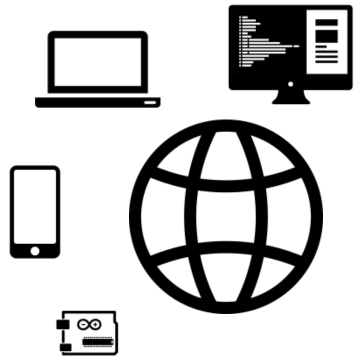
Introduction à Angular.js

ANGULARJS

Qu’est ce qu’Angular.js ?

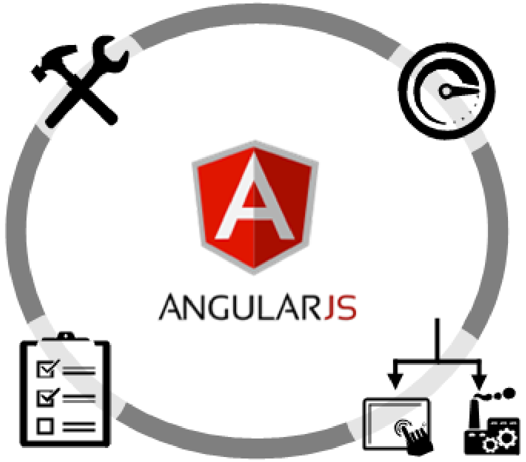
Un internet de Services

- Nécessiter de fournir des contenus adaptés à chaque terminal:
  - Terminaux mobiles (android, ios, windows)
  - Web browser
  - Machine to Machine (M2M)
- Découpler l’interface utilisateur de la logique métier
- Utiliser pleinement la puissance des web browsers
- Applications Web offline

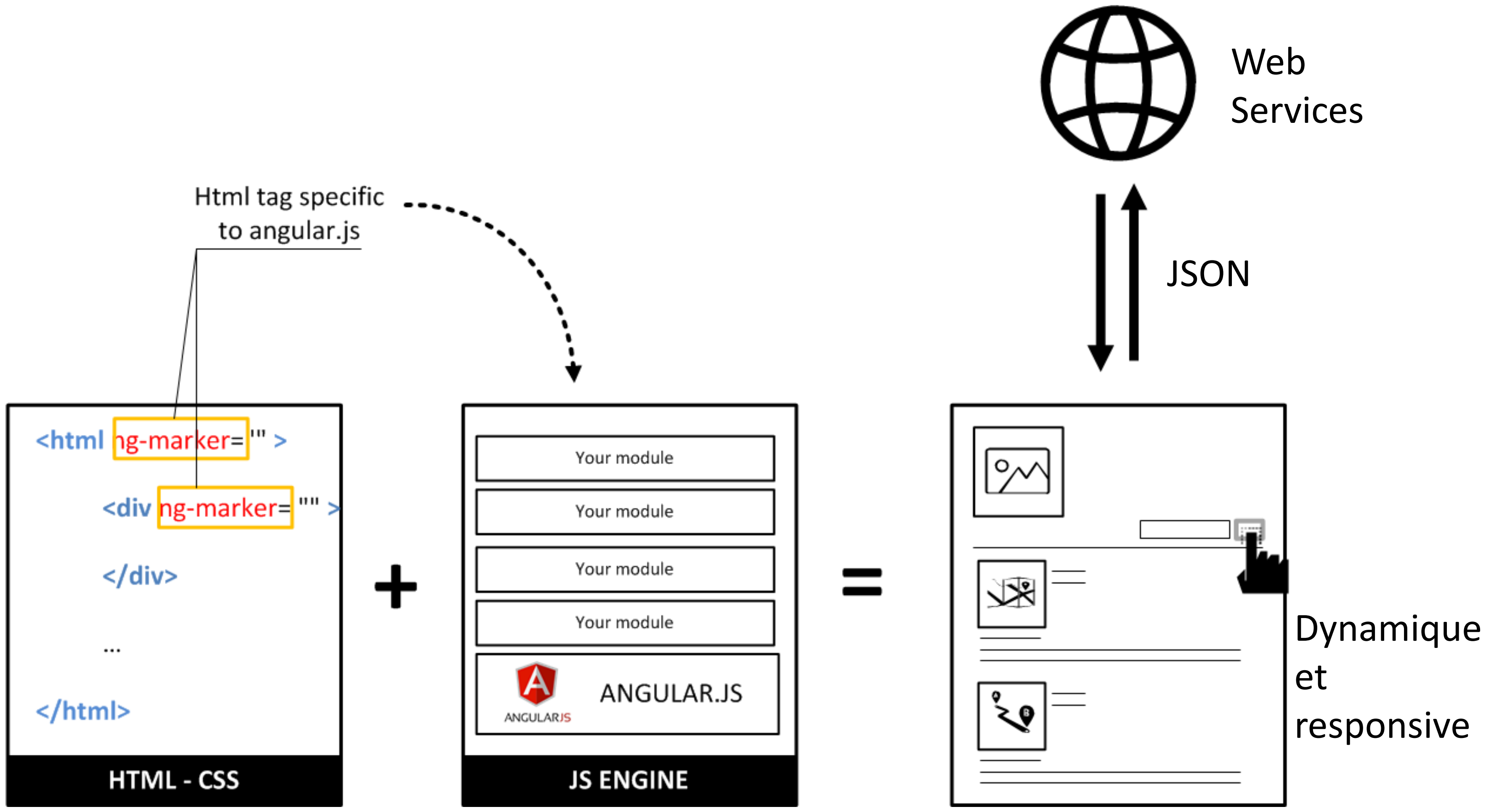


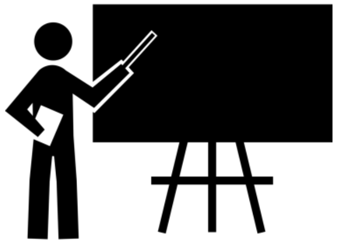
Qu'apporte Angular.js ?

- ❑ Organiser le code javascript
- ❑ Réelle séparation Modèle Vue Controleur
- ❑ Du javascript facilement testable
- ❑ Possibilité de réutiliser très facilement des modules
- ❑ Créer des applications web dynamique et responsive



Comment fonctionne Angular.js ?





Angular.js: les bases

Les Composants (1/2):

- ❑ Une **librarie** javascript (https://angularjs.org/)
- ❑ Des tags html spécifiques à Angular.js appelés **directives**
- ❑ Des **expressions**
- ❑ Du code javascript répartit en **module**

Angular.js: les bases

```
<!DOCTYPE html>
<html ng-app="sampleApp"   >
   <head>
    <link href="css/bootstrap.min.css" rel="stylesheet">
   </head>
   <body>
      <div ng-controller="sampleCtrl as crt">

         <h1> {{ "Hello" + "," + "nice to met you" }}</h1>
         <h2> 1 {{ 1+1 }} {{ 1+1+1 }}</h2>

         <button type="button"
               class="btn btn-default btn-lg"
               ng-click="crt.clickFunction()">
               Click me
         </button>
      </div>

      <script src="angular.min.js"></script>
      <script src="component.js"></script>
   </body>
</html>
```

Tags spécifique Angular.js applés **DIRECTIVES**

« Framework» Angular.js

Angular.js: les bases

```
<!DOCTYPE html>
<html ng-app="sampleApp" >
   <head>
    <link href="css/bootstrap.min.css" rel="stylesheet">
   </head>
   <body>
      <div ng-controller="sampleCtrl as crt " >

         <h1>{{ "Hello" + "," + "nice to met you" }}</h1>
         <h2> 1{{ 1+1 }} {{ 1+1+1 }}</h2>

         <button type="button"
               class="btn btn-default btn-lg"
               ng-click="crt.clickFunction()">
               Click me
         </button>
      </div>


      <script src="angular.min.js"></script>
      <script src="component.js"></script>
   </body>
</html>
```

Usage du module **application** SampleApp

Usage du **controller** SampleCrt ayant comme **label** crt

Eléments interprétés par le Framework Angular.js appelé **EXPRESSION**

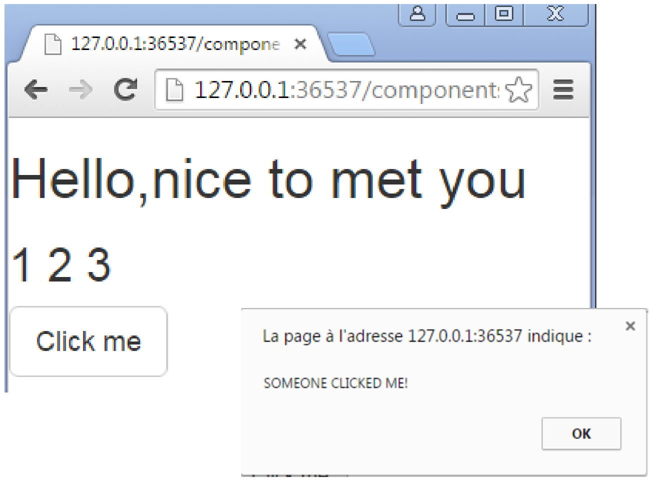
Déclenchement de la fonction du **controller** sampleCrt lors d'un click

Ajout de notre code javascript décrivant les modules sampleApp et sampleCrt

Angular.js: les bases

```html
<!DOCTYPE html>
<html ng-app="sampleApp">
   <head>
    <link href="css/bootstrap.min.css" rel="stylesheet">
   </head>
   <body>
      <div ng-controller="sampleCtrl as crt">

         <h1> {{ "Hello" + "," + "nice to met you" }}</h1>
         <h2> 1 {{ 1+1 }} {{ 1+1+1 }}</h2>

         <button type="button"
               class="btn btn-default btn-lg"
               ng-click="crt.clickFunction()">
               Click me
         </button>
      </div>

      <script src="angular.min.js"></script>
      <script src="component.js"></script>
   </body>
</html>
```

```javascript
// NOT GOOD PRACTICE
var app = angular.module('sampleApp',[]);

app.controller('sampleCtrl',function(){

   this.clickFunction=function () {

      alert('SOMEONE CLICKED ME!');

   };
});
```


127.0.0.1:36537/compone
127.0.0.1:36537/component

Hello,nice to met you

1 2 3

Click me

La page à l'adresse 127.0.0.1:36537 indique :

SOMEONE CLICKED ME!

OK

Les Composants (2/2):

- Module
  - « Main » de notre application Web
  - Possibilité de charger d’autres modules en tant que dépendances
- Controllers
  - Fonction liée à un module
  - Utilisé pour alimenter le SCOPE du module
    - Initialisation du scope
    - Association de comportement au scope
- Scope
  - Container d’objets et de données de l’application
  - Organisation hiérarchique de scopes
  - Visible via les expressions

Angular.js: les bases

```
// NOT GOOD PRACTICE

var app = angular.module('sampleApp',[]);



app.controller('sampleCtrl',function() {

   this.clickFunction=function () {

      alert('SOMEONE CLICKED ME!');

   };

this.currentShip=
                 {
                 name: {
                    nickname:'pagme Starship',
                    fullname:'Naboo Royal
                              Starship'
                 },

                 location: 'Naboo',
                 length:76
                 };

});
```

Création d'un module

Liste de dépendances

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

Utilise la lib angular.js

Nom du module

Nom du controlleur

```
app.controller('sampleCtrl',function() {});
```

Association d'un controller
Au module courant.

Description du comportement du contrôleur

```
this.clickFunction=function () {};
this.currentShip={ … };
```

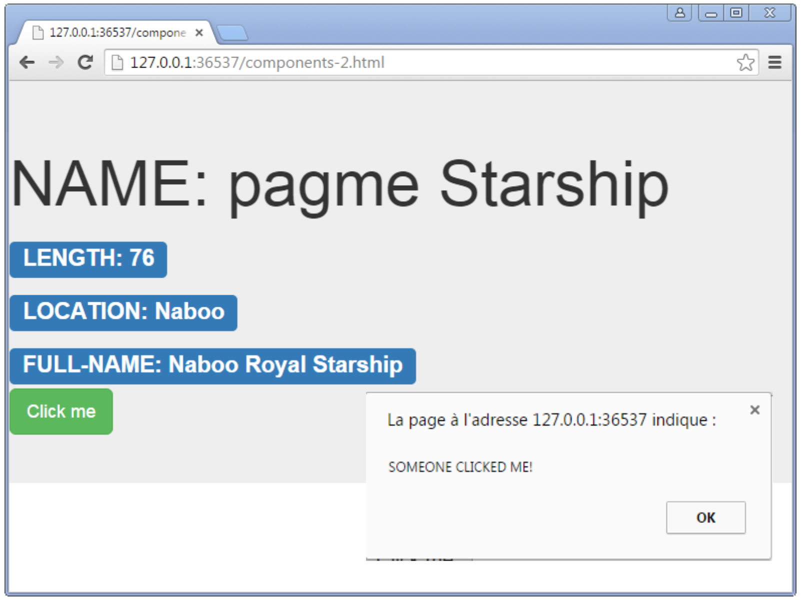
Ajout d'une donnée au scope

Ajout d'une fonction au scope

Angular.js: les bases

```html
<!DOCTYPE html>
<html ng-app="sampleApp">
   <head>
    <link href="css/bootstrap.min.css"  rel="stylesheet">
   </head>
   <body>
      <div ng-controller="SampleCtrl as crt" class="jumbotron">
         <h1> NAME: {{ crt.currentShip.name.nickname }}</h1>
         <h2> <span class="label label-primary">
                    LENGTH: {{ crt.currentShip.length }}
                     </span>
         </h2>
         <h2> <span class="label label-primary">
                    LOCATION: {{ crt.currentShip.location }}
                     </span>
        </h2>
         <h2> <span class="label label-primary">
                    FULL-NAME: {{ crt.currentShip.name.fullname }}
                    </span>
          </h2>
         <button type="button"
               class="btn btn-success btn-lg"
               ng-click="crt.clickFunction()">
               Click me
         </button>
      </div>
      <script src="angular.min.js"></script>
      <script src="component-2.js"></script>
   </body>
</html>
```

```javascript
// NOT GOOD PRACTICE
var app = angular.module('sampleApp',[]);
app.controller('sampleCtrl',function() {

   this.clickFunction=function () {
      alert('SOMEONE CLICKED ME!');
   };
this.currentShip=
                  {
                  name: {
                     nickname:'pagme Starship',
                     fullname:'Naboo Roya Starship'
                  }, location: 'Naboo', length:76 };
});
```


127.0.0.1:36537/compone ×
127.0.0.1:36537/components-2.html

NAME: pagme Starship

LENGTH: 76

LOCATION: Naboo

FULL-NAME: Naboo Royal Starship

Click me

La page à l'adresse 127.0.0.1:36537 indique :

SOMEONE CLICKED ME!

OK

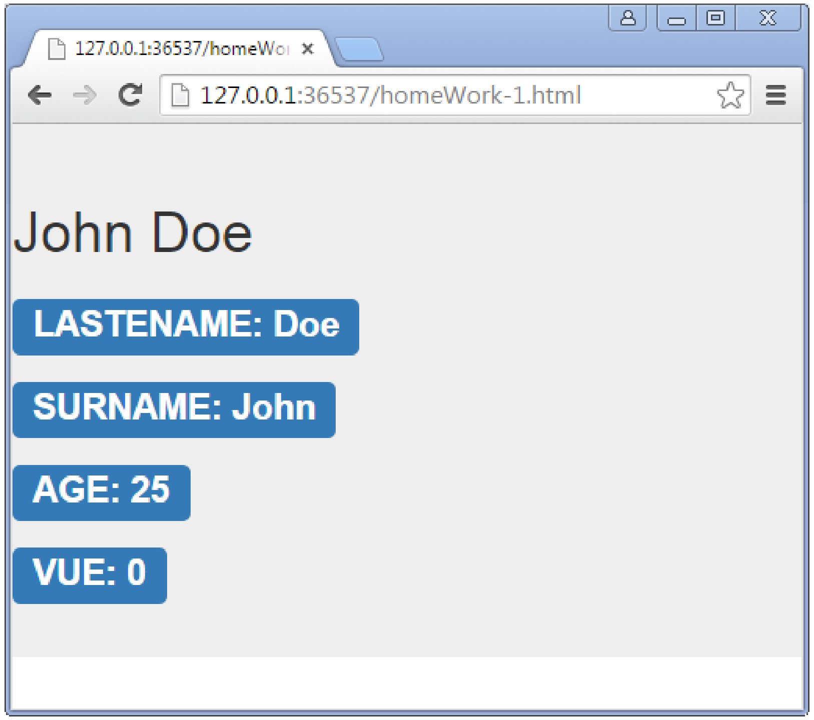
Sum up

DIRECTIVES:	Tags HTML déclenchant des comportements javascript
MODULES:	Conteneur de notre application (données et logique métier)
CONTROLLER:	Définition des comportements de l'application
EXPRESSION:	Décrit comment les données seront affichées dans une page

A vous de jouer !

- Créer un canvas d'application angular.js
  - .html
  - ng-app, ng-controller
  - .js
- Afficher les propriétés d'un utilisateur
  - nom, prenom, age ,nbre_de_vue
- Lier une méthode permettant d'incrémentée le nbre_de_vue à chaque clic sur la div courante

Angular.js: les bases


127.0.0.1:36537/homeWork-1.html

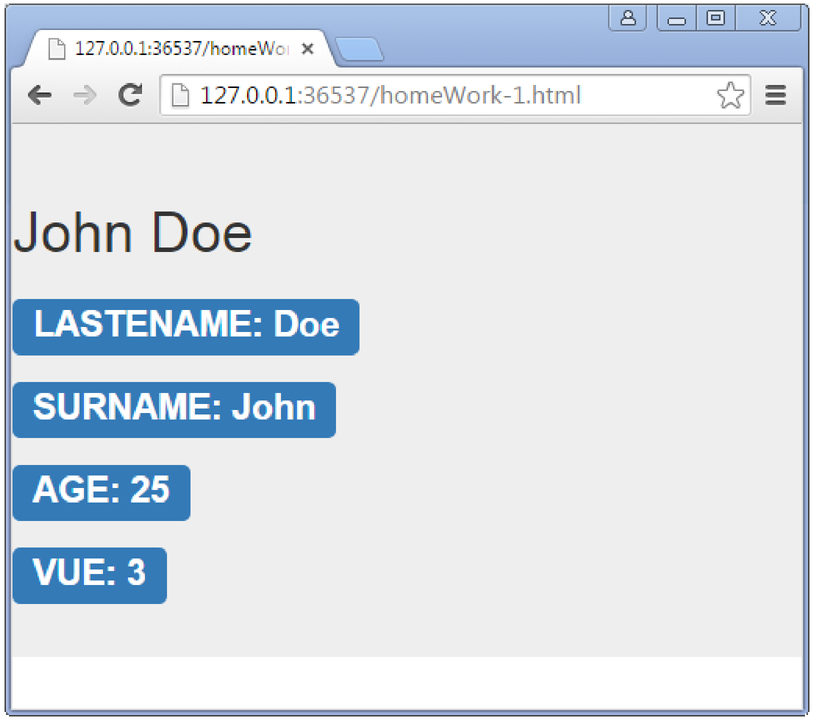
John Doe

LASTENAME: Doe

SURNAME: John

AGE: 25

VUE: 0


127.0.0.1:36537/homeWork-1.html

John Doe

LASTENAME: Doe

SURNAME: John

AGE: 25

VUE: 3

Affichage et Manipulation de données

Les Directives d’affichage:

- ng-repeat
  - Affichage de liste d’information

```
<div ng-repeat="obj in objList">
      {{obj.att1}}
      {{obj.att2}}
</div>
```

```
<div ng-repeat="(key, value) in objMap">
  {{key}}
  {{value.att1}}
</div>
```

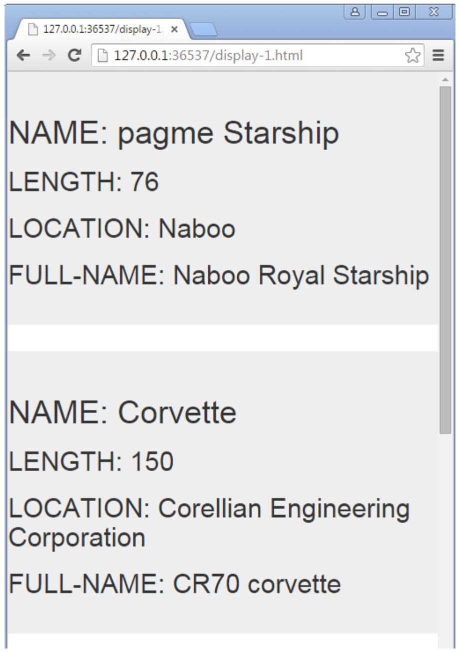
- ng-show, ng-hide
  - Affichage conditionnel d’un bloc

```
<div ng-show="objList.length  < 1">
            <h1> No object in database</h1>
</div>
```

Affichage et manipulation de données

```html
<!DOCTYPE html>
<html ng-app="sampleApp">
  <head>
   <link href="css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <div ng-controller="SampleCtrl as crt">

      <div ng-hide="crt.shipList.length >  0">
        <h1>
          <span class="label label-danger">
            No object in database</span>
        </h1>
      </div>

      <div class="jumbotron" ng-repeat="ship in crt.shipList">
      <h1> NAME: {{ ship.name.nickname }}</h1>
      <h2>  LENGTH: {{ ship.length }} </h2>
      <h2>  LOCATION: {{ ship.location }} </h2>
      <h2>  FULL-NAME: {{ ship.name.fullname }} </h2>

      </div>
</div>

    <script src="angular.min.js"></script>
    <script src="display-1.js"></script>
  </body>
</html>
```

```javascript
var pagmShip={  name: {    nickname:'pagme Starship',
                  fullname:'Naboo Royal Starship'     },
                location: 'Naboo',
                length:76
                };
var corvetteShip={  name: {  nickname:' Corvette',
                  fullname:'CR70 corvette'       },
                location: 'Corellian Engineering Corporation',
                length:150
                };
var app = angular.module('sampleApp',[]);
    app.controller('SampleCtrl',function() {
            this.shipList=[pagmShip,corvetteShip];
});
```

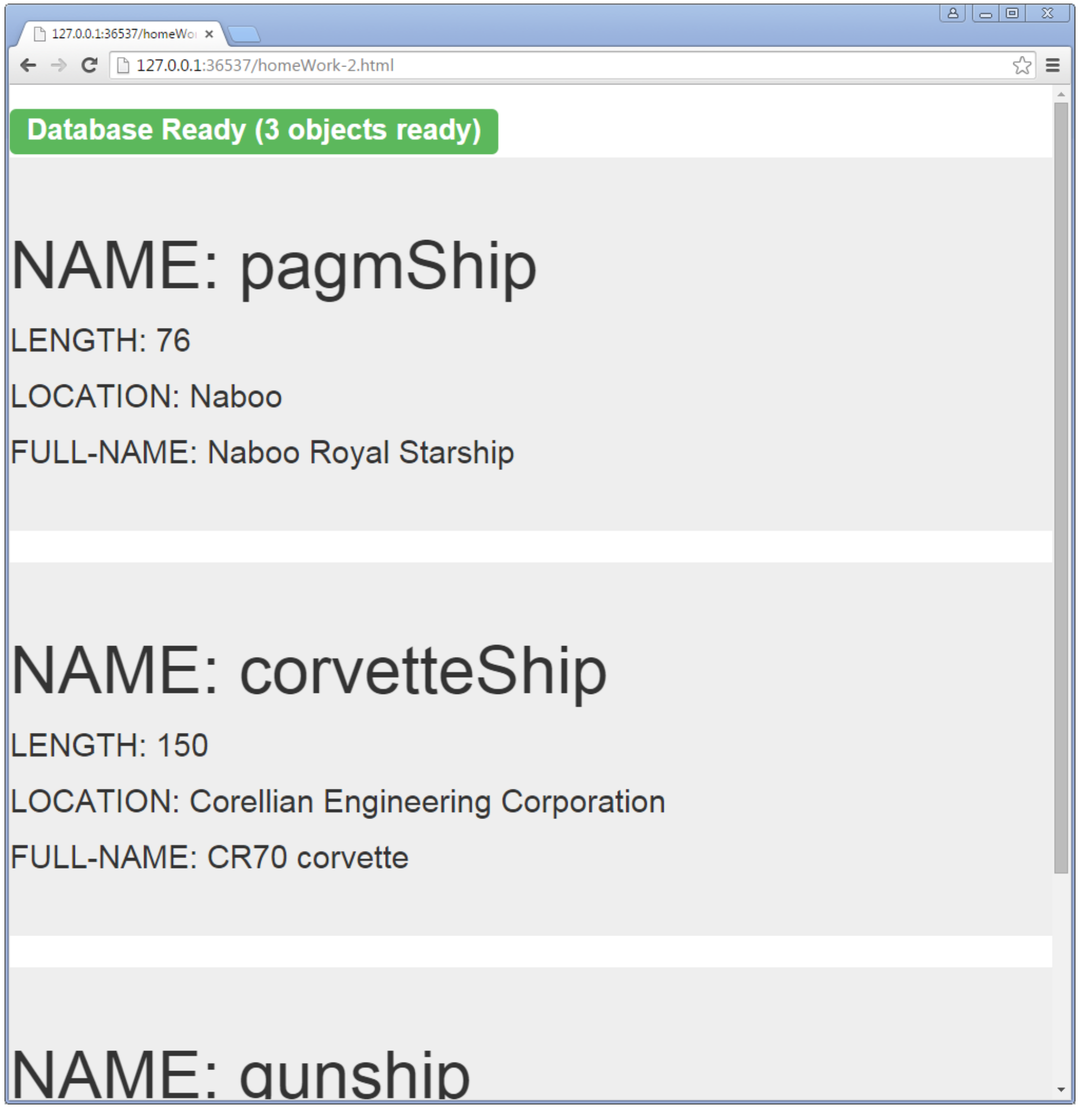


```
127.0.0.1:36537/display-1.html

NAME: pagme Starship
LENGTH: 76
LOCATION: Naboo
FULL-NAME: Naboo Royal Starship

NAME: Corvette
LENGTH: 150
LOCATION: Corellian Engineering
Corporation
FULL-NAME: CR70 corvette
```

A vous de jouer !

- ❑ Afficher une liste d'information provenant d'un dictionnaire de données (Map)
- ❑ Afficher « DATABASE READY (n objects available)» si la taille de la map >0

Affichage et manipulation de données


127.0.0.1:36537/homeWo

127.0.0.1:36537/homeWork-2.html

Database Ready (3 objects ready)

NAME: pagmShip

LENGTH: 76

LOCATION: Naboo

FULL-NAME: Naboo Royal Starship

NAME: corvetteShip

LENGTH: 150

LOCATION: Corellian Engineering Corporation

FULL-NAME: CR70 corvette

NAME: gunship

Les filtres

- Filtre / formatage des données d'expression

```
{{ data | filter:options }}
```

- Date

```
{{ '1388123412323' | date:'MM/dd/yyyy @ h:mma' }}        12/27/2013 @ 12:50AM
```

- Uppercase / lowercase

```
{{ 'spaceship commander' | uppercase}}                    SPACESHIP COMMANDER
```

- LimitTo

```
{{'My Description' | limitTo:8}}                         My Descr

<li ng-repeat="ship in crt.shipList | limitTo:2" >
```

- orderBy

```
<li ng-repeat="ship in crt.shipList | orderBy: ' -length' " >
```

Les filtres

- Listes des options

currency	date	json	uppercase
lowercase	number	limitTo	
orderBy	filter		

https://docs.angularjs.org/guide/filter

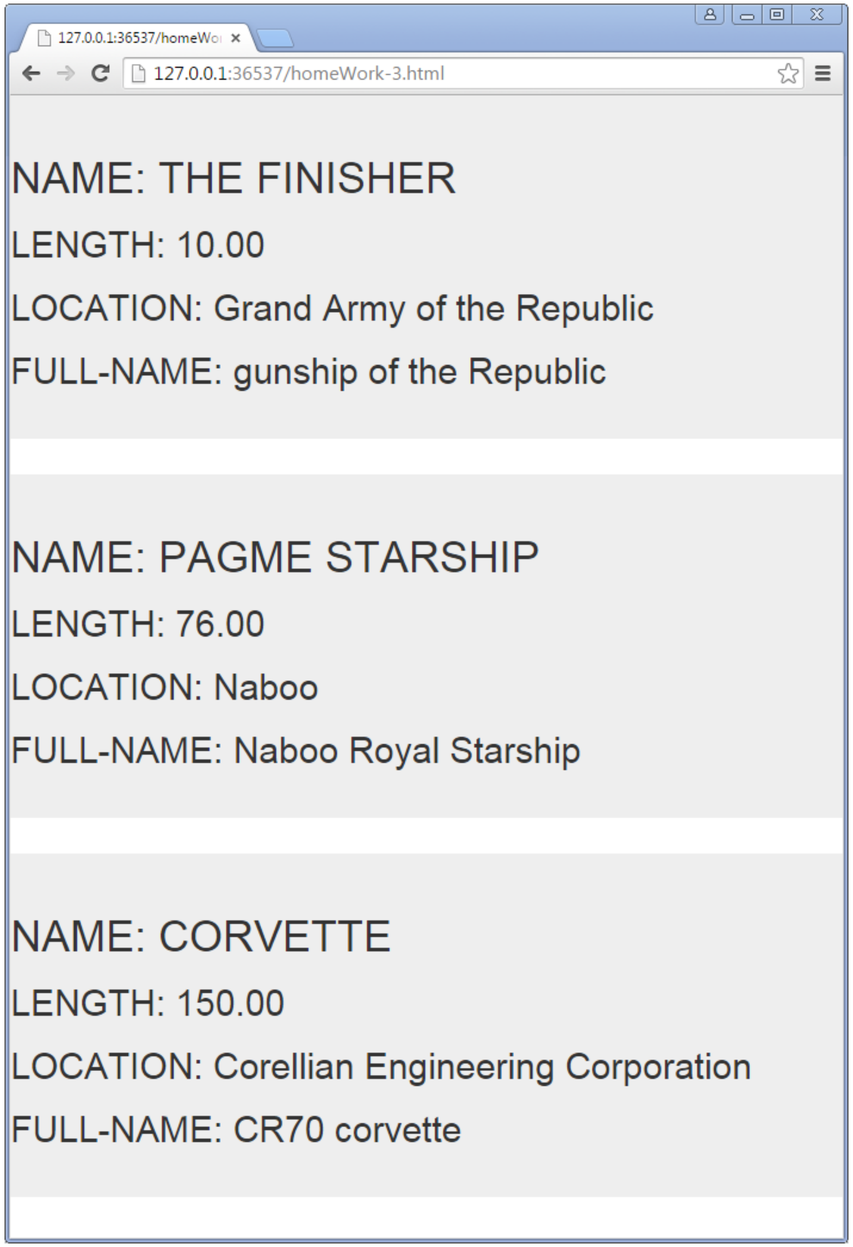
- The filter option
  - Recherche un pattern dans une liste contenant une expression spécifique

```
<li ng-repeat="ship in crt.shipList | filter: 'ship' " >

<li ng-repeat="ship in crt.shipList | filter: { location:'Naboo', lenght:76}" >
```

A vous de jouer !

- ❑ Afficher une liste d'information provenant d'une liste de données
- ❑ Ordonnée les données par ordre croissant sur un attribut numérique (e.g lenght)
- ❑ Afficher le nom principal en capitale
- ❑ Afficher le champ numérique avec 2 décimales

Affichage et manipulation de données



NAME: THE FINISHER

LENGTH: 10.00

LOCATION: Grand Army of the Republic

FULL-NAME: gunship of the Republic

NAME: PAGME STARSHIP

LENGTH: 76.00

LOCATION: Naboo

FULL-NAME: Naboo Royal Starship

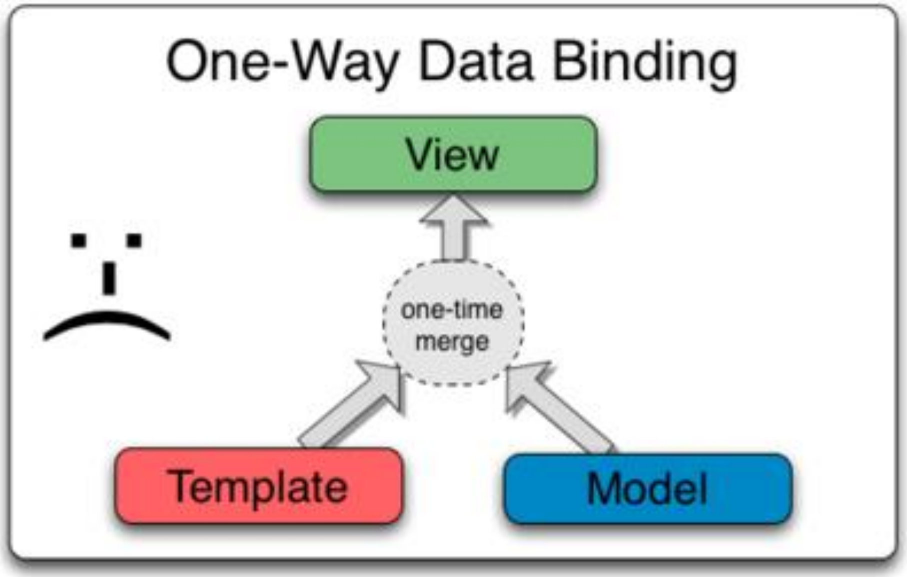
NAME: CORVETTE

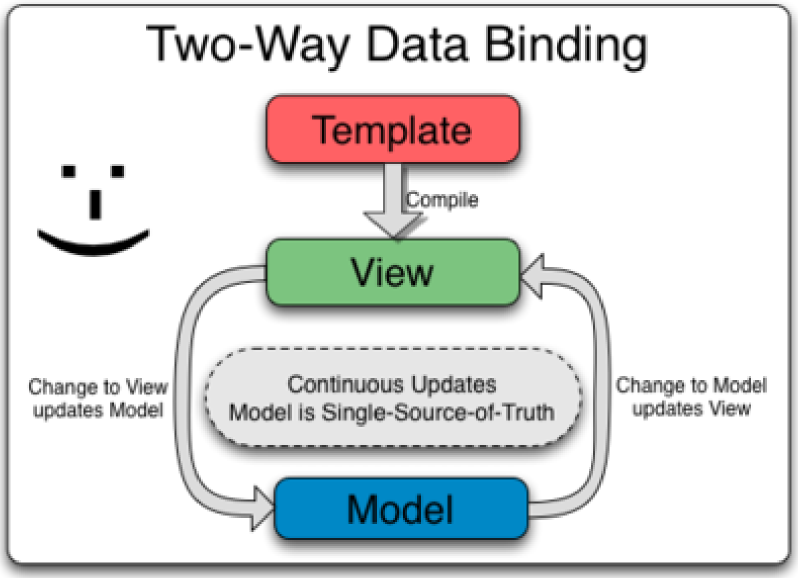
LENGTH: 150.00

LOCATION: Corellian Engineering Corporation

FULL-NAME: CR70 corvette

Binding de données



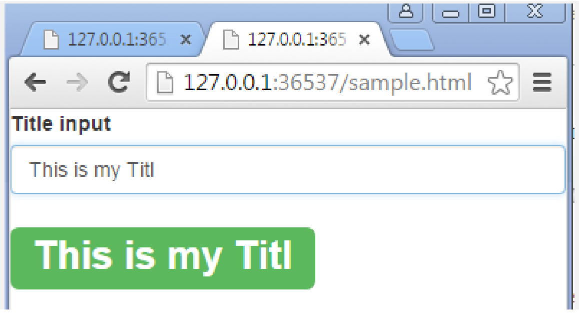

Two-Way Data Binding
Template
Compile
View
Change to View updates Model
Continuous Updates Model is Single-Source-of-Truth
Change to Model updates View
Model

https://docs.angularjs.org/guide/databinding

Binding de données

- ng-model and inputs
  - Assigne une expression angular.js au binding de données

```
<form>
    <label for="titleInput">Title input</label>
    <input  id="titleInput" ng-model="crt.title">
</form>
<h1> {{crt.title}} </h1>
```


127.0.0.1:365
127.0.0.1:365
127.0.0.1:36537/sample.html

Title input

This is my Titl

This is my Titl

- Directives supplémentaires sur Input
  - Outils supplémentaires de contraintes sur l'input

ng-required **ng-minlength** **ng-maxlength** **ng-pattern**

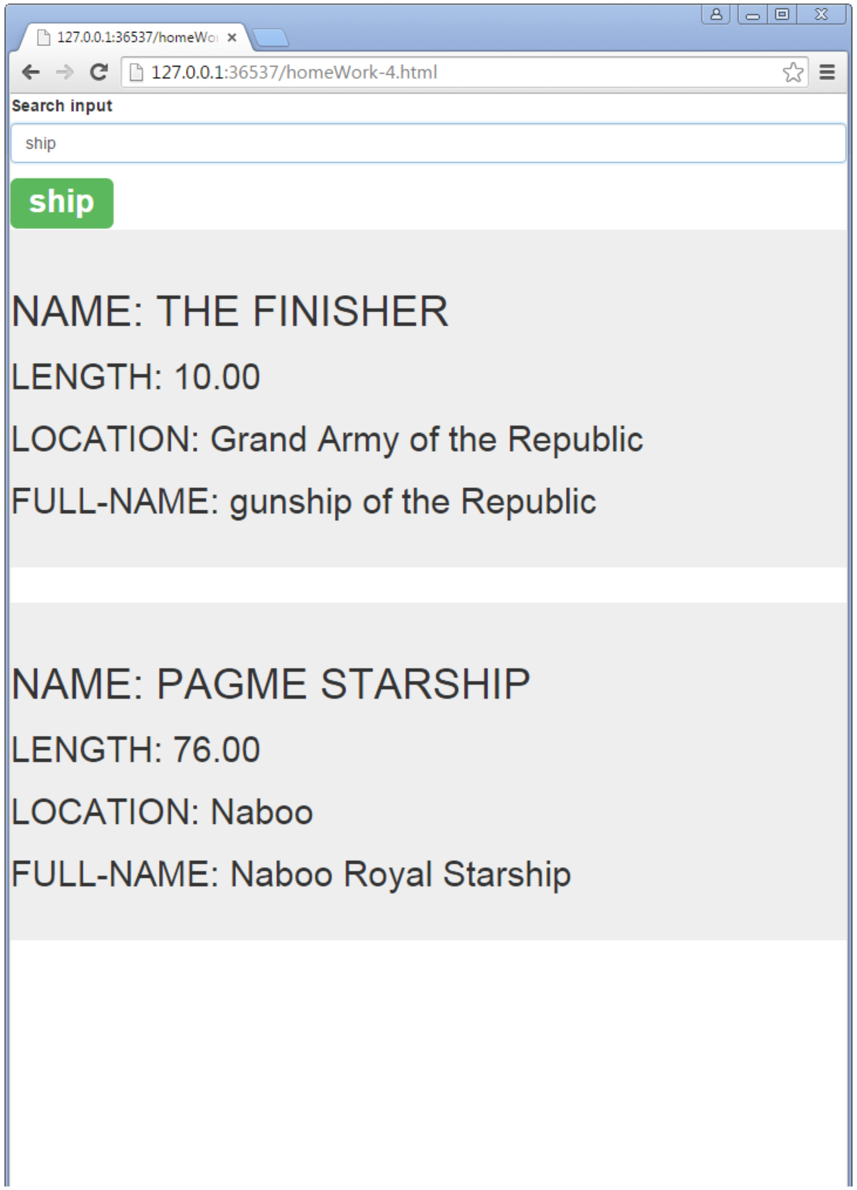
```
<input  id="titleInput"
            ng-model="crt.title"  ng-required=« {{crt.shipList.length >0}}"
            ng-ng-minlength="3" ng-maxlength="10"
            ng-pattern=" ' [a-zA-Z-_.]+' ">
```

ÉCOLE SUPÉRIEURE DE CHIMIE PHYSIQUE ÉLECTRONIQUE DE LYON

A vous de jouer !

- ❑ Crée un input permettant de définir la chaine de caractère à chercher dans une liste d'objet

Affichage et manipulation de données



Search input

ship

ship

NAME: THE FINISHER

LENGTH: 10.00

LOCATION: Grand Army of the Republic

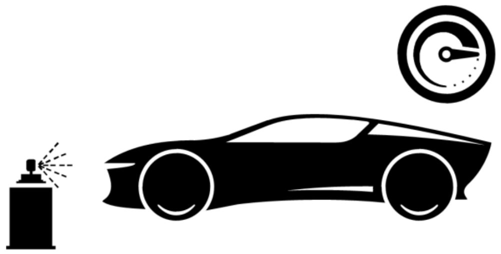
FULL-NAME: gunship of the Republic

NAME: PAGME STARSHIP

LENGTH: 76.00

LOCATION: Naboo

FULL-NAME: Naboo Royal Starship



Customize your Angular!

Les Services

- ❑ Boites à outils d'angular.js
- ❑ Ajoutés en tant que dépendances
  - Uniquement instancié slors qu'on composant en dépend
  - Une seule instance pour toute l'application (Singleton)
- ❑ Un ensemble d'outils (buildin service start always with $)

$http	$route	$log	$q
$window	$animate	$filter	…

https://docs.angularjs.org/api/ng/service

Les Services

- Usage pour les services ng

```
var app = angular.module('sampleApp',[]);

app.controller('SampleCtrl',['$scope','$log','$window',function($scope,$log,$window) {

                $log.info('constuction made');
}]);
```

- Usage pour les services présents dans d’autres modules

```
var app = angular.module('sampleApp',[  'ngRoute'   ]  );

app.controller('SampleCtrl',    ['$route',  function( $route  )  {


}]);
```

https://docs.angularjs.org/api/ng/service

Créer vos propres Services

- ❑ Différents type de services
  - ❑ **Factory** : utiliser pour créer des objets, ajouter des propriétés et les retourner
  - ❑ **Service**: utiliser pour fournir des fonctionnalités au controller
  - ❑ **Provider**: utiliser pour modifier la configuration avant de fournir des fonctionnalités aux controllers

```
app.factory('MyFactory', function(){
  var service={};

   service.createUser=function(){
      return {surname:'john',lastname:'doe'};
   };
   return service;

});
```

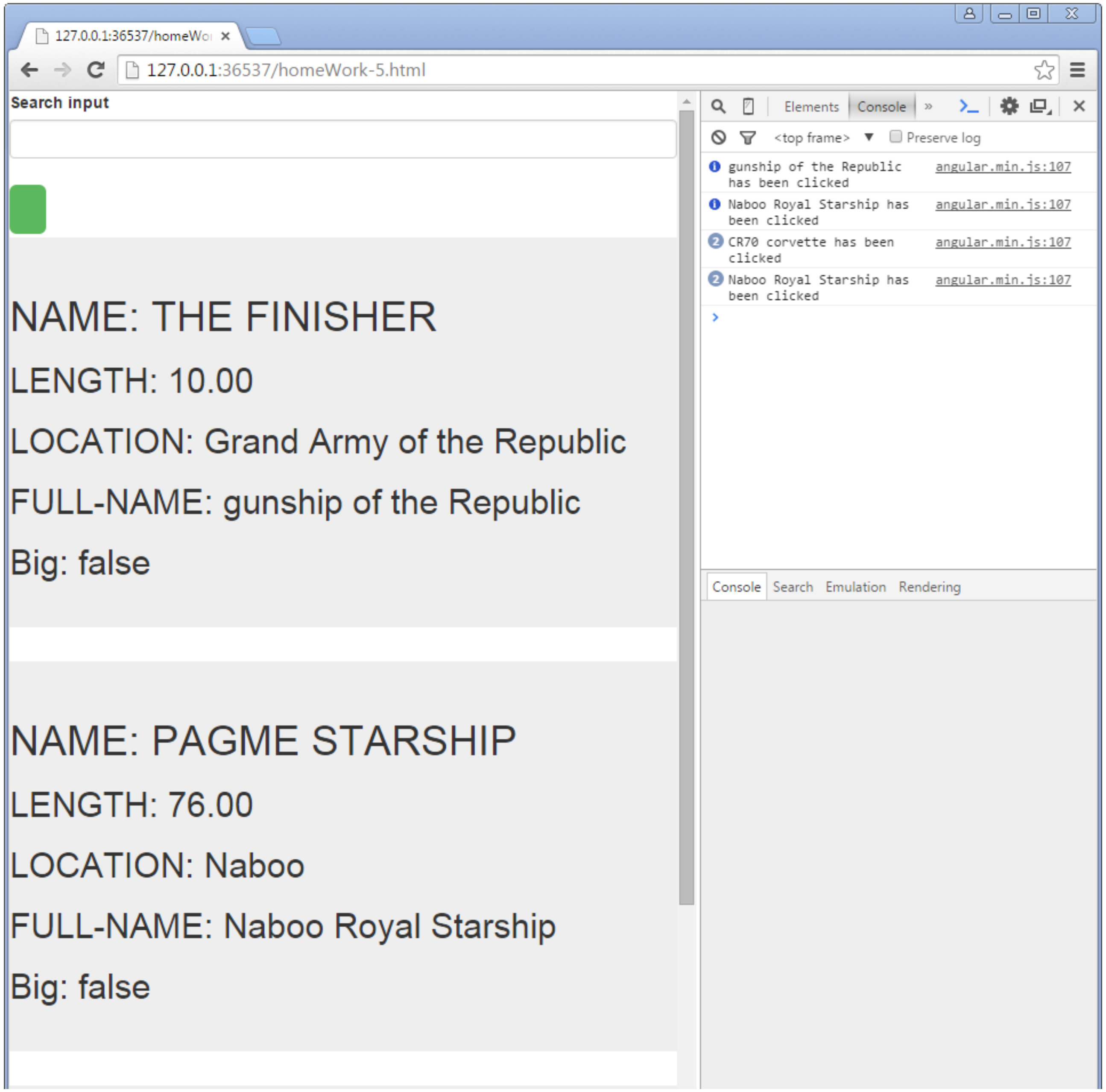
```
app.service('MyService', function($http,$log){
  var service={};

   service.sendData=function(data){
      $http.post('/savePres',data).
         success(function(data, status,
                                   headers, config) {
         $log.info('Success data sent');
      });
   };
   return service;
});
```

```
app.provider('MyProvider', function(){


});
```

A vous de jouer !

- Créer un service permettant de répondre à la question « it is a BigShip ? » retournant vrai si length >100
- Utiliser le service $log, afin d'afficher dans la console le nom du vaisseau cliqué

Customize your Angular !



Search input

NAME: THE FINISHER

LENGTH: 10.00

LOCATION: Grand Army of the Republic

FULL-NAME: gunship of the Republic

Big: false

NAME: PAGME STARSHIP

LENGTH: 76.00

LOCATION: Naboo

FULL-NAME: Naboo Royal Starship

Big: false

```
gunship of the Republic has been clicked      angular.min.js:107
Naboo Royal Starship has been clicked         angular.min.js:107
2 CR70 corvette has been clicked              angular.min.js:107
2 Naboo Royal Starship has been clicked       angular.min.js:107
>
```

Créer vos propres Directives

- ❑ Etendre l'usage d'HTML et facilité l'usage d'angular.js
- ❑ Créer des composants (template) réutilisable facilement

```
app.directive('infoBox', function() {          Création de la directive
 return {
   scope: true,                                Utilisation d'un scope qui hérite du scope parent
   restrict: 'AEC',                            Définit comment la directive peut être appelée
   templateUrl : 'infoMsg.html'                Rendu html inséré
 };
});
```

```
<div info-box></div>             <!--restrict A -->
<info-box></info-box>            <!--restrict E -->
<div class="info-box"></div>     <!--restrict C -->
```

Customize your Angular !

index.html

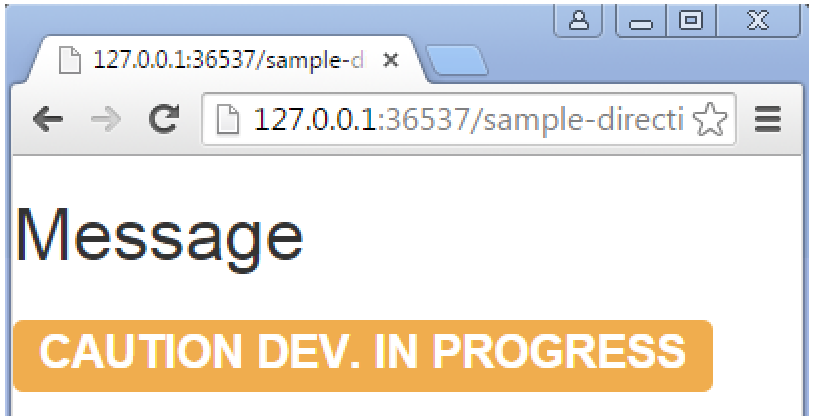
```
<!DOCTYPE html>
<html ng-app="myapp">
   <head>
    <link href="css/bootstrap.min.css"
          rel="stylesheet">
   </head>
   <body>
      <div ng-controller="myController as crt">
         <div info-box></div>
      </div>

      <script src="angular.min.js"></script>
      <script src="sample-directive.js"></script>
   </body>
</html>
```

Sample-directive.js

```
var app = angular.module('myapp', []);
app.controller('myController',['$scope',function($scope){
      $scope.infoMsg={
            type:'WARNING',
             msg:'CAUTION DEV. IN PROGRESS'
      };
}]);

app.directive('infoBox', function() {
      return {
          scope: true,
          restrict: 'AEC',
          templateUrl : 'infoMsg.html'
      };
});
```

InfoMsg.html

```
<h1>Message</h1>
<h2 ng-show="infoMsg.type == 'WARNING'" >
   <span class="label label-warning"> {{infoMsg.msg}}
   </span>

</h2>
<h2 ng-show="infoMsg.type == 'INFO'" >
   <span class="label label-primary"> {{infoMsg.msg}}
   </span>
</h2>
```



Message

CAUTION DEV. IN PROGRESS

Créer vos propres Directives

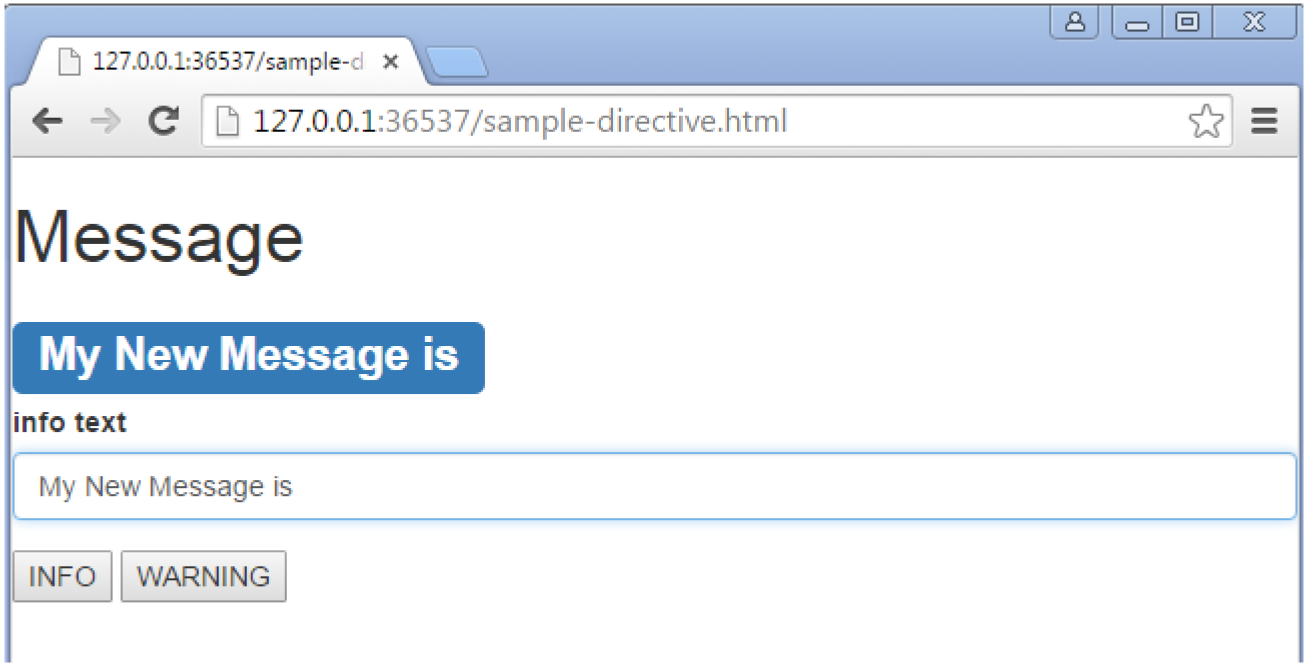
- Et bien plus…
- Interaction avec les scopes, usage des attributs de l'élément réaction sur évènements…

https://docs.angularjs.org/guide/directive

A vous de jouer !

- ❑ Créer votre propre directive permettant d’afficher un message Warning si type=« WARNING», info si type =« INFO»
- ❑ Créer une page Html permettant d’afficher la directive
- ❑ Créer 2 boutons 1 settant type=INFO, l’autre type=Warning
- ❑ Crée un champ de saisie permettant de modifier le texte du message

Affichage et manipulation de données

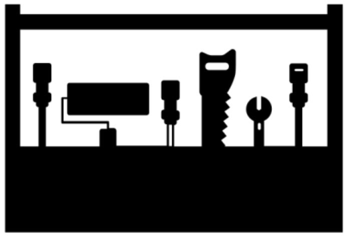


Message

My New Message is

info text

My New Message is



Boites à outils et Scope

Comprendre les scopes

- Considéré comme le modèle de l'application
- Représente la glue entre les controllers et la vue
- Fournit une boite d'outils pour
  - réagir à des changements ($watch)
  - propager des changements ($apply)
- Structuré de façon hiérarchique

Boites à outils et scope

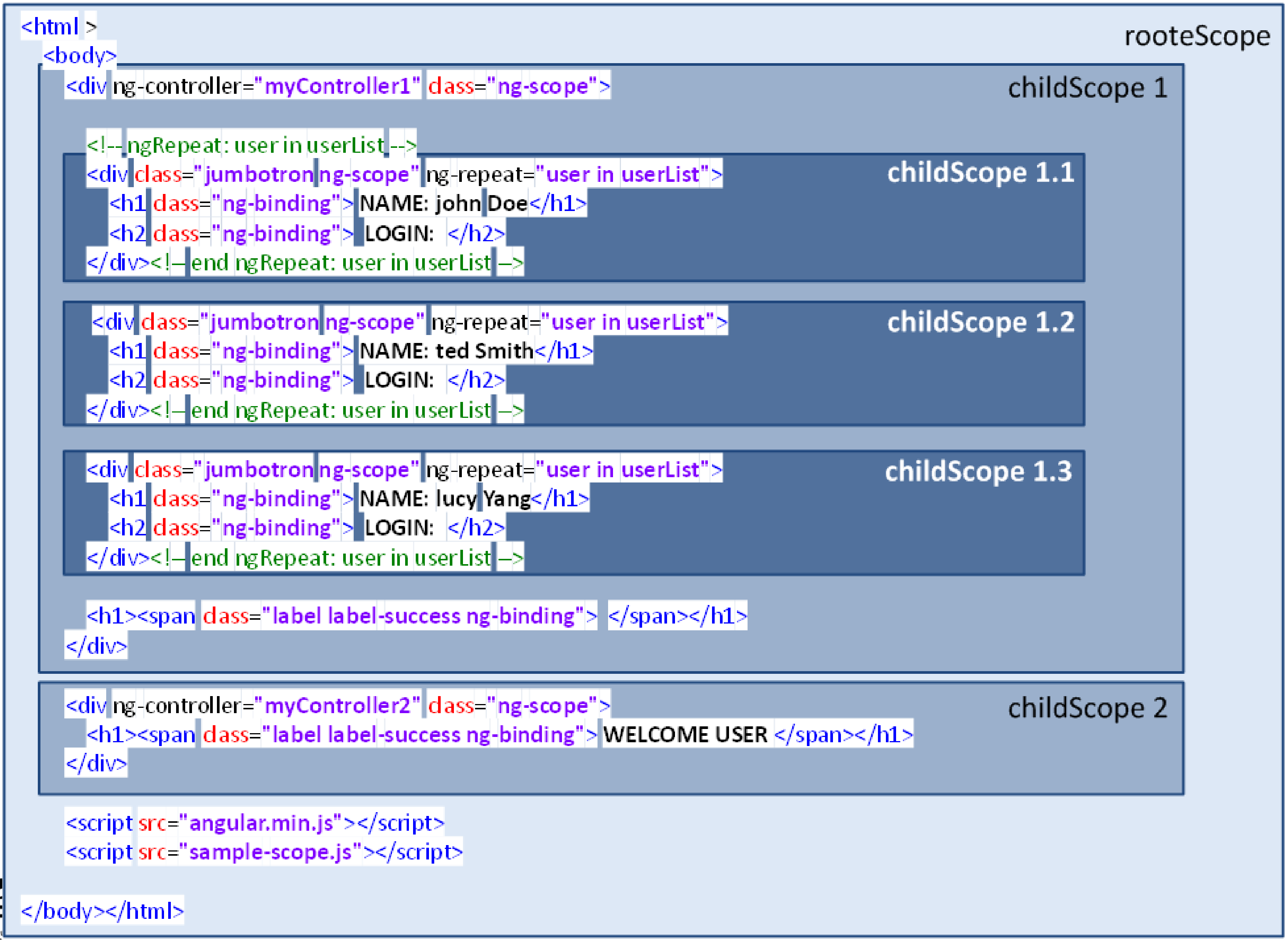
```html
<!DOCTYPE html>
<html ng-app="myapp">
   <head>
    <link href="css/bootstrap.min.css" rel="stylesheet">
   </head>
   <body>
      <div ng-controller="myController1">
         <div class="jumbotron" ng-repeat="user in userList">
            <h1> NAME: {{ user.fullName }}</h1>
            <h2>  LOGIN: {{ ship.login }} </h2>
         </div>
         <h1>
                  <span class="label label-success">
                  {{infoMsg}} </span>
          </h1>
      </div>
      <div ng-controller="myController2">
         <h1>
                  <span class="label label-success">
                   {{infoMsg}} </span>
         </h1>
      </div>

      <script src="angular.min.js"></script>
      <script src="sample-scope.js"></script>
   </body>
</html>
```

```javascript
var app = angular.module('myapp', []);
app.controller('myController1',
                  ['$scope',function($scope){
$scope.userList=[
      {fullName:'john Doe',
                  login:'jDoe'},
       {fullName:'ted Smith',
                   login:'tSmith'},
       {fullName:'lucy Yang',
                   login:'lYang'}
      ];
}]);
app.controller('myController2',
                  ['$scope',function($scope){
                  $scope.infoMsg='WELCOME USER';
}]);
```

Boites à outils et scope

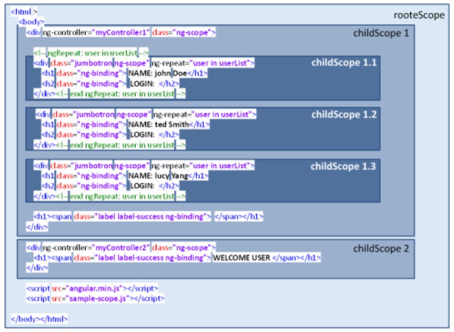
```
<html >
   <body>
      <div ng-controller="myController1" class="ng-scope">

         <!– ngRepeat: user in userList –>
         <div class="jumbotron ng-scope" ng-repeat="user in userList">
            <h1 class="ng-binding"> NAME: john Doe</h1>
            <h2 class="ng-binding">  LOGIN:  </h2>
         </div><!-- end ngRepeat: user in userList -->

          <div class="jumbotron ng-scope" ng-repeat="user in userList">
            <h1 class="ng-binding"> NAME: ted Smith</h1>
            <h2 class="ng-binding">  LOGIN:  </h2>
         </div><!-- end ngRepeat: user in userList -->

         <div class="jumbotron ng-scope" ng-repeat="user in userList">
            <h1 class="ng-binding"> NAME: lucy Yang</h1>
            <h2 class="ng-binding">  LOGIN:  </h2>
         </div><!-- end ngRepeat: user in userList -->

         <h1><span class="label label-success ng-binding">  </span></h1>
      </div>

      <div ng-controller="myController2" class="ng-scope">
         <h1><span class="label label-success ng-binding"> WELCOME USER </span></h1>
      </div>

      <script src="angular.min.js" ></script>
      <script src="sample-scope.js"></script>

</body></html>
```

Boites à outils et scope

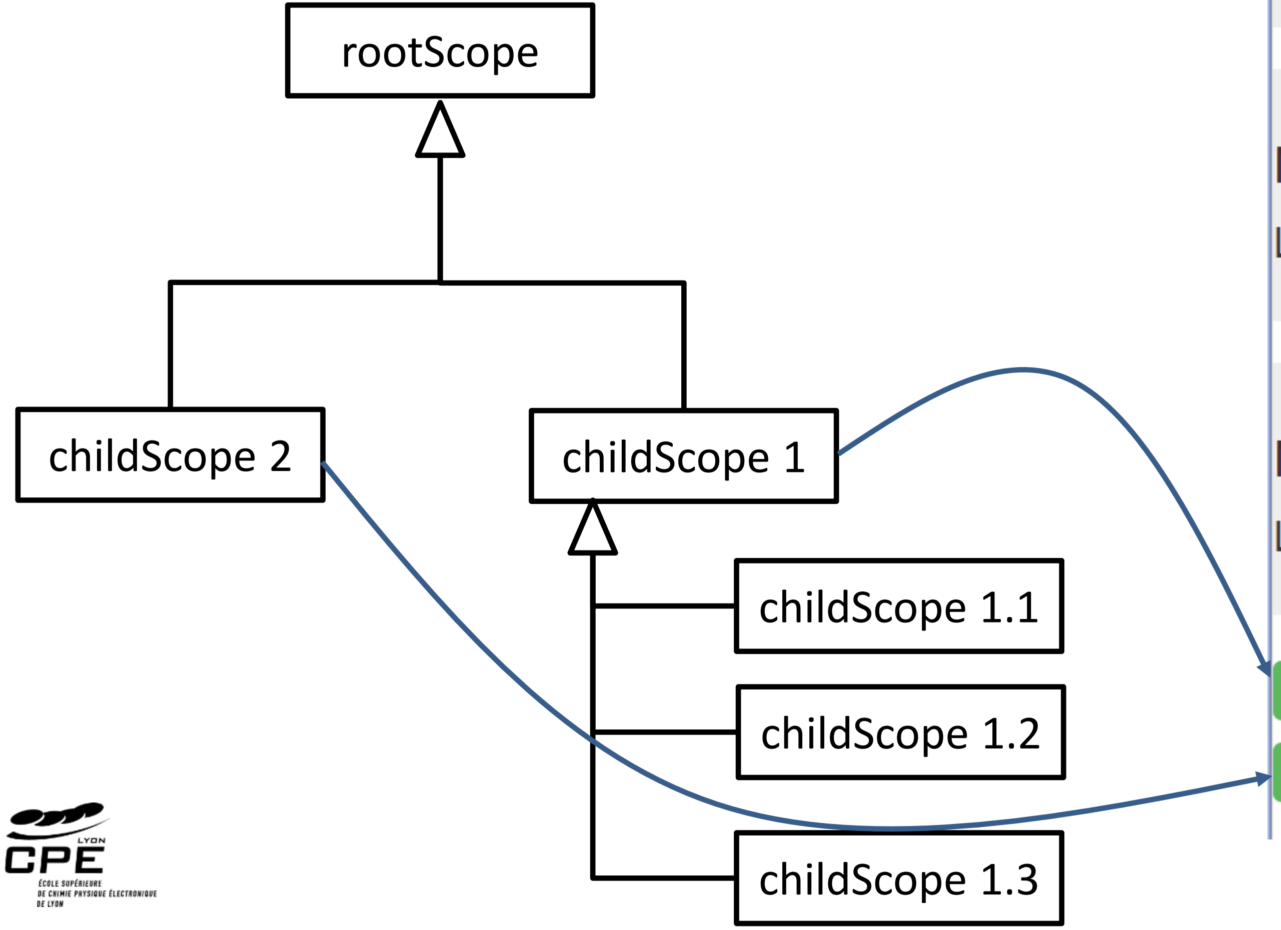


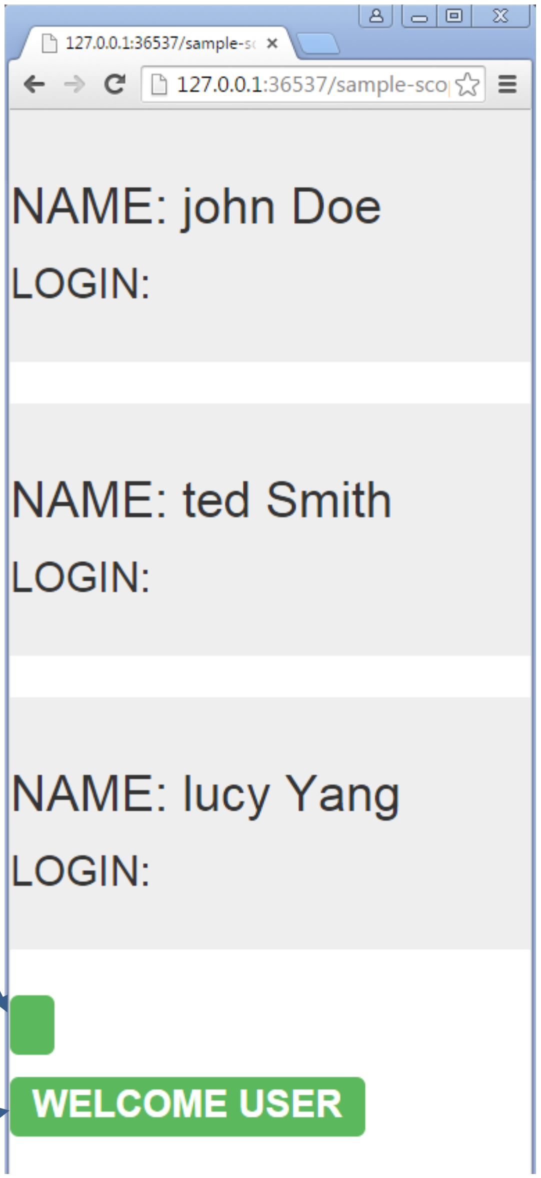
```
<html >
  <body>
    <div ng-controller="myController1" dass="ng-scope">

      <!-- ngRepeat: user in userList -->
      <div class="jumbotron ng-scope" ng-repeat="user in userList">
        <h1 dass="ng-binding">NAME: john Doe</h1>
        <h2 dass="ng-binding"> LOGIN:  </h2>
      </div><!—end ngRepeat: user in userList —>

       <div dass="jumbotron ng-scope" ng-repeat="user in userList">
        <h1 dass="ng-binding">NAME: ted Smith</h1>
        <h2 dass="ng-binding"> LOGIN:  </h2>
      </div><!—end ngRepeat: user in userList —>

      <div class="jumbotron ng-scope" ng-repeat="user in userList">
        <h1 dass="ng-binding">NAME: lucy Yang</h1>
        <h2 dass="ng-binding"> LOGIN:  </h2>
      </div><!—end ngRepeat: user in userList —>

      <h1><span dass="label label-success ng-binding"> </span></h1>
    </div>

    <div ng-controller="myController2" dass="ng-scope">
      <h1><span dass="label label-success ng-binding">WELCOME USER </span></h1>
    </div>

    <script src="angular.min.js"></script>
    <script src="sample-scope.js"></script>

</body></html>
```

Boites à outils et scope


rooteScope
childScope 1
childScope 1.1
childScope 1.2
childScope 1.3
childScope 2

```
<html >
 <body>
  <div ng-controller="myController1" class="ng-scope">

   <!-- ngRepeat: user in userList -->
   <div class="jumbotron ng-scope" ng-repeat="user in userList">
     <h1 class="ng-binding">NAME: john Doe</h1>
     <h2 class="ng-binding"> LOGIN:  </h2>
   </div><!-- end ngRepeat: user in userList -->

    <div class="jumbotron ng-scope" ng-repeat="user in userList">
     <h1 class="ng-binding">NAME: ted Smith</h1>
     <h2 class="ng-binding"> LOGIN:  </h2>
   </div><!-- end ngRepeat: user in userList -->

   <div class="jumbotron ng-scope" ng-repeat="user in userList">
     <h1 class="ng-binding">NAME: lucy Yang</h1>
     <h2 class="ng-binding"> LOGIN:  </h2>
   </div><!-- end ngRepeat: user in userList -->

   <h1><span class="label label-success ng-binding"> </span></h1>
  </div>
  <div ng-controller="myController2" class="ng-scope">
   <h1><span class="label label-success ng-binding">WELCOME USER </span></h1>
  </div>

  <script src="angular.min.js"></script>
  <script src="sample-scope.js"></script>

</body></html>
```





NAME: john Doe

LOGIN:

NAME: ted Smith

LOGIN:

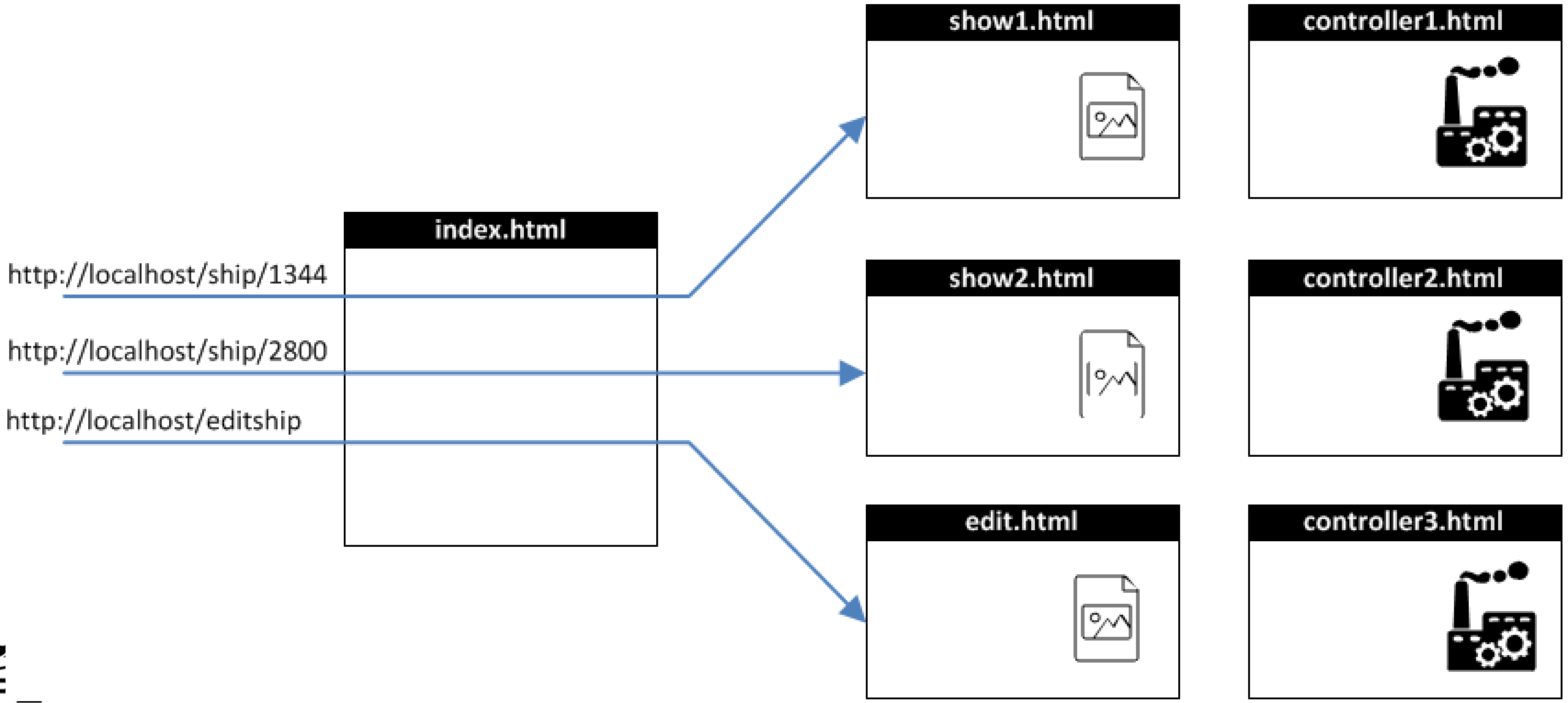
NAME: lucy Yang

LOGIN:

WELCOME USER

Routing:

- ❑ Appel de plusieurs vues au sein de notre application
- ❑ Spécification des controllers relatifs à chaque vue
- ❑ Passage d'information entre les vues

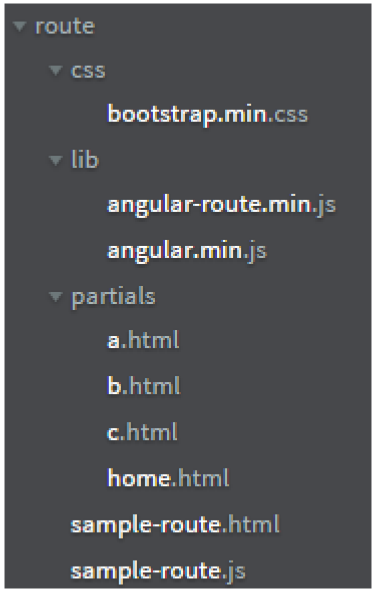


Routing:

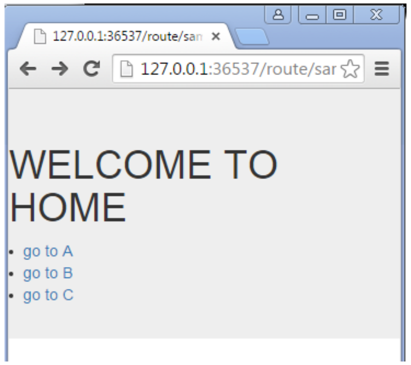
- ❑ Utilisation de **angular-route** ajouter le module ngRoute
  - https://docs.angularjs.org/api/ngRoute
- ❑ Usage de **ngView**
  - directive permettant l'ajout de vues dans le template courant
- ❑ Usage de **$routeProvider**
  - ❑ Service permettant d'orchestrer le comportement lié à des URL
- ❑ Définition de **$routeParams**
  - ❑ Définition d'éléments liés à l'URL e.g /ship/:shipId , toutes les variables définies par **:** sont insérées dans **$routeParams**

Expected Behavior

File Structure



```
route
  css
    bootstrap.min.css
  lib
    angular-route.min.js
    angular.min.js
  partials
    a.html
    b.html
    c.html
    home.html
  sample-route.html
  sample-route.js
```



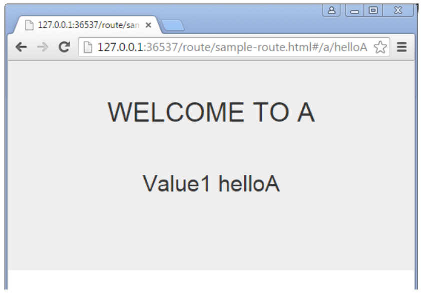
127.0.0.1:36537/route/san

WELCOME TO HOME

- go to A
- go to B
- go to C

```
<div class="jumbotron ">
      <h1>{{message}}</h1>
      <li>
        <a href="#a/helloA"> go to A</a>
       </li>
       <li>
        <a href="#b/helloB"> go to B</a>
       </li>
    <li><a href="#c"> go to C</a></li>
</div>
```

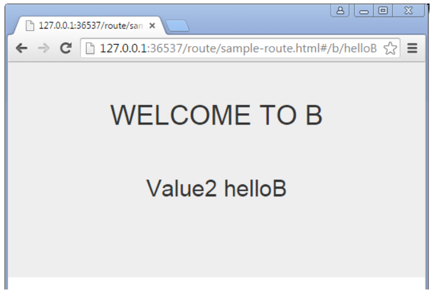
+ MainCtrl



127.0.0.1:36537/route/sample-route.html#/a/helloA

WELCOME TO A

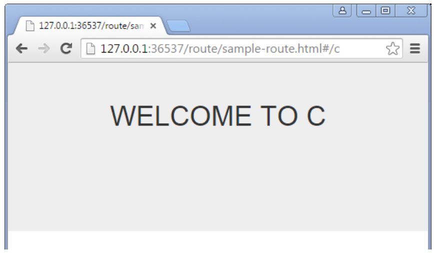
Value1 helloA



127.0.0.1:36537/route/sample-route.html#/b/helloB

WELCOME TO B

Value2 helloB



127.0.0.1:36537/route/sample-route.html#/c

WELCOME TO C

```
<div class="jumbotron ">
      <h1>{{message}}</h1>
       <div class="jumbotron" >
      <h2>
          Value1 {{value1}}
       </h2>
      </div>
  </div>
```

+ ACtrl

```
<div class="jumbotron ">
      <h1>{{message}}</h1>
       <div class="jumbotron" >
      <h2>
          Value2 {{value2}}
       </h2>
      </div>
  </div>
```

+ BCtrl

```
<div class="jumbotron ">
      <h1>{{message}}</h1>
</div>
```

+ CCtrl

Boites à outils et scope

Module

```
var app = angular.module('sampleApp',['ngRoute']);
```

Injection du module de routage

Configuration

```
app.config(['$routeProvider',  function($routeProvider) {

   $routeProvider.
     when('/', {
       templateUrl: 'partials/home.html',
       controller: 'MainCtrl'
     }).
    when('/a/:param1', {
       templateUrl: 'partials/a.html',
       controller: 'ACtrl'
     }).
     when('/b/:param2', {
       templateUrl: 'partials/b.html',
       controller: 'BCtrl'
     }).
     when('/c', {
       templateUrl: 'partials/c.html',
       controller: 'CCtrl'
     }).
     otherwise({
       redirectTo: '/'
     });
  }]);
```

Définition du comportement sur l'appel du chemin

Définition d'un paramètre passé à $routeParams

Définition de la vue cible

Définition du contrôleur de la vue cible

CPE
ÉCOLE SUPÉRIEURE DE CHIMIE PHYSIQUE ÉLECTRONIQUE DE LYON

Boites à outils et scope

Controllers

```
app.controller('MainCtrl',['$scope‘
,function($scope) {
   $scope.message='WELCOME TO HOME';

}]);

app.controller('ACtrl',['$scope','$routeParams',
function($scope,$routeParams) {

   $scope.message='WELCOME TO A';
   $scope.value1=$routeParams.param1;
}]);

app.controller('BCtrl',['$scope','$routeParams',
function($scope,$routeParams) {

   $scope.message='WELCOME TO B';
   $scope.value2=$routeParams.param2;
}]);

app.controller('CCtrl',['$scope',
function($scope) {

   $scope.message='WELCOME TO C';

}]);
```

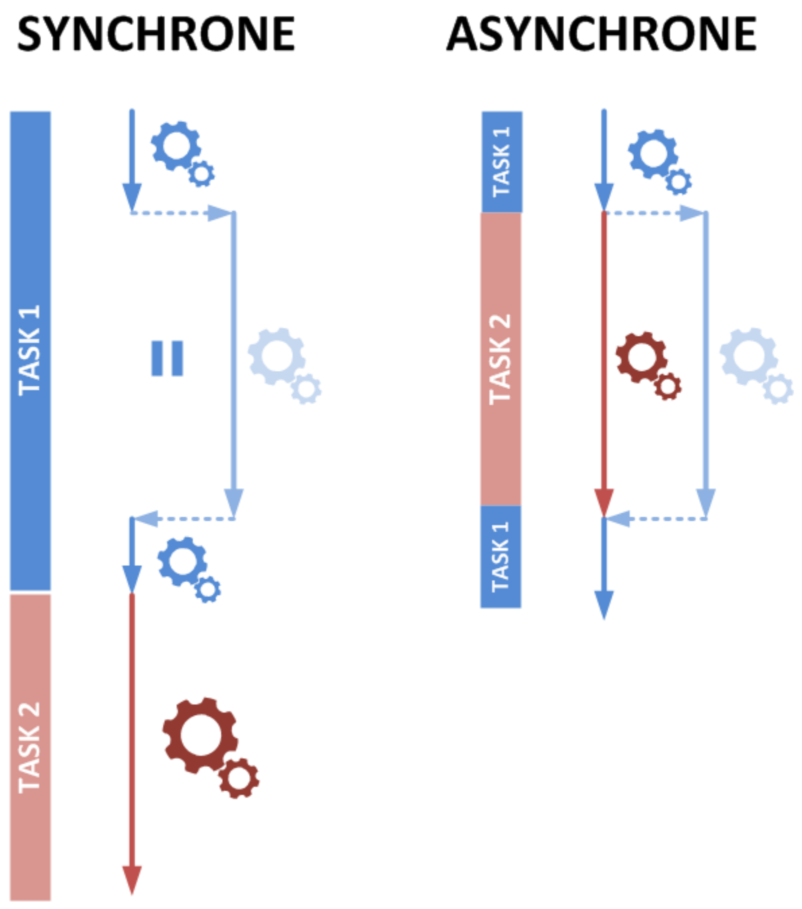
Définition des contrôleurs

Injection du service

Récupération du paramètre

Promise:

- ❑ Retourne un valeur dès que celle-ci est disponible
- ❑ Conserve le caractère asynchrone de l'application



Promise:

- ❑ Disponible via la service $q
- ❑ Création d'un container de données
- ❑ Retour du container de données qui sera rempli ultérieurement
- ❑ Mise à jour du container de données si la donnée est disponible
- ❑ Information du container de données en cas d'erreur

```
var deferred = $q.defer();
```

```
return deferred.promise;
```

```
deferred.resolve(data);
```

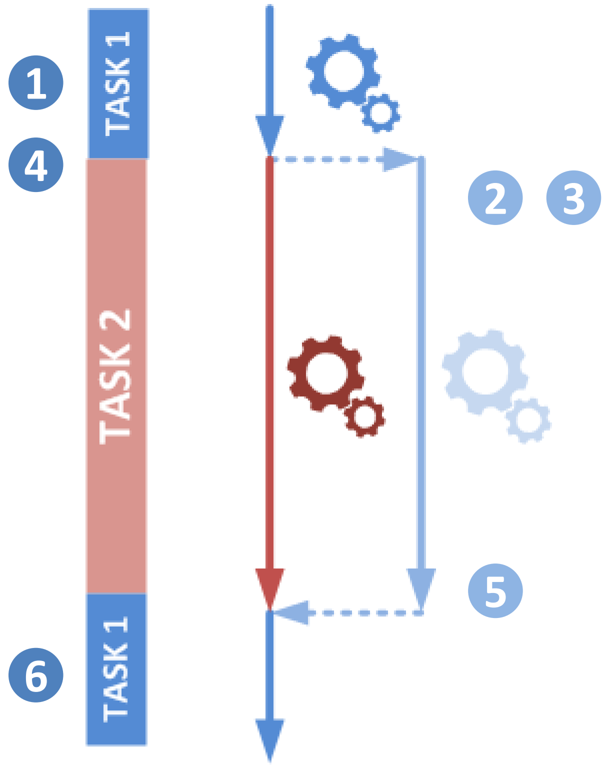
```
deferred.reject(status);
```

Promise:

```
1  var future_content=processData(param1,param2);

4  future_content.then(
       function(payload) {
6              //TODO
       },
       function(errorPayload) {
6              //TODO
       });
```

```
   function processData(param1,param2){
2      var deferred = $q.defer();

               //Processing data take time
               $http.get('/resources_list').
        success(function(data, status, headers, config) {
5              //Set resolve in case of success
               deferred.resolve(data);
       }).
       error(function(data, status, headers, config) {
5             //OR set reject in case of failure
               deferred.reject(status);
        });
      //Return container that will be fill later
3     return deferred.promise;
   };
```

ASYNCHRONE



References

References

https://angularjs.org/
https://docs.angularjs.org/tutorial
http://campus.codeschool.com/courses/shaping-up-with-angular-js/intro
https://docs.angularjs.org/guide/module
http://www.w3schools.com/angular/default.asp
https://docs.angularjs.org/tutorial/step_04
https://docs.angularjs.org/tutorial/step_07
https://docs.angularjs.org/api/ng/service
http://www.sitepoint.com/practical-guide-angularjs-directives/
https://docs.angularjs.org/guide/directive
https://scotch.io/tutorials/single-page-apps-with-angularjs-routing-and-templating

Jacques Saraydaryan

Jacques.saraydaryan@cpe.fr