



TRANSCENDEZ VOS DÉVELOPPEMENTS WEB AVEC ANGULARJS

- HANDS ON -

Thierry Lau / @laut3rry

Sébastien Letélie / @sebmade

Thierry Chatel / @ThierryChatel

PRÉ-REQUIS

- un client git pour récupérer le code source
- node.js pour lancer l'application
- un navigateur Chrome, Firefox ou IE9+
- votre éditeur de code favori

RÉCUPÉRER LES SLIDES

<http://lauterry.github.com/slides-handson-angular>

RÉCUPÉRER LE CODE

`git clone https://github.com/lauterry/angularmovie.git`

ou

Récupérer à partir de Google Drive

DÉROULEMENT

Présentation des concepts clé d'AngularJS ...
... et 14 exercices pour les mettre en pratique

SIMPLES PAGES HTML STATIQUES

Ma vidéothèque 4 films



Avatar

Année de sortie : 2010-09-01

Réalisateur : James Cameron

Acteurs : Sam Worthington, Zoe Saldana, Sigourney Weaver, Stephen Lang, Michelle Rodriguez

Synopsis : Sur la lointaine planète de Pandora, Jake Sully, un héros malgré lui, se lance dans une quête de rédemption, de découverte, d'amour inattendu, dont l'issue sera un combat héroïque pour sauver toute une civilisation.



Seigneur des Anneaux : La Communauté de l'Anneau

Année de sortie : 2003-12-03

Réalisateur : Peter Jackson

Acteurs : Elijah Wood, Sean Astin, Ian McKellen, Sala Baker, Viggo Mortensen

Synopsis : Frodon le Hobbit hérite de l'Anneau Unique, un instrument de pouvoir absolu qui permettrait à Sauron, le Seigneur des ténèbres, de régner sur la Terre du Milieu. Commence alors un vaste périple visant à la destruction de l'objet.



The Grudge

Année de sortie : 2004-12-29

APPLICATION WEB MODERNE

Ma vidéothèque 9 films

[Ajouter un film](#)[Trier par année](#)[Trier par titre](#)

SEIGNEUR DES ANNEAUX : LA COMMUNAUTÉ DE L'ANNEAU



Année de sortie : 2001

Réalisateurs : Peter Jackson

Acteurs : Elijah Wood, Sean Astin, Ian McKellen, Sala Baker, Viggo Mortensen

Synopsis : Frodon le Hobbit hérite de l'Anneau Unique, un instrument de pouvoir absolu qui permettrait à Sauron, le Seigneur des ténèbres, de régner sur la Terre du Milieu. Commence alors un vaste périple visant à la destruction de l'objet.

Note : ★★★★★



THE GRUDGE



Année de sortie : 2004

Réalisateurs : Takashi Shimizu

Acteurs : Sarah Michelle Gellar, Jason Behr, Clea DuVall, Kadee Strickland, Bill Pullman

Synopsis : Dans ce qui paraît être une paisible maison de Tokyo se cache un épouvantable fléau. Quiconque franchit le seuil de la demeure est aussitôt frappé par une malédiction qui ne tardera pas à le tuer dans un

POUR CHAQUE CONCEPT :

1. Présentation
2. Enoncé de l'exercice
3. A vous de jouer !
4. Live coding de la réponse + explications

A LA TRaine ?

14 exos = 14 branches git

```
git checkout -f step-#
```

C'EST PARTI !

```
git checkout -f step-0
```

INITIALISATION ET DOUBLE DATA BINDING

Charger angularjs dans votre page ...

```
<html>
  <head>
    <script src="path/to/angular.js"></script>
  </head>
  <body>

    </body>
</html>
```


... le rendre actif sur celle-ci ...

```
<html ng-app>
  <head>
    <script src="path/to/angular.js"></script>
  </head>
  <body>

    </body>
</html>
```

... et vérifier que ça marche

```
<html ng-app>
  <head>
    <script src="path/to/angular.js"></script>
  </head>
  <body>
    {{ 1 + 1 }}
  </body>
</html>
```



Bonjour

JAVASCRIPT

```
<html>
  Bonjour <span id="name"></span>
  <input type="text"/>
</html>
```

```
window.onload = function(){

  var span = document.getElementById('name');
  var input = document.getElementsByTagName('input')[0];

  input.onkeyup = function(){
    if (span.textContent || span.textContent === "") {
      span.textContent = input.value;
    } else if(span.innerText || span.innerText === "") { // IE
      span.innerText = input.value;
    }
  };
};
```

JQUERY

```
<html>
  Bonjour <span id="name"></span>
  <input type="text"/>
</html>
```

```
$(document).ready(function() {
  var $input = $('input');
  var $span = $('#name');

  $input.keyup(function (event) {
    $span.text(event.target.value);
  });
});
```

ANGULARJS

```
<html ng-app>
  <div>
    <input type="text" ng-model="name">
    <p>Bonjour {{name}}</p>
  </div>
</html>
```

EXERCICE 1 : SALUEZ QUI VOUS VOULEZ !

Changer dynamiquement le nom par le contenu d'un champ de saisie

- Pensez à initialiser AngularJs dans la page `index.html`
- La librairie `angularjs` est disponible dans le dossier `js/lib`
- Rajouter un champ de saisie
- Nommons par exemple la variable `{{name}}`
- Remplacer le titre `AngularMovie` par `{{name}}`

CONTROLLER ET SCOPE

```
<html ng-app>
  <div ng-controller="myController">
    <input type="text" ng-model="name">
    <p>Hello {{name}}</p>
  </div>
</html>
```

```
function myController($scope){
  $scope.name = 'World';
}
```

EXERCICE 2 : IMPORTANCE DU SCOPE

Gardez les mêmes variables `{ {name} }` mais leur affecter des valeurs différentes

- Créez un fichier `app/js/app.js` et y écrire vos 2 controllers
- Attention à l'endroit où vous déclarez vos controllers dans la page HTML

```
<!DOCTYPE html>
<html ng-app>
  <head>...</head>
  <body ng-controller="mainController">

    <h1>{{title}}</h1>

    <div ng-controller="helloController">
      Hello {{name}}
    </div>

    <div ng-controller="nameController">
      My name is {{name}}
      and i'm talking about {{title}}
    </div>

  </body>
</html>
```

AngularJS

Hello GDG Paris

My name is Thierry LAU and i'm talking about AngularJS

```
<!DOCTYPE html>
```

```
<html ng-app>
```

```
<head>...</head>
```

```
<body ng-controller="mainController">
```

```
<h1>{{title}}</h1>
```

```
<div ng-controller="helloController">
```

```
  Hello {{name}}
```

```
</div>
```

```
<div ng-controller="nameController">
```

```
  My name is {{name}}
```

```
  and i'm talking about {{title}}
```

```
</div>
```

```
</body>
```

```
</html>
```

\$scope

\$scope

title = "AngularJS"

\$scope

name = "GDG Paris"

\$scope

name = "Thierry LAU"

title = "AngularJS"

AngularJS

Hello GDG Paris

My name is Thierry LAU and i'm talking about AngularJS

MODULE & NAMESPACING

```
<html ng-app="myApp">
  <div ng-controller="myController">
    <input type="text" ng-model="greeting">
    <p>{{greeting}}</p>
  </div>
</html>
```

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

```
myApp.controller('myController', function($scope){
  $scope.greeting = 'Hello World';
});
```

SOLUTION

```
git checkout -f step-1
```

TEMPLATING AVANCÉE

NG-REPEAT

Itère dans une collection et génère un template par élément

```
<ul>
  <li ng-repeat="fruit in fruits">
    /* répétition du template li par élément fruit */
    {{ fruit.name }}
  </li>
</ul>
```

EXERCICE 3 : NG-RÉPÉTER LES FILMS

Afficher la liste des films en utilisant la directive `ng-repeat`

- AngularJs est déjà actif dans la page `movie-list.html`
- Créer un nouveau contrôleur nommé `moviesController` pour la liste des films
- Créer une collection de films dans le contrôleur. Aidez vous de la slide suivante !
- Pensez à rendre dynamique le nombre de films affichés
- JS : accéder aux propriétés d'un objet avec la notation pointée (ex : `car.color`)

```
[
  {
    "id":1,
    "title":"Avatar",
    "releaseYear":"2010",
    "poster":"img/avatar.jpg",
    "directors":"James Cameron",
    "actors":"Sam Worthington, Zoe Saldana, Sigourney Weaver, Stephen Lang, Michelle Rodriguez",
    "synopsis":"Sur la lointaine planète de Pandora, Jake Sully, un héros malgré lui, se lance dans une quête de rédemption, de découverte, d'amour inattendu, dont l'issue sera un combat héroïque pour sauver toute une civilisation.",
    "rate":"3"
  },
  {
    "id":2
```

SOLUTION

```
git checkout -f step-2
```


GESTION DES ÉVÉNEMENTS

NG-CLICK

```
<li ng-click="faireUnTruc(arg1, arg2, ...)"></li>
```

```
function myController($scope){  
    $scope.faireUnTruc = function(arg1, arg2, ...){  
        /* faire un truc */  
    };  
}
```

```
git checkout -f step-3
```

Ajout d'un modal Twitter Bootstrap

EXERCICE 4 : GÉRER UN CLICK

Afficher une `window.alert()` avec quelques infos du films saisies dans le formulaire

- Créer un nouveau controller nommé `movieFormController` pour gérer le formulaire
- N'oubliez pas de binder les champs de saisies au `$scope`

EXERCICE 5 : AJOUTER UN FILM DANS LA LISTE

Ajouter un film dans la liste des films et l'afficher dans la page

- Pensez à la hiérarchie des `$scope`
- Surveillez la position de vos directives `ng-controller`
- Pensez à réinitialiser le formulaire après chaque film rajouté
- Utiliser la méthode `array.push(object)` pour ajouter le film dans le tableau `$scope.movies`

SOLUTION

```
git checkout -f step-3-solution
```

PROPAGATION DES ÉVÉNEMENTS \$SCOPE

Un évènement peut être "diffusé" aux `$scope` enfants ...

```
$scope.$broadcast('myEvent', param1, param2, ...);
```

... ou "émis" aux `$scope` parents

```
$scope.$emit('anotherEvent', param1, param2, ...);
```

L'écoute des évènements se fait comme suit :

```
$scope.$on('yetAnotherEvent', function(event, param1, param2, ...){  
    console.log(param1);  
});
```


EXERCICE 6 : AJOUTER UN FILM DANS LA LISTE (AUTRE SOLUTION)

Ajouter un film dans la liste des films et l'afficher dans la page en utilisant les événements du scope

- \$emit ou \$broadcast ?
- Pensez au \$rootScope, parent de tous les \$scope
- Cette solution peut remplacer celle de l'exercice 5

SOLUTION

```
git checkout -f step-3-scope-event
```

XHR & INJECTION DE DÉPENDANCES

Pour effectuer des appels XHR, Angular dispose du service
\$http ...
... injectable comme le \$scope

```
function myController($scope, $http){  
};
```

Usages

```
$http.get(url, options)
```

```
$http.post(url, data, options)
```

```
$http.put(url, data, options)
```

```
$http.delete(url, options)
```

Gérer les retours

```
$http.get(url, options)
  .success(function(data, status, headers, config){
    /* do something */
  })
  .error(function(data, status, headers, config){
    /* handle error */
  });
```

EXERCICE 6 : FINI LES DONNÉES EN DUR.

Récupérer les films à partir du serveur. Idem pour l'ajout de film

- GET `/server/api/movies`
- POST `/server/api/movies`
- Attention au format du JSON reçu

SOLUTION

```
git checkout -f step-4
```


NAVIGATION CÔTÉ CLIENT

<http://localhost:3001/#/home>

Bonjour Thierry LAU

Bienvenue sur AngularMovie, votre cinéthèque développée avec



<http://localhost:3001/#/movies>

Ma vidéothèque 9 films

Ajouter un film

Rechercher



Trier par année

Trier par titre



AVATAR



Année de sortie : 2010

Réalisateurs : James Cameron

Acteurs : Sam Worthington, Zoe Saldana, Sigourney Weaver, Stephen Lang, Michelle Rodriguez

Synopsis : Sur la lointaine planète de Pandora, Jake Sully, un héros malgré lui, se lance dans une quête de rédemption, de découverte, d'amour inattendu, dont l'issue sera un combat héroïque pour sauver toute une civilisation.

Note : ★★★



SEIGNEUR DES ANNEAUX : LA COMMUNAUTÉ DE L'ANNEAU



<http://localhost:3001/#/movies/edit/1>

Modification

Titre :

Avatar

Année de sortie :

2010

Réalisateur :

James Cameron

Acteurs :

Sam Worthington, Zoe Saldana,

Synopsis :

Sur la lointaine planète de Pandora, Jake Sully, un héros

Note :

3

Annuler

Modifier

SERVICE \$ROUTEPROVIDER

Permet de configurer le template HTML à charger avec son contrôleur associé pour une URL donnée

```
$routeProvider
  .when('/route1', {
    templateUrl: 'path/to/view1.html',
    controller : 'view1Controller'
  })
  .when('/route2', {
    templateUrl: 'path/to/view2.html',
    controller : 'view2Controller'
  })
  .when('/route2/:id', {
    templateUrl: 'path/to/view3.html',
    controller : 'view3Controller'
  })
  .otherwise({
    redirectTo: '/route1'
  });
```


OÙ L'INJECTER ?

Dans la factory config de votre module

```
/* Dans le fichier de définition de votre module app.js */  
var angularMovieApp = angular.module('angularMovieApp', []);  
  
angularMovieApp.config(function($routeProvider) {  
    /* routes configurations */  
});
```

A utiliser avec la directive ng-view

```
<body>
  <header>
    /* Common part for all pages.*/
  </header>
  <section ng-view>
    /* this is the dynamic part */
  </section>
  <footer>
    /* Common part for all pages.*/
  </footer>
</body>
```

Attention : Vous ne pouvez utiliser qu'une seule directive ng-view

EXERCICE 6 : UNE NAVIGATION CÔTÉ CLIENT

La navigation d'une page à l'autre peut être améliorée. Utiliser le service `$routeProvider` et la directive `ng-view` pour ne garder qu'un fichier `index.html`

- Identifier dans votre page `index.html` les portions qui sont communes à toutes les pages et la partie spécifique
- Créer un répertoire `app/partials` où stocker les fragments de page spécifique
- Utiliser soit l'attribut `controller` du service `$routeProvider`, soit la directive `ng-controller`
- Penser à corriger les attributs `href` des liens de la barre de menu supérieure

SOLUTION

```
git checkout -f step-5
```

EXERCICE 7 : SUPPRIMER UN FILM

```
git checkout -f step-5-delete
```

Un bouton de suppression vient d'être ajouté pour chaque film. Implémentez son action

- API:DELETE /server/api/movies/:id
- Astuce : Vous pouvez accéder au numéro d'index courant de la collection dans ng-repeat via \$index
- Utiliser la méthode `array.splice(from, nbToDelete)` pour retirer le film de `$scope.movies`

SOLUTION

```
git checkout -f step-5-delete-solution
```

EXERCICE 8 : MODIFIER UN FILM

```
git checkout -f step-5-edit
```

Un bouton de modification vient d'être ajouté pour chaque film. Implémentez son action

- Petite contrainte : Pas de popup. La page d'édition doit être accessible via l'url `/#/movies/edit/:id`
- Utiliser un nouveau fichier HTML pour la page d'édition et lui affecter un nouveau controller
- `PUT /server/api/movies`
- `GET /server/api/movies/:id`
- Utiliser le service `$routeParams` dans le controller pour récupérer l'id du film à modifier
- Utiliser le service `$location.path('/path')` pour naviguer vers `/#/path`

SOLUTION

```
git checkout -f step-5-edit-solution
```

RÉCAPITULATIFS

- Double Data Binding
 - `$scope` : seule source de vérité
 - Elimination du code de manipulation du DOM (60 à 80% de votre code applicatif)
- Modèle Vue Contrôleur
 - Modèle = simple object javascript
 - Vue = HTML valide
 - Contrôleur = fonction javascript
- Single Page Application
 - Navigation client side
 - Agnostique au Back end

VERS DES CONCEPTS PLUS AVANCÉES

CRÉER VOTRE PROPRE SERVICE

Fournit des tâches spécifiques à l'application

Exemple : \$scope, \$http

CRÉER SES PROPRES SERVICES

```
myModuleApp.factory("CountService", function (dep1, dep2, ...) {  
  
    // privée  
    var counter = 0;  
  
    // publique  
    return {  
        increment : function(){  
            counter = counter + 1;  
        };  
    };  
});
```

```
function myController($scope, CountService){  
    CountService.increment();  
};
```

EXERCICE 9 : CRÉER VOTRE PROPRE SERVICE

Actuellement, nous utilisons le service `$http` en l'injectant dans les controllers et nous dupliquons à chaque fois l'url de l'API. Améliorons ceci en créant notre propre service pour nous connecter au back end pour les opérations de CRUD sur les films.

- Appelons ce service `Movies`
- Par exemple, dans le contrôleur, au lieu d'appeler `$http.get("/server/api/movies")`, j'aimerais appeler `Movies.fetch()` ;
- De manière analogue, idem pour les autres opérations CRUD

SOLUTION

```
git checkout -f step-7
```

FILTRES ANGULARJS

Formate les données avant de les afficher

hello
HELLO

1368730200

Jeudi 16 mai 2013 20H50

1234.562342

1 234,56 €

[Thierry, David, Nelly, Alex, Claire]

[Alex, Claire, David, Nelly, Thierry]

[pommes, pâtes, farine, PQ]

[PQ]

**TRANSFORMER LA DONNÉE AVANT DE
L’AFFICHER À L’UTILISATEUR**

FILTRES ANGULARJS

- currency
- date
- lowercase
- uppercase
- limitTo
- filter
- orderBy

FILTRE FILTER

Permet de filtrer un sous ensemble d'une collection et retourne ce sous ensemble.

```
<div>{{collection | filter:predicat}}</div>
```

Ce filtre accepte un predicat (string) en argument. Toute chaine de caractère ou propriété qui contient ce prédicat sera retournée dans une nouvelle collection.

EXERCICE 10 : RECHERCHER UN FILM

```
git checkout -f step-8
```

Une barre de recherche est disponible. Implémentez une fonction de recherche en utilisant le filtre `filter`

- Tirez partie du data-binding
- Pas de code nécessaire dans le controller, seulement dans la page HTML

FILTRE ORDERBY

Tri une collection selon un predicat

```
<div>{{collection | orderBy:predicat:reverse}}</div>
```

Ce filtre accepte un predicat (string) en argument. Ce prédicat désigne la propriété sur lequel est fait le tri. Un deuxième argument (boolean) permet d'inverser le sens du tri

EXERCICE 11 : TRIER LES FILMS PAR ANNÉE ET PAR TITRE

2 boutons de tri sont disponibles. Utilisez le filtre `orderBy` pour implémentez les tris correspondants

- Tirez partie du data-binding
- Les filtres peuvent être chaînés
- Pas de code nécessaire dans le controller, seulement dans la page HTML

SOLUTION

```
git checkout -f step-8-solution
```

CRÉER SES PROPRES FILTRES

```
angularMovieApp.filter('greet', function () {  
    return function(inputValue, arg1, arg2, ....) {  
        return 'Hello ' + inputValue;  
    };  
});
```

```
<div>{{name | greet}}</div>  
<div>{{name | greet:arg1:arg2}}</div>
```

EXERCICE 12 : JE VEUX DE VRAIES ÉTOILES

```
git checkout -f step-9
```

Créer un filtre permettant d'afficher le nombre d'étoiles correspondant à la note.

- Voici la caractère unicode pour une étoile : "\u2605"
- Créez le filtre dans un nouveau fichier filters.js

EXERCICE 13 : UNE IMAGE VIDE PAR DÉFAULT

Dans le répertoire `img`, se trouve une image `no-poster.jpg`.
Créer un filtre qui affiche cette image lorsqu'aucun poster
n'est disponible pour un film.

- Mmmmmh, sur quoi vais-je appliquer le filtre ?

SOLUTION

```
git checkout -f step-9-solution
```

DIRECTIVES ANGULARJS

PERMET DE :

Étendre le langage HTML

Créer ses propres composants

DIRECTIVE NG-SWITCH

Change la structure du DOM de manière conditionnel

```
<section ng-switch on="value">
  <div ng-switch-when="one">Valeur1</div>
  <span ng-switch-when="two">Valeur2</span>
  <div ng-switch-default >Default</div>
</section>
```

EXERCICE 14 : UNE LISTE DE FILMS PLUS COMPACTE

```
git checkout -f step-10
```

Vous allez permettre à l'utilisateur de basculer vers une vue plus compacte de la liste des films.

- Un bouton vient d'être rajouté.
- Utiliser la directive `ng-switch`
- Tirez partie du data-binding
- Aidez vous du snippet de code de la slide suivante pour une vue en mode tableau

```
<table class="table table-striped">
  <thead>
    <tr>
      <th>#</th>
      <th>Titre</th>
      <th>Réalisateur</th>
      <th>Année de sortie</th>
      <th>Note</th>
    </tr>
  </thead>
  <tbody>
    <tr ng-repeat="movie in movies | orderBy:tri:reverse | filter : search ">
      <td>{{$index + 1}}</td>
      <td>{{movie.title}}</td>
      <td>{{movie.directors}}</td>
      <td>{{movie.releaseYear}}</td>
```

SOLUTION

```
git checkout -f step-10-solution
```

DIRECTIVES VUES

- ng-repeat
- ng-model
- ng-controller
- ng-switch

PLUSIEURS TYPES D'INVOCATIONS POSSIBLES

```
<div navbar ></div>
```

```
<navbar></navbar>
```

```
<div class="navbar" ></div>
```

```
<!-- directive : navbar -->
```

Primary tab

Second tab

Some other tab

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pharetra semper lorem et porta. Maecenas vehicula ullamcorper sed aliquam purus sollicitudin et. Pellentesque feugiat. Fusce dapibus diam eu nisl hendrerit id scelerisque quam. Fermentum turpis vitae nunc iaculis euismod. Morbi congue tellus euismod eget porta risus vulputate. Curabitur sed

```
<ul class="nav-tabs">
  <li class="active">
    <a href="#">Primary tab</a>
  </li>
  <li>
    <a href="#">Primary tab</a>
  </li>
  <li>
    <a href="#">Primary tab</a>
  </li>
</ul>
```


Primary tab

Second tab

Some other tab

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pharetra semper lorem et porta. Maecenas vehicula ullamcorper sed aliquam purus sollicitudin et. Pellentesque feugiat. Fusce dapibus diam eu nisl hendrerit id scelerisque quam. Fermentum turpis vitae nunc iaculis euismod. Morbi congue tellus euismod eget porta risus vulputate. Curabitur sed

```
<tabs>
  <pane title="Primary tab" active>
    <!-- contenu -->
  </pane>
  <pane title="Secondary tab">
    <!-- contenu -->
  </pane>
  <pane title="some other tab">
    <!-- contenu -->
  </pane>
</tabs>
```


LIBÉREZ VOTRE IMAGINATION

```
<calendar></calendar>
```

```
<progress-bar></progress-bar>
```

```
<dialog></dialog>
```

```
<datepicker></datepicker>
```

```
<accordion></accordion>
```

CRÉER SES PROPRES DIRECTIVES

- Scope
- Linking function
- Compile
- Restriction
- Transclusion




CRÉER VOTRE PROPRE DIRECTIVE


```
myModuleApp.directive("MaDirective", function (dep1, dep2) {  
  return {  
    template : "<div></div>",  
    templateUrl : "/path/to/ma-directive.html",  
    replace : true,  
    restrict : 'E,A,C,M',  
    transclude : true,  
    scope : {},  
    link : function(scope, element, attributes){  
  
    }  
  };  
});
```

{{ DEMO }}

OUTILS AUTOUR D'ANGULARJS

BATARANG

 Chrome Web Store



thierlau@gmail.com 

batarang



Résultats de recherche pour "batarang" dans la catégorie :

TousApplicationsExtensionsThèmes

Extensions1 RÉSULTATS D'EXTENSIONS SUR 1



AngularJS Batarang
Auteur : AngularJS
Extends the Developer Tools, adding tools for debugging and profiling AngularJS applications.

 ÉVALUER 

Outils de développement
★★★★★

YEOMAN



Home



Why Yeoman?



Getting Started (1.0)



- Road to 1.0
- Migrating from 0.9
- Deployment
- Roadmap
- Support
- Package Manager
- About



MODERN WORKFLOWS FOR MODERN WEBAPPS



YO



GRUNT



BOWER

1.0 BETA

Yeoman 1.0 is more than just a tool. It's a workflow; a collection of tools and best practices working in harmony to make developing for the web even better.

KARMA (TESTACULAR)

Karma

Spectacular Test Runner for JavaScript

[intro](#)[config](#)[plus](#)[dev](#)[about](#)[v0.8](#)

// On the AngularJS team, we rely on testing and we always seek better tools to make our life easier. That's why we created Karma - a test runner that fits all our needs.

[View project on GitHub](#)

```
npm install -g karma
```

Things should be simple. We believe in testing and so we wanna make it as simple as possible.

The main goal for Karma is to bring a productive testing environment to developers. An environment, where they don't have to set up many things and rather just write the code and get an instant feedback. Because getting a quick feedback is what makes you productive and **creative**.



Test on Real Devices

Test your code on real browsers and real devices such as phones, tablets or headless PhantomJS.



Remote Control

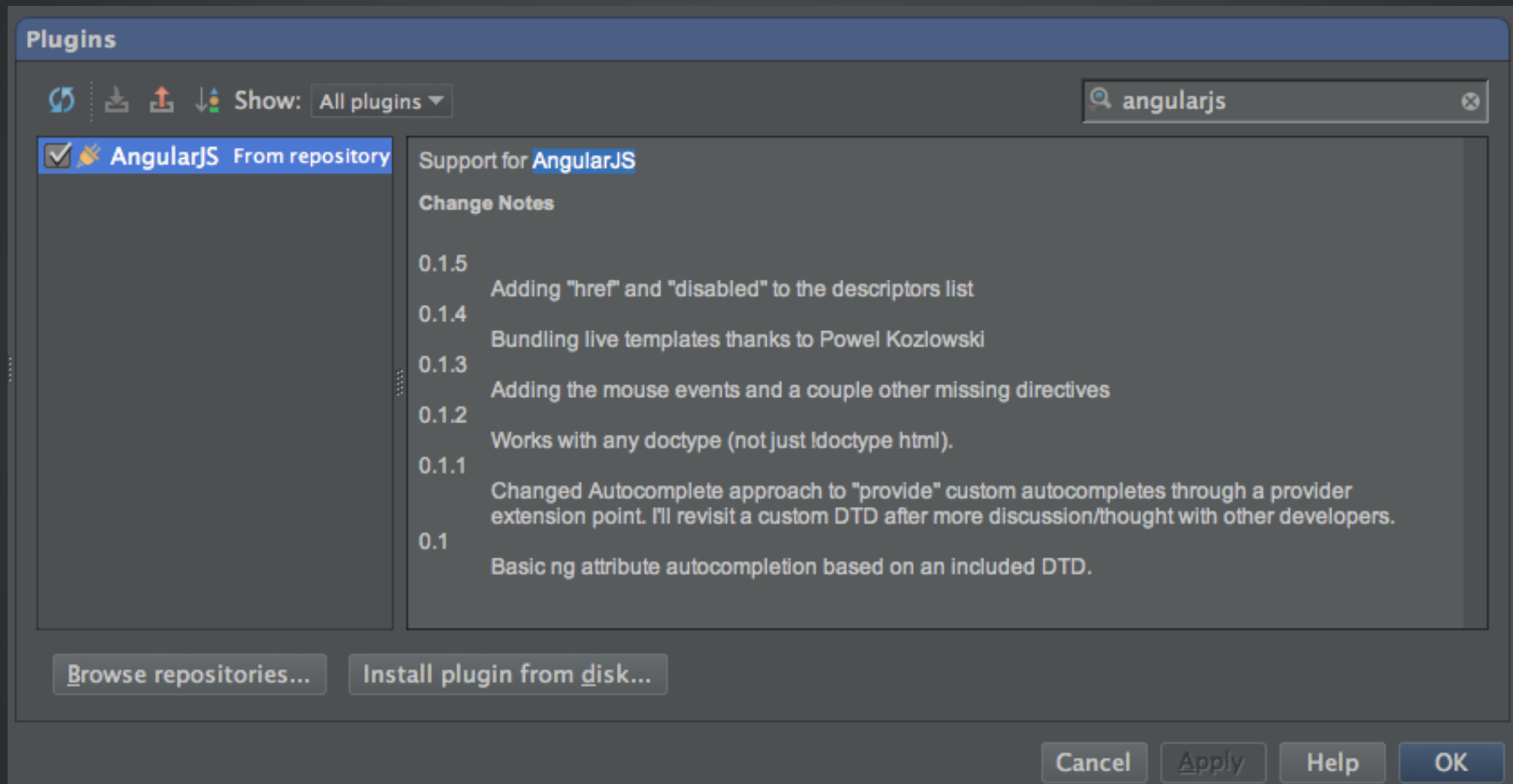
Control the whole workflow from the command line or your IDE - just save a file and Karma will run all the tests.



Testing Framework Agnostic

Describe your tests with Jasmine, Mocha, QUnit. Or write a simple adapter for any framework you like.

PLUGIN INTELLIJ / WEBSTORM



ANGULARUI

AngularUI ▾

Home

Directives ▾

Filters ▾

Contributors

Need Help? ▾

AngularUI

The companion suite to the **AngularJS** framework.



Code on Github



Download

Create a Build



Star 1,713



Fork 284



Tweet 325



+1 212

Check out the new

Bootstrap Project

by the AngularUI Team!

ANGULARSTRAP

AngularStrap

Bootstrap directives for **AngularJS**.

Star 897

Fork 113

Tweet 428

+1 342

[Modal](#)[Dropdown](#)[Navbar](#)[Tab](#)[Tooltip](#)[Popover](#)[Alert](#)[Buttons](#)

Modal directives/modal.js

Overview and examples

Fetches an external html partial (or an inline `ng-template`) and populates the modal with its content.

Use any button/link to trigger a modal by appending a `bs-modal` attribute.

Both `$scope.show()` and `$scope.dismiss()` are available inside the modal to toggle its visibility.

Example

Edit in plunker

RESSOURCES

- **Tutorial Officiel**
- **Chaîne Youtube AngularJS Officiel**
- **Year Of Moo**
- **AngularFR** , par Thierry Chatel
- **I.T Aware** , par Sébastien Letélié
- **egghead.io** Plein de vidéos réalisées par John Lindquist
- **Communauté Google+ française AngularJS**

POURQUOI ANGULARJS ?

- Beaucoup moins de code à écrire grâce au data-binding bi-directionnel
- Modularisation du code avec les modules et les services injectables : Pas de pollution de l'espace global
- AngularJS peut être actif sur une portion de la page HTML
- Créer ses propres composants réutilisable grâce aux directives
- Développement guidée
- Framework qui promeut les tests unitaires
- API bien documentée pour certains aspects
- **Excellent tutorial pour débiter**

DIFFICULTÉS

- Documentation officielle pauvre sur certaines notions
- Ecriture des directives pouvant être compliquée si on ne maîtrise pas son cycle de vie
 - Compilation
 - Link
 - Transclusion
 - Restriction
- Le data-binding bi-directionnel ne fonctionne pas hors du "monde" AngularJS. Il faut utiliser `$scope.$watch()` et `$scope.$apply()` pour surveiller et mettre à jour manuellement une valeur
- Intégration librairies UI jQuery pouvant être problématique
 - Je la wrappe dans une directive ?
 - ... ou je crée une directive ?

MERCI DE VOTRE ATTENTION !

Thierry Lau - [@laut3rry](#)

Sébastien Letélié - [@sebmade](#)

Thierry Chatel - [@ThierryChatel](#)