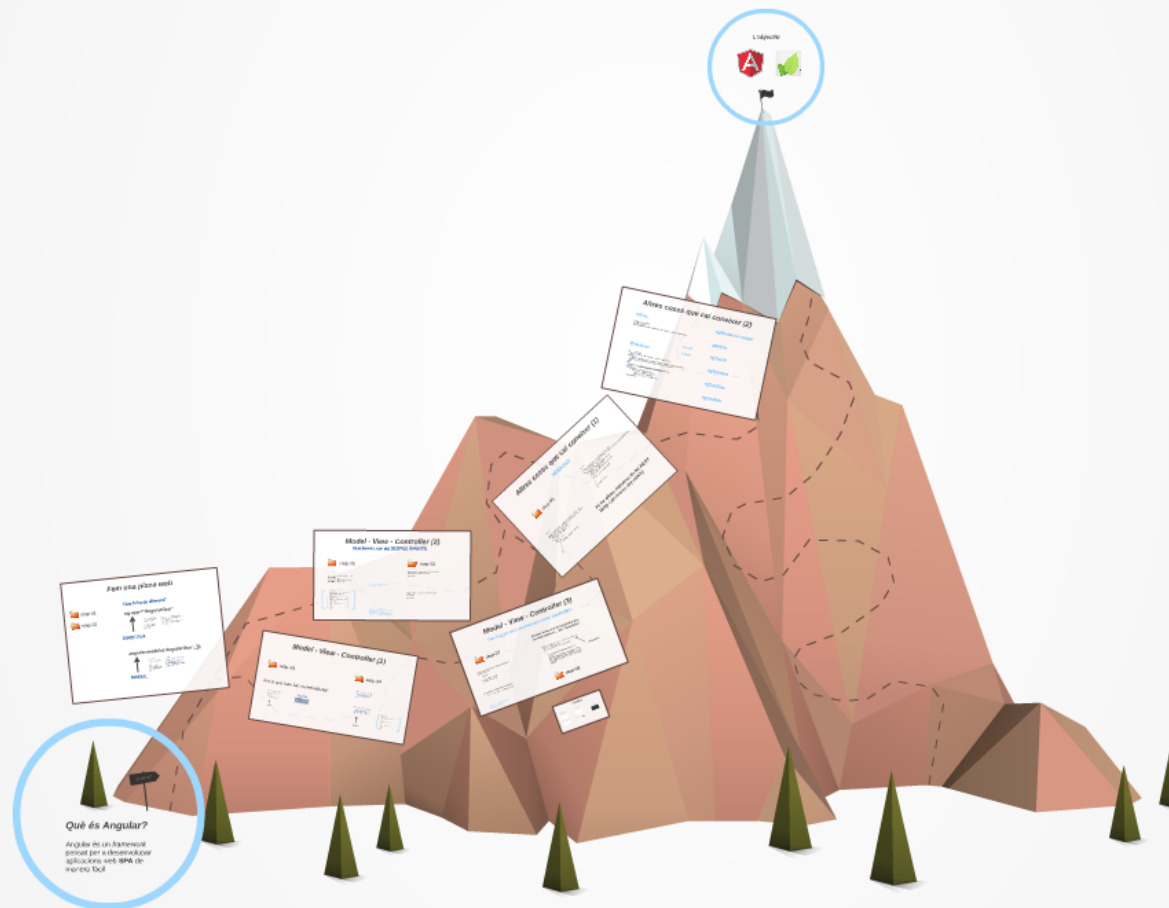
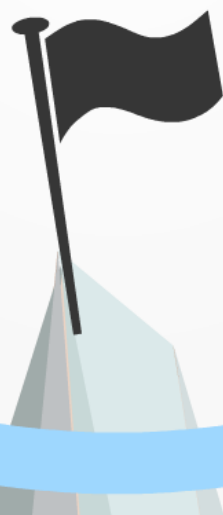


Seminari Angular & Leaflet

Aplicacions SPA amb component Geogràfic

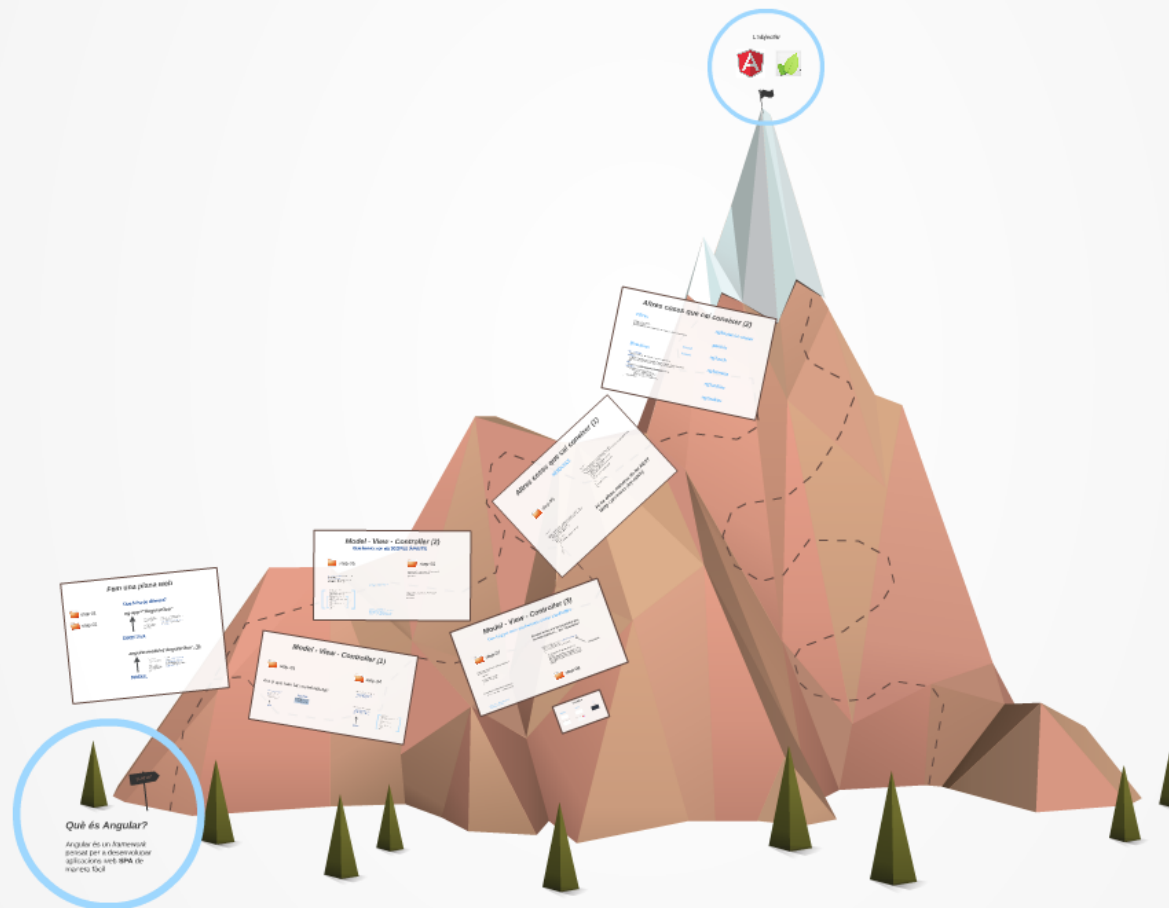


L'objectiu



Seminari Angular & Leaflet

Aplicacions SPA amb component Geogràfic





Què és Angular?

Angular és un *framework* pensat per a desenvolupar aplicacions web **SPA** de manera fàcil

Fem una plana web



step-01



step-02

Què hi ha de diferent?

`ng-app="AngularGeo"`



DIRECTIVA

Una directiva és una funció que ens torna un objecte del DOM

Les directives pròpies d'angular comencen totes per **ng-**

Com que una directiva torna un objecte del DOM també pot tornar objectes lligats a etiquetes html.

`<login>`, `<widget-weather>`, `<leaflet>`
com si fossin natives de l'estàndard html
``, ``, `<a>`

`angular.module('AngularGeo', []);`



MÒDUL

Un mòdul és un component de codi dins de la nostra aplicació

Tots els mòduls no deixen de ser un camp dins d'angular i per tant tenen un getter i un setter.

El setter:
`angular.module('AngularGeo', []);`

El getter:
`angular.module('AngularGeo');`
recordeu què hem dit de ng-?
Doncs què estem fent quan cridem
`ng-app="AngularGeo"`

Fem una plana web

Què hi ha de diferent?

ng-app="AngularGeo"



Una directiva és una funció que ens torna un objecte del DOM

Les directives pròpies d'angular comencen totes per **ng-**

Com que una directiva torna un objecte del DOM també pot tornar objectes lligats a etiquetes html.

`<login>`, `<widget-weather>`, `<leaflet>`
com si fossin natives de l'estàndard html
``, ``, `<a>`

DIRECTIVA



step-01



step-02



step-01



step-02



step-01



step-02

m una plana web

Què hi ha de diferent?

ng-app="AngularGeo"



Una directiva és una funció que ens torna un objecte del DOM

Com que una directiva del DOM també pot t...

Què hi ha de diferent?

ng-app="AngularGeo"



Una directiva és una funció que ens torna un objecte del DOM

Les directives pròpies d'angular comencen totes per **ng-**

Com que una directiva torna un objecte del DOM també pot tornar objectes lligats a etiquetes html.

`<login>`, `<widget-weather>`, `<leaflet>`
com si fossin natives de l'estàndard html
``, ``, `<a>`

DIRECTIVA

Una directiva és una funció que ens torna un objecte del DOM

Les directives pròpies

l'atribuïdo que ens torna un
objecte del DOM

Les directives pròpies
d'angular comencen
totes per **ng-**

Com que una directiva torna un objecte del DOM també pot tornar objectes lligats a etiquetes html.

`<login>`, `<widget-weather>`, `<leaflet>`

com si fossin natives de l'estàndard html

``, ``, `<a>`

DIRECTIVA

angular.module('AngularGeo', []);



MÒDUL

Un mòdul és un component de codi dins de la nostra aplicació

Tots els mòduls no deixen de ser un camp dins d'angular i per tant tenen un getter i un setter.

El setter:

`angular.module('AngularGeo', []);`

El getter:

`angular.module('AngularGeo');`

recordeu què hem dit de ng-?

Doncs què estem fent quan cridem `ng-app="AngularGeo"`

Un mòdul és un component de codi dins de la nostra aplicació

Tots els mòduls no

deixen de ser un component

de la nostra aplicació

Tots els mòduls no deixen de ser un camp dins d'angular i per tant tenen un getter i un setter.

El setter:

```
angular.module('AngularGeo', []);
```

El getter:

```
angular.module('AngularGeo');
```

recordeu què hem dit de ng-?

Doncs què estem fent quan cridem

```
ng-app="AngularGeo"
```




step-03

Què hem après en el pas 3?

Hem creat un input amb la directiva de model `ng-model="name"`
hem mostrat el contingut amb `{{}}`
`{{name}}`

- L'aplicació té un model de dades, que no hem declarat ;) javascript
- Angular s'encarrega de mostrar les dades

Què hem après en el pas 3?

Hem creat un input amb la directiva de model
`ng-model="name"`
hem mostrat el contingut amb `{{}}`
`{{name}}`

- L'aplicació té un model de dades, que no hem declarat ;) javascript
- Angular s'encarrega de mostrar les dades



MODEL

Regla d'OR

**Nosaltres en encarreguem
de les DADES, Angular ja
s'encarrega de manipular el
DOM**



step-04

Fem-ho hen fet declarem la variable name

Fem-ho ben fet, declarem la variable name...

```
angular.module('AngularGeo', [])  
.run(function($rootScope){  
  var name = "Sorpresa";  
});
```


Fem-ho ben fet, afegim una variable...

```
angular.module('AngularGeo', [])  
.run(function($rootScope){  
  $rootScope.name = "Sorpresa";  
});
```



Fem-ho ben fet, afegim una variable...

```
angular.module('AngularGeo', [])  
.run(function($rootScope){  
  $rootScope.name = "Sorpresas";  
});
```



ÀMBIT

```
<!doctype html>
<html lang="ca" ng-app="AngularGeo">
<head>
  <meta charset="utf-8">
  <title>Angular & Leaflet (Seminaris FIB - Empresa) - Pas 2</title>
  <script src="lib/angular/angular.min.js"></script>
  <script src="js/app.js"></script>
</head>
<body>
  <div>
    <h1>Angular & Leaflet</h1>
    Creació d'aplicacions SPA amb component Geogràfic
  </div>
  <div>
    Escriu el què vulguis
  </div>
  <div>
    <input type="text" ng-model="name"></input>
  </div>
  <div>
    Hola {{name}}
  </div>
</body>
</html>
```

Model - View - Controller (2)

Que bonics són els SCOPES (ÀMBITS)



step-05

Hem afegit els controladors!! Que per suposat, són un mòdul d'Angular

Hem canviat els àmbits!!

Anem a jugar amb el `{{name}}` en el títol de la plana
Primer al step-04 i després al step-05

```
<doctype html>
<html lang="ca" ng-app="AngularGeo">
<head>
  <meta charset="utf-8">
  <title>Angular &amp; Leaflet (Seminaris FIB - Empresa) - Pas 5-05</title>
  <script src="//angular.min.js"></script>
  <script src="//angular-ui.min.js"></script>
  <script src="//leaflet.js"></script>
</head>
<body ng-controller="AngularGeoController">
  <div>
    <h1>Angular &amp; Leaflet</h1>
    Creació d'aplicacions SPA amb component Geogràfic
    <div>
      <div>
        Escriu el que vulguis
      </div>
      <div>
        <input type="text" ng-model="name"></input>
      </div>
      <div>
        Hola {{name}}
      </div>
    </div>
  </div>
</body>
</html>
```

Que guai... però és evident el què has fet.
Has declarat name al controlador!!

Que guai... Però com es menja això?
Es l'herència prototipada de JavaScript que no té res a veure amb l'herència clàssica de la OOP
La lectura es 'mou' de fills cap a pares (cap amunt)
L'escriptura es 'mou' de pares cap a fills (cap avall)



step-06

Exercici vostre: Afegiu name al `$rootScope` de l'app i pose-ho al títol de la plana

Què passa?

Exercici vostre: treieu name del `$scope` del controlador

Què passa?



step-05

Hem afegit els controladors!! Que per suposat, són un

Hem afegit els controladors!! Que per suposat, són un mòdul d'Angular

Hem canviat els àmbits!!

**Anem a jugar amb el `{{name}}` en el títol de la plana
Primer al step-04 i després al'step-05**

```
<!doctype html>
<html lang="ca" ng-app="AngularGeo">
<head>
  <meta charset="utf-8">
  <title>Angular & Leaflet (Seminari FIB - Empresa) - Pas 5</title>
  <script src="lib/angular/angular.min.js"></script>
  <script src="js/app.js"></script>
  <script src="js/controllers/main.js"></script>
</head>
<body ng-controller="AngularGeoController">
  <div>
    <h1>Angular & Leaflet</h1>
    Creació d'aplicacions SPA amb component Geogràfic
  </div>
  <div>
    Escribeix el que vulguis
  </div>
  <div>
    <input type="text" ng-model="name"></input>
  </div>
  <div>
    Hola {{name}}
  </div>
</body>
</html>
```


**Que guai... però és evident el què has fet.
Has declarat name al controlador!!**



step-06

Exercici vostre: Afegiu name al \$rootScope de l'app i pose-ho al títol de la plana

Què passa?



Exercici vostre: treieu name del \$scope del controlador

Què passa?

Que guai... Però com es menja això?

És l'herència prototipada de JavaScript que no té res a veure amb l'herència clàssica de la OOP

La lectura es 'mou' de fills cap a pares (cap amunt)

L'escriptura es 'mou' de pares cap a fills (cap avall)

Model - View - Controller (3)

Com faig per tenir una herència similar a la d'OOP?



step-07

Hem creat un objecte que funcionarà com un model de dades

```
'use strict';  
  
function NameModel() {  
  this.name = 'Jaume';  
}
```

L'instanciem a l'app, però no al controller

```
$rootScope.dades = new NameModel();
```

Com s'està accedint a les dades?
Què passa realment?

Encara hi ha una funcionalitat que no hem explicat.... les "injections"

```
angular.module('AngularGeo', ['AngularGeo.Controllers'])  
  .run(function($rootScope){  
    $rootScope.dades = new NameModel();  
  });
```

- Les injections són les referències a les llibreries.
- Quan carreguem l'HTML carreguem el JS i li diem a angular que té un conjunt de mòduls.
- Les injections diuen on es fan servir aquests mòduls.

INJECTION



step-08



step-07

Hem creat un objecte que funcionarà com un

Hem creat un objecte que funcionarà com un model de dades

```
'use strict';
```

```
function NameModel() {  
  this.name = 'Jaume';  
}
```

}

L'instanciem a l'app, però no al controller

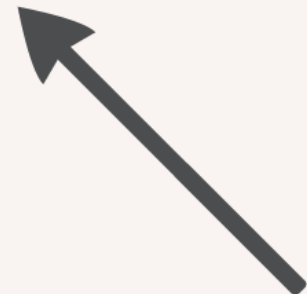
```
$rootScope.dades = new NameModel();
```

**Com s'està accedint a les dades?
Què passa realment?**

La herència similar a la d'OOP?

Encara hi ha una funcionalitat que no hem explicat.... les "injections"

```
angular.module('AngularGeo', ['AngularGeo.Controllers'])  
  .run(function($rootScope){  
    $rootScope.dades = new NameModel();  
  });
```



• *Les injections són les referències a les*

hem explicat.... les "injections"

```
angular.module('AngularGeo', ['AngularGeo.Controllers'])  
  .run(function($rootScope){  
    $rootScope.dades = new NameModel();  
  });
```



INJECTION

- *Les injections són les referències a les llibreries.*
- *Quan carreguem l'HTML carreguem el JS i li diem a angular que té un conjunt de mòduls.*
- *Les injections diuen on es fan servir aquests mòduls.*



mòdul.



step-08

Exemples

Bootstrap



<http://angular-ui.github.io/bootstrap/>



Leaflet



<http://tombatossals.github.io/angular-leaflet-directive/#!/>



Ionic



<http://ionicframework.com/>

Bootstrap



Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web.

[Download Bootstrap](#)

Currently v3.3.2



Become an engineer - attend our 12-week immersive JavaScript coding school.
ads via Carbon

Designed for everyone, everywhere.

<http://angular-ui.github.io/bootstrap/>

UI Bootstrap Directives ▾ Getting started

UI Bootstrap

Bootstrap components written in pure AngularJS by the AngularUI Team

[Code on Github](#)

[Download \(0.12.0\)](#)

[Create a Build](#)

[Star](#) 7,708

[Fork](#) 4,118

[Twitter](#)

[G+1](#) 1,448

Getting started

Dependencies

This repository contains a set of **native AngularJS directives** based on Bootstrap's markup and CSS. As a result no dependency on jQuery or Bootstrap's JavaScript is required. The **only required dependencies** are:

- [AngularJS](#) (requires AngularJS 1.2.x, tested with 1.2.16)
- [Bootstrap CSS](#) (tested with version 3.1.1). This version of the library (0.12.0) works only with Bootstrap CSS in version 3.x. 0.8.0 is the last version of this library that supports Bootstrap CSS in version 2.3.x.

Files to download

Build files for all directives are distributed in several flavours: minified for production usage, un-minified for development, with or without templates. All the options are described and can be [downloaded from here](#).

Alternatively, if you are only interested in a subset of directives, you can [create your own build](#).

Leaflet



Star 9,618 Tweet Follow 11.8K followers Like 5.3k

An Open-Source JavaScript Library for Mobile-Friendly Interactive Maps

Overview Features Tutorials API Download Plugins Blog GitHub Twitter Forum

Leaflet is a modern open-source JavaScript library for mobile-friendly interactive maps. It is developed by [Vladimir Agafonkin](#) with a team of dedicated [contributors](#). Weighing just about 33 KB of JS, it has all the [features](#) most developers ever need for online maps.

Leaflet is designed with *simplicity, performance and usability* in mind. It works efficiently across all major desktop and mobile platforms out of the box, taking advantage of HTML5 and CSS3 on modern browsers while still being accessible on older ones. It can be extended with a huge amount of [plugins](#), has a beautiful, easy to use and [well-documented API](#) and a simple, readable [source code](#) that is a joy to [contribute](#) to.

Used by: Flickr foursquare Pinterest craigslist Data.gov IGN Wikimedia OSM Meetup WSJ Mapbox CartoDB GIS Cloud ...



In this basic example, we create a map with tiles of our choice, add a marker and bind a popup with some text to it:

```
// create a map in the "map" div, set the view to a given place and zoom
var map = L.map('map').setView([51.505, -0.09], 13);

// add an OpenStreetMap tile layer
L.tileLayer('http://{s}.tile.osm.org/{z}/{x}/{y}.png', {
  attribution: '&copy; <a href="http://osm.org/copyright">OpenStreetMap</a> contributors'
}).addTo(map);

// add a marker in the given location, attach some popup content to it and open the popup
L.marker([51.5, -0.09]).addTo(map)
  .bindPopup('A pretty CSS3 popup. <br> Easily customizable.');
```

<http://tombatossals.github.io/angular-leaflet-directive/#!!>

angular-leaflet-directive

Getting started Examples How to extend Github repository



Simple to use directive for easy embedding and interacting with a map managed with the [leaflet map library](#) on an [AngularJS](#) application.

This directive allows you to embed a map on your [AngularJS](#) application and interact bi-directionally with it via the [AngularJS scope](#) and the [leaflet map library API](#). In its current state, it's not a full-featured library, although it's easy to use and extend, and a good starting point to learn to code a directive into the great [AngularJS](#) library.

The code of the directive and this page has been inspired by the [Angular Google Maps](#) directive developed by [mlaplante](#). The initial code has been (and is being) evolved and improved by [these contributors](#).

Getting started

Before using [angular-leaflet-directive](#), you must include some javascript files to your main HTML:

- the [AngularJS](#) library,
- the [Leaflet](#) library (javascript & CSS),
- and the [angular-leaflet-directive.js](#) script.

Something like this:

Javascript includes CSS includes

```
<script src="http://cdn.leafletjs.com/leaflet-0.7.1/leaflet.js"></script>
<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.2.8/angular.min.js"></script>
<script src="/js/angular-leaflet-directive.min.js"></script>
```

And that's all. Now you can start embedding maps to your application adding the custom tag `<leaflet>` with some attributes on your HTML code, and interacting with the `scope` of your controller. Let's see how it works and how you can use it in the [examples page](#), or take a look of what can be accomplished with it on the [demos](#) below.



lonic

<http://ionicframework.com/>



[Products](#) [Getting Started](#) [Docs](#) [Showcase](#) [Forum](#) [Blog](#)¹

Create amazing apps

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

[Get Started](#)

[Docs & Help](#)

v1.0.0-beta.14 "magnesium-mongoose" · 2014-12-15 · MIT Licensed



Ionic View

[13,303 stars](#) [1,843 forks](#) [16.8K followers](#)

[Subscribe](#)



Coming soon: Push notifications made easy for your Ionic apps

[RESERVE YOUR SPOT](#)

Altres coses que cal coneixer (1)

RESOURCE



step-09

```
'use strict';

angular.module('NameAPI', ['ngResource'], function($provide) {
  $provide.factory('NameHTTP', ['$resource', function($resource) {
    //return $resource('name.json');
    return $resource('name.json', {}, {
      save: {
        method: 'POST',
        headers: {
          'Content-Type': 'multipart/form-data',
        }
      }
    });
  }]);
});
```

```
'use strict';

angular.module('AngularGeo.Controllers', ['NameAPI']).
  controller('AngularGeoController', ['$scope', 'NameHTTP', function($scope, NameHTTP) {
    $scope.getName = function () {
      var response = NameHTTP.get(function () {
        if (response === undefined) {
          console.error("ERROR getting name.txt");
        } else {
          $scope.name = response.name;
        }
      });
    };
    $scope.getName();
  }]);
```

***Hi ha altres maneres de fer REST
\$http i promises (try-catch)***



step-09

```
'use strict';
```

```
angular.module('NameAPI', ['ngResource'], function($provide) {  
  $provide.factory('NameHTTP', ['$resource', function($resource) {  
    //return $resource('name.json');  
    return $resource('name.json', {}, {  
      save: {  
        method: 'POST',  
        headers: {  
          'Content-Type': 'multipart/form-data',  
        }  
      }  
    });  
  }]);  
});
```

```
'use strict';

angular.module('AngularGeo.Controllers', ['NameAPI']).
  controller('AngularGeoController', ['$scope', 'NameHTTP', function($scope, NameHTTP) {
    $scope.getName = function () {
      var response = NameHTTP.get(function () {
        if (response === undefined) {
          console.error("ERROR getting name.txt");
        } else {
          $scope.name = response.name;
        }
      });
    };
    $scope.getName();
  }]);
```

na altres maneres de fer REST


```
var response = NameHTTP.get(function () {
  if (response === undefined) {
    console.error("ERROR getting name.txt");
  } else {
    $scope.name = response.name;
  }
});
$scope.getName();
});
```

***Hi ha altres maneres de fer REST
\$http i promises (try-catch)***

Altres coses que cal coneixer (2)

Filtres

```
{{ name | uppercase }}  
{{ 12 | currency }}  
{{1288323623006 | date:"MM/dd/yyyy 'at' h:mmam"}}: 10/29/2010 at 5:40AM  
...
```

Directives

```
ngStyle i ngClass  
ng-class="{strike: deleted, bold: important, red: error}"  
ngRepeat  
<li class="animate-repeat" ng-repeat="friend in friends | filter:q as results">  
  {{{ $index + 1 }}} {{friend.name}} who is {{friend.age}} years old.  
</li>  
ngSrc  
  
ngPluralize  
<ng-pluralize count="personCount"  
  when="{0: 'Nobody is viewing.',  
  'one': '1 person is viewing.',  
  'other': '{} people are viewing.'}">  
</ng-pluralize>  
...
```

ngRouter/ui-router

partials

\$watch

\$event

ngTouch

ngAnimate

ngSanitize

ngCookies

Filtres

{{ name | uppercase }}

{{ 12 | currency }}

551288223623006 | date:"MM/dd/yyyy "

Directives

ngStyle *ngClass*

ng-class="{strike: deleted, bold: }

\$watch

\$event

Directives

ngStyle i *ngClass*

```
ng-class="{strike: deleted, bold: important, red: error}"
```

ngRepeat

```
<li class="animate-repeat" ng-repeat="friend in friends | filter:q as results">  
  [{{$index + 1}}] {{friend.name}} who is {{friend.age}} years old.  
</li>
```

ngSrc

```

```

ngPluralize

```
<ng-pluralize count="personCount"  
  when="{0: 'Nobody is viewing.',  
        'one': '1 person is viewing.',  
        'other': '{} people are viewing.'}">  
</ng-pluralize>
```

...

conexer (2)

ngRouter/ui-router

partials

ngRouter/ui-router

partials

ngTouch

partials

ngTouch

ngAnimate

ngTouch

ngAnimate

ngSanitize

ngAnimate

ngSanitize

ngCookies

\$watch

\$event



Seminari Angular & Leaflet

Aplicacions SPA amb component Geogràfic

