AngularJS中的$http

# 更新的$http

官网学习：<https://docs.angularjs.org/api/ng/service/$http>

$http({

method: 'GET',

url: '/someUrl'

})**.then(function successCallback(response) {**

**// this callback will be called asynchronously**

**// when the response is available**

**}, function errorCallback(response) {**

**// called asynchronously if an error occurs**

**// or server returns response with an error status.**

**});**

The response object has these properties:

* **data** – {string|Object} – The response body transformed with the transform functions.
* **status** – {number} – HTTP status code of the response.
* **headers** – {function([headerName])} – Header getter function.
* **config** – {Object} – The configuration object that was used to generate the request.
* **statusText** – {string} – HTTP status text of the response.



# AngularJS中$http服务的简单用法

我们可以使用内置的$http服务直接**同外部**进行通信。**$http服务**只是简单的封装了浏览器原生的**XMLHttpRequest**对象（简称XHR）。

## 链式调用

$http服务是**只能接受一个参数的函数**，这个参数是一个对象，包含了用来生成**HTTP请求的配置内容**。这个函数返回一个**promise对象**，具有**success和error**两个方法。

示例：

**$http({**

**url:'data.json',**

**method:'GET'**

**}).success(function(data,header,config,status){**

**//响应成功**

**}).error(function(data,header,config,status){**

**//处理响应失败**

**});**

## 返回一个promise对象

**var promise=$http({**

**method:'GET',**

**url:"data.json"**

**});**

**由于$http方法返回一个promise对象，我们可以在响应返回时用then方法来处理回调。如果使用then方法，会得到一个特殊的参数，它代表了相应对象的成功或失败信息，还可以接受两个可选的函数作为参数。或者可以使用success和error回调代替。**

**promise.then(function(resp){**

**//resp是一个响应对象**

**},function(resp){**

**//带有错误信息的resp**

**});**

**或者这样：**

**promise.success(function(data,status,config,headers){**

**//处理成功的响应**

**});**

**promise.error(function(data,status,hedaers,config){**

**//处理失败后的响应**

**});**

**then()方法与其他两种方法的主要区别是，它会接收到完整的响应对象，而success()和error()则会对响应对象进行析构。**

## 快捷的get请求

$http.get('/api/users.json');

get()方法返回**HttpPromise对象**。

## 也可以将$http当做函数来使用，这时需要传入一个设置对象，用来说明如何构造XHR对象。

$http({

method:'GET',

url:'/api/users.json',

params:{

'username':'tan'

});

其中设置对象可以包含以下主要的键：

①method可以是：**GET/DELETE/HEAD/JSONP/POST/PUT**

②url:绝对的或者相对的请求目标

③params(字符串map或者对象)

这个键的值是一个字符串map或对象，会被转换成查询字符串追加在URL后面。如果值不是字符串，会被JSON序列化。

//参数会转为？name=ari的形式

$http({

params:{'name':'ari'}

});

④data(字符串或者对象)

这个对象中包含了将会被当作消息体发送给服务器的数据。**通常在发送POST请求时使用。**

**从AngularJS 1.3开始，它还可以在POST请求里发送二进制数据。**要发送一个blob对象，你可以简单地通过使用data参数来传递它。

例如：

var blob=new Blob(['Hello world'],{type:'text/plain'});

$http({

method:'POST',

url:'/',

data:blob

});

# 响应对象

响应对象

AngularJS传递给**then（）方法的响应对象**包含了四个属性。

## ◇data：这个数据代表转换过后的响应体（如果定义了转换的话）

## ◇status：响应的HTTP状态码

## ◇headers：这个函数是头信息的getter函数，可以接受一个参数，用来获取对应名字值

例如，用如下代码获取X-Auth-ID的值：

$http({

method: 'GET',

url: '/api/users.json'

}).then (resp) {

// 读取X-Auth-ID

resp.headers('X-Auth-ID');

});

## ◇config：这个对象是用来生成原始请求的完整设置对象。

◇statusText（字符串）：这个字符串是响应的HTTP状态文本。

# 缓存HTTP请求

**默认情况下，$http服务不会对请求进行本地缓存。**在发送单独的请求时，我们可以通过向$http请求**传入一个布尔值或者一个缓存实例**来启用缓存。

$http.get('/api/users.json',**{ cache: true }**)

.success(function(data) {})

.error(function(data) {});

第一次发送请求时，$http服务会向/api/users.json发送一个GET请求。第二次发送同一个GET请求时，$http服务会从缓存中取回请求的结果，而不会真的发送一个HTTP GET请求。

在这个例子里，由于设置了启用缓存，AngularJS默认会使用$cacheFactory,这个服务是AngularJS在启动时自动创建的。

如果想要对AngularJS使用的缓存进行更多的自定义控制，可以向请求传入一个自定义的缓存实例代替true。

# 调用

$http(config).success(function(**data,status,headers,config**){}).error(function(data,status,headers,config){});

## config为一个JSON对象，其中主要包含该请求的url、data、method等，如{url:"login.do",method:"post",data:{name:"12346",pwd:"123"}}。 •method {String} 请求方式 "GET"."POST"等等。

•url {String} **请求的URL地址**

•params {key,value} **请求参数**，将在URL上被拼接成？key=value

•data {key,value} **数据**，将被放入请求内发送至服务器

•**cache {boolean}** 若为true，**在http GET请求时采用默认的$http cache**，否则使用$cacheFactory的实例

•**timeout {number}** 设置超时时间

## success为请求成功后的回调函数，error为请求失败后的回调函数，这里主要是对返回的四个参数进行说明。

•data 响应体

•status 相应的状态值

•headers 获取getter的函数

•config 请求中的config对象，同上第1点

# 简化

为了方便大家与HTTP服务器进行交互，angularJS提供了各个请求方式下方法。

## $http.put/post(url,data,config) url、name必填，config可选

## $http.get/delete/jsonp/head(url,confid) url必填，config可选

　　url、data、config与$http的参数一致。

# 示例

<!DOCTYPE HTML>

<html lang="zh-cn" >

<head>

<meta charset="UTF-8">

<title>CSSClasses</title>

<script src="angular.min.js" type="text/javascript"></script>

<script type="text/javascript">

function ctrl($http,$scope){

$scope.login = function(user){

$http.post("login.do",user).success(function(data, status, headers, config){

alert("success");

}).error(function(data, status, headers, config){

alert("error");

})

}

$scope.login1 = function(user){

$http({url:"login.do",data:user}).success(function(data, status, headers, config){

alert("success");

}).error(function(data, status, headers, config){

alert("error");

})

}

}

</script>

</head>

<body ng-app>

<div ng-controller="ctrl">

<form name="loginFm">

Name:<input ng-model="user.name" />

pwd: <input ng-model="user.pwd" />

<input type="button" value="login" ng-click="login(user)" />

<input type="button" value="login1" ng-click="login1(user)" />

</form>

</div>

</body>

</html>