



AngularJS 實作與應用

Day 4



資訊服務處劉智漢

2015.04.30

@中央研究院 人文館

day3 複習

- directives 介紹
 - AngularJS 內建的directives(ng-*)
 - 自訂directives
 - restrict(ACEM)、scope(=@&)
 - link:function(\$scope, \$element,\$attrs,ngModel)
- Form 資料檢核
- AngularJS 組態設定(config)

AngularJS 開發實戰：新手入門篇

- **AngularJS 觀念與架構**
 - 認識 AngularJS 與核心觀念
 - 了解 jQuery 與 AngularJS 的差異
 - 體驗以物件模型為基礎的網頁操作模式
 - 認識 MVC 架構 (Models, Views, Controllers)
 - 認識 DI 設計模式 (Dependency Injection)
- **AngularJS 基礎開發入門**
 - 執行 AngularJS 的多種方式
 - 認識 Angular 表達式 (Angular Expression)
 - 認識 Angular 過濾器 (Filters)
 - 認識 Angular 控制器 (Controllers)
 - 認識雙向資料繫結 (Two-way data binding)
 - 認識內建的 Directives (HTML 指令)
 - 呼叫後端 API 的開發技巧 (使用 \$http 服務)

AngularJS 開發實戰：模組開發篇

- **AngularJS 核心觀念**
 - 認識 Angular 模組化架構
 - 了解 DI 與 Strict DI Mode
 - 了解 Controller 與 as 語法
 - 了解 Angular 執行生命週期
 - 了解 Angular 啟動過程
 - 了解 Scope (物件存取範圍) 與 Digest (消化事件迴圈)
 - 認識雙向資料繫結與單向資料繫結
- **AngularJS 模組化技術**
 - 深入了解 Angular 模組 (Module)
 - 深入了解 Angular 服務 (Service, Factory, Provider)
 - 深入了解 Angular 過濾器 (Filter)
 - 深入了解 Angular 指令 (Directive)
- **AngularJS 進階議題**
 - 介紹 AngularJS 好用開發工具
 - 了解 AngularJS 效能調校方法

javascript strict mode

- javascript 嚴謹模式，2009 年ECMAScripts 加入
- IE 10 之前的版本並不支援此模式
- 語法："use strict";

- 變數使用前一定要先用var宣告

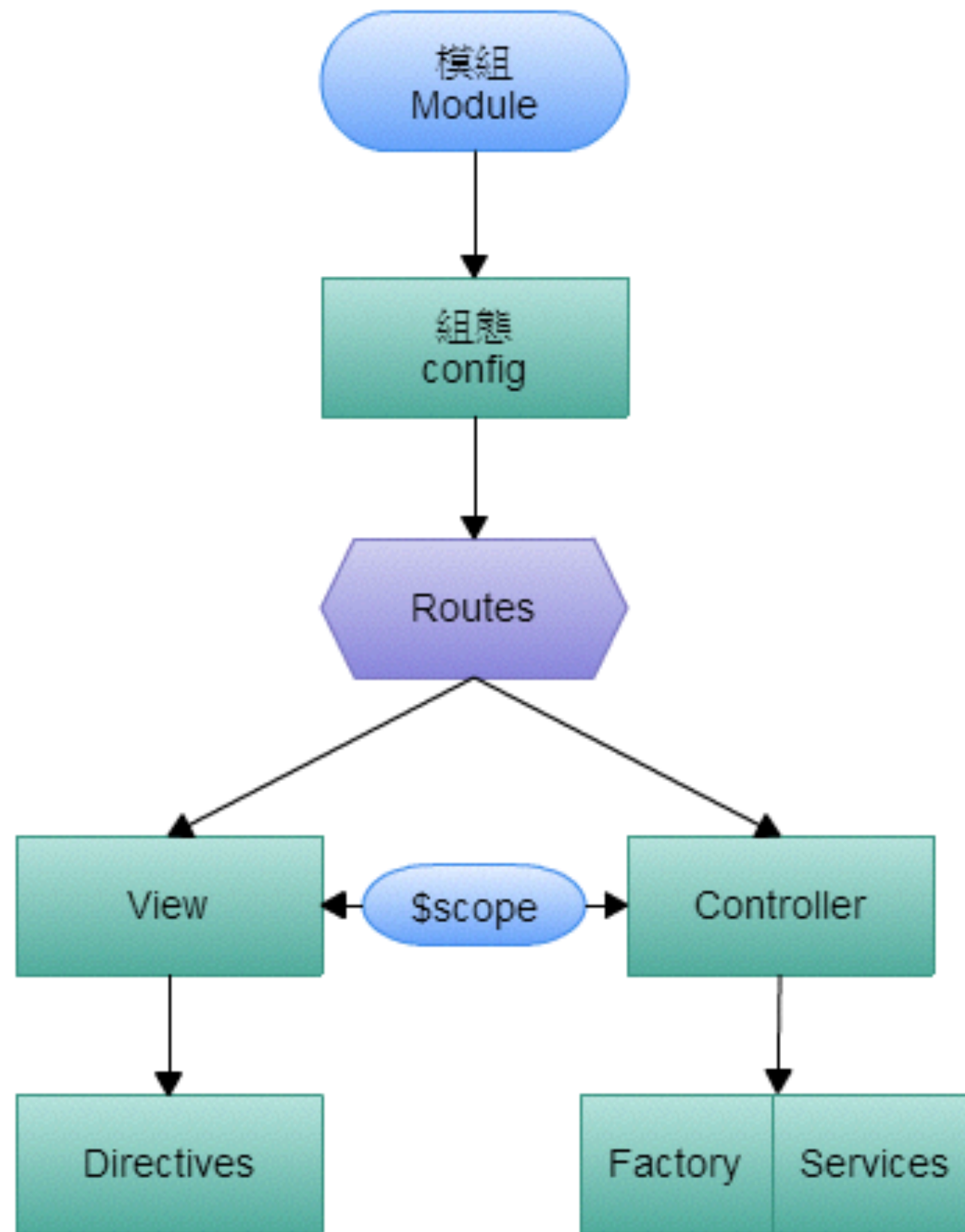
- 有作用域概念

```
function() {  
    "use strict";  
}
```

=====

```
"use strict";  
function() { ... }
```

- 八進位制數字被禁止使用，010代表10而非8
- 參數中不可以使用delete



AngularJS routing 原理

- `` 內部錨點 ``
- ``
- 內部錨點參考範例：資訊服務處網站
- 使用 `hash` 來儲存對應的 URL
 - `{" / ": "index.html", "/dir/app": "app.html" }`
- 透過 `XMLHttpRequest` 取得遠端網頁資料
- 讀取到的資料是 `cache` 內的資料，非最新資料

routing 類別

- **ngRoute**

- 透過瀏覽器網址來進行routing方式

- **ngTemplate**

- 透過ng-template方式來進行網頁內部routing

- **uiRoute** (第三方開發)

- 透過程式紀錄狀態(state)進行routing方式

ngRoute 使用步驟

- 需至官網額外下載angular-route.js檔案
- `<script src="libs/angular-route.js"></script>`
- 前端設定ngView
- 後端設定config、\$routeProvider

route config 語法

```
• angular.module('myApp', ["ngRoute"])\n  .config(function($routeProvider) {\n    $routeProvider\n      .when(\n        path, {\n          template: "" or templateUrl: "url"\n          controller: "helloworld",\n          controllerAs: "",\n          resolve: function() {}\n        })\n      .otherwise({redirectTo: "/"});\n  });
```


Lab12. ngRoute 設定

- 設計兩個選單，讓使用者可以點擊後作頁面切換
- 前端設定：ng-view
- config 設定 \$routeProvider

Lab12-1. ngRoute 練習



- 將水庫及ubike資料改用ngRoute呈現
- 建立兩個獨立頁面分別放水庫及ubike顯示畫面
- 將內容改用templateUrl呈現

controllerAs

```
angular.module('myApp', ['ngRoute'])
.config(['$routeProvider', function($routeProvider) {
  $routeProvider
    .when(
      "/save/:id",
      {
        controller: "helloworld as ctrl",
        //controller: "helloworld",
        //controllerAs: 'ctrl'
      })
}]);
```

為何要使用 controllerAs ?

- 程式碼不用重寫，且可以有新的instance (類似 new)

- controller 內也需要改寫：self = this; self.water = ...;

resolve 語法

```
angular.module('myApp', ['ngRoute'])  
  .config(['$routeProvider', function($routeProvider) {  
    $routeProvider  
      .when(  
        "/save/:id",  
        {  
          resolve: {  
            key: [dependency inject function]  
            // key: ['$http', function($http) {  
              }]  
          }  
        }  
      )  
  }])  
});
```

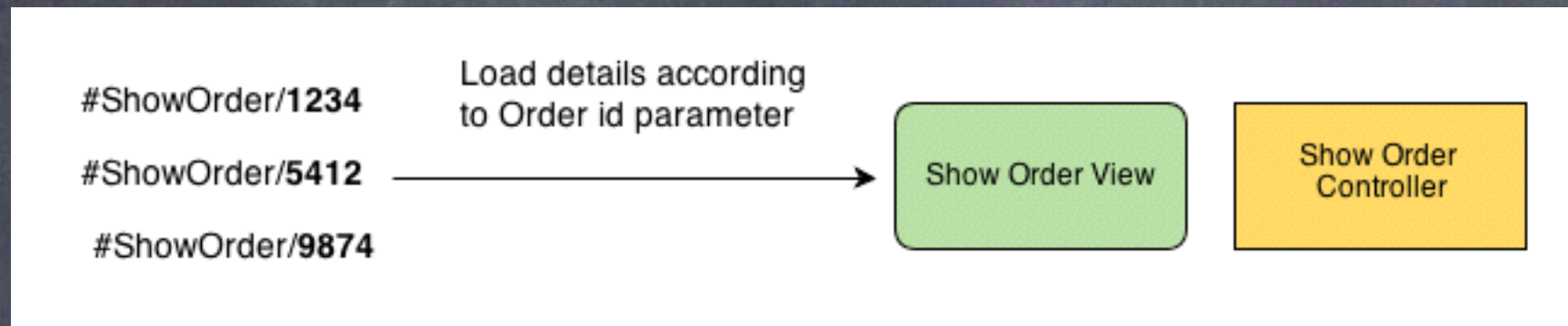

resolve 說明

- AngularJS 會在載入此路徑模組前呼叫它
- 其回傳值為：`true`、`false`或`promise`，只有回傳值為真時，才會繼續載入此繞送路徑相關模組
- `resolve`可以有多个函数，AngularJS會等到所有函数都有回傳值後才會繼續載入`route`，若其中有一個回傳`false`，則不會載入該`route`
- 載入失敗時，AngularJS仍會載入該`route`的`template` 內容，但不會進行編譯及建立HTML DOM

Lab12. 將水庫資料與ubike 資訊搭配bootstrap做成選單

- 1. 先使用bootstrap製作選單
- 2. 將directives移到單獨頁面
- 3. 切換頁面

網址傳入參數用法



- web 參數傳遞法 (ex:php) :

`http://oo.xx.com/save.php?id=3`

- AngularJS routing 傳遞 :

`http://oo.xx.com/index.html#/save/:id"`

controller 中需搭配 `$routeParams` 使用

參數傳遞 config端

- <http://oo.xx.com/index.html#/page/12345>
- ```
angular.module('myApp', ['ngRoute'])
.config(['$routeProvider', function($routeProvider) {
 $routeProvider
 .when(
 "/page/:id",
 {
 controller: "helloworld", //controllerAs
 templateUrl: "views/index.html"
 })
 .otherwise({redirectTo: "/" });
}]);
```
- `$routeParams.id`    `// 12345`



- `$routeParams {  
 id:",  
 cd:",  
}`

- `$routeParams.id`

- `$routeParams.cd`



# 參數傳遞 controller

- <http://oo.xx.com/index.html#/page/12345>
- ```
angular.module('myApp', ["ngRoute"])
.controller('helloWorld', ['$routeParams',
function($routeParams) {
    console.log($routeParams.id);
}]);
```
- `$routeParams.id` // 12345

ngRoute 使用上注意事項

- 每個ngRoute都必需要指定template或templateUrl，不可以有空的內容
- \$routeParams 取得的參數型態都為string，所以在使用===作比較時需注意型別
- "123" == 123
- "123" === 123
- 每個應用程式只能有個ng-view

補充：\$location 介紹

- AngularJS 提供的service，主要用來處理browser的 address bar 內的資料(url)，在controller內使用
- .protocol() : https、http
- .port() : 3306
- .host() : www.sinica.edu.tw
- .absUrl() : 回傳完整的url(包含http://)
- .url() : 回傳主機後的網址 /index.html

另類切換畫面方式

ng-template

ng-template 使用方式

- index.html

```
<script type="text/ng-template" id="add_order.html">  
  <h2> Add Order </h2>  
  {{message}}  
</script>
```


Lab13. ngTemplate 練習

- 設計兩個功能，讓使用者可以點擊後作頁面切換

Lab13. ngTemplate 練習



- 將水庫及ubike資料改用ngTemplate呈現