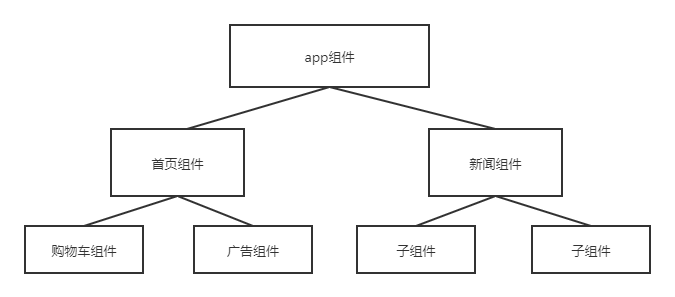
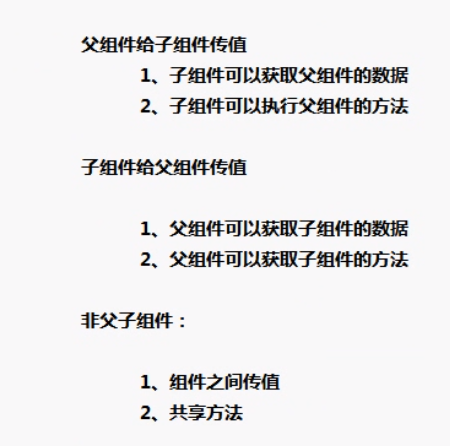
# Angular父子组件

父子组件是一种相对关系，如下图所示，app组件相对于首页组件是父组件，首页组件相对于app组件是子组件；在首页组件引入购物车组件，首页组件相对于购物车组件就是父组件。



父子组件之间的传值分为下面几种情况：



## 父组件给子组件传值@input

父组件不仅可以给子组件传递简单的数据，还可以把自己的方法以及整个父组件传给子组件。

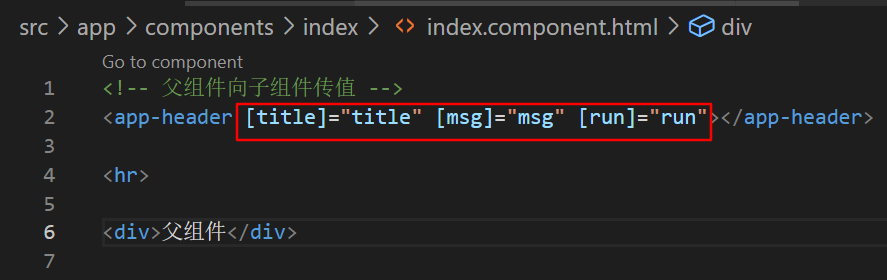
父组件：index.component

子组件：header.component

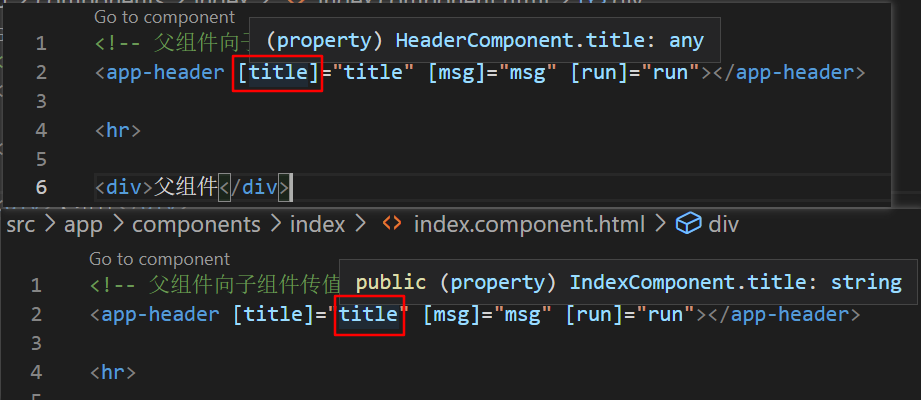
1、在父组件中定义属性和方法



2、父组件引入子组件



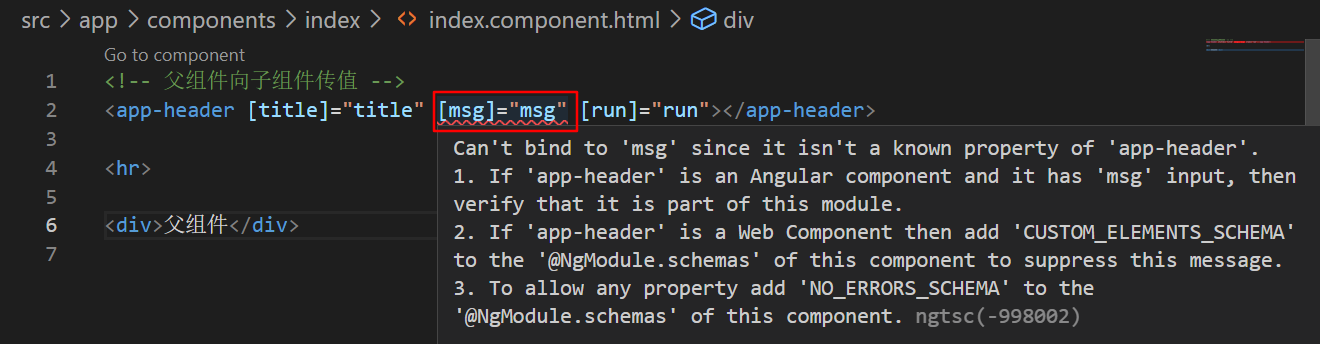
[]内是子组件的属性名称，""内是父组件的属性名称。如下：



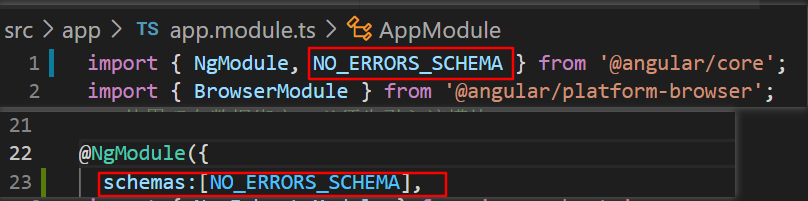
3、子组件接收父组件传入的数据



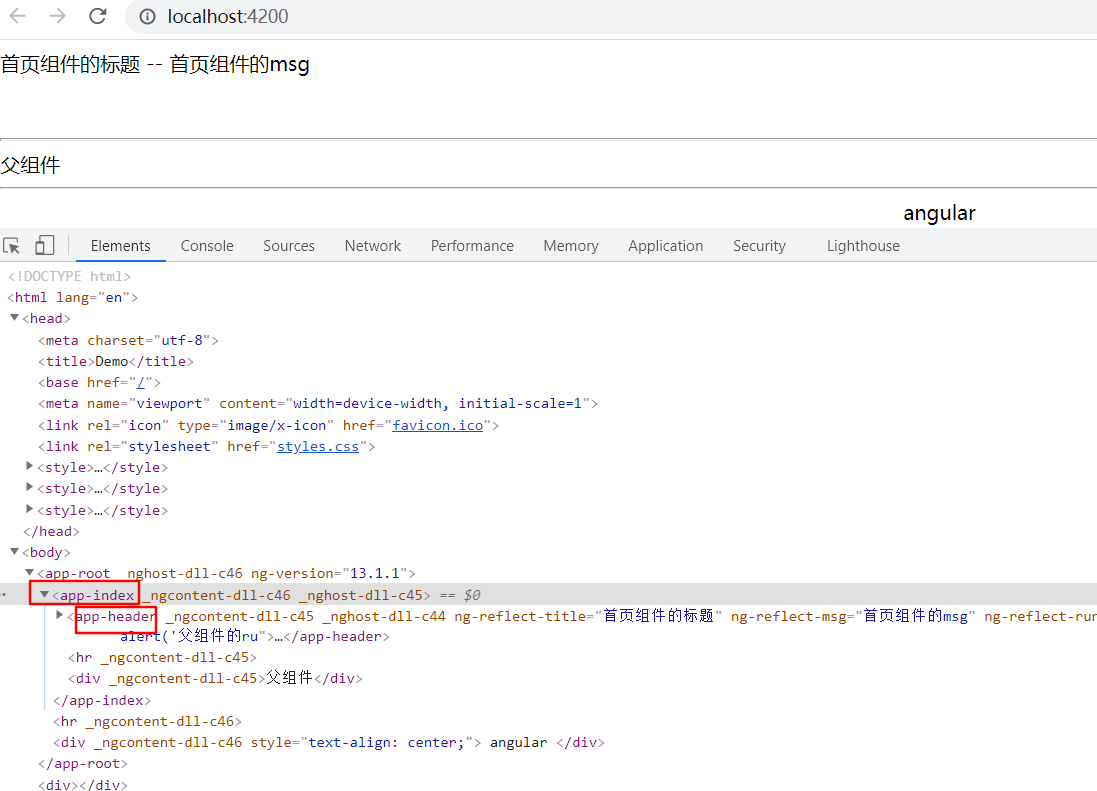
【注意】如果父组件向子组件传值时，如果子组件还未定义相应的属性名，编译器会报错。此时可以在app.module.ts中添加schemas:[NO\_ERRORS\_SCHEMA]关闭该提示。



关闭错误提示，不建议这样做。

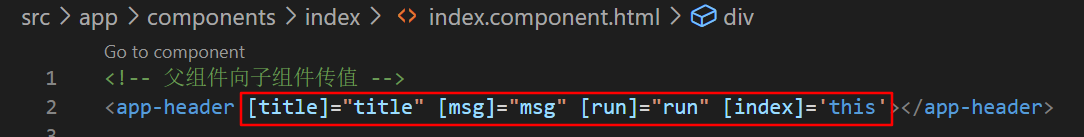


ng serve --open运行：

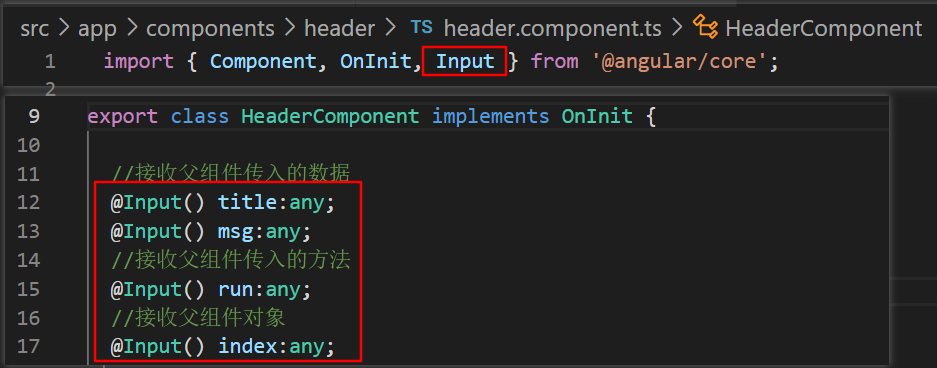


## 父组件向子组件传值小结

1、在父组件引入子组件的地方使用[]传入值(数据、方法、对象)



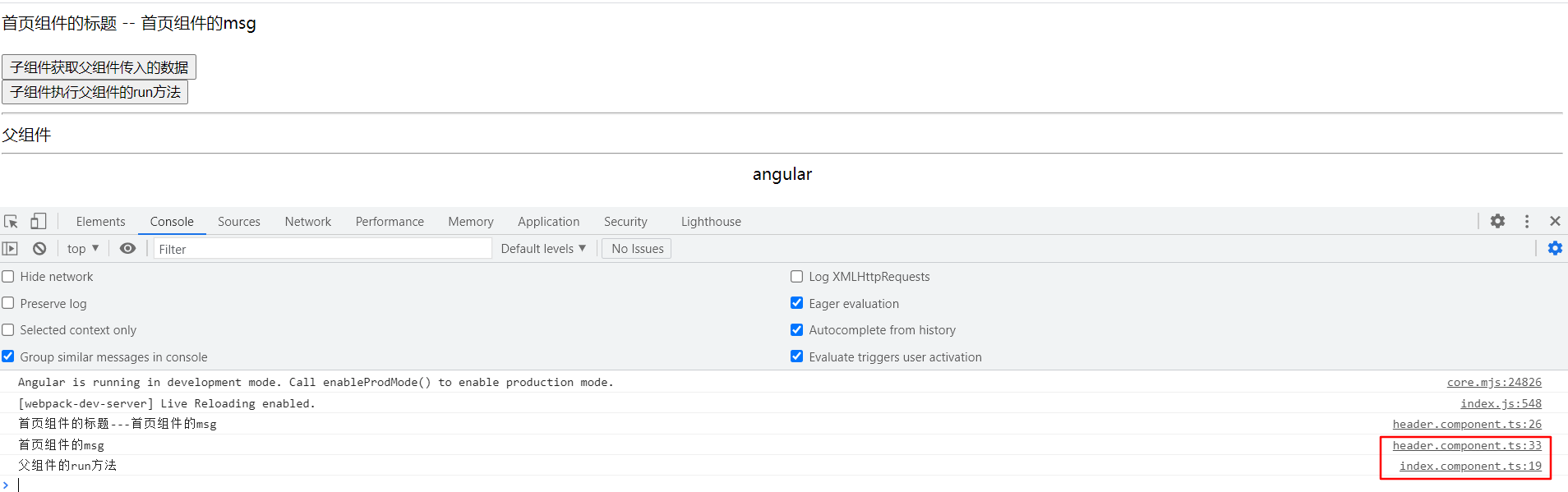
2、子组件导入Input，接收父组件传来的值



3、在子组件中进行调用



运行截图



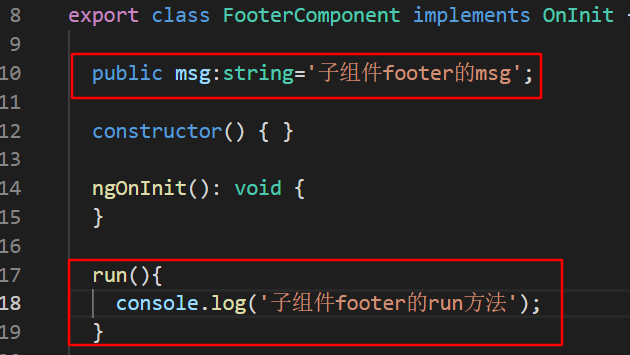
## 父组件从子组件获取数据和方法@ViewChild

1、父组件引入子组件时，给子组件设置标识

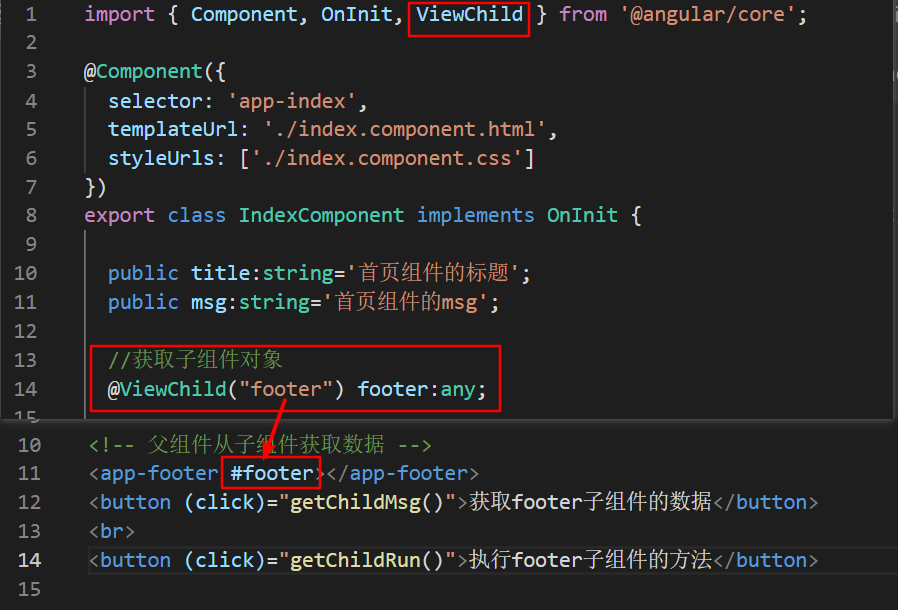
index.component.html



子组件footer.component.ts中定义属性和方法



2、父组件导入ViewChild，获取子组件对象



3、父组件使用获取到的子组件对象调用属性和方法

  getChildMsg(){

    console.log(this.footer.msg);

  }

  getChildRun(){

    this.footer.run();

  }

## 子组件使用@Output触发父组件的方法

### 1.子组件引入output&EventEmitter

//footer.component.ts

import { Component, OnInit, Output, EventEmitter } from '@angular/core';

### 2.子组件实例化EventEmitter

//footer.component.ts

@Output() public outer = new EventEmitter();

### 3.在父组件中监听

//index.component.html

<app-footer #footer (outer)="parentRun($event)"></app-footer>

父组件中监听的对象必须和子组件中定义的一样，如此处为outer。

//index.component.ts

  parentRun(e:any){

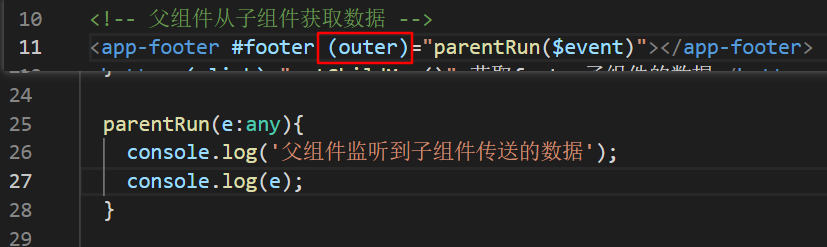
    console.log('父组件监听到子组件传送的数据');

    console.log(e);

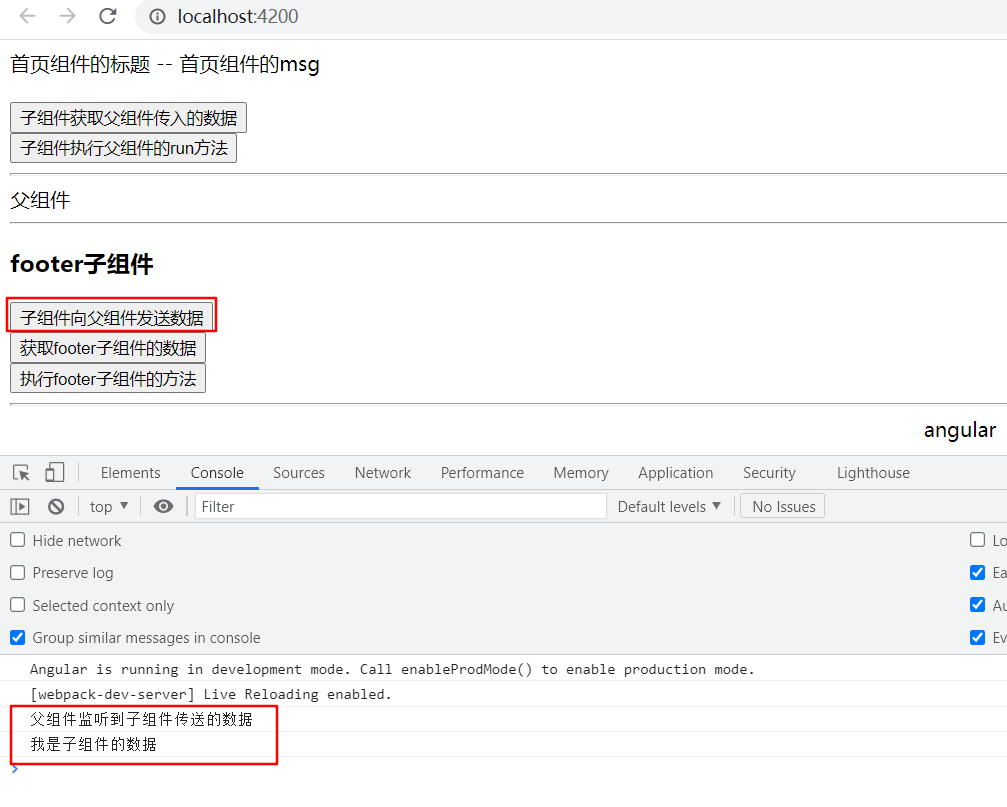
  }

子组件中广播数据时，会触发父组件的方法。





运行截图



# 组件间通讯小结

1、父组件给子组件传值

使用@Input

2、子组件给父组件传值

使用@ViewChild（推荐）或者@Output(较复杂，一般不用)。

3、非父子组件间通讯：

1）使用LocalStorage

2）通过Angular服务

# 提交记录

git仓库：<https://github.com/Angeliau/angular-demo.git>

