$location &window.location.url 1

$stateProvider & $stateParams & $urlRouterProvider 1

$stateProvider 1

urlRouterProvider 2

$localStorage & $cookieStorage & $sessionStorage 2

微信接口 2

Decorator: 2

类的装饰器 3

Function decorator 3

Getter/setter 3

Directive 3

ng-change 4

Angular 2 - Typescript 4

1.1 config 4

1.1.1 webpack.config.json--angular 4

1.1.1.1文件 4

1.1.1.2命令行 4

1.1.2 tsconfig.json—typescript 4

1.2 项目框架 5

1.3 内置模块 5

# $location &window.location.url

参考：<https://docs.angularjs.org/api/ng/service/$location>

$location.url vs $windows.location.href

前者依据window.location而来，且为双向绑定。Location.replace不会添加到history中。

后者添加items到history中，window.location是

# $stateProvider & $stateParams & $urlRouterProvider

angular的服务依赖： $stateParam

## $stateProvider

.state('contacts', {

url: "/contacts",

templateUrl: 'contacts.html',

Controller:’’

})

.state('contacts', {

url: "/contacts",

templateUrl: 'contacts.html',

Controller:’’

})

.state('contacts', {

url: "^/contacts",

templateUrl: 'contacts.html',

Controller:’’

})

----- 放置与url前，表示绝对path，不包括host

-----对于想要让某个变量值在所有的url的controllers中使用，使用resolve，参考：

https://github.com/angular-ui/ui-router/wiki/URL-Routing

创建了一个名为contacts的state对象，用于监听，当链接的path符合url时，将template渲染到界面。

使用transitionTo(“contacts”)：那么链接地址将会被更新到/contacts

Angular的地址请求中可以使用**{}**

在{}中可以应用正则表达式。如/user/{id: [0-9]{1,9}?}，注意不能使用捕获组，但是可以使用非捕获组?:

地址中的信息可以使用angular的模块$stateParams进行获取，如

<http://localhost:4002/#/invoice/invoice-detail?detailId=a163bfd0-fdac-11e6-8f9d-db8ca1d9fa95&method=add>

对于处理该网页的控制器，使用$stateParams可以获得$stateParams.detailId和$stateParams.method

url: '/users/:id/details/{type}/{repeat:[0-9]+}?from&to'

那么$stateParams的值为{ id:'123', type:'', repeat:'0' }

## urlRouterProvider

负责监听$location

$urlRouterProvider.when(state.url, ['$match', '$stateParams', function ($match, $stateParams) {

if ($state.$current.navigable != state || !equalForKeys($match, $stateParams)) {

$state.transitionTo(state, $match, false);

}

}]);

# $localStorage & $cookieStorage & $sessionStorage

提供键值形式数据，存、取时自动调用了angular的toJson和fromJson

ngStorage包括了两个服务：$localStorage & $sessionStorage

$localStorage客户端存到本地文件中，最大5M

# 微信接口

# Decorator:

固定格式：

Return function(){}

## 类的装饰器

比如：可以利用该装饰器，创建angular的service服务。单击此处输入文字。

Function decorator\_name{

Return function(constructor: Function){}

}

## Function decorator

调用该函数前，执行decorator，decorator可以重新定义被装饰方法的实现，

Function(){

Return function(target:any, propertyKey:any, desc?:PropertyDescriptor ){}

}

# Getter/setter

typescript中的getter/setter方法，在调用时，无需添加get字段

class foo {

private \_bar:boolean = false;

get bar():boolean {

return this.\_bar;

}

set bar(theBar:boolean) {

this.\_bar = theBar;

}

}

在调用时，只需要myfoo=new foo(); console.log(myfoo.bar) 即可获取到值

==============================================================================================

正则：

非捕获组：

？：

Gulp

# Directive

原生的scroll到低端后，引发load more操作

## ng-change

用来控制输入框发生变化后出发提供的函数，参数提供一个function，如ng-change=”function”, 一般与ng-model连用

对于输入框中的操作，赋值，将由ng-model的值进行保存，该变量的值可用作ng-change的参数函数的参数。

例如： <input type="text" ng-change="myFunc(myValue)" ng-model="myValue" />

# Angular 2 - Typescript

## 1.1 config

### 1.1.1 webpack.config.json--angular

#### 1.1.1.1文件

程序的入口配置文件webpack.config.js, exports出json对象

module.exports = {

devServer: {

inline: true, //同命令中的参数 –inline相同

hot: true, //

}

}

1.1.1.2命令行

webpack-dev-server参数：

--content-base dir: 指明需要编译的源码

--inline/iframe: 两种自动刷新页面模式。

1. Inline：内置webpack-dev-server client入口到bundle中（浏览器地址会变化，实时显示状态信息于console）

2: 内置页面到iframe中，根据change自动刷新（app中的地址变化不会反映到浏览器地址中）

对于inline模式：启用，使用装饰器@Component时，指定template为对象，不启用，则只需要传递path到templateUrl

### 1.1.2 tsconfig.json—typescript

angular使用typescript，需要在webpack.config.json的angular的配置文档中引入tsconfig.json, 最为解析.ts文档的配置选项。

{

“compileOnSave”: true,

“compilerOptions”: {

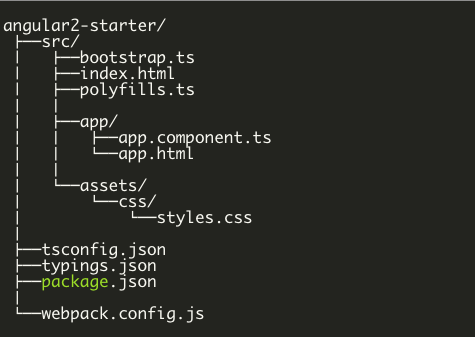
“typeRoots”: []

}

}

当tsconfig文件需要分离成多个文件，且不在root目录下，此时需要在root目录下的tsconfig.json文档中添加配置

## 1.2 项目框架



## 1.3 内置模块

angular中默认的root module 为AppModule， 对于angular中声明的class，都需要使用@ngmodule进行装饰。Angular中存在三种view class： components、directive、pipes

装饰器：@ngmodule

declarations:

providers：创建service，同时同步到global collection of services