

AngularJS

Présentation du framework AngularJS

AngularJS, qu'est-ce que c'est ?

AngularJS est un framework JavaScript, créé en octobre 2010 par des développeurs de chez Google et sous la licence MIT.

Il propose une architecture basée sur le pattern MVC (Model View Controller) pour des SPA (Single Page Application), autrement dit des applications web où la navigation se fait sur une même et unique page (exemple : Gmail, Dropbox, iCloud, etc.).

Il a été créé avec la philosophie suivante :

Le HTML est très bien pensé pour les documents statiques mais il n'est pas du tout adapté pour un usage dynamique tel qu'on le voit dans les applications web. AngularJS enrichie le HTML avec une grammaire similaire, spécialement conçue pour des RIA (Rich Internet Application). Le résultat est un code parfaitement expressif (on comprend chaque action effectuée à sa simple lecture), lisible (ça reste du HTML) et facile à écrire.

La principale caractéristique de ces frameworks est qu'un grand nombre d'actions effectuées sur le serveur tel que le rendu du moteur de template, la récupération des données, leur (pré) validation et la navigation dans une application, sont désormais déportés côté client.

Le serveur se limite à traiter, vérifier, valider et envoyer les données aux clients dans un format générique (JSON, XML, etc.)

Cela permet d'avoir une charge sur les serveurs nettement moins importante et une fluidité de navigation chez le client.

Gestion de la vue

AngularJS se différencie de ses concurrents par une gestion facultative du DOM (Document Object Model) de l'application. Contrairement à Backbone.js et d'autres frameworks, AngularJS permet de développer une RIA sans devoir utiliser des sélecteurs d'éléments HTML afin de rendre une application interactive.

Syntaxe

La syntaxe des templates de la vue se résume à du HTML, avec des attributs spécifiques (ng-*) et un affichage de variables à l'aide d'accolades

Ex : `<h1>The sum of 2+3 = {{2 + 3}}!</h1>`

Two-way data binding

Une de ses fonctionnalités phares est le two-way data binding qui permet de notifier tous les éléments qui font référence à une variable, de son changement. Ainsi sans coder explicitement du JavaScript, on peut mettre à jour en temps réel un titre qui fait référence à un champ texte.

Ex : `<input type="text" ng-model="yourName"><h1>Hello {{yourName}}!</h1>`

Ng-*

AngularJS propose une liste d'attributs propres aux éléments HTML comme ngdisabled pour les inputs, ng-multiple pour les selects et ng-click, ng-class, ngstyle pour tous les éléments (ng-* est un préfixe permettant de différencier les vrais attributs HTML de ceux d'AngularJS) permettant d'effectuer une action

tel qu'appeler une fonction, assigner une valeur à une variable, choisir une classe suite à un évènement (clic, survol, envoi de formulaire) ou à un changement de variable.

L'attribut `ng-repeat` permet l'affichage d'éléments d'un tableau d'une manière très simple, en l'affectant à un template d'une balise HTML qui représente un élément du tableau. (tab représente un tableau et el un de ses éléments)

Ex : `<li ng-repeat="friend in friends">`

Les filtres

Actuellement, lors d'un affichage spécifique d'une donnée dans une page, on effectue un traitement de la variable à afficher à l'aide d'une fonction dont on insert le résultat dans une variable intermédiaire et enfin, on affecte le `innerHTML` d'un élément.

AngularJS permet d'une manière plus aisée ce type d'affichage à l'aide du caractère « `|` » suivi de la fonction qui transforme la donnée.

Ex : `<p>default currency symbol ($): {{amount | currency}}</p>`

AngularJS intègre déjà plusieurs filtres (date, monnaie, nombre, caractères minuscules) mais aussi des filtres d'affichage (nombre maximum d'éléments, recherche, tri).

Ex : `<tr ng-repeat="friend in friends | limitTo: 10">`

La partie logique

Les services

Les services sont des fonctions qui exécutent des tâches communes aux contrôleurs. Ils permettent d'éviter une redondance du code. Ce sont des singletons. AngularJS en propose plusieurs par défaut (routing, i18n, ajax, scope, filter, etc.).

Les contrôleurs

Les contrôleurs sont tout simplement des fonctions qui acceptent en paramètre des services prédéfinis comme `$scope` (qui permet d'avoir accès aux propriétés et méthodes de la vue) ou `$route` (qui permet d'avoir accès au moteur de gestion de navigation entre pages). Il peut être défini pour un élément de la vue à l'aide de l'attribut `ng-controller`.

Le routing

Le système de gestion de la navigation d'une RIA s'appelle le « routing ». Dans AngularJS, on peut paramétrer le choix du contrôleur et le template choisi pour la vue (valeur en dur ou page externe) pour une route donnée.

Quels sont ses inconvénients ?

Malgré sa facilité d'utilisation, la syntaxe utilisée et l'emplacement choisi (directement dans la page HTML) suscitent une certaine repousse des développeurs qui sont habitués à utiliser des moteurs de templates plus classiques tel que EJS (syntaxe similaire à Smarty).

De plus, une bonne partie des librairies permettent d'ajouter des événements, modifier des éléments HTML avec les méthodes basiques JavaScript, là où AngularJS nous pousse à utiliser ses méthodes qui sont assez différentes.

Il ne propose pas d'interface graphique comme ses concurrents Dojo et ExtJS (framework permettant de coder entièrement une SPA sans utiliser du HTML).

Sa jeunesse et le manque d'entreprises l'utilisant en production lui font défaut contrairement à Backbone.js qui a acquis à sa cause LinkedIn et Foursquare.

Pourquoi choisir ce framework ?

AngularJS arrive à se démarquer de ses concurrents malgré la forte concurrence qu'il existe entre les différents frameworks JavaScript.

Tout d'abord, le projet est développé par des développeurs de chez Google et il est sponsorisé par leur entreprise, ce qui n'est pas négligeable pour la pérennité du projet.

Comme cités précédemment, ses fonctionnalités et leur mise en place font énormément gagner du temps aux développeurs.

AngularJS propose aussi les modules qui permettent de découper son SPA en parties distinctes comme les namespaces et aussi les directives qui permettent de redéfinir et ainsi étendre leurs possibilités (ex : `<calendar>` remplacerait un widget complet en HTML, ce qui simplifie grandement la lisibilité du code).

De plus, AngularJS est utilisé en production sur l'application Youtube de la PlayStation 3 ainsi que dans la nouvelle version de DoubleClick (régie publicitaire appartenant à Google, destinée aux grands comptes).