



链滴

angular4.x 中如何封装组件

作者: [Zhongxiangyun](#)

原文链接: <https://ld246.com/article/1575282043648>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

###1.需要实现的效果图展示（如果只是需要实现组件封装，请直接浏览4）：



###2.组件的思想说明：

组件化并不是前端所特有的，一些其他的语言或者桌面程序等，都具有组件化的先例。确切的说，只有UI层的展示，就必定有可以组件化的地方。简单来说，组件就是将一段UI样式和其对应的功能作为立的整体去看待，无论这个整体放在哪里去使用，它都具有一样的功能和样式，从而实现复用，这种体化的思想就是组件化。不难看出，组件化设计就是为了增加复用性，灵活性，提高系统设计，从而高开发效率

###3.说说抽离组件的好处和需要注意的事项：

好处——解决以前代码里面需要嵌套N个列表的问题：

- 1.组件的思想，可以理解为积木，拼图的一个零部件，哪里需要哪里搬，这样带来的好处就是复用性强
- 2.节省时间，不用去写重复的界面代码
- 3.便于维护和应对新的需求变更

注意事项

1.组件带来方便的同时，也带来了一系列的问题，最最核心的就是传参问题：

第一种情况，父组件向子组件传值操作

第二种情况，子组件向父组件传值操作

2.组件的设计最好在项目初期就制定好，组件的原则，尽量单一，不涉及界面的业务逻辑，通俗的说组件尽量只用来做展示，一业务逻辑尽量通过业务代码来控制实现。

3.当然，要抽离组件，是一个比较耗费时间的工作，需要合理安排时间来来进行

###4.进入今天的主题

#####step1.首先定义一个组件模块

ts

```
import {Component, Input} from '@angular/core'
```

```
@Component({  
  selector: 'snail-showcase-nav',  
  templateUrl: './nav.component.html',
```

```

    styleUrls: ['./nav.component.css'],
  })
  export class MemberNavComponent {

    @Input() memberId: any ////会员id
    @Input() memberName: any ////会员id
  }

  /*
  *
  * 1.引入该组件 <snail-showcase-nav [memberId]="memberId" [memberName]="memberDetail.name"></snail-showcase-nav>
  * 2.需要多配置一个会员名字的参数 memberId: any; //会员id memberName: string = ""; //会员name
  * 3.在父组件中 加入: encapsulation: ViewEncapsulation.None//去掉对应Component打包出来的css作用域, 引入自定义组件才能有样式效果
  *
  * */

```

html

```

<ul class="header clearfix">
  <li class="nav over-ell">
    <a routerLink="/member" class="return">会员</a>
    &gt;
    <span class="name">{{memberName}}</span>
  </li>
  <a class="box" routerLink="/member/detail/{{memberId}}" routerLinkActive="active">
    <div class="button">&emsp; 概要 &emsp;</div>
  </a>
  <a class="box" routerLink="/member/page/cost/{{memberId}}" routerLinkActive="active">
    <div class="button">消费记录</div>
  </a>
  <a class="box" routerLink="/member/page/store/{{memberId}}" routerLinkActive="active">
    <div class="button">余额流水</div>
  </a>
  <a class="box" routerLink="/member/page/count/{{memberId}}" routerLinkActive="active">
    <div class="button">次卡记录</div>
  </a>
  <a class="box" routerLink="/member/page/shopping/{{memberId}}" routerLinkActive="active">
    <div class="button">购物卡</div>
  </a>
  <a class="box" routerLink="/member/page/Coupon/{{memberId}}" routerLinkActive="active">
    <div class="button">优惠券</div>
  </a>

```

```
<a class="box" routerLink="/member/page/message/{{memberId}}" routerLinkActive="active">
  <div class="button">短信记录</div>
</a>
<a class="box" routerLink="/member/page/up/{{memberId}}" routerLinkActive="active">
  <div class="button">等级变更记录</div>
</a>
</ul>
```

css 省略

#####step2.在公共的 @NgModule 里面引入,大概长下面这样

```
declarations: [
  MemberShoppingLog
],
```

#####step3.进入需要展示这个效果的界面,引入子组件

如下:

```
<snail-showcase-nav [memberId]="memberId" [memberName]="memberDetail.name"></snail-showcase-nav>
```

参数说明: 组件传参模式

[memberId]="memberId" //需要定义好 memberId

[memberName]="memberDetail.name" //需要定义好 memberName

#####step4.子组件接收参数设置:

```
@Input() memberId: any ////会员id
@Input() memberName: any ////会员id
```

#####step5.关于样式失效问题,请看前面的文章说明。

#####step6.关于跳转并选中,借用路由标签特性:

routerLinkActive="active"

```
<a class="box" routerLink="/member/page/store/{{memberId}}" routerLinkActive="active">
  <div class="button">余额流水</div>
</a>
```

本文完。