptions.js:48](#))
- at new NoBindingError ([exceptions.js:70](#))
- at Injector.execute.\$__export._throwOrNull ([injector.js:640](#))
- at Injector.execute.\$__export._getByKeySelf ([injector.js:645](#))
- at Injector.execute.\$__export._getByKey ([injector.js:629](#))
- at Injector.execute.\$__export._getByDependency ([injector.js:621](#))
- at Injector.execute.\$__export._instantiate ([injector.js:518](#))
- at Injector.execute.\$__export._new ([injector.js:491](#))
- at InjectorInlineStrategy.execute.\$__export.instantiateBinding ([injector.js:283](#))

ERROR CONTEXT: [browser_adapter.js:106](#)

[browser_adapter.js:106](#)

▼ _Context {element: show-error.ng-binding, componentElement: template-driven-forms, injector: Injector}

- ▶ componentElement: template-driven-forms
- ▶ element: show-error.ng-binding
- ▶ injector: Injector
- ▶ __proto__: _Context

The injector of the element where Angular failed to instantiate a directive

> |

依赖注入

- 每一个HTML标签上面都会有一个注射器实例
- 注射是通过constructor进行的
- @Component是@Injectable的子类

<https://vsavkin.com/dependency-injection-in-angular-1-and-angular-2-d69589979c18#.3ptlx48zf>

<https://my.oschina.net/mumu/blog/775695>

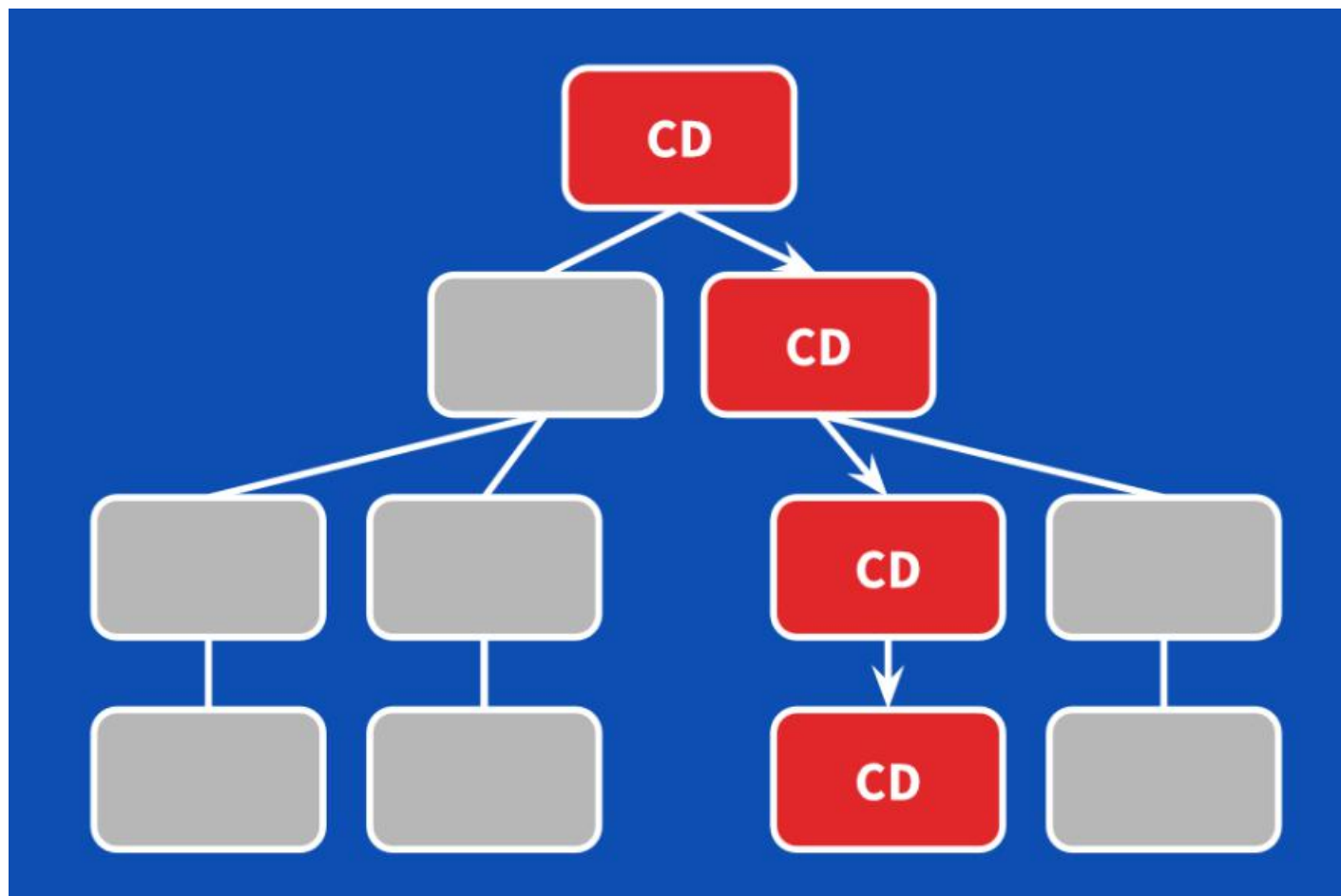
数据绑定



Immutable Data

简而言之，新版本的Angular已经彻底重写了脏检查机制

数据绑定



极其高效的脏检查机制

数据绑定

<http://blog.thoughttram.io/angular/2016/02/22/angular-2-change-detection-explained.html>

<http://teropa.info/blog/2015/03/02/change-and-its-detection-in-javascript-frameworks.html>

<https://vsavkin.com/change-detection-in-angular-2-4f216b855d4c>

第四块：UI库

UI Libraries

- [PrimeNG](#) : 到目前为止最完善的开源免费UI组件库
- [NG-Zorro](#) : 来自阿里云的开源组件库
- [Jigsaw](#) : 来自ZTE中兴通讯的开源组件库
- [Clarity](#) : 来自Vmware的组件库
- [Angular-Material](#) : Angular官方提供的组件库
- [Ionic](#) : 专门为Angular设计的移动端组件库

UI Libraries

The screenshot shows the PrimeNG website interface. On the left is a dark sidebar with a navigation menu containing icons and labels for: Input, Button, Data, Panel, Overlay, Menu, Charts, Messages, Multimedia, DragDrop, and Misc. The main content area has a blue header with the PrimeNG logo and navigation links for Setup, Free Themes, and Forum. Below the header, the PrimeNG title is displayed in red, followed by a description: "PrimeNG is a collection of rich UI components for Angular 2. PrimeNG is a sibling of the popular JavaServer Faces Component Suite, PrimeFaces." It also mentions that all widgets are open source under the Apache License 2.0 and that the library is developed by PrimeTek Informatics. Two prominent buttons are shown: "Download" (yellow) and "View on GitHub" (orange). The right side of the page features a grid of eight feature cards, each with an icon, a title, and a brief description: PRIMEFACES UI (Derived from the mighty PrimeFaces), WIDGETS (60+ Components, Easy to Use, Accessible), PRODUCTIVITY (Simple, Lightweight, Powerful), MOBILE (Responsive, Cross Browser, Touch Optimized), COMMUNITY (Active, Vibrant, Open Source), and THEMES (35+ Free Themes, Premium Themes, Theme Creator Tool).

PrimeNG

PrimeNG is a collection of rich UI components for Angular 2. PrimeNG is a sibling of the popular JavaServer Faces Component Suite, [PrimeFaces](#).

All widgets are open source and free to use under Apache License 2.0, a commercial friendly license.

PrimeNG is developed by [PrimeTek Informatics](#), a company with years of expertise in developing open source UI components. For project news and updates, follow us on [twitter](#).

[Download](#)

[View on GitHub](#)

PRIMEFACES UI
Derived from the mighty PrimeFaces

WIDGETS
60+ Components
Easy to Use
Accessible

PRODUCTIVITY
Simple
Lightweight
Powerful

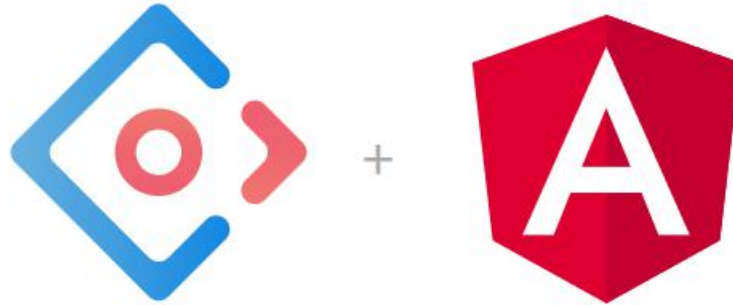
MOBILE
Responsive
Cross Browser
Touch Optimized

COMMUNITY
Active
Vibrant
Open Source

THEMES
35+ Free Themes
Premium Themes
Theme Creator Tool

[PrimeNG](#)

UI Libraries



[NG-Zorro](#)

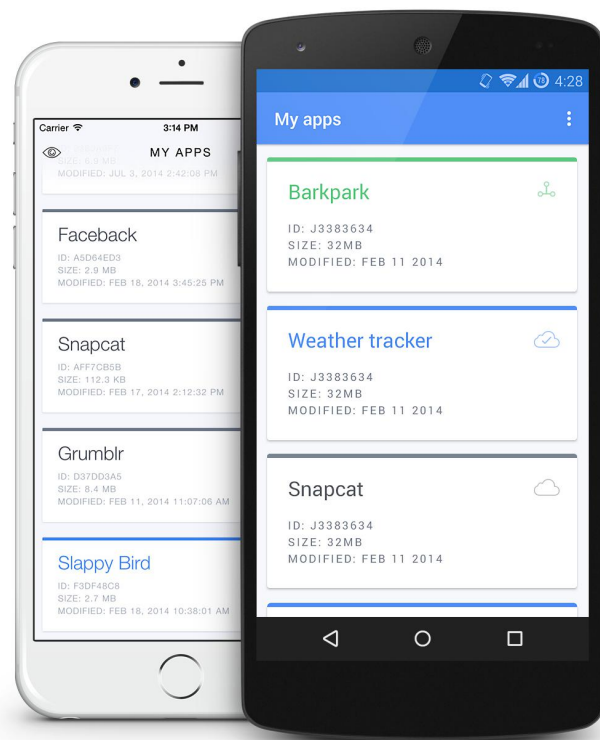
UI Libraries

Jigsaw-七巧板

npm package	1.0.0-beta.16	build	passing	coverage	82%	documentation	3%
components	44	directives	8	injectables	3	demo	137
						e2e testcases	26

[Jigsaw](#)

移动端开发神器



ionic , 底层是NativeScript

第五块：参考资源推荐

特别推荐：ng2-admin



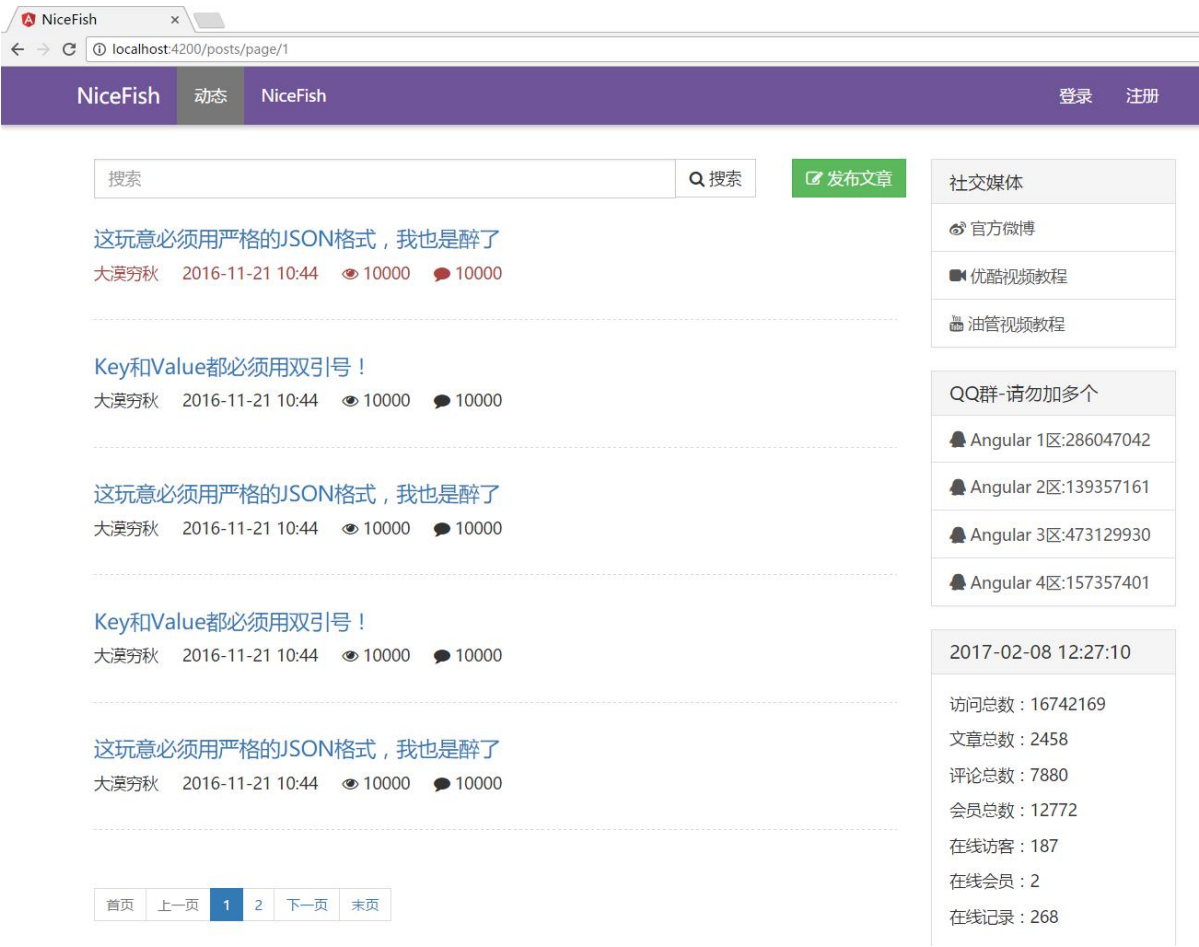
<https://github.com/akveo/ng2-admin>

特别推荐：JHipster-后端基于SpringMVC



<https://jhipster.github.io/>

一般推荐：NiceFish



<http://git.oschina.net/mumu-osc/NiceFish>

谢谢

<https://damoqiongqiu.github.io/>