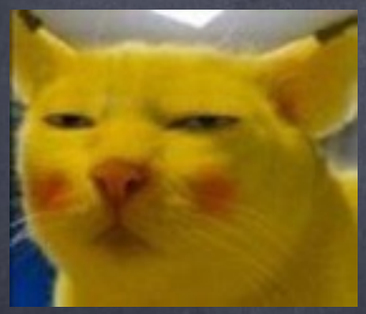
AngularJS 實作與應用

Day 5



資訊服務處劉智漢

2015.05.13

@中央研究院 人文館

day4 複習

- Routing 原理
- ngRouting 搭配config進行rout設定
- ngRouting、ngTemplate、uiRouter 用法

uiRouter

uiRouter 介紹

- 非AngularJS官方提供的函式庫
- 為狀態導向，很多情況下不會改變Url
- 適用時機：
  - 頁面多個view中需要各自獨立處理且各自有各自不同的Url來源時
  - 頁面上的view需要各自針對不同情況顯示或隱藏
- 多使用於手機版本上

- 選擇release目錄，下載angular-ui-router.js

- 在網頁中

如何使用uiRouter

1.<script src="libs/angular-ui-router.js"></script>
2.<div ui-view>

- angular.module('myApp', ["ui.router"])
.config($stateProvider) {
    $stateProvider
        .state('myState', {
        }
};

```
angular.module('myApp', ["ui.router"])
.config($stateProvider) {
    $stateProvider
        .state('myState', {
        }
};
```

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

uiRoute 語法

- $stateProvider.state('stateName', {
    template/templateUrl:"",
    controller: function or "functionName",
    controllerAs:"asName",
    controllerProvider:function //return name
    resolve: function,
    onEnter: function,
    onExit: function,
    parent: "parent state name"
});

template

- $stateProvider.state('stateName', {
    template:" I'm template "
    templateUrl: "template/a.html"
    templateUrl: function($stateParams) {
      return "template/"+ $stateParams.page;
    }
  });

- templateUrl若使用function，則必須要回傳url(字串)

controller

- $stateProvider.state('stateName', {
    controller: 'myController',
    controller: function($scope) {
      $scope.title = "Scope title";
    }
  });

- 若template未定義，則controller不會啟動

resolve

- $stateProvider.state('stateName', {
    resolve: {
      key: factory(string|function)
    }
  });

- 若factory為string，則表示為service name

resolve 範例

- $stateProvider.state('stateName', {
    resolve: {
        keyA: function() { return something},
        keyB: function() { return something}
    },
    controller: function($scope, keyA,keyB) {
    }
});

- 若factory為string，則表示為service name

onEnter、onExit

- $stateProvider.state('stateName', {
    resolve: { title: 'Hi Title' },
    onEnter: function(title) {
    },
    onExit: function(title) {
    }
});

- 若factory為string，則表示為service name

如何啟用state

- $state.go()
  $state.go('lazy.state', {a:1, b:2},
  {inherit: false});
- ui-sref：透過Url的語法來啟動state
- url 呼叫

Nested states

- $stateProvider
  .state('rootState', {})
  .state('rootState.subState, {});

- $state.Provider
  .state('rootState', {})
  .state('subState', {
    parent: 'rootState';
  });

lab days_0.
Texted state練習

- Nexted state 示範
- 手機程式示範及說明

multi name views

Root

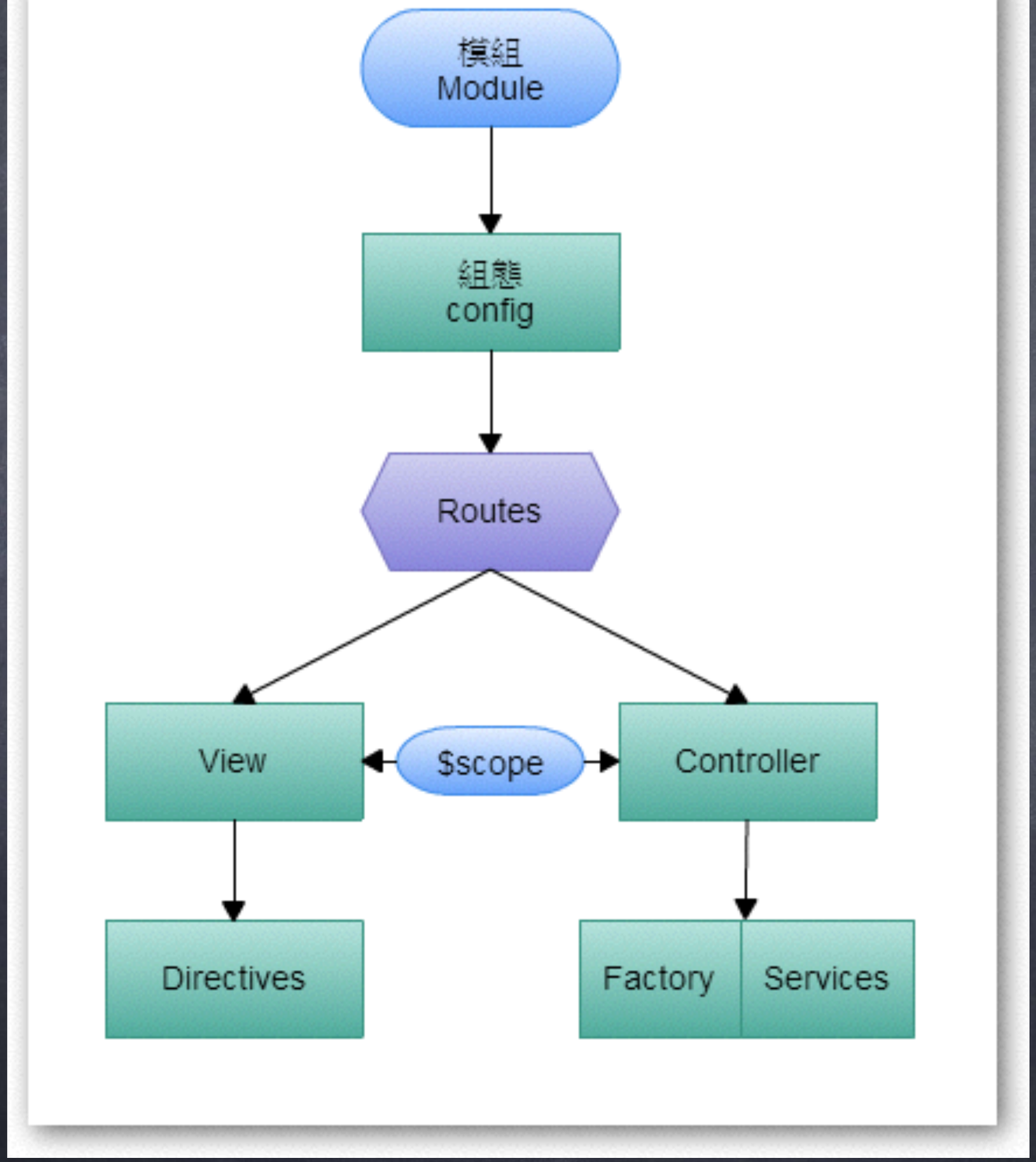
filters

Type: Employee Age Range 3 to 35 Date: 01/01/2013

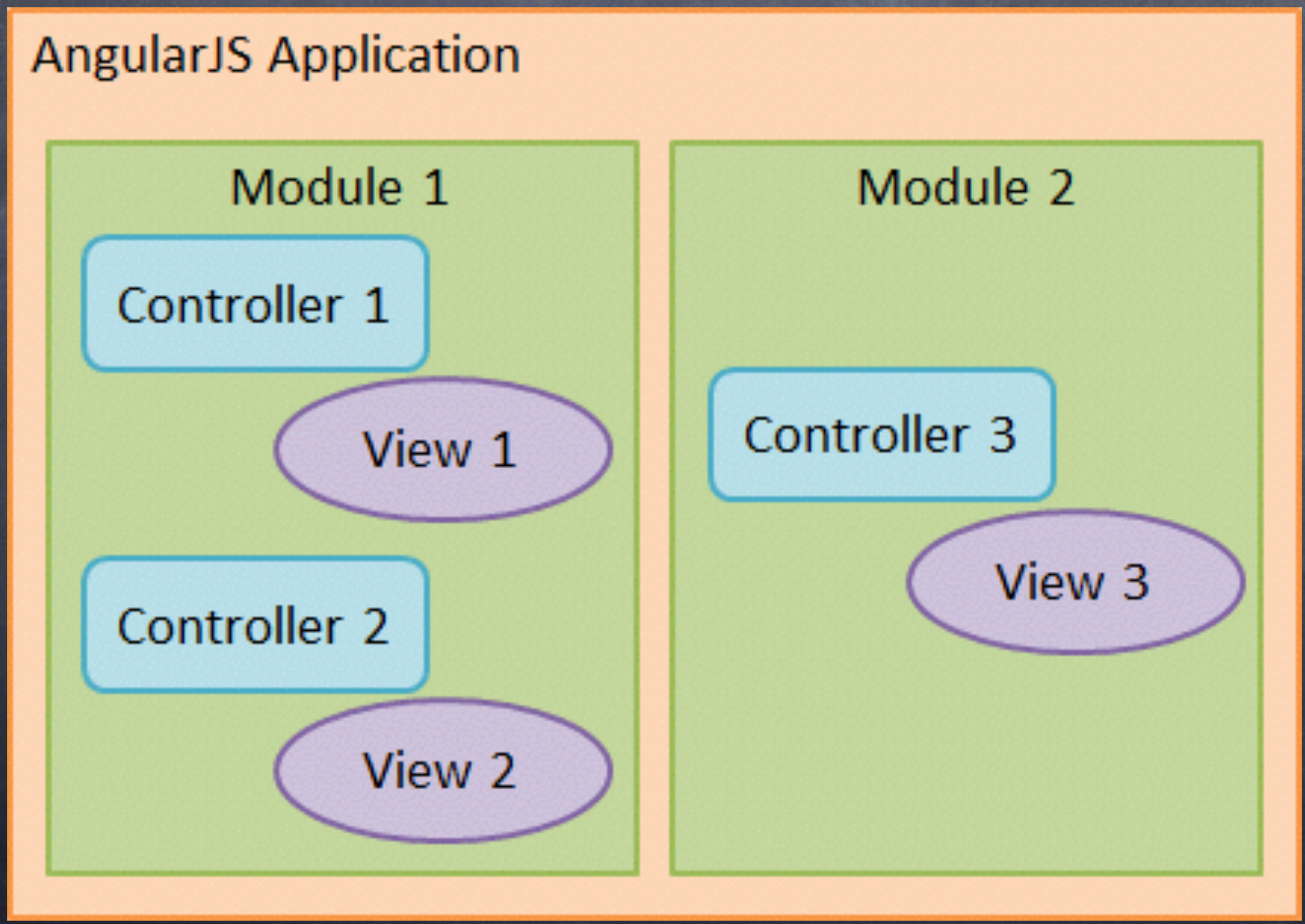
tabledata

Name (job title)	Age	Nickname	Employee
Giacomo Guilizzoni Founder & CEO	34	Peldi	☑
Guido Jack Guilizzoni	4	The Guids	☐
Marco Botton Tuttofare	31		☑
Mariah Maclachlan Better Half	35	Patata	☑
Valerie Liberty COO, WOW! Division	:)	Val	☑

graph


模組
Module
組態
config
Routes
View
$scope
Controller
Directives
Factory
Services

controller 介紹



controller 主要負責工作

- 為UI畫面至主機抓取合適的資料以便呈現(取得那些資料)
- 決定那些資料要呈現在畫面上（如何顯示）
- 決定呈現的邏輯，如HTML元件、UI畫面等
- 與使用者互動及反應(ngClick、ngChange...)
- controller 會與HTML元件作某種關係的連結
- 沒有與HTML元件連結的程式碼建議寫在service中

controllerAS 介紹

Controller

```
function InvoiceController {
  this.pay = function…
  this.total = function…
  this.cost=2.5;
  this.qty=1;
}
```

Scope

```
invoice:
  new InvoiceController
```

View (DOM)

```
<div ng-controller=
  "InvoiceController as invoice">
  <input ng-model=“invoice.qty">
  <input ng-model=“invoice.cost">
  {{invoice.total('USD')}}
  <button ng-click=
        "invoice.pay()">
</div>
```

controller 語法

- ```
  angular.module('myApp', [])
  .controller('ctrlName', ['DIs', function(DIs)
  {
  }]);
  ```
  ```

controller 使用上注意事項

- 盡量使用controllerAs語法
- 使用this需小心，盡量先指到一個區域變數(self)再開始使用

```
var self = this;
```

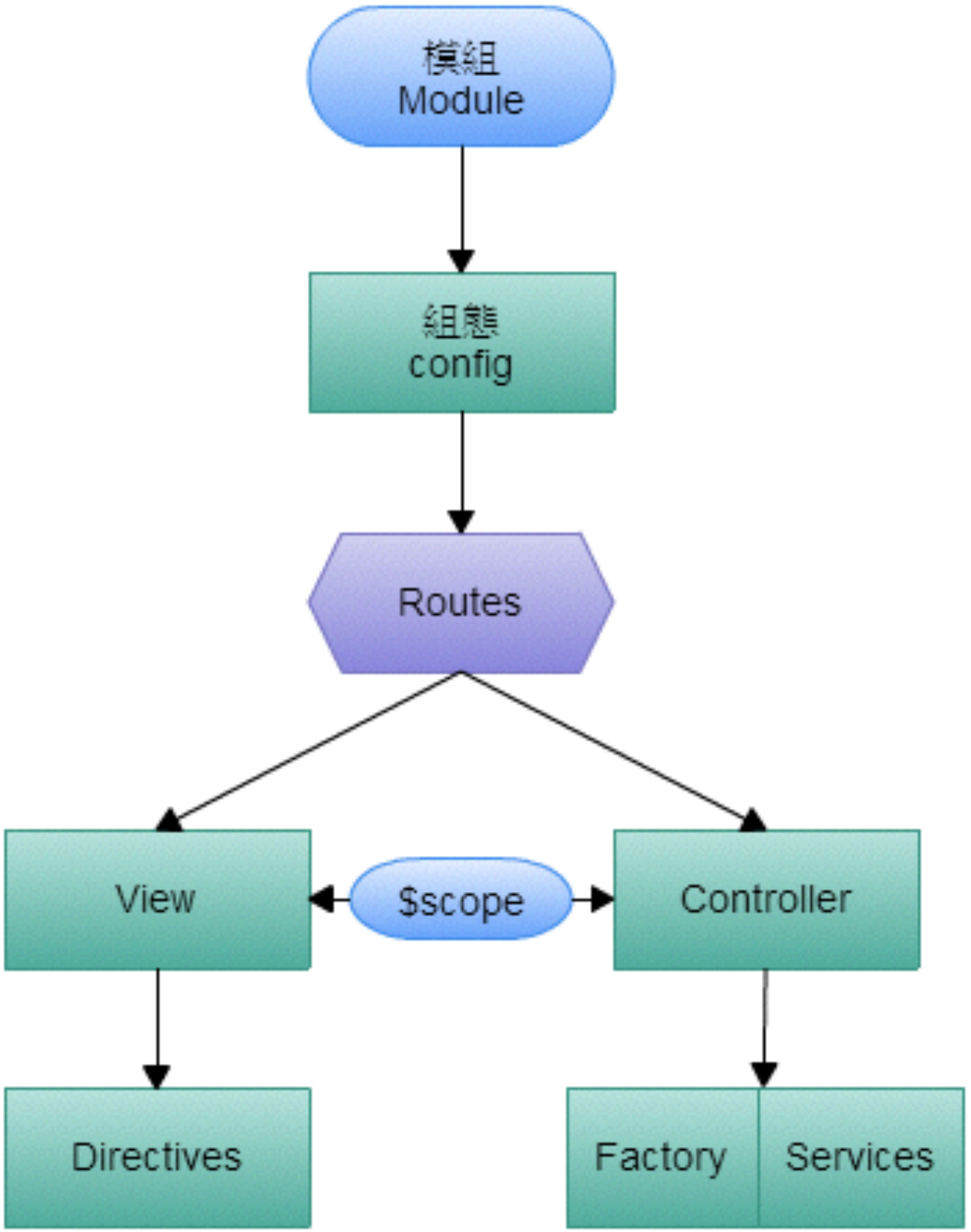
- 盡量不要在controller中使用jquery語法

多個controllers使用

- app.js
  angular.module('myApp', ['starter.controllers', ‘starter.services'])

- controllers.js

```
angular.module('starter.controllers', [])
  .controller('ACtrl', function($scope) {})
  .controller('BCtrl', function($scope) {})……..
```


模組
Module
組態
config
Routes
View
$scope
Controller
Directives
Factory
Services

service

- AngularJS 提供的相關服務
- 為AngularJS中負責特定任務的獨立物件或函數
- 每個service只會被實體化一次，可重複被使用
- 相同行為、共享狀態、快取、factories等都適合使用service撰寫
- 整體service V.S services

service 使用原則

- 與工作邏輯有關而與畫面無關的程式碼
- 可重複使用
- 主要提供伺服器存取資料、驗證邏輯、資料儲存
- 可在應用程式中任意地方被呼叫，
  如:controller、directives、filters等

內建的service

- $location：前面已經介紹過
- $http
- $sce

$http 介紹

- 底層仍是透過XMLHttpRequest進行遠端連線
- XMLHttpRequest->AJAX->Jquery
- HTTP 語法：
  - GET(select)、POST(insert)、PUT(update)、DELETE(delete)

$http 語法

- $http.get(url), $http.post(url, data)
- $http({url:url, method:"post", param:params})
- 成功、失敗函數

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

http status回應代碼

- 200：OK
- 201：建立成功（搭配POST）
- 400：Bad Request
- 401：未認證
- 403: Forbidden
- 404：not found 找不到

$q 物件

- AngularJS 負責：
  - 註冊：var qjob = $q.defer();
  - 通知：qjob.resolve("ready");
- 程式設計師：
  - 取得資料：qpromise = qjob.promise;
  - qpromise.then("do something");
- 參考：days/labs_4

$sce

- AngularJS 基本上不允許程式設計師直接透過參數將HTML code透過變數顯示在畫面上
- 正確的作法是透過directives來實現
- 實作上，為了輸出簡單的HTML CODE實作directives，太麻煩且不符合經濟效益
- 所以 AngularJS 提供$sce service 來達到這件事
- 主要功能為用來輸出HTML程式碼

$sce 範例

- HTML 端：

```
<div ng-bind-html="myHtml"></div>
```

- javascript 端：

```
angular.module('myApp', [])
.controller('helloWorld', ["$scope", "$sce",
function($scope, $sce)  {
    var htmlCode = "<b>$sce範例</b>";
    $scope.myHtml = $sce.trustAsHtml(htmlCode);
});
```

自訂的service

- angular.module('myApp', [])
  .service('serviceName', [', function() {
  }]);

- 使用Service時，會得到新的instance，用法類似:
  new service()...

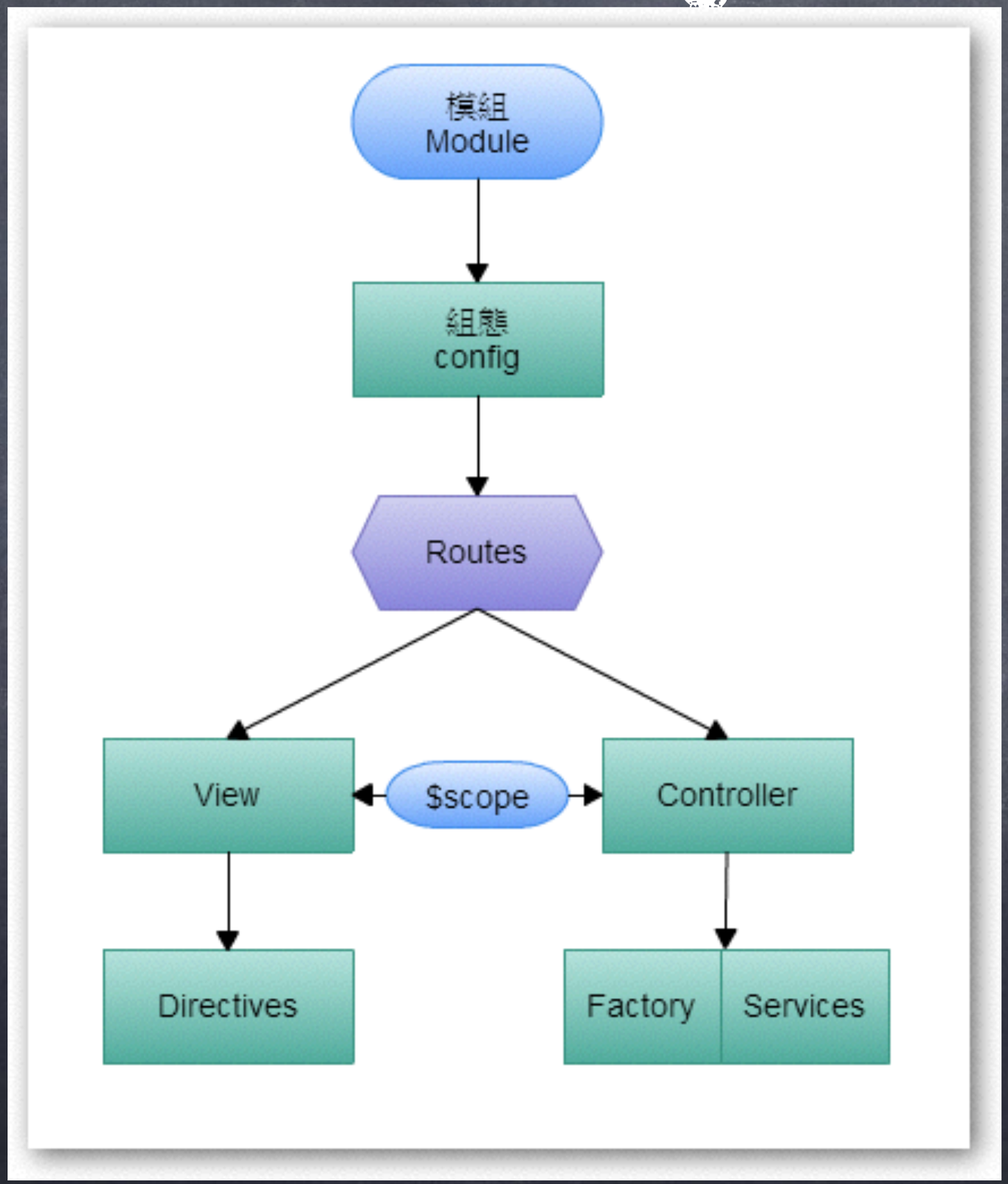
lab days_2.
service 範例

- 簡單service撰寫法
- 輸入一個內容三次後，顯示該內容

labs_3 service練習

- 將ubike服務移到service中
- 建立一個名為 ubike 的 service
- 在ubike.service中設計CRUD功能
  - get、post、put、delete

factory


模組
Module
組態
config
Routes
View
$scope
Controller
Directives
Factory
Services

factory

- angular.module('myApp', [])
  .factory('factoryName', [function() {
  }]);
- 建立factory時，會得到value of the returned

service V.S factory相似點

- 兩者都允許使用者建立可以在應用程式中重複使用的物件

- 兩者建立Instance的方式都採用singleton（一個service/factory只有一個Instance）

service V.S factory相異點

- factory回傳由functions組合的物件;service 則是回傳物件的constructor functions(use new)
- factory call function; service new function

service V.S factory 範例

- see labs_1

service V.S factory 誰好？

- javascript 程式設計師通常喜歡使用function來回傳物件（factory 作法）
- CoffeeScript、TypeScript則建議使用constructor function（service作法）

providers

- angular.module('myApp', [])
  .provider('providerName', [function() {
  }]);
- 建立provider時，可以從provider().$get()
- provider的好處是可以在應用程式組態階段時，進行相關設定（例如：config中的$routeProvider）

lab days_4.
services 範例

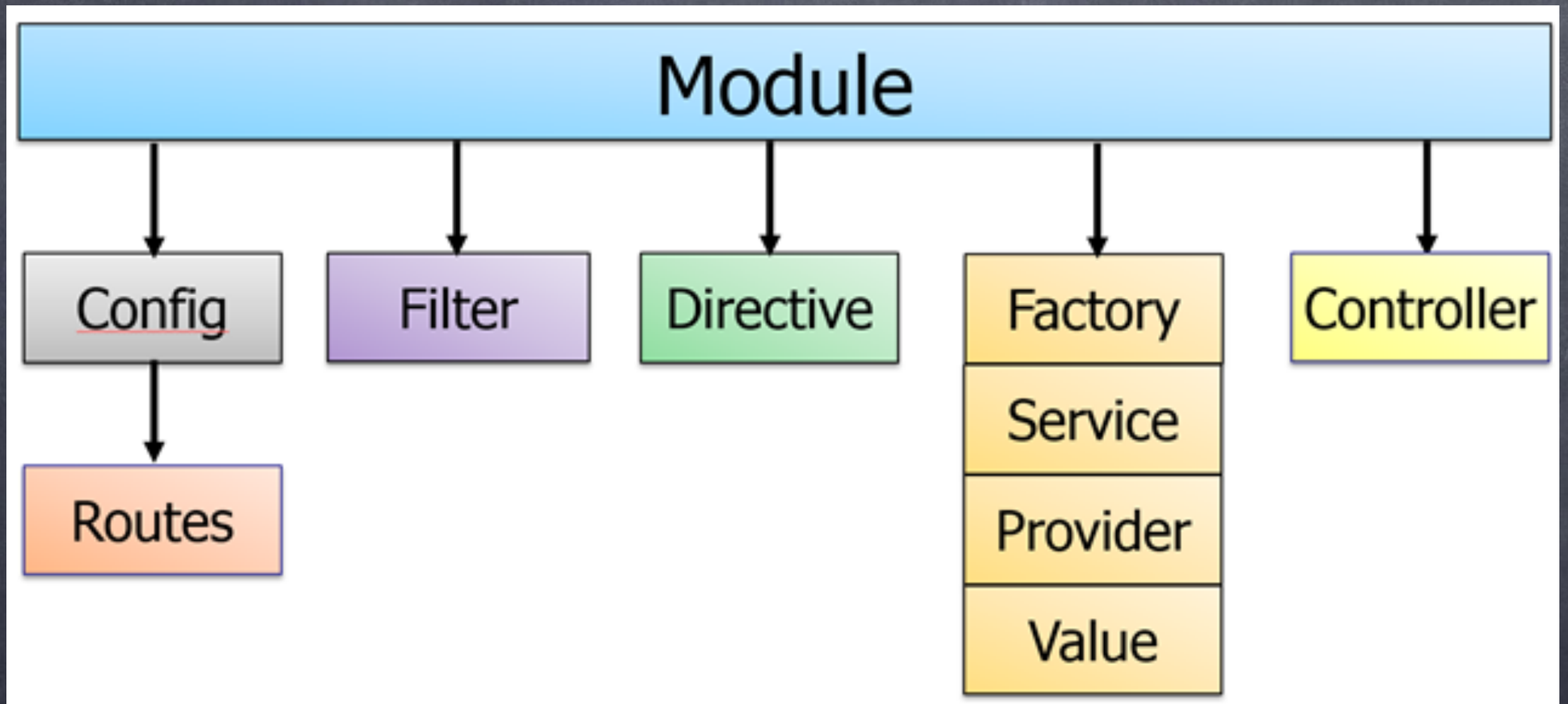
- service V.s factory v.s provider
- http://jsbin.com/ohamub/1/edit?html,output

lab days_5.
share data 範例

- 不同模組間分享資料可以透過service 或 factory 來進行

AngularJS service 類別

- service : return the actual function
- factory : return the function's return value
- providers : return the output of the function's $get function


Module
Config
Filter
Directive
Factory
Controller
Routes
Service
Provider
Value