



AngularJS 實作與應用

Day 3



資訊服務處網頁技術及出版組劉智漢

andyliu@sinica.edu.tw

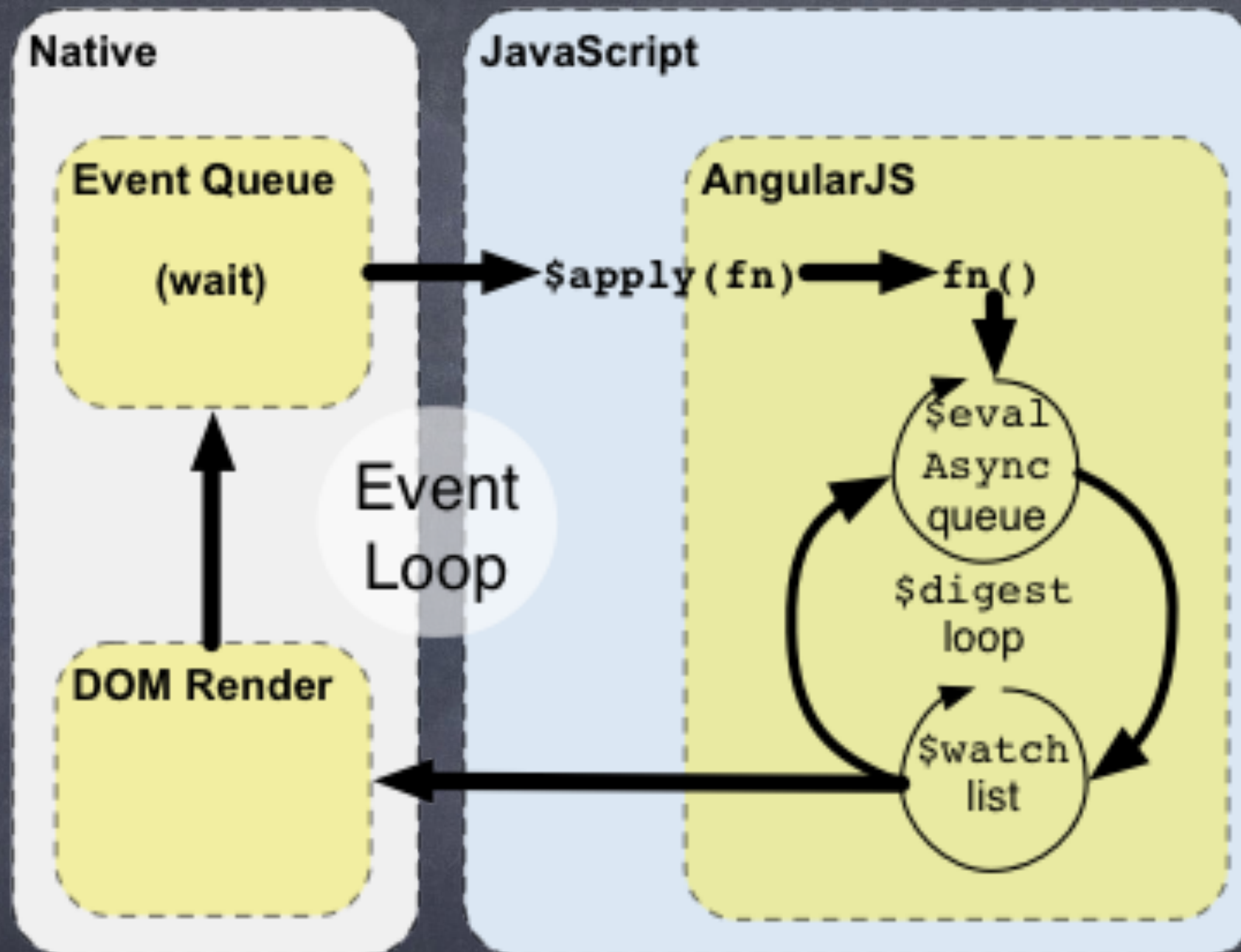
<http://www.facebook.com/justjii>

2015.04.21

@中央研究院 人文館

Day1 複習

- MVC、SPA
- AngularJS 啟動迴圈、\$Digest (消化事件迴圈)
- AngularJS ng-app、controller、\$scope 作用域
- {}、[] 以及資料顯示方式 {{ }}
- two-way binding 資料雙向繫結



- dirty check
- digest loop 消化事件迴圈

day2 複習

- orderBy
- \$filter
- bootstrap css framework
- directives
 - AngularJS內建的directives(ng開頭的)

補充：如果你使用IE...

Internet Explorer Compatibility

 [Improve this Doc](#)

Note: AngularJS 1.3 has dropped support for IE8. Read more about it on [our blog](#). AngularJS 1.2 will continue to support IE8, but the core team does not plan to spend time addressing issues specific to IE8 or earlier.

This document describes the Internet Explorer (IE) idiosyncrasies when dealing with custom HTML attributes and tags. Read this document if you are planning on deploying your Angular application on IE.

The project currently supports and will attempt to fix bugs for IE9 and above. The continuous integration server runs all the tests against IE9, IE10, and IE11. See [Travis CI](#) and [ci.angularjs.org](#).

We do not run tests on IE8 and below. A subset of the AngularJS functionality may work on these browsers, but it is up to you to test and decide whether it works for your particular app.

To ensure your Angular application works on IE please consider:

1. Use `ng-style` tags instead of `style="{{ someCss }}"`. The latter works in Chrome and Firefox but does not work in Internet Explorer ≤ 11 (the most recent version at time of writing).

補充：立即函數表達式

immediately invoked
function expression

立即函數 IIFE

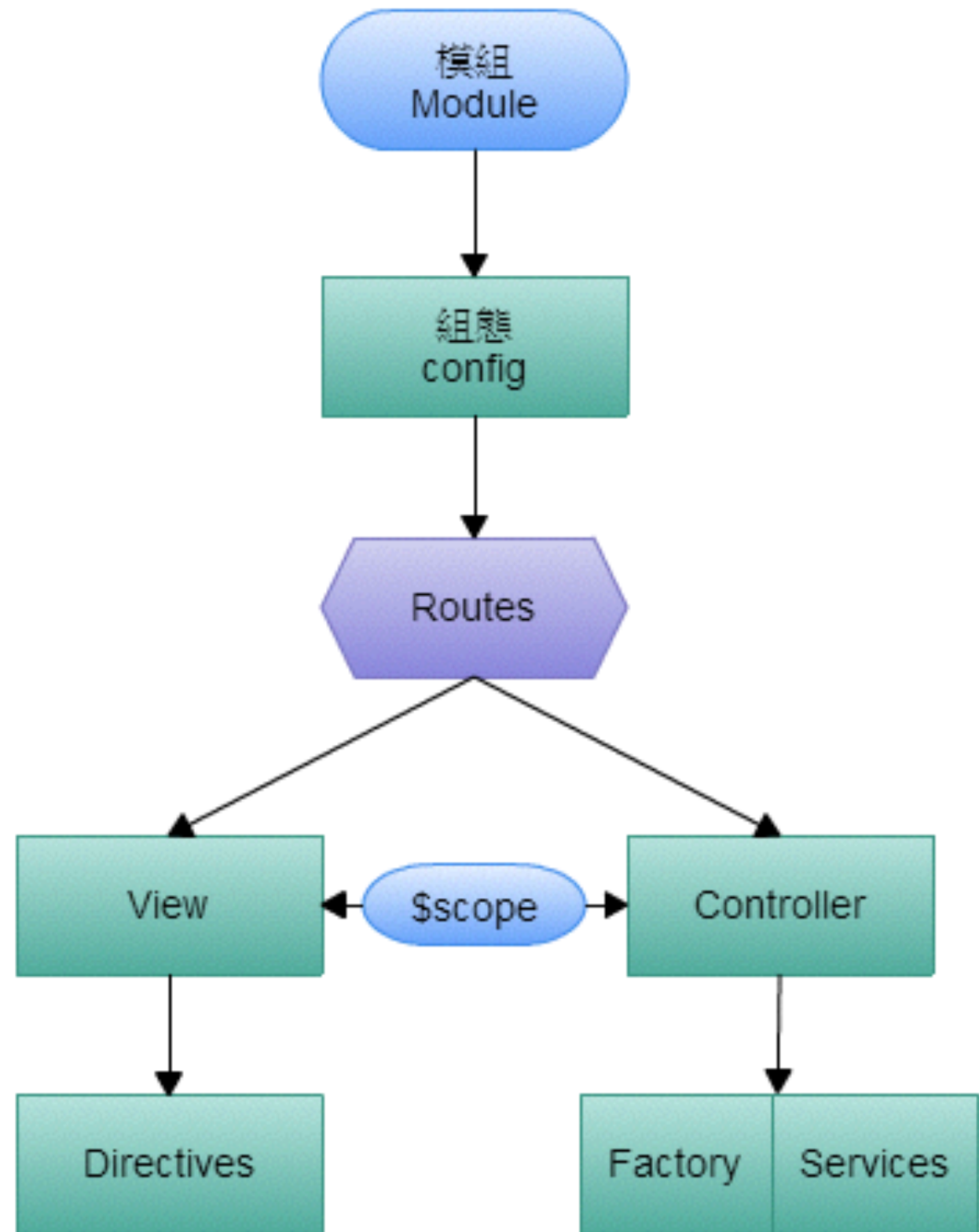
- 在程式碼或函式時常見到這種寫法
- 是javascript中一種立即可以執行的函數
- `(function() {.....})()`;
- 主要用途：
 - 減少全域變數的使用，全域變數區域化
 - 常被用於只執行一次的程式中，如初始化

範例

```
/**
 * @license AngularJS v1.3.15
 * (c) 2010-2014 Google, Inc. http://angularjs.org
 * License: MIT
 */
(function(window, angular, undefined) {'use strict';

/**
 * @ngdoc module
 * @name ngRoute
 * @description
 *
 * # ngRoute
 *
 * The `ngRoute` module provides routing and deeplinking services and directives for angular apps.
 *
 * ## Example
 * See {@link ngRoute.$route#example $route} for an example of configuring and using `ngRoute`.
 *
 * <div doc-module-components="ngRoute"></div>
 */
/* global -ngRouteModule */
var ngRouteModule = angular.module('ngRoute', ['ng']).
    provider('$route', $RouteProvider),
    $routeMinErr = angular.$$minErr('ngRoute');
```


AngularJS Directives



自訂 Directives

命名習慣

- 使用獨特的前置名稱
- 不要使用`ng`開頭的前置名稱
- 命名習慣
 - AngularJS的UI，通常使用`ui-`
 - 單位名稱(ex:ASCC) 可以使用`ascc-`

directives 格式

```
angular.module("demo", [])  
  .directive('myEditor', [function() {  
    return {  
      restrict: 'E'(A,C,M),  
      scope: {data: "@bikeData"},  
      template: "html code",  
      link: function($scope, $element, $attrs) { }  
    };  
  }]);
```


restrict 類別

- E(HTML Element)
`<my-editor></my-editor>`
- A(attribute) 預設
- `<div my-editor></div>`
- C(class)
`<div class='my-editor'></div>`
- M(mark 註解)
`<!-- directive:my-editor -->`
- 可以單獨或同時使用 : strict: 'EACM'

directives template

- template: "<div>...</div>"
- templateUrl: "views/oo.html"

directives 的 scope

- ◉ 語法：scope:
 - ◉ false：預設值，使用/繼承程式的\$scope
 - ◉ true：繼承程式的\$scope，但在directives異動，不會改變上層\$scope的內容
 - ◉ object：使用傳入{key:value}形式的物件
 - ◉ {directives中使用的dataName: 外部資料}
 - ◉ 外部資料來源可以有三種方式取得：
 - ◉ one-way binding：@
 - ◉ two-way binding：=
 - ◉ behavior binding：& (可以想成是reference)

scope 外部資料取得

- = : 表示傳入的資料會被當成 json object, 所有在 parent 的改變 directives 也會同步改變 (two-way)

```
<div ng-repeat="List in Lists">
```

```
  <my-editor player="List"></my-editor>
</div>
```

- scope: { myData: "=player" }

- scope: { myData: '= ' }

- <my-editor myData="List"></my-editor>

scope 外部資料取得(續)

- @ : 表示傳入的資料會被當成string，所有在parent的改變directives也會同步改變(one-way binding)

```
<div ng-repeat="List in Lists">  
  <my-editor player="{{List.name}}"  
  color="{{ List.color }}"></my-editor>  
</div>
```

- scope: { myData: "@player", myColor: "@color" }

scope 外部資料取得(續)

- & : Behaviour binding/method binding

```
<div ng-repeat="List in Lists">
```

```
  <my-Editor player="onSelect(List)"></my-Editor>
```

```
</div>
```

- scope: { myData: "&player" }

scope 外部資料取得(續)

- scope: {
 name: '@', // one-way binding
 color: '=', // two-way binding
 reverse: '&' // behavior binding
}
- <my-editor
 name="{ {name} }"
 color="color" // object
 reverse="reverseName()
>
- </my-editor>

Lab10

將ubike 資料顯示改用directives方式改寫

- `<div bike-datas="List.retVal"></div>`
- `scope: { bikeDatas: '=' }`
- directives 會建立stockData供內部使用
- stockData 的值會被設為 HTML 端bike-datas的傳入值 (List.retVal)

Lab10-2 傳入第二組參數

- 將 `ubike` 資料顯示改用 `directives` 方式改寫
- 增加下拉選單 `filter` 功能，讓使用者可以選擇地區

Lab.10-1 directives 練習



- 比照ubike範例，將水庫進水量也以
directives 方式改寫（使用restrice:'E' 方
式）
- <http://140.109.6.150:10080/>

Link function

- directives的Link function 就像是view與controller內function的關係
- 定義directives需要的API及資料異動
- 每次directives被啟動，就會執行Link function，各個directives Link function 各自獨立，互不干擾
- 傳入參數固定(\$scope, \$element, \$attrs)

Lab11 directives 範例

- 增加與directives互動的getBalance function，取得UBike空位剩餘數

Link function與scope的迷思

- 既然Link function預設會傳入parent 的 `$scope`，為何我們還需要`scope: {}` 指定資料是由外部傳入？
- 若外部`$scope`傳入的名稱改變，忘了更改到 `directives` 內部的變數名稱時，程式會出錯
- `directives` 有可能會誤用相同的參數或不小心改寫資料

Directives 使用tips

- 若程式碼少，盡量使用template而非templateURL，因為templateURL會多一次request
template: '<div>.....</div>'
templateURL: '/tmp/a.html'
- 盡量使用ng-if 而非 ng-show/hide，ng-if只會在true時才建立DOM元素
- 盡量把directives當作是顯示資料UI使用，主要的商業邏輯還是放在controller中
- 為了降低與parent的關聯，盡量少用scope:&

form 資料檢核

form

- HTML 元件
- 表單內的元件，盡量以{ } 的方式來蒐集資料
- 使用ng-submit來取代submit
- 盡量使用AngularJS驗證方式

form example

- ```
<form ng-submit="submitIT()">
 <input ng-model="user.userName">
 <input type='password' ng-model='user.password'>
 <input type='submit' value='submit'>
</form>
```
- Angularjs 會自動幫我們建立 user object



# 表單欄位驗證設定

- `ng-required` : 此欄位為必填欄位
- `ng-minlength` : 最少必須輸入資料的長度
- `ng-maxlength` : 最常可輸入的資料長度
- `ng-pattern` : 檢查資料的類型
- `type='email'` `type='number'` `type='date'`  
`type='url'`



# 表單狀態 form state

- `$invalid` : 檢查是否有欄位不符合規則
- `$valid` : 檢查是否所有欄位都符合規則
- `$pristine` : 是否該輸入的欄位都有輸入值
- `$dirty` : 是否有更動任何內容
- `$error` : 各欄位的檢查結果



# Lab11. form 驗證練習



- 使用bootstrap設計三個輸入欄位（姓名、Email、自訂密碼）
- 依序加入：必填、Email驗證、密碼驗證
- 送出按鍵需要都通過驗證後，才能enable

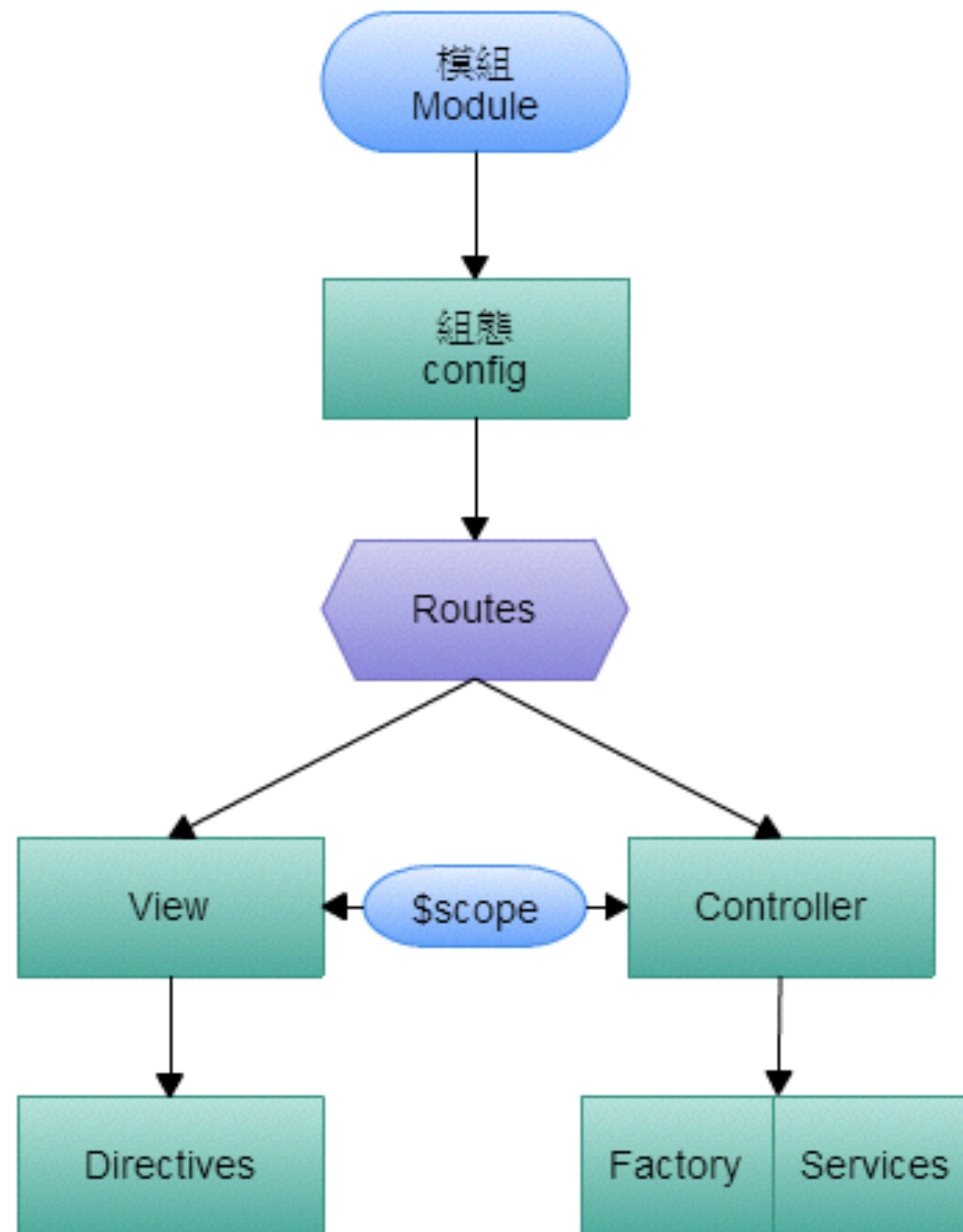


# form設計上注意事項

- 只讓使用者填寫需要的資訊
- 使用single-column layout 讓使用者一次專注一個輸入項目
- 馬上反應的錯誤訊息
- 明確的說明及提示內容
- 不要使用CAPTCHA
- 最好讓行動裝置也能填寫



# AngularJS config · Routing





# AngularJS Config

- 用來設定程式組態
- 設定需要使用的模組
- ```
angular.module('myApp', [])  
  .config(function() {  
    // 應用程式的相關設定  
    .....  
  });
```


AngularJS Routing

何謂Routing

- 透過改變瀏覽器的URL來改變頁面內容
- http://xxx.xx.xx.xx/ => index.html
- http://xxx.xx.xx.xx/pagea.html
- http://xxx.xx.xx.xx/dir/page.html

SPA Routing

- 改變頁面內容時，並不會改瀏覽器網址，僅更改需要異動的部分
- `http://xxx.xx.xx.xx/index.html`
- `http://xxx.xx.xx.xx/index.html#pageA`
- `http://xxx.xx.xx.xx/index.html#pageB`

AngularJS route

- AngularJS 本身內建強大的routing功能
- 屬於service中的providers類別
- 使用前需要額外include angular-route.js
函式庫

如何在AngularJS啟動routing

- 連上官網：<http://angularjs.org/>
- 開啟首頁的download
- 找到：Extras Browse additional modules
- 下載angular-route.js
- `<script src="libs/angular-route.js"></script>`

route 格式

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


UI 的搭配 ng-view

- 在 `config` 設定繞送路徑，需要搭配一個 `container` 才能讓網頁內容顯示在其中
- `directives` 的用法及概念
- `ng-view`
`<div ng-view></div>`
- 每個頁面只能有一個 `ng-view`

改寫myappjs-config

```
angular.module('myApp', ['ngRoute'])  
  .config(function($routeProvider) {  
    $routeProvider  
      .when(  
        "/",  
        {  
          controller: "helloworld",  
          templateUrl: "views/index.html"  
        })  
      .otherwise({redirectTo: "/"});  
  });
```


改寫myappjs-前端

- `<div ng-view></div>`