

Le Mans Université

Architectures Logicielles pour le Web

Licence Informatique - L3

Valérie Renault
prenom.nom@univ-lemans.fr

28 février 2018



1 - CMS : Content Management Systems

2 - Architectures logicielles pour le Web

CM3

Valérie Renault
prenom.nom@univ-lemans.fr

28 février 2018



- ▶ CMS : Content Management System ;
- ▶ SGC : Système de Gestion de Contenu ;

Définition

Un ou des programmes (web ou non) qui permettent à un ou des utilisateurs de **tous niveaux de compétences** de gérer rapidement, efficacement et simplement le **contenu**, le **graphisme** et les **modules dynamiques** d'un ou plusieurs sites Internet/Intranet.



- ▶ Éliminer le goulot d'étranglement de la production web ;
- ▶ Faciliter la production de contenu ;
- ▶ Gérer la qualité de l'information (via le workflow - étape de validation / modération) ;
- ▶ Organiser la production de contenu (via le workflow - rôles : rédacteurs, modérateurs, relecteurs) ;
- ▶ Permettre la multi-diffusion et la mutualisation ;



► Sites éditoriaux

- Magazine de voyage - <http://www.atacamag.com/> (WordPress) ;
- Quotidien de l'économie Toulousaine - <http://www.touleco.fr/> (SPIP) ;

► Communautés en ligne

- Facebook - <http://www.facebook.com> (Facebook platform) ;

► Bases de connaissances

- Wikipedia - <http://www.wikipedia.fr> (MediaWiki) ;

► Entreprises

- Inserm - <http://www.inserm.fr> (Zope/Plone) ;

► E-learning

- UMTice - <http://umtice.univ-lemans.fr> (Moodle) ;
- Univ. Paris-Dauphine - <http://www.dauphine.fr/fr/index.html> (Typo3) ;

► ...

Applications



ATACAMAG

VOLAGES SPORTS INTERVIEW NEWS MATÉRIELS OUTDOOR LES SEMMES MOUS

Tentes de randonnée, des conseils pour choisir

LE MAG DU LIBRE

PARTICIPER VOUS AUSSI À LA SEMME DUMAG

Bates

Academic Campus Admissions Alerts Parents

Admission: Mission Home, Society, Scholarship, News, Events, Faculty, Healthy Youth, School of the Week, Parents of Bates, Athletics

What's Happening: Events, Dining Info, Office of Institutional Effectiveness, Big Red Conference, Music & Art, Village Hall, Chapel Hill, Pleasanton, Clubs & Organizations

Office & Resources: Therapy & Wellness, Office of Institutional Effectiveness, Office of the President, Student Support Office, Housing Office, Student Employment, Career Development, Student Center, Student Center, Student Center

Faculty & Support: Security Office, Health Center, Student Center, Office Services, Faculty Services

Inserm

Institut national de la santé et de la recherche médicale

Qu'est-ce que l'Inserm ?

Neurosciences, sciences cognitives, neurologie, psychiatrie

Cancer

Microbiologie et maladies infectieuses

Circulation, métabolisme, nutrition

Immunologie, hématologie, pneumologie

Santé publique

Technologies pour la santé

Basics moléculaires et structurales du vivant

Actualités

La recherche

Ouverture de la campagne de recrutement

Concours chercheurs 2013 : 115 postes ouverts

Dossiers d'information

De A à Z, retrouvez nos dossiers d'information santé-recherche

Zoom

Glaucome

Les glaucomes constituent la seconde cause de cécité en France. Quels sont les facteurs de risque et les traitements ? Consultez notre dossier d'information



- ▶ Productivité de la fabrication des sites ;
- ▶ Qualité des informations ;
- ▶ Multi-diffusion, mutualisation ;
- ▶ Souplesse (séparation des gabarits et des contenus) ;
- ▶ Qualité perçue du travail à accomplir ;
- ▶ Evolutivité, potentialités (extensions) ;



► Manque de lisibilité de l'offre :

- <http://www.cmsmatrix.org> - site comparateur de CMS ;
- Compare environ 1200 produits différents ;
- Critères de comparaisons (10aines de points pour chaque critère) : système requis, sécurité, support technique, facilité d'utilisation, performances, management, interopérabilité, flexibilité, add-on existant, add-on commerciaux.

	Joomla! 2.5.4	Plone 4.0	SPIP 1.8.1	spip 2.0	TYPO3 Enterprise CMS 4.6	WordPress 3.3.2
Last Updated	5/2/2012	5/5/2011	8/1/2005	3/31/2010	10/25/2011	5/29/2012
System Requirements	Joomla! 2.5.4	Plone 4.0	SPIP 1.8.1	spip 2.0	TYPO3 Enterprise CMS 4.6	WordPress 3.3.2
Application Server	CGI	Zope	Apache	Apache	Apache	blank
Application Cost	Free	Free	Free	Free	Free	Free
Database	MySQL	Other	MySQL	MySQL	MySQL	MySQL
License	Open Source	Open Source	Open Source	Open Source	Open Source	Open Source
Operating System	Platform	Platform	Platform	Platform	Platform	Platform
Programming Language	PHP	Python	PHP	PHP	PHP	PHP
Root Access	No	No	Yes	No	No	No
Shell Access	No	Yes	Yes	No	No	No
Web Server	Any	Any	Apache	Apache	Apache	blank
Security	Joomla! 2.5.4	Plone 4.0	SPIP 1.8.1	spip 2.0	TYPO3 Enterprise CMS 4.6	WordPress 3.3.2
Anti-Fraud	No	Yes	No	Yes	Yes	Limited
Captcha	Free Add On	Free Add On	No	Free Add On	Free Add On	Free Add On
Content Approval	Yes	Yes	No	Free Add On	Yes	Yes
Email Verification	Yes	Yes	Yes	Yes	Yes	Yes
Guest User Privileges	Yes	Yes	Yes	Free Add On	Yes	Yes
Kerberos Authentication	No	Free Add On	No	No	Free Add On	No
LDAP Authentication	Yes	Yes	Yes	Yes	Free Add On	Free Add On
Login History	Yes	Free Add On	Yes	No	Yes	Free Add On
NTL Authentication	No	Free Add On	No	No	Free Add On	No
NTLM Authentication	No	Free Add On	No	No	Free Add On	No
Pluggable Authentication	Yes	Yes	Yes	Yes	Free Add On	Free Add On
Provisioning	No	Free Add On	No	No	Yes	Free Add On
Sanitization	No	Yes	No	Free Add On	Yes	Limited
Session Management	Yes	Free Add On	Yes	Limited	Yes	Free Add On



- ▶ Manque de lisibilité de l'offre ;
- ▶ Produits bogués et/ou inachevés ;
- ▶ Interfaces utilisateurs de qualité variable ;
- ▶ Coûts variables ;
- ▶ Puissance demandée ;
- ▶ Coûts d'adaptation et de développements complémentaires (extensions) ;



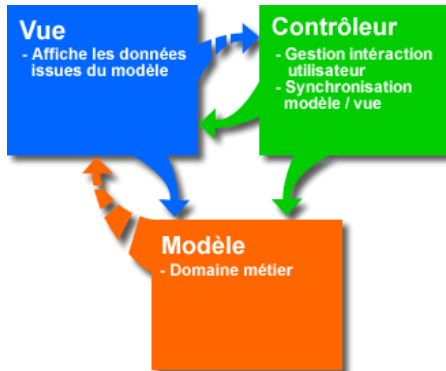
- ▶ Back-Office :
 - ▶ Administration ;
 - ▶ Gestion de contenu ;
 - ▶ Gestion des utilisateurs ;

- ▶ Front-Office :
 - ▶ Publication ;
 - ▶ Communication ;



- ▶ Outils d'extension : ajouts de plugin / module ; possibilité pour un développeur de créer ses propres plugins ;
- ▶ Outils de mise à jour : faire évoluer les versions du CMS ;
- ▶ Gestionnaire de sécurité et d'erreurs ;
- ▶ Séparation Vue et Document (modèle MVC) ;

Modèle Modèle-Vue-Contrôleur :





- ▶ Séparation fonds/forme ;
- ▶ Structure des articles et définition des types d'articles ;
- ▶ Organisation des contenus ;
- ▶ La gestion des contributeurs ;
- ▶ Workflow et chaine de validation ;
- ▶ Cycle de vie des articles ;
- ▶ Gestion des versions ;



- ▶ Gestion des utilisateurs et de leur droits ;
- ▶ Contrôle d'accès ;
- ▶ Mémorisation de certaines actions utilisateurs (date et durée de connexion, dépôt de fichiers, pages consultées, ...) ;
- ▶ Statistique sur les utilisateurs ;



- ▶ Gabarits/Templates de présentation ;
- ▶ Sélection de contenu ;
- ▶ Moteurs de recherche : moteur interne, référencement par des moteurs classiques ;
- ▶ Syndication : échanges de contenus entre des sites (flux RSS) ;
- ▶ Statistiques du site ;

Qu'est-ce que le Workflow ?



- ▶ La **gestion de contenu** (Content Management) définit les systèmes permettant de collecter, centraliser, conserver et exploiter les données sous forme électronique.
- ▶ Les services de Content Management prennent en charge la totalité du **cycle de vie** de l'information, de sa création jusqu'à sa diffusion, en passant par la validation, la mise en forme et la mise à jour.

Qu'est-ce que le Workflow ?



Workflow : " **circuit de navigation d'un document**".

Par exemple :

- ▶ le **workflow de validation** : suite d'actions comprenant la rédaction d'un texte, sa validation, sa mise en ligne, les avertissements aux différents rédacteurs et validateurs, ... = "politique éditoriale" du site.
- ▶ le **workflow de syndication** tel que celui d'un flux RSS : suite d'actions comprenant la rédaction d'un texte, sa mise à disposition, sa réutilisation dans un site distant, ...



- ▶ UMTICE
- ▶ WordPress
- ▶ Prestashop
- ▶ Hébergement : OVH, Gandi, ...



- ▶ Choisir un CMS n'est pas une chose triviale : il n'y a pas un CMS mieux que les autres ;
- ▶ Beaucoup d'éléments à prendre en compte : L'usage visé, Les usagers visés, Les coûts nécessaires vs possibles ;



Un framework pour le web, c'est quoi ?

Définition et caractéristiques (Wikipedia)

- ▶ modéliser l'architecture des logiciels applicatifs, des applications web, des middlewares et des composants logiciels ;
- ▶ ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture et des patterns, l'ensemble formant ou promouvant un « squelette » de programme.
- ▶ c'est souvent bibliothèque logicielle et accompagnée du plan de l'architecture cible du framework

Framework de Single Page Application - SPA

- ▶ 1 page web Web
- ▶ application



- ▶ Ecrire du JavaScript côté serveur, au même titre que PHP et Java ;
- ▶ Interagir avec les ressources de la machine serveur ;



- ▶ Création d'un serveur qui aura le rôle de "servir" les fichiers JavaScript, HTML, CSS, ... ;
- ▶ Installation : [https ://nodejs.org/en/](https://nodejs.org/en/) ;
- ▶ JavaScript runtime built on Chrome's V8 JavaScript engine ;
- ▶ Modèle "event-driven" ;
- ▶ Modèle "non-blocking I/O" ;
- ▶ Package 'npm' de librairies opensource ;
- ▶ npm : gestionnaire de paquets : gère automatiquement les dépendances de vos paquets ;
- ▶ Pour Angular : installation de bower + yo
- ▶ Pour Angular : installation de générateur d'applications Angular generator-angular



- ▶ Framework écrit en Java Script par Google en 2009 ;
- ▶ Libre et open-source ;
- ▶ Améliorer la syntaxe de JavaScript et la productivité du développeur ;
- ▶ Utilise JQuery : bibliothèque JavaScript qui permet de créer des effets dynamiques sur vos pages web comme des changements de couleur, des animations, et des effets de fondu. ;



- ▶ Angular 1 et 2 : rupture de compatibilité entre Angular 1 et 2, nouveaux modèles, réécriture totale du framework ;
- ▶ Angular 4.0 - mars 2017
- ▶ Angular 5.0 - novembre 2017 : création d'applications Web progressives (PWA)
- ▶ Google prévoit une version tous les 6 mois...
- ▶ <https://angularjs.org/>

Application Web Progressives : collaboration Google et Mozilla, avoir des applications web sur mobile de qualités identiques voir meilleures d'une application native.

Exemple d'applications : Gmail, Maps, Calendar, ...



Caractéristiques :

- ▶ Basé sur la **architecture MVC** (modèle vue controller) :
 - ▶ Modèle : données métiers ;
 - ▶ Vue : présentation des données - vue de l'utilisateur ;
 - ▶ Contrôleur : actions à effectuer sur ces données ;
- ▶ **Data Binding** ;
- ▶ **Injection de dépendances** ;
- ▶ **Template côté client** ;
- ▶ **Manipulation du DOM** au moyen de directives ;



AngularJS repose sur une approche tests :

- ▶ Conserver un code maintenable, compréhensible ;
- ▶ Fournit une suite de tests à faire avant de mettre le produit en production ;
- ▶ Applications Web : Projets plus ambitieux qu'un simple site Web ;
- ▶ SPA : Single-Page Application : dédiée à une application web à une seule page ;



3 fichiers de configuration

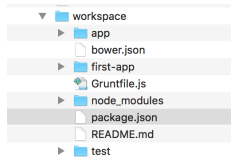
- ▶ package.json : fichier de configuration utilisé par le npm - le nom de l'application et ses dépendances
- ▶ bower.json : fichier de configuration pour les dépendances purement front-end
- ▶ Gruntfile.js : le "Task Runner" - fournir au développeur le moyen d'automatiser les tâches répétitives comme la minification, la compilation, l'exécution des tests unitaires

Structure d'un projet - package.json

Angular JS



27

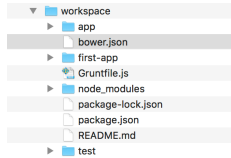


- ▶ nom + version ;
- ▶ Fichier de configuration utilisé par npm ;
- ▶ Dependences : de fonctionnement et de développement ;
- ▶ 'engines' : version du 'node' (JS) nécessaire à l'application ;

```
package.json
{
  "name": "workspace",
  "private": true,
  "devDependencies": {
    "autoprefixer-core": "^5.2.1",
    "grunt": "^0.4.5",
    "grunt-angular-templates": "^0.5.7",
    "grunt-concurrent": "^1.0.0",
    "grunt-contrib-clean": "^0.6.0",
    "grunt-contrib-concat": "^0.5.0",
    "grunt-contrib-connect": "^0.9.0",
    "grunt-contrib-copy": "^0.7.0",
    "grunt-contrib-cssmin": "^1.0.2",
    "grunt-contrib-htmlmin": "^0.4.0",
    "grunt-contrib-imagemin": "^1.0.0",
    "grunt-contrib-jshint": "^0.11.0",
    "grunt-contrib-uglify": "^0.7.0",
    "grunt-contrib-watch": "^0.6.1",
    "grunt-filerev": "^2.1.2",
    "grunt-google-cdn": "^0.4.3",
    "grunt-jscs": "^1.8.0",
    "grunt-newer": "^1.1.0",
    "grunt-ng-annotate": "^0.9.2",
    "grunt-postcss": "^0.5.5",
    "grunt-svgmin": "^2.0.0",
    "grunt-usemin": "^3.0.0",
    "grunt-wiredep": "^2.0.0",
    "jasmine-core": "^3.0.0",
    "jit-grunt": "^0.9.1",
    "jshint-stylish": "^1.0.0",
    "karma": "^2.0.0",
    "karma-jasmine": "^1.1.1",
    "karma-phantomjs-launcher": "^1.0.4",
    "phantomjs-prebuilt": "^2.1.16",
    "time-grunt": "^1.0.0"
  },
  "engines": {
    "node": ">=0.10.0"
  },
  "scripts": {
    "test": "karma start test/karma.conf.js"
  }
}
```

Structure d'un projet - bower.json

Angular JS



- ▶ nom + version ;
- ▶ Dependences : du front-end ;
- ▶ Version d'angular ;

```
bower.json
{
  "name": "workspace",
  "version": "0.0.0",
  "dependencies": {
    "angular": "^1.4.0"
  },
  "devDependencies": {
    "angular-mocks": "^1.4.0"
  },
  "appPath": "app",
  "moduleName": "workspaceApp"
}
```



Fournir au développeur le moyen d'automatiser les tâches répétitives comme :

- ▶ la minification ;
- ▶ la compilation ;
- ▶ l'exécution des tests unitaires.

Au lancement de la commande "grunt" :

- ▶ tâches de vérification de code (exple : jshint) ;
- ▶ tâche de test ;
- ▶ tâche de build (exple : cssmin - minification) ;

```
grunt.registerTask('test', [  
  'clean:server',  
  'wiredep',  
  'concurrent:test',  
  'postcss',  
  'connect:test',  
  'karma'  
]);  
  
grunt.registerTask('build', [  
  'clean:dist',  
  'wiredep',  
  'useminPrepare',  
  'concurrent:dist',  
  'postcss',  
  'ngtemplates',  
  'concat',  
  'ngAnnotate',  
  'copy:dist',  
  'cdnify',  
  'cssmin',  
  'uglify',  
  'filerev',  
  'usemin',  
  'htmlmin'  
]);  
  
grunt.registerTask('default', [  
  'newer:jshint',  
  'newer:jscs',  
  'test',  
  'build'  
]);  
};
```



Lancement du serveur en ligne de commande : grunt serve

Côté terminal :

```
Gryffindor:first-app vrenault$ grunt serve --force
Running "serve" task

Running "clean:server" (clean) task
>> 0 paths cleaned.

Running "wiredep:app" (wiredep) task
Warning: Error: Cannot find where you keep your Bower packages. Used --force, continuing.

Running "wiredep:test" (wiredep) task
Warning: Error: Cannot find where you keep your Bower packages. Used --force, continuing.

Running "concurrent:server" (concurrent) task

  Running "copy:styles" (copy) task
  Copied 1 file

  Done, without errors.

Execution Time (2018-02-20 15:26:41 UTC+1)
loading tasks          95ms ██████████ 79%
loading grunt-contrib-copy 12ms █████ 18%
copy:styles           14ms █████ 12%
Total 121ms

Running "postcss:server" (postcss) task
>> 1 processed stylesheet created.

Running "connect:livereload" (connect) task
Started connect web server on http://localhost:9000

Running "watch" task
Waiting...
[]
```

Côté navigateur :

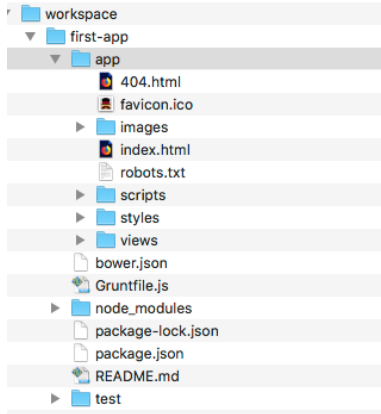
Toggle navigation [firstApp](#)

- [Home](#)
- [About](#)
- [Contact](#)

from the Yeoman team

Structure d'un projet - Organisation de l'arborescence

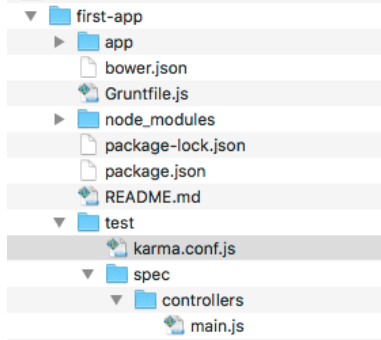
Angular JS



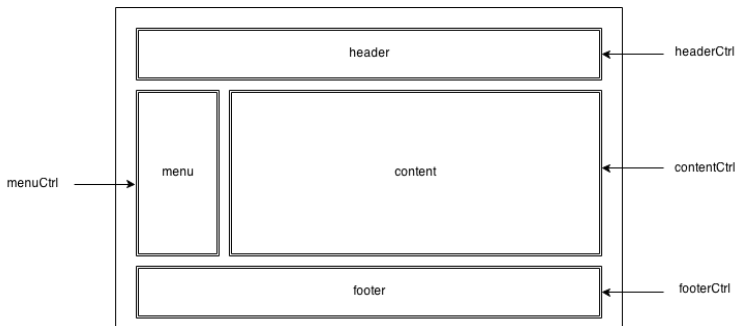
- ▶ images : toutes les images du projet ;
- ▶ scripts : le coeur de l'application avec les contrôleurs, les services et les directives Angular ;
- ▶ styles : l'ensemble des feuilles de styles ;
- ▶ views : les templates HTML
- ▶ index.html : le point d'entrée de l'application - une page HTML



Le répertoire test :



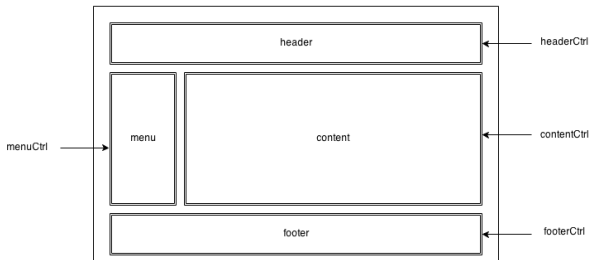
- ▶ karma.conf.js : "Test Runner"
- automatisation de l'exécution des tests ;
- ▶ spec : les tests du contrôleur principal ;



- ▶ Chaque composant de la page va posséder son propre contrôleur ;
- ▶ But : Disposer de plusieurs petits contrôleurs indépendants ;



Directive : ng-controller



Côté HTML :

```
<section ng-controller="headerCtrl">
  <h1>HEADER</h1>
</section>
```

Côté JavaScript :

```
angular.module('firstAppApp')
  .controller("headerCtrl", function () {
    //instructions
  });
```

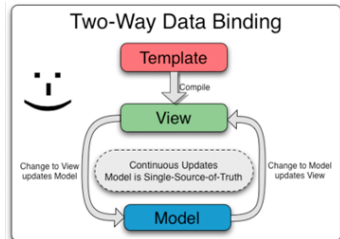
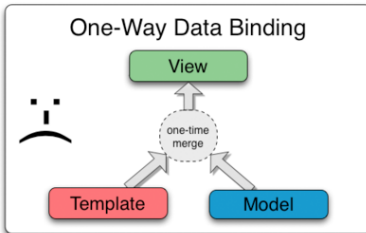


Frameworks MVVM (Modèle-Vue-ViewModèle)

on ne modifie pas le DOM (Cf cours de JavaScript) mais on passe par les bindings.

Principe permettant de lier la partie vue à la partie logique

- ▶ Présent dans la plupart des frameworks pour créer des applications web ;
- ▶ Two-way data binding / liaison de données à double-sens :
 - ▶ les actions sur la page HTML mettent à jour les variables JavaScript ;
 - ▶ quand les variables changent dans le JavaScript, la vue dans HTML se met à jour immédiatement





HTML ▼

```
1 <div ng-app="myApp">
2   <div ng-controller="exemple1Ctrl">
3     <input ng-model="age"/>
4     <span>Vous êtes <b ng-bind="majeurOrMineurText()"></b></span>
5   </div>
```

CSS ▼

```
1
```

JavaScript + AngularJS 1.2.1 ▼

```
1 var myApp = angular.module('myApp', []);
2
3 myApp.controller("exemple1Ctrl", function($scope){
4   $scope.age = 0;
5   $scope.majeurOrMineurText = function(){
6     return ($scope.age >= 18) ? "majeur" : "mineur";
7   };
8 });
```

19

Vous êtes **majeur**

HTML ▼

```
1 <div ng-app="app">
2   <div ng-controller="myController">
3     <label>Afficher le paragraphe</label>
4     <input type="checkbox" ng-model="showContent"/>
5     <p ng-show="showContent">Lorem ipsum dolor sit amet, consectetur
adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore
magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla
pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa
qui officia deserunt mollit anim id est laborum.</p>
6   </div>
7 </div>
```

JavaScript + AngularJS 1.4.8 ▼

```
1 var app = angular.module("app", []);
2
3 app.controller("myController", function($scope){
4   $scope.showContent = true;
5 });
```

CSS ▼

```
1 p{
2   font-size:12px;
3 }
```

Afficher le paragraphe ☒

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



HTML ▼

```
1 <div ng-app="myApp">
2   <div ng-controller="exemple1Ctrl">
3     <input ng-model="age"/>
4     <span>Vous êtes <b ng-bind="majeurOrMineurText()"></b></span>
5 </div>
```

directives : permettent de faire le "binding" - précédées de "ng"

- ▶ **ng-model** : directive permettant de lier la valeur d'un champ input à une variable
- ▶ **ng-bind** : directive liant une fonction retournant une chaîne de caractères
- ▶ **ng-show** : directive prenant un booléen en paramètre et affiche le contenu si le booléen est vrai, sinon le contenu est caché



- ▶ Permettre à des modules de ne pas se soucier de l'instanciation des modules dont ils dépendent ;
- ▶ Il suffit d'appeler les dépendances, et Angular se charge de les instancier ;
- ▶ Concept en Angular JS, Symfony2 ; etc.
- ▶ `$scope` : une dépendance/module native du langage ;
- ▶ Les dépendances natives commencent par `$` ;
- ▶ Créer des petits modules unitaires pour les assembler ensuite ;
- ▶ Permettre de réutiliser des modules (notamment des services) ;



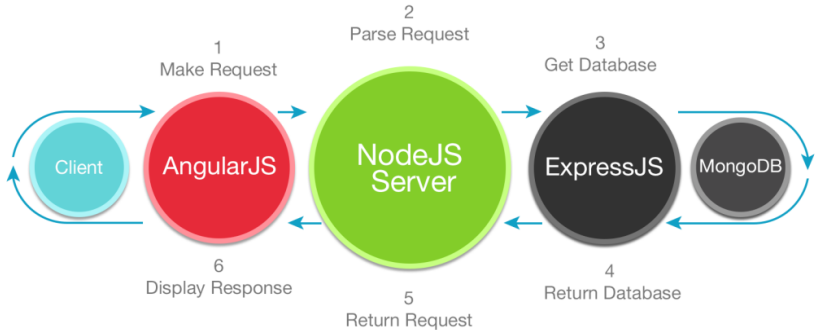
```
1 var myApp = angular.module('myApp', []);
2
3 myApp.controller("exemple1Ctrl", function($scope){
4     $scope.age = 0;
5     $scope.majeurOrMineurText = function(){
6         return ($scope.age >= 18) ? "majeur" : "mineur";
7     };
8 });
```

- ▶ Dépendance du contrôleur
- ▶ Toute donnée qui n'est pas attachée à \$scope n'appartient pas au modèle et ne peut donc pas être exposée à la vue !
- ▶ Le \$scope : le contexte courant. Chaque contrôleur possède son propre objet Scope

Synthèse : MEAN



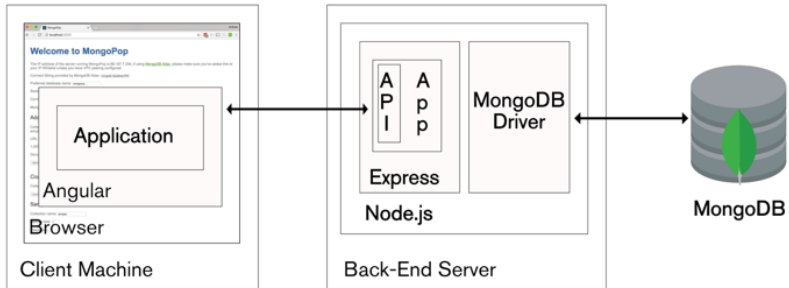
MEAN = Mongo, Express, Angular, Node



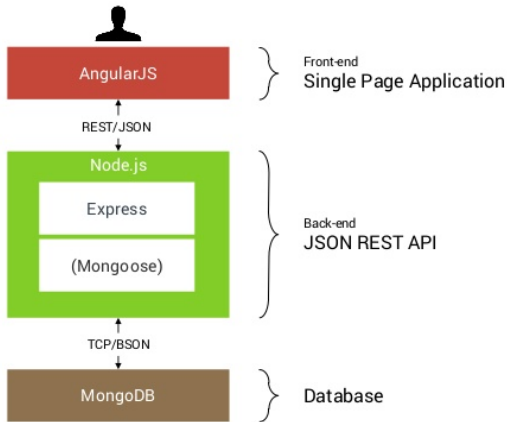
Synthèse : MEAN



43



Mean is an architecture



Synthèse : MEAN

