使用Angular.js開發企業級應用程式

陳小風

tw MVC

講者介紹 – 陳鋒逸（小風）

- 91mai就要買資深軟體工程師
- JSDC 2013 講師
- TechDay 2014 講師
- ✓IT邦幫忙鐵人賽優選
  - 使用Jenkins打造.Net CI Server(2012)
  - 使用Asp.Net MVC打造Api(2013)



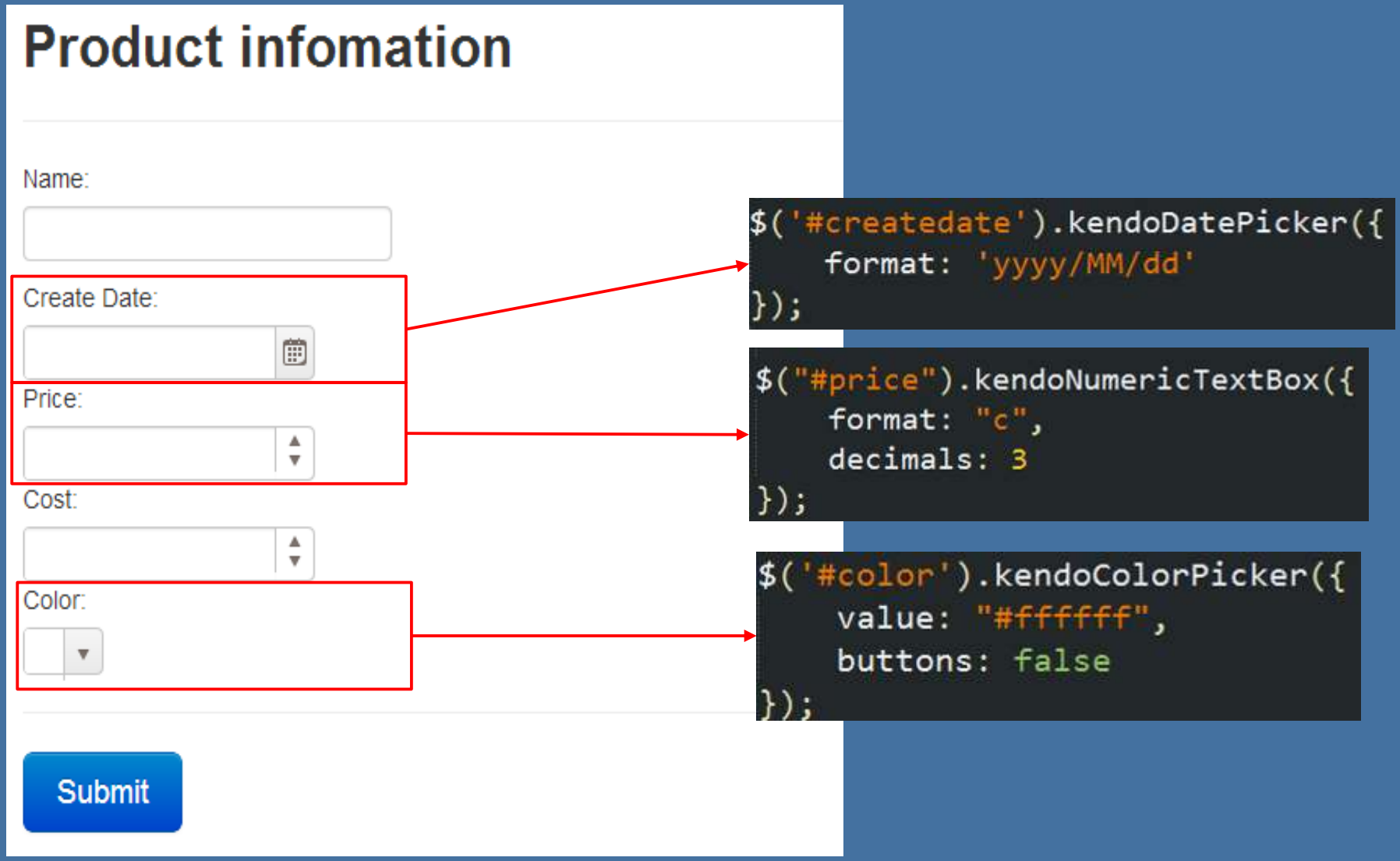
九易宇軒

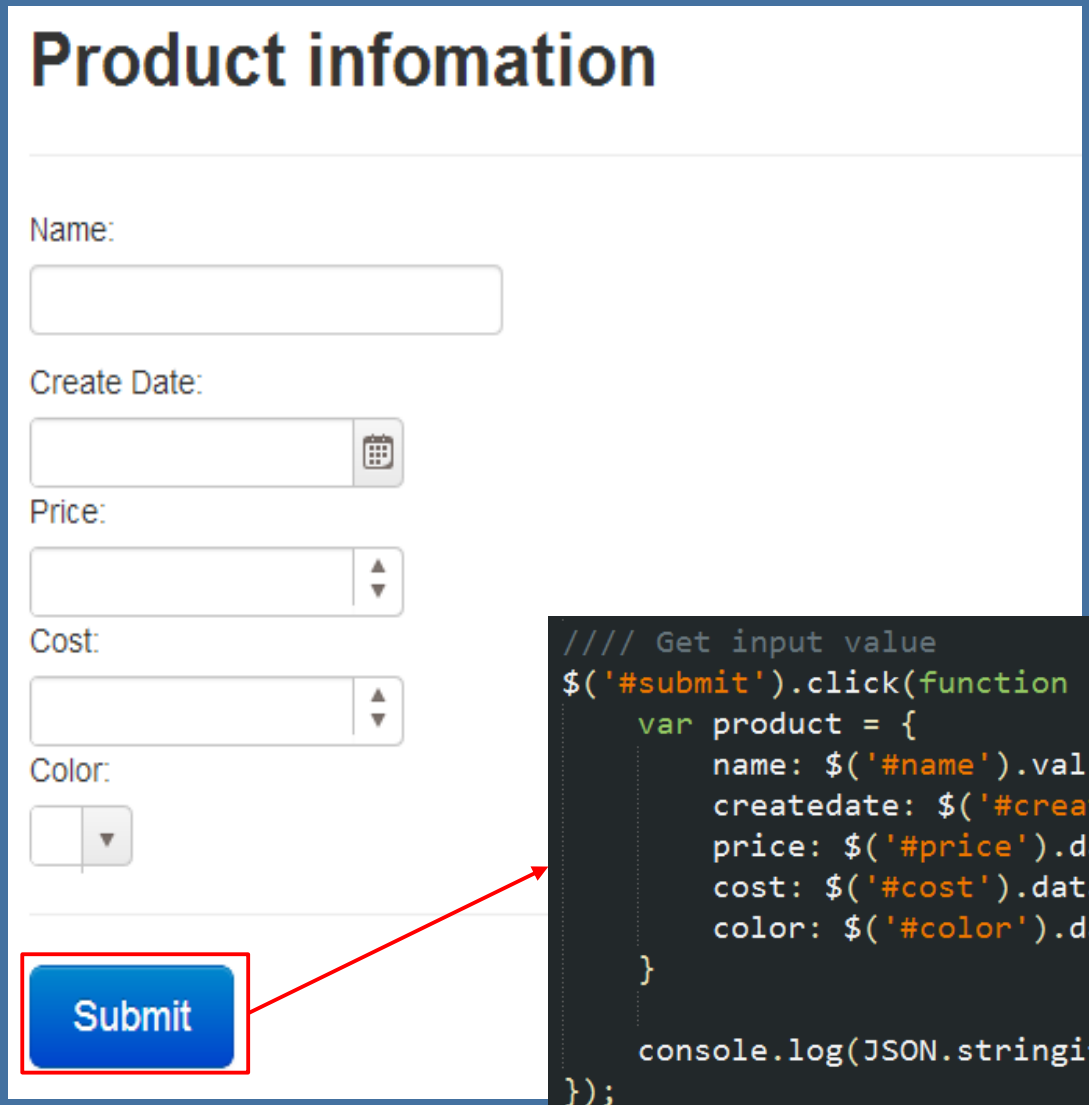


什麼是Angular.js？

Angular.js

- ✓開源 JavaScript函式庫，由Google維護
- ✓幫助我們快速開發，良好的支援測試
- ✓支援開發單一頁面應用程式（SPA）
- ✓良好的支援物件導向程式設計，讓程式碼耦合性降低
- ✓友善的支援JQuery


Product infomation
Name:
Create Date:
Price:
Cost:
Color:
Submit
$('#createdate').kendoDatePicker({
format: 'yyyy/MM/dd'
});
$("#price").kendoNumericTextBox({
format: "c",
decimals: 3
});
$('#color').kendoColorPicker({
value: "#ffffff",
buttons: false
});




```
//// Get input value
$('#submit').click(function () {
    var product = {
        name: $('#name').val(),
        createdate: $('#createdate').data('kendoDatePicker').value(),
        price: $('#price').data('kendoNumericTextBox').value(),
        cost: $('#cost').data('kendoNumericTextBox').value(),
        color: $('#color').data('kendoColorPicker').value()
    }

    console.log(JSON.stringify(product));
});
```

JQuery的特色



- 直覺
- 與DOM綁定深
- View修改，Code也要改
- 程式碼較多

```
$('#createdate').kendoDatePicker({
    format: 'yyyy/MM/dd'
});
```


參考: http://jquery.com/

JQuery的特色



- 直覺
- 與DOM綁定深
- View修改，Code
- 程式碼較多

```
//// Initialize input control
$('#createdate').kendoDatePicker({
    format: 'yyyy/MM/dd'
});

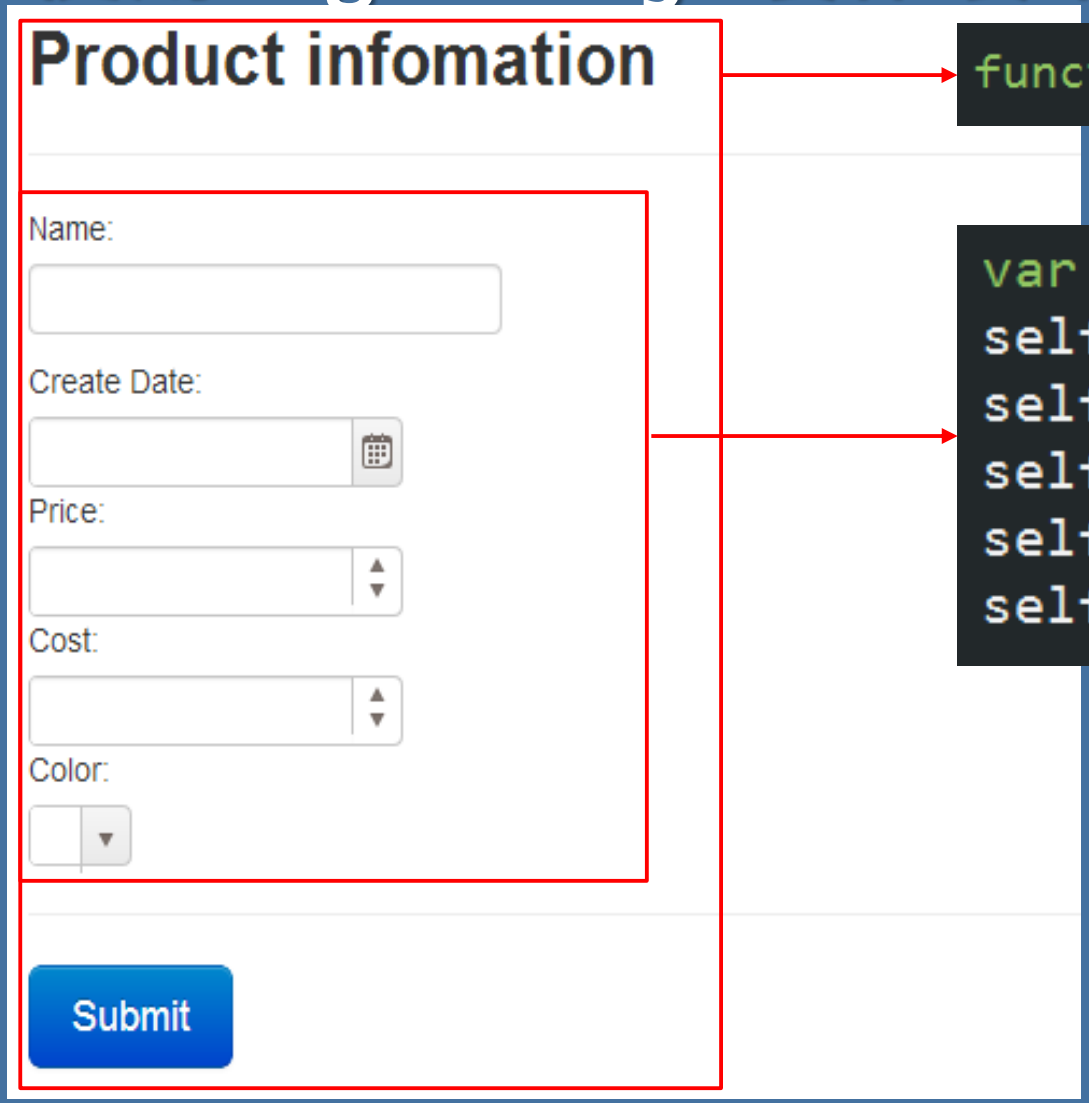
$("#price").kendoNumericTextBox({
    format: "c",
    decimals: 3
});
$('#price').data('kendoNumericTextBox').value()

$("#cost").kendoNumericTextBox({
    format: "c",
    decimals: 3
});
$('#cost').data('kendoNumericTextBox').value()

$('#color').kendoColorPicker({
    value: "#ffffff",
    buttons: false
});
```


參考: http://jquery.com/

使用Angular.js製作表單

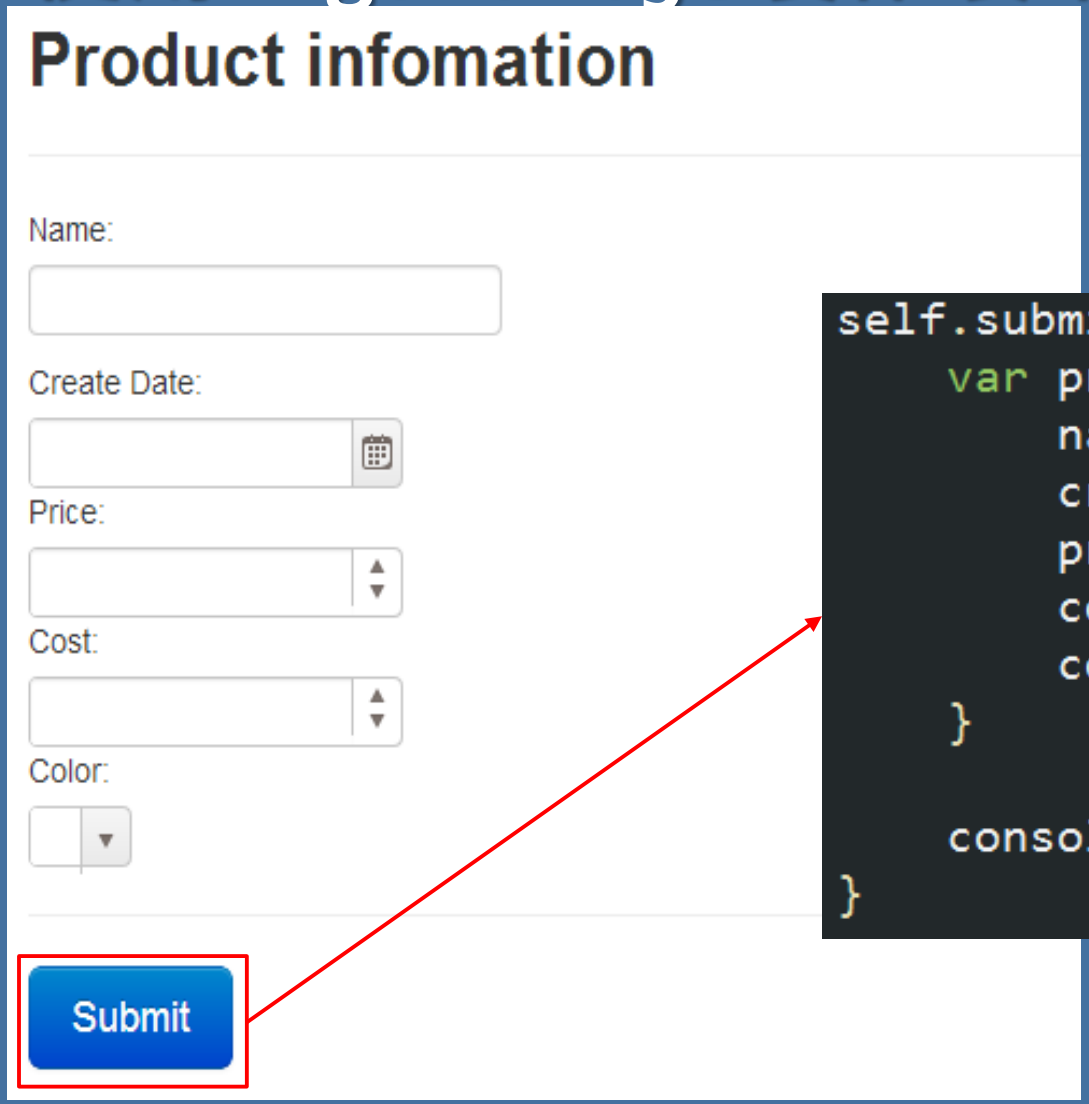



```
function MainController() {
```

```
var self = this;
self.name = '';
self.createdate = '';
self.price = 0;
self.cost = 0;
self.color = '#ffffff';
```

Model

使用Angular.js製作表單

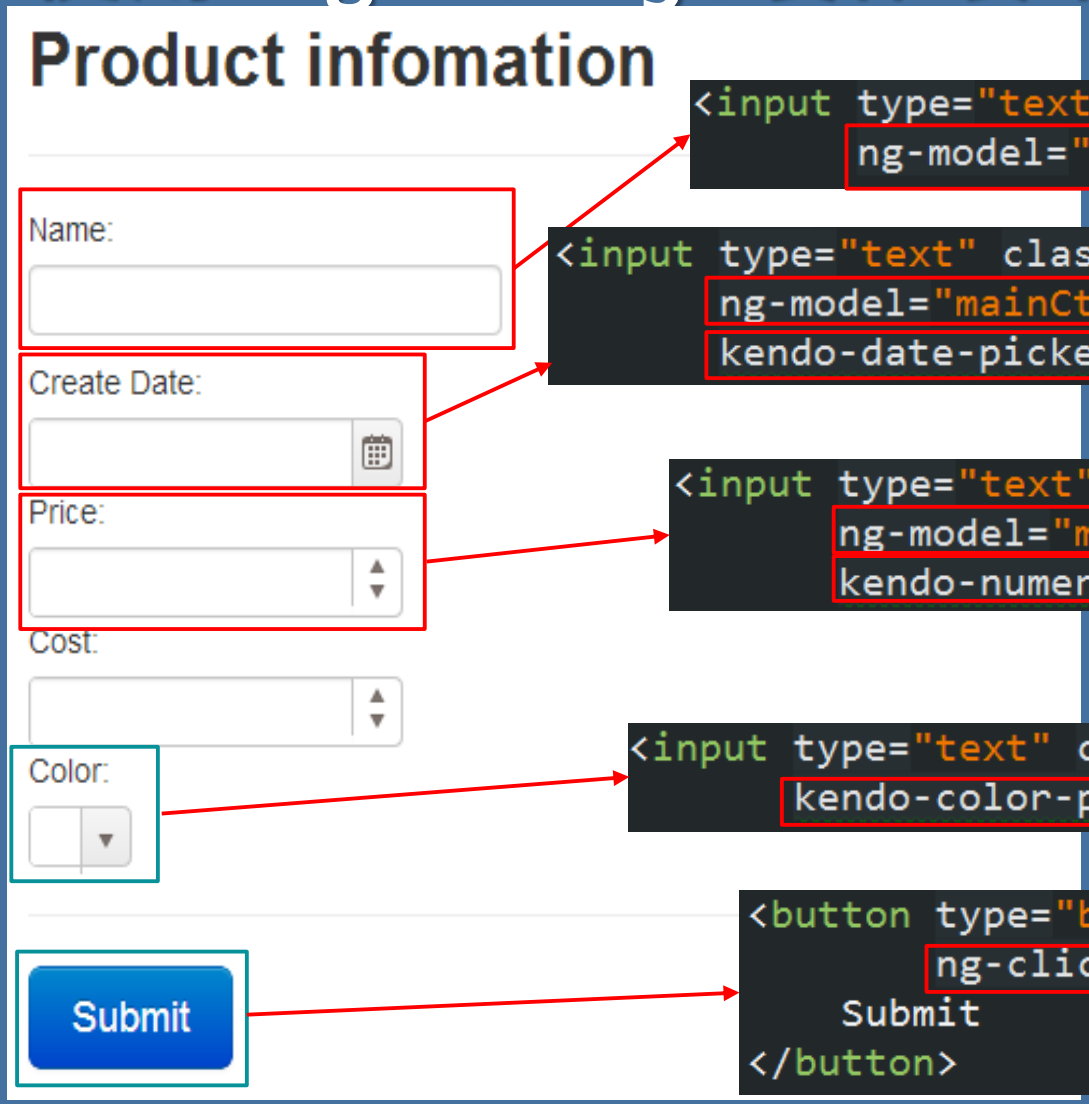



```
self.submit = function () {
    var product = {
        name: self.name,
        createdate: self.createdate,
        price: self.price,
        cost: self.cost,
        color: self.color
    }

    console.log(product);
}
```

Action

使用Angular.js製作表單




```
<input type="text" class="form-control" id="name"
       ng-model="mainCtrl.name">
```

```
<input type="text" class="form-control" id="createdate"
       ng-model="mainCtrl.createdate"
       kendo-date-picker k-format="'yyyy/MM/dd'">
```

```
<input type="text" class="form-control" id="price"
       ng-model="mainCtrl.price"
       kendo-numeric-text-box >
```

```
<input type="text" class="form-control" id="color"
       kendo-color-picker ng-model="mainCtrl.color">
```

```
<button type="button" class="btn btn-primary"
        ng-click="mainCtrl.submit()">
    Submit
</button>
```

宣告式開發



- 定義Controller
- 用Directive套用Library
- 繫結Model
- 互相連動
- View和Code分離

```
<input type="text" class="form-control" id="price"
       ng-model="mainCtrl.price"
       kendo-numeric-text-box >
```

小結

- 套版快
- 分工明確
- 好維護



更方便的開發方式

工程師的困擾

- 時間不夠
- 需求吹又生
- 功能不好用
- 明天能好嗎？

靈異現象?

```
var setting = {
    show: true,
    override: false,
}

$('createDateTime').datetimePicker(setting);
```

```
> window.setting
< 2014-09-03 10:57:12.899 Object {show: true, override: false}
>
```

但是Plug In越用越多…

```
var setting = {
    message: 'Hello World'
}

$.blockUI(setting);
```

```
var setting = {
    width: 100,
    height: 200
};

$('.widget').size(settting);
```

Setting會是什麼？



https://farm4.staticflickr.com/3099/2610267740_c718f6a644_o.jpg

到處都是Callback

```
var a = 1, b = 2, c = 3, d = 4;
$.ajax({
    url: '/api/calculate',
    type: 'GET',
    data: { a: a, b: b },
    dataType: "json",
    success: function (response1) {
        $.ajax({
            url: '/api/calculate',
            type: 'GET',
            data: { a: response1.sum, b: c },
            dataType: "json",
            success: function (response2) {
                $.ajax({
                    url: '/api/calculate',
                    type: 'GET',
                    data: { a: response2.sum, b: d },
                    dataType: "json",
                    success: function (response3) {
                        console.log($("#result"));
                        $("#result").text(response3.sum);
                    }
                });
            }
        });
    }
});
```

折扣=>運費=>分期

我...在第幾層？



梭哈式開發法

- 一個頁面，一隻程式！
- 職責分離？
- 物件導向？
- 設計模式？

四字真言

會　動　就　好



http://photo.l99.com/bigger/11/1341838228447_n3ex30.jpg

http://pic.pimg.tw/z642319240/1383384010-3293348412_n.jpg



混亂的根源

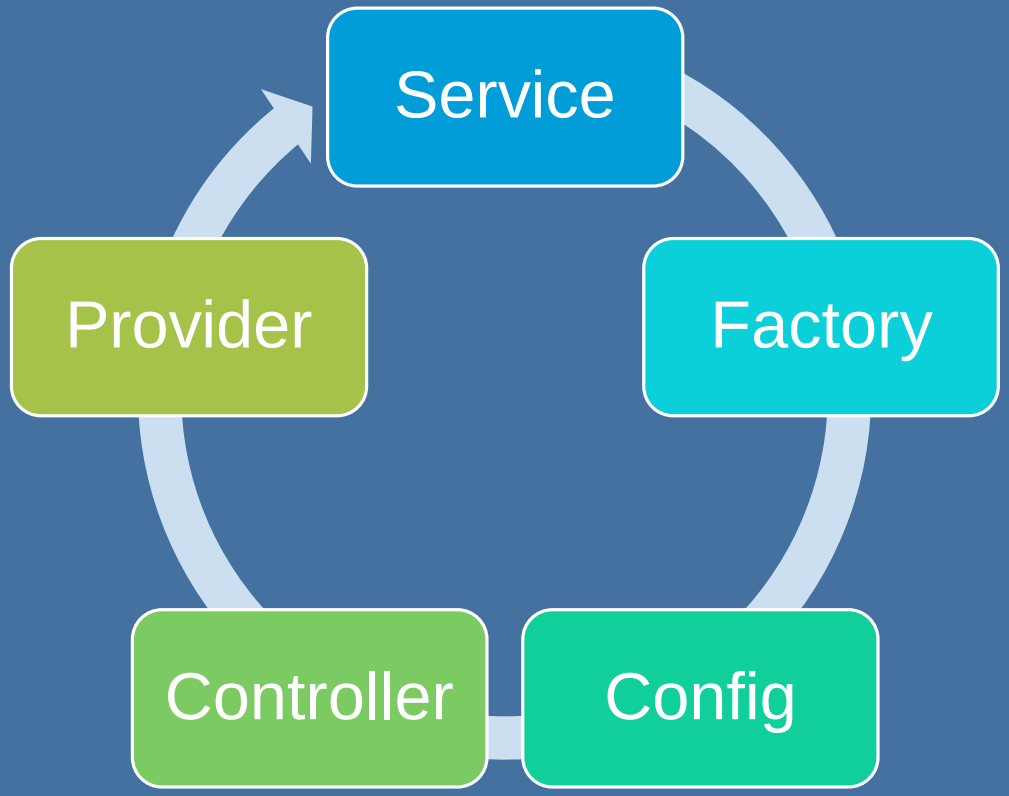
- 缺少Coding Standard
- 沒有統一的開發流程
- 每個人使用不同的寫法
- 程式碼零散不明確



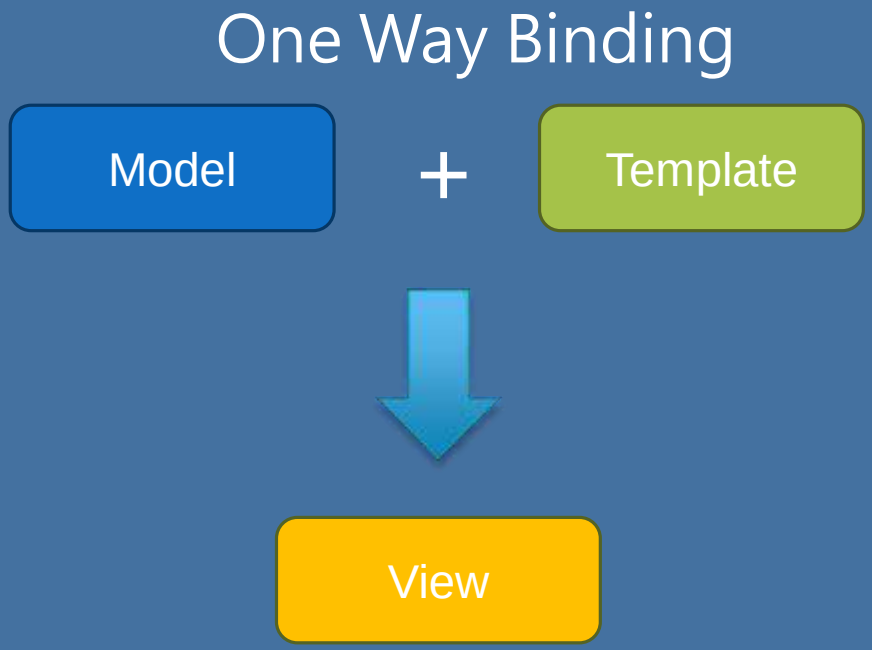
開發三步驟

- 決定角色
- 撰寫劇本
- 交給導演

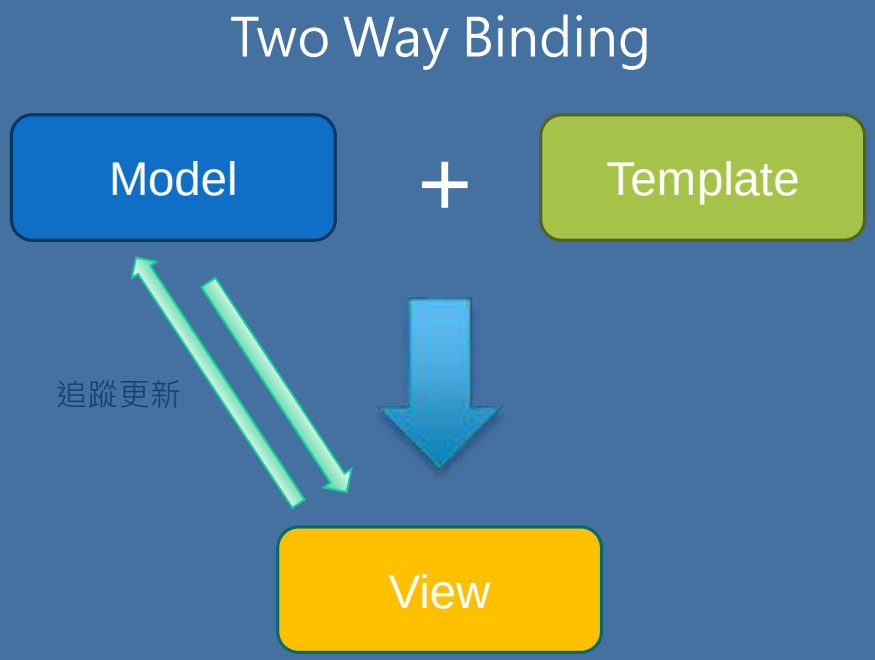




單向繫結

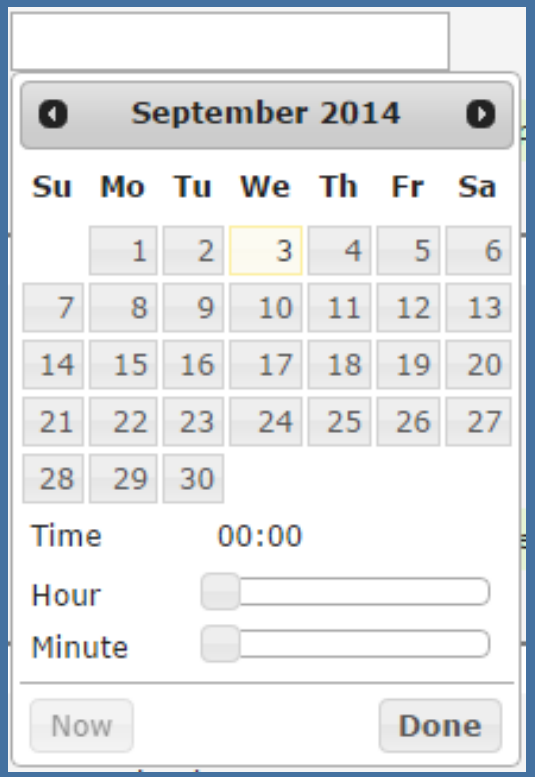


雙向繫結





把複雜的語法打包




```
app.directive('datetimePicker', function () {
    return {
        restrict: 'A',
        require: 'ngModel',
        link: function (scope, element, attrs, ngModelCtrl) {
            element.datepicker({
                dateFormat: 'DD, d  MM, yy',
                onSelect: function (date) {
                    scope.date = date;
                    scope.$apply();
                }
            });
        }
    };
});
```

```
<input type="text" datetime-picker datetime-format="yyyy/MM/dd" />
```

好找的程式碼

```
app.config(function ($routeProvider) {

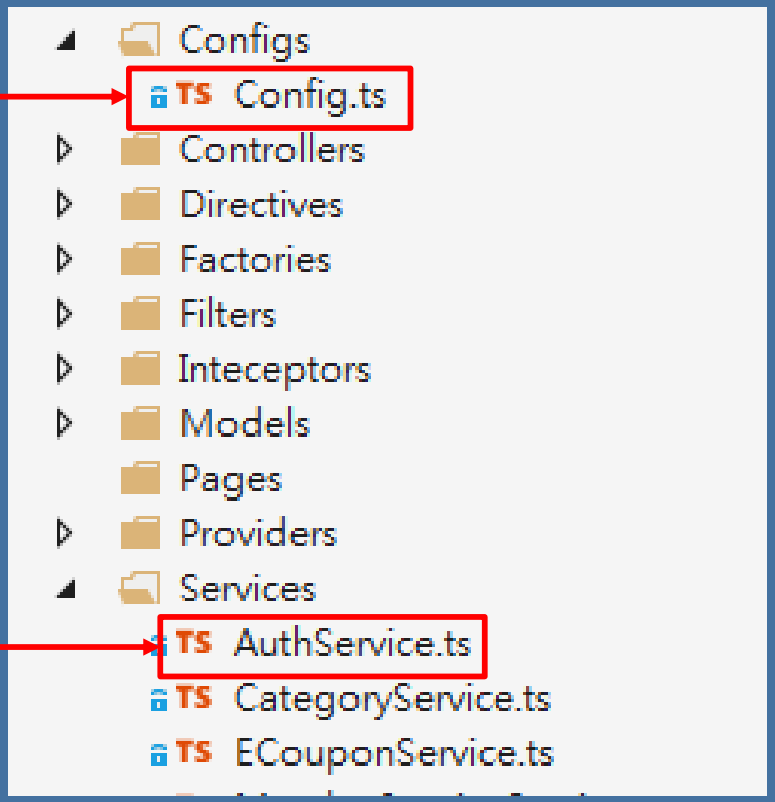
    /// 程式碼啟動設定

});
```

```
app.service('AuthService', function ($http) {

    /// Service程式碼

})
```

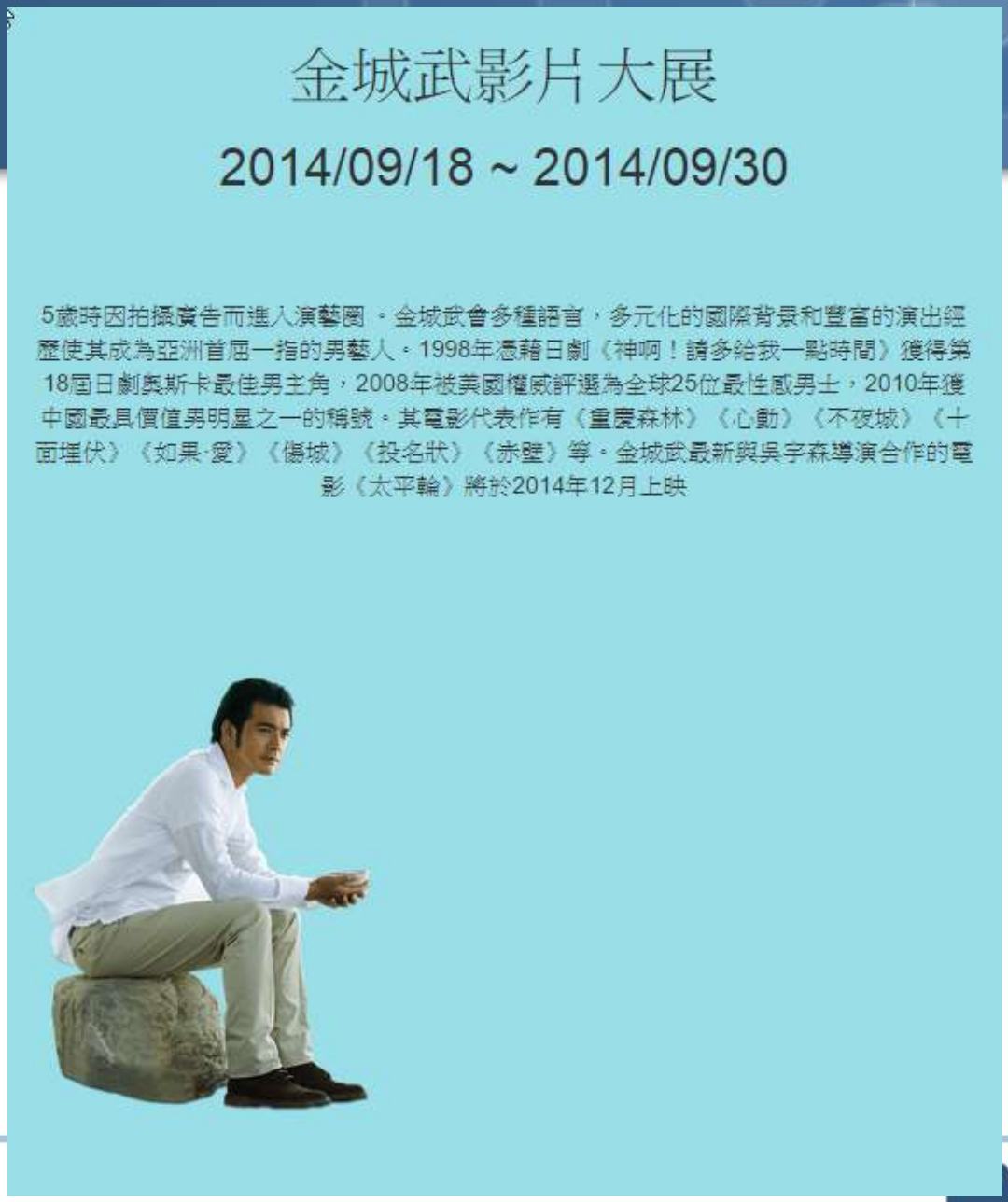


Demo

- 廣告編輯器

海報產生器

- 編輯同時預覽
- 不用JavaScript
- 動態換廣告樣式



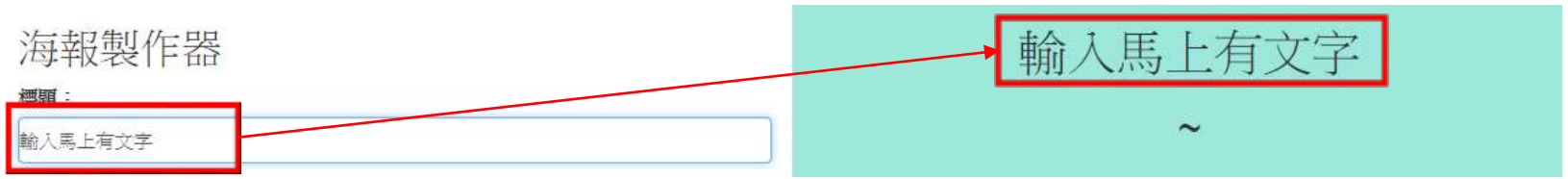
快速完成表單

- 決定欄位
- 套上Directive
- 收工！



連動更新

- 修改就可以看到效果



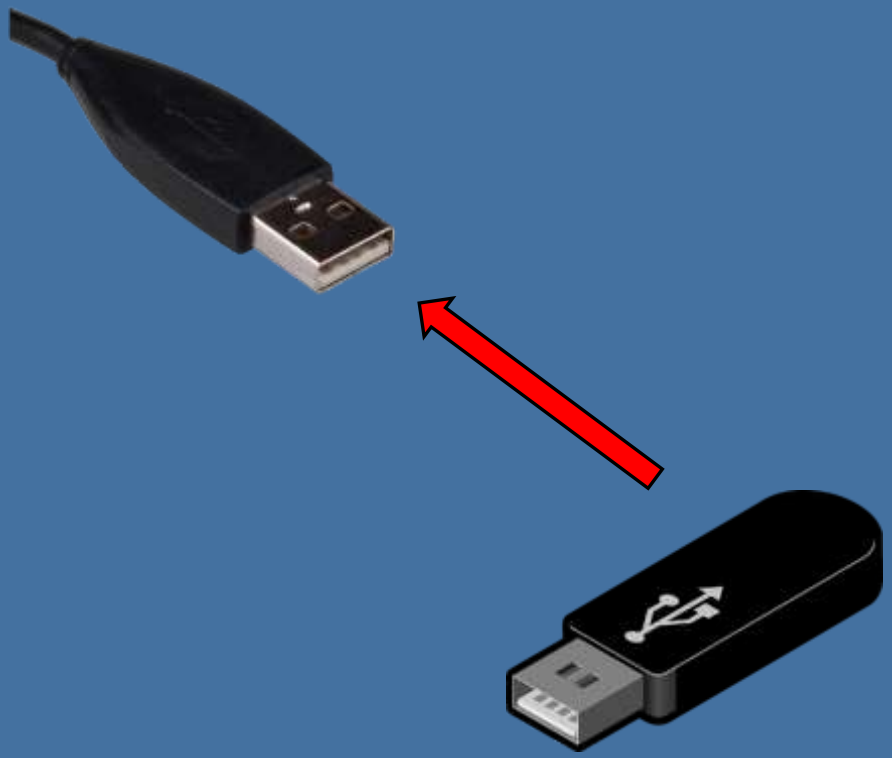
小結

- 內建角色
- 雙向繫結
- 化繁為簡



提供良好的彈性

可組合的設計（DI）






SUPER SLIM
タイプ
スリムな2段弁当箱で使える。
カバンの中にもすっきり！
COMPACT
タイプ
お弁当ライフの必需品！
USBあったか
弁当箱ポーチ

什麼是相依性注入（DI）？



線上購物

超商清單

內建DI

註冊為
超商清單

Angular.js如何實現DI

```
app.controller('ShoppingCartController', function (CVSService) {
    var self = this;

    self.cvslist = CVSService.getList();
});
```

- 判斷參數名稱
- 誰註冊為CVSService
- 注入CVSService的實體

使用7-11做為超商

```
app.service('CVSService', function (sevenEleven) {

    //// 取得7-11超商清單
});
```

SevenEleven.js

```
<script src="~/Scripts/ShoppingCartController.js"></script>
<script src="~/Scripts/SevenEleven.js"></script>
```

向Angular註冊為CVSService

將超商更換為全家

```
app.service('CVSService', function (family) {

    //// 取得全家超商清單
})
```

Family.js

```
<script src="~/Scripts/ShoppingCartController.js"></script>
<!--<script src="~/Scripts/SevenEleven.js"></script>-->
<script src="~/Scripts/Family.js'></script>
```

向Angular註冊為CVSService

其他程式完全不用改!!

專屬的HTML




```
<accordion>
    <accordion-group heading="分類A" is-open="isOpen">
        這是分類A的內容
    </accordion-group>
    <accordion-group heading="分類B">
        這是分類B的內容
    </accordion-group>
    <accordion-group heading="分類C">
        <p>這是一個段落段落</p>
        <button class="btn btn-default btn-sm">按鈕</button>
    </accordion-group>
</accordion>
```

- 自己取名
- 容易讀懂

抽換的彈性

- View和Code分離
- 程式功能獨立
- 耦合性低
- 可切換

Demo

- 切換資料來源

商品展示頁

- 前後端並行





產品說明

產品內容產品內容產品內容

根據環境切換

```
app.controller('ProductController',
    function ($scope, ProductDataService) {

    $scope.products = [];

    ProductDataService.getProducts()
        .then(function (data) {
            $scope.products = data;
        });
});
```

資料來源名稱

取得資料

```
<!-- 假資料來源，API好之前使用 -->
<script src="scripts/productDataServiceFake.js"></script>
<!-- 真的呼叫Api取得資料 -->
<script src="scripts/productDataService.js"></script>
```




開發時期

- 使用假資料

```
//// Fake, For Test
app.service('ProductDataService', function ($q) {
    var self = this;

    self.getProducts = function () {

        var defer = $q.defer();

        var data = [];
        for (var i = 0; i < 10; i++) {
            var item = {
                image: 'http://lorempixel.com/200/200/sports/' + i,
                title: '產品' + i,
                desc: ' 產品內容產品內容產品內容產品內容產品內容產品內容產
            }

            data.push(item);
        }

        defer.resolve(data);

        return defer.promise;
    }
});
```

資料來源名稱

假資料

正式環境

- 資料由Api取得

```
//// Real, Call Api
app.service('ProductDataService', function ($q, $http) {
    var self = this;

    self.getProducts = function () {
        var defer = this.$q.defer();
        var url = 'api/Product/getProducts';

        this.$http.get(url)
            .success(function (result) {
                if (result.isSuccess) {
                    defer.resolve(result.Data);
                }
                else {
                    defer.reject();
                }
            })
            .error(defer.reject);

        return defer.promise;
    }
});
```

資料來源名稱

呼叫Api

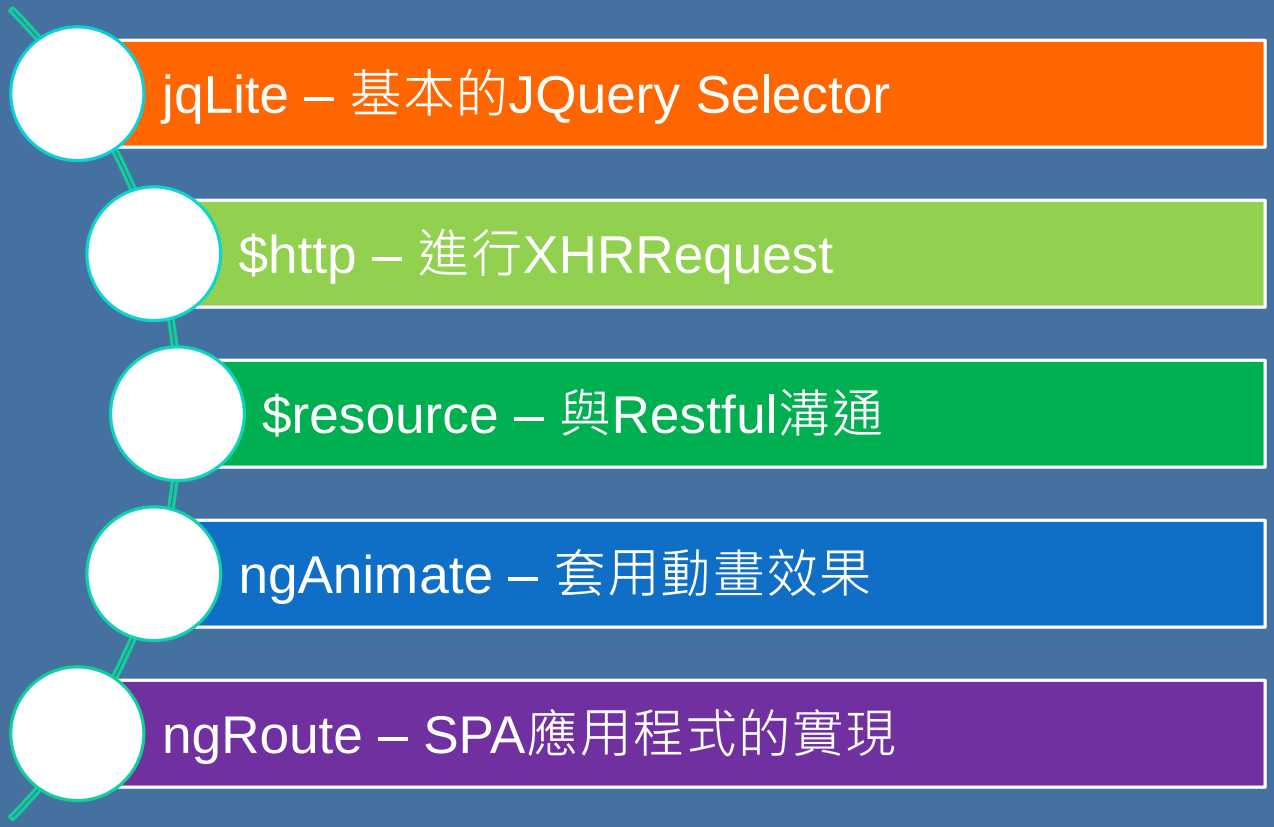
小結

- DOM自己取名
- 功能可抽換
- 前後端並行



強大的火力支援

具備網頁開發常用的工具庫

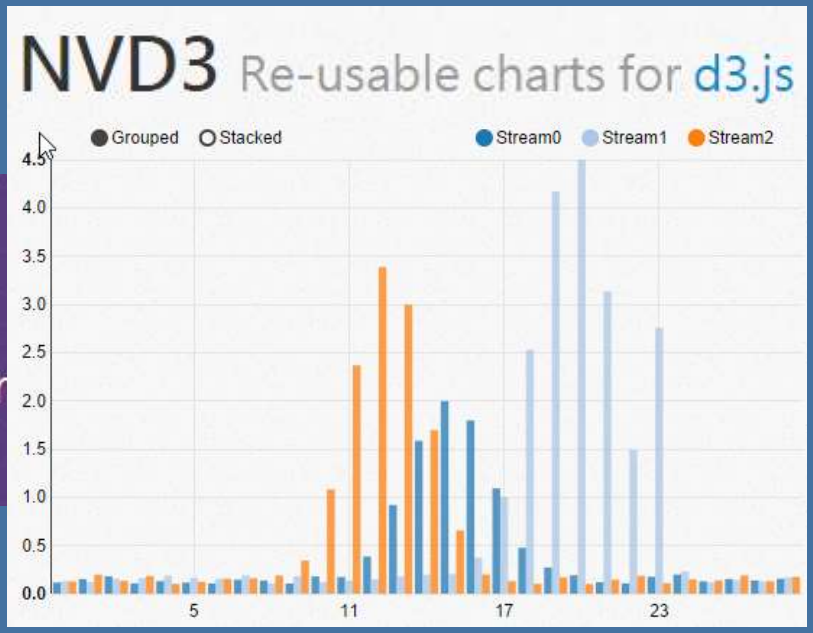


好用Library支援

UI Bootstrap

Bootstrap components written in pure Angular

AngularUI Team





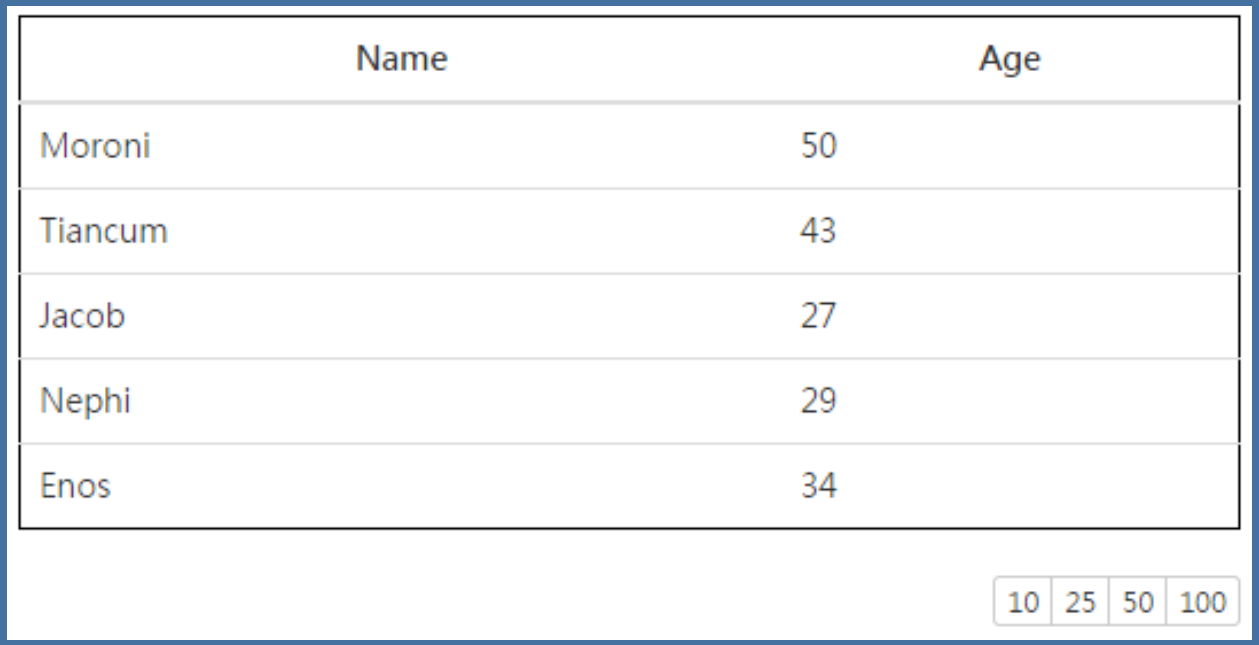
Angular-Kendo

ionic

Create amazing apps

The beautiful, open source front-end framework for developing hybrid mobile apps with HTML5.

容易使用



可分頁的Grid

容易使用

```
$scope.users = [
    { name: "Moroni", age: 50 },
    { name: "Tiancum", age: 43 },
    { name: "Jacob", age: 27 },
    { name: "Nephi", age: 29 },
    { name: "Enos", age: 34 }
];
```

資料

+

```
<table ng-table class="table">
    <tr ng-repeat="user in users">
        <td data-title="'Name'">{{user.name}}</td>
        <td data-title="'Age'">{{user.age}}</td>
    </tr>
</table>
```

繫結

使用JQuery呼叫Restful API

```
function query() {
    var blogs = $.ajax({
        url: self.apiUrl,
        type: 'GET',
        contentType: 'application/json; charset=utf-8',
    });

    return {
        $promise: blogs
    };
}
```

讀取

```
function save(blog) {
    $.ajax({
        url: self.apiUrl,
        type: 'POST',
        contentType: 'application/json; charset=utf-8',
        data: JSON.stringify(blog)
    });
}
```

新增

```
function update(id, blog) {
    var data = $.extend({}, id, blog);

    return $.ajax({
        url: self.apiUrl + id.id,
        type: 'PUT',
        contentType: 'application/json; charset=utf-8',
        data: JSON.stringify(blog)
    })
}
```

修改

```
function remove(blog) {
    $.ajax({
        url: self.apiUrl + blog.Id,
        type: 'DELETE',
        contentType: 'application/json; charset=utf-8'
    });
```

刪除

讓CRUD變簡單

```
function BlogService($resource) {
    var self = this;

    var Blog = $resource(
                "/api/Blogs/:Id",
                { Id: "@Id" },
                { "update": { method: "PUT" } }
            );

    self.query = Blog.query;
    self.create = Blog.save;
    self.update = Blog.update;
    self.delete = Blog.delete;
}
```

透過$resource將CRUD功能快速實現

Demo

- 使用WebApi建立Blog的Api
- 快速實現CRUD

小結

- 內建常用功能
- 許多現成套件
- 簡單易用



輕鬆的表單驗證

簡單的加上驗證

```
<form name="userForm">
    <div class="field">
        <label for="emailAddress">Enter your email address:</label>
        <input type="email"
               name="emailAddress"
               required
               ng-minlength="5"
               ng-maxlength="30"
               />
                                    錯誤提示訊息
        <div ng-messages="userForm.emailAddress.$error">
            <div ng-message="required">此欄位為必填欄位</div>
            <div ng-message="minlength">輸入的內容太短</div>
            <div ng-message="maxlength">輸入的內容太長</div>
            <div ng-message="email">輸入的Email格式不正確</div>
        </div>
    </div>

    <input type="submit" />
</form>
```

- Required
- ng-minlength
- ng-maxlength
- ng-pattern
- 自訂驗證

可共用的驗證訊息

```
<form name="userForm">
    <div class="field">
        <label for="emailAddress">Enter your email address:</label>
        <input type="email"
               name="emailAddress"
               required
               ng-model="email"
               ng-minlength="5"
               ng-maxlength="30"
               />

        <div ng-messages="userForm.emailAddress.$error"
             ng-messages-include="error-messages.html"></div>



    </div>

    <input type="submit" />
</form>
```

可共用的驗證訊息

- 建立常用錯誤訊息樣版

error-messages.html

```
<div ng-message="required">此欄位為必填欄位</div>
<div ng-message="minlength">輸入的內容太短</div>
<div ng-message="maxlength">輸入的內容太長</div>
<div ng-message="email">輸入的Email格式不正確</div>
```

自訂驗證

```
angular.module("App")
    .directive('cellPhone', function() {
        return {
            require : 'ngModel',
            link : function($scope, element, attrs, ngModel) {

                ngModel.$validators.cellPhone = function(value) {
                    //// 手機格式 Regular Expression
                    var pattern = /[0-9]{4}\-[0-9]{3}\-[0-9]{3}/

                    //// 輸入值是否符合格式
                    return pattern.test(value);
                };

            }
        }
    });
```

輸入值

回傳驗證是否成功

```
<input type="tel" cell-phone />
```

套上驗證

Demo

- 套用表單驗證
- 共用錯誤提示樣版



支援自動化測試

我以為工作後測試軟體是...



實際上測試軟體有點像……

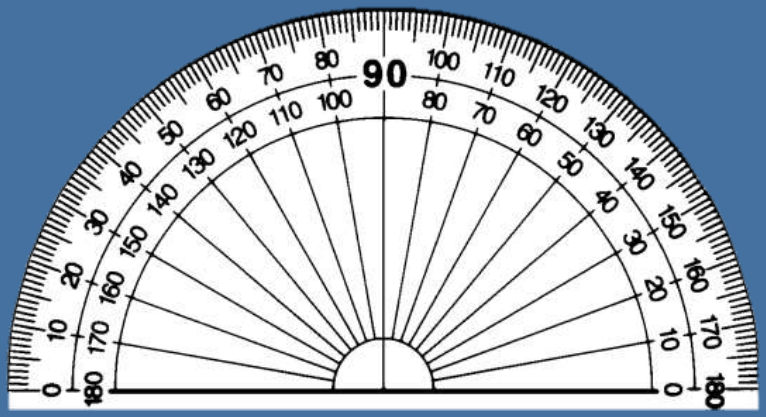


為什麼需要自動化測試？



使用Protractor

- 量身打造
- 模擬人類操作
- 支援多瀏覽器



複雜表單的測試

- 測試一個這樣的表單需要花多少時間？



交給自動化測試

- Just One Click



Demo

- 自動化測試

小結

- 減少重複的動作
- 省下大量的時間
- 多瀏覽器支援



總結

選擇Angular.js的理由

- 關注點分離（SoC）
- 架構設計良好（SOLID）
- 程式碼好維護
- 社群活動踴躍
- 相容性極佳

工商服務



公司介紹
http://goo.gl/6P7FmV

- APP開店平台
- PC、Mobile Web、IOS、Android一次搞定
- 誠徵
  - F2E
  - .Net Developer (Asp.Net Mvc)
  - Ohters

工作機會
http://goo.gl/sp9JPj

好活動需要支持

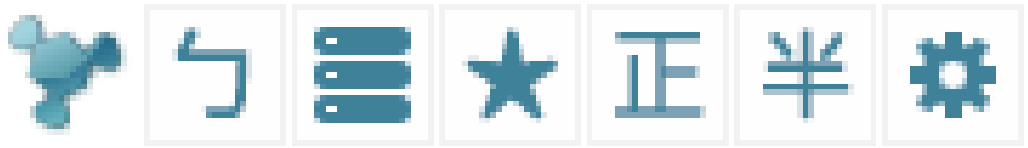


感謝 KKTIX 贊助 twMVC 活動報名平台

好輸入法需要露出



http://skilltree.my/Sponsor/xin_zi_ran



好課程需要支持



http://skilltree.my

謝謝各位

http://mvc.tw

- 本投影片所包含的商標與文字皆屬原著作者所有。
- 本投影片使用的圖片皆從網路搜尋。
- 本著作係採用姓名標示-非商業性-相同方式分享 3.0 台灣授權。閱讀本授權條款，請到 http://creativecommons.org/licenses/by-nc-sa/3.0/tw/，或寫信至Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.