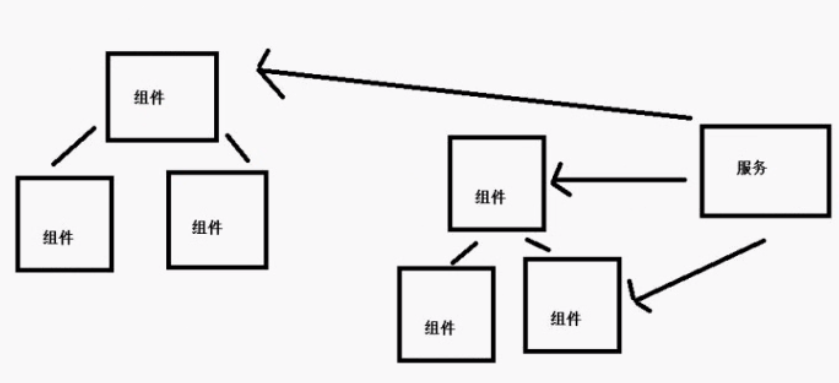
# Angular服务

## angular中的服务



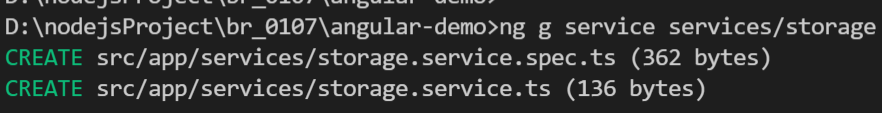
angular是基于模块化、组件化的开发方式，默认情况下一个组件无法调用另一个组件内的方法，组件间是相互隔离的。将公共的功能封装起来放到服务中去，在需要调用的组件中，只要引入该服务，就可以调用公共方法。

## 创建服务

ng g service service-name

创建到指定目录下：

ng g service services/storage



## 引入并配置服务

在app.module.ts中引入服务，并配置

import { StorageService } from './services/storage.service';

@NgModule({

…

providers: [StorageService],

…

})

storage.service.ts中有如下方法：

export class StorageService {

  constructor() { }

  get(){

    return 'this is a service';

  }

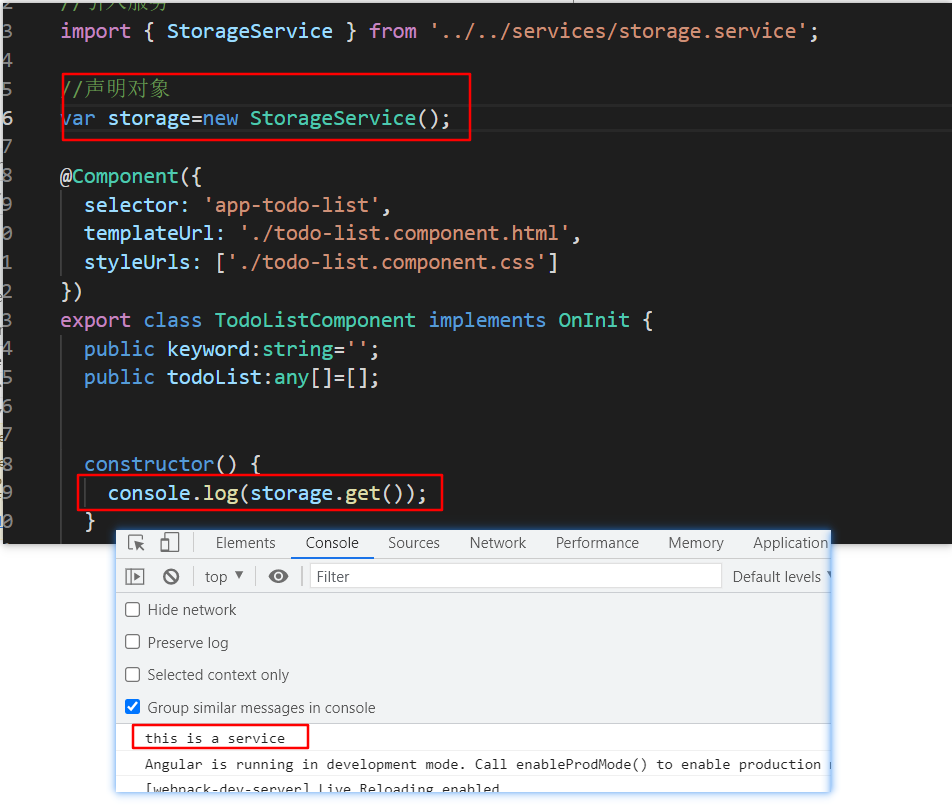
}

## 在组件中使用服务

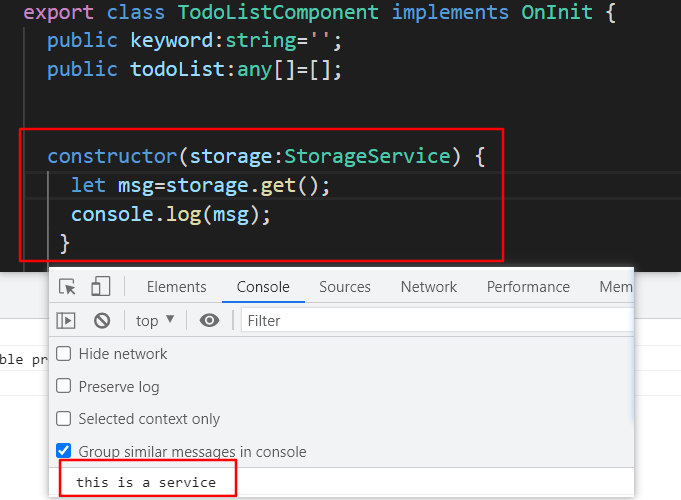
需要在要调用服务的组件中引入服务，



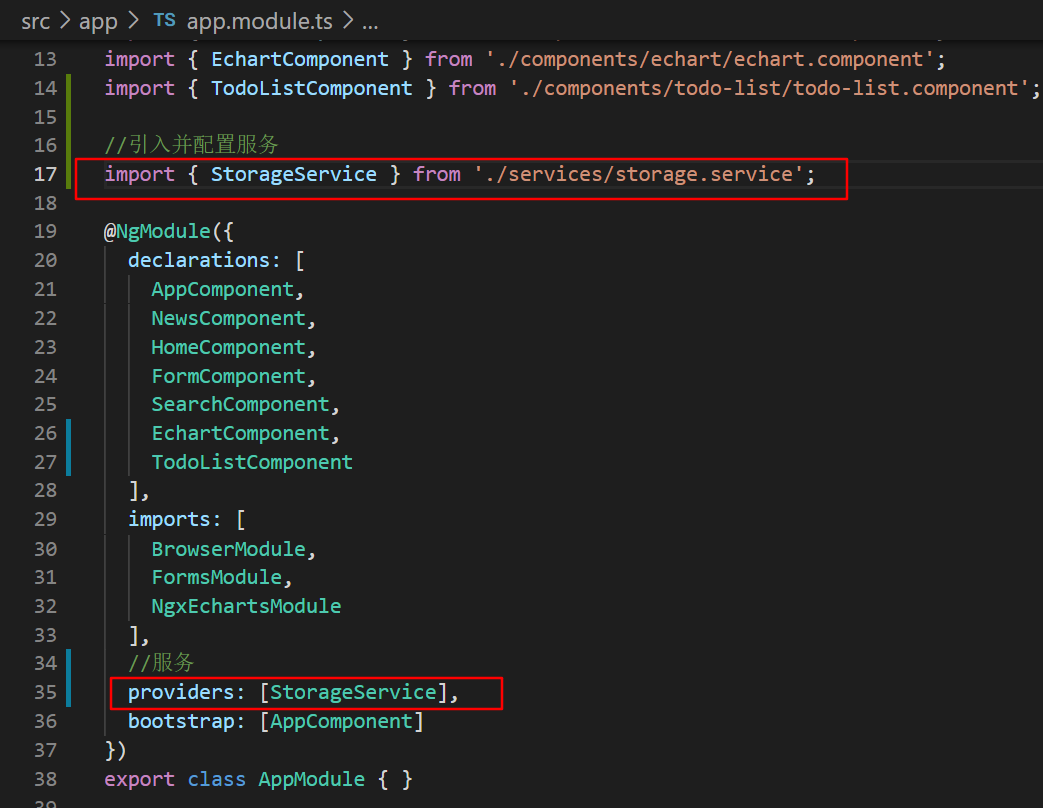
### 不推荐的调用方式

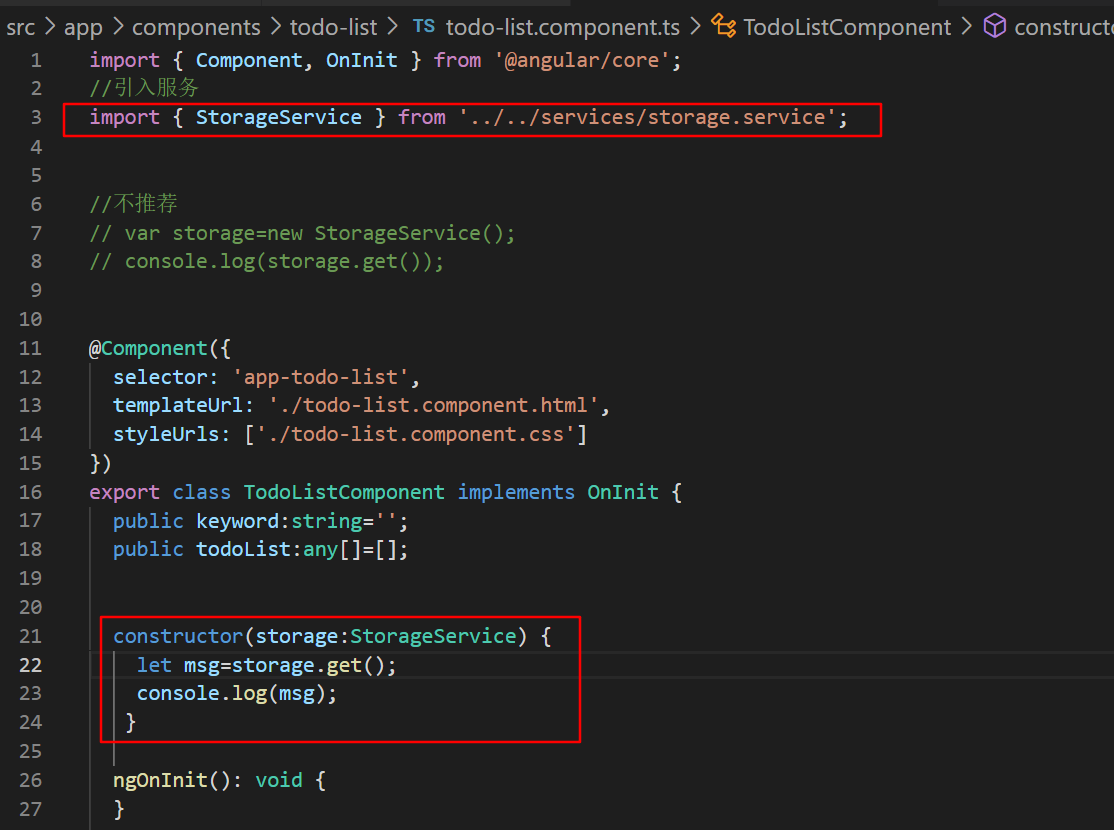


### 官方推荐用法



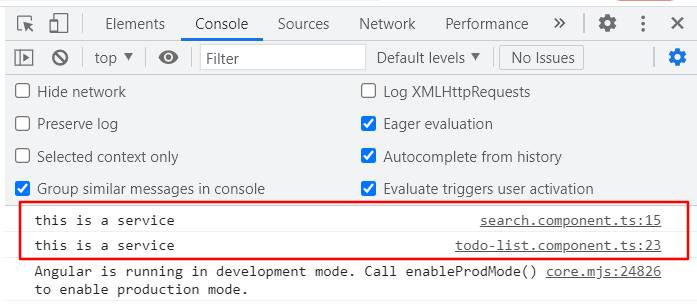
【注意】在组件中使用服务时，需要引入两次服务。一次是在app.module.ts引入并配置，一次是在对应组件的xxx.component.ts中引入。





### 在多个组件中调用服务

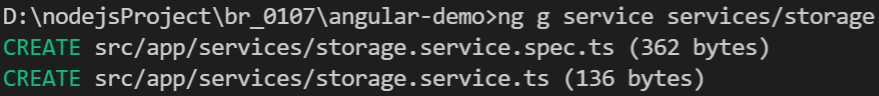
search.component.ts和todo-list.component.ts中分别引入服务，调用get()：



## angular服务小结

1、创建服务

ng g service services/服务名



2、app.module.ts引入创建的服务，并声明服务

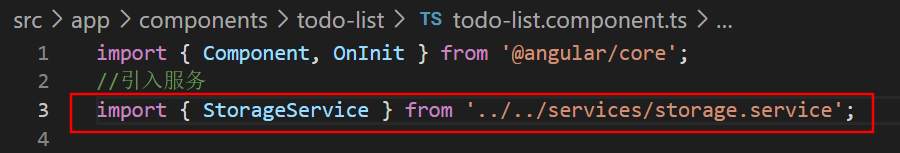
//引入服务

import { StorageService } from './services/storage.service';

//声明服务

  providers: [StorageService],

3、在用到的组件里面，引入服务



4、在构造函数中初始化

  constructor(public storage:StorageService) {

    let msg=this.storage.get();

    console.log(msg);

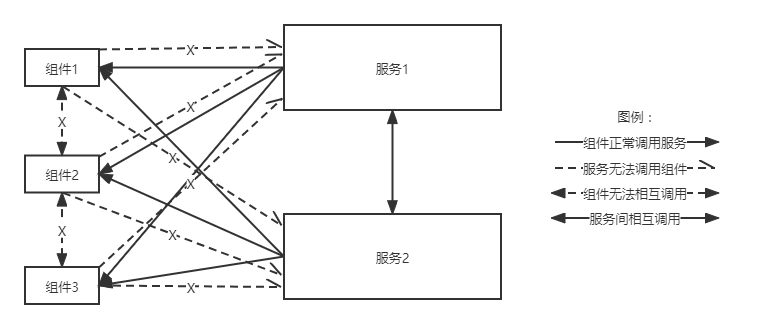
   }

angular中，所有的组件都可以调用服务，只需要引入服务即可。

服务和服务之间可以相互调用。

反过来，服务无法调用组件中的方法；组件之间的方法也无法相互调用。

调用关系如下图所示：



# angular数据持久化

需求：实现页面刷新后，历史记录仍然存在。

未实现数据持久化前，输入的内容刷新后就没有了，如下图：



数据持久化后，在页面刷新后，数据依旧存在



## 在服务中缓存数据

  set(key:string,value:any){

    localStorage.setItem(key,JSON.stringify(value));

  }

  get(key:string){

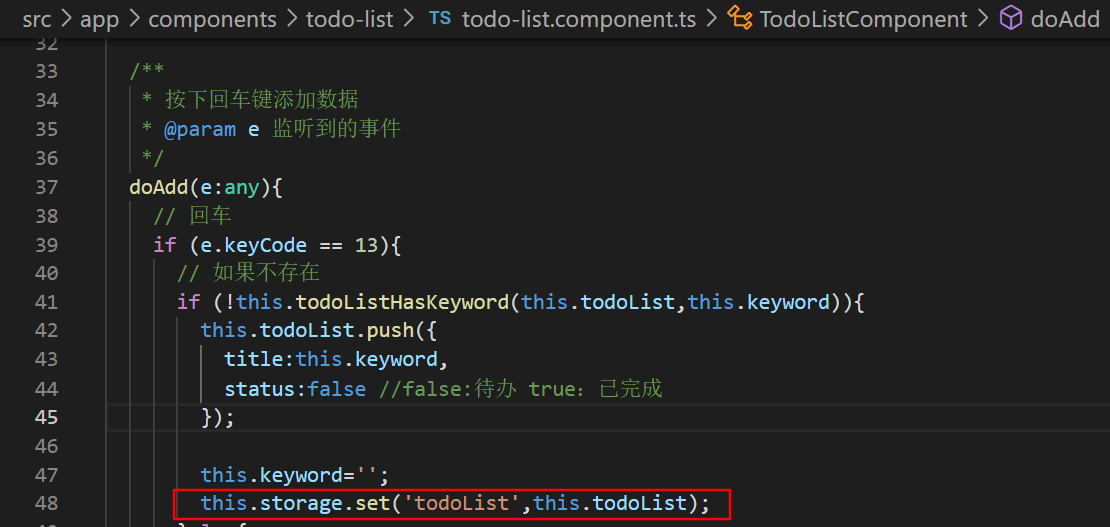
    let str = localStorage.getItem(key);

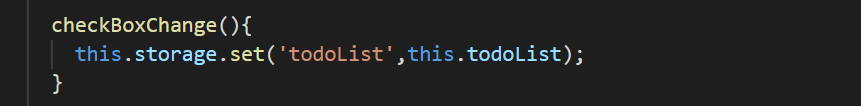
    return JSON.parse((str != null ? str : ""));

  }

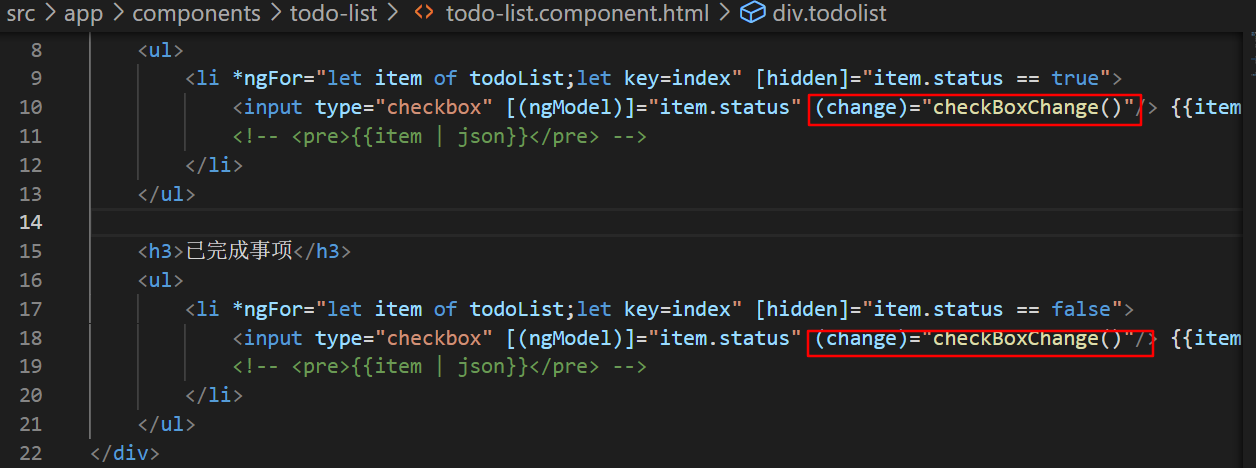
## 组件调用服务

todolist.component.ts

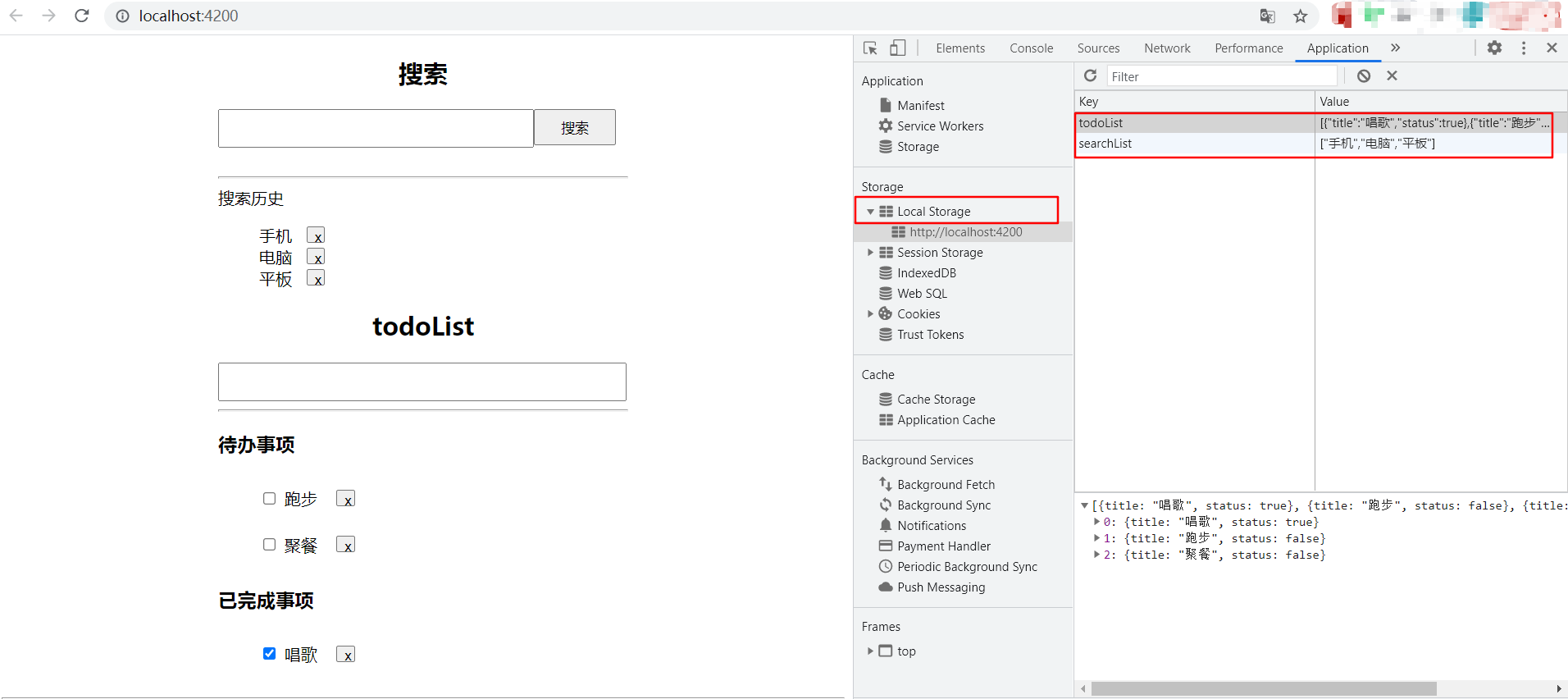




todolist.component.html



运行结果：



# 提交记录

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

