



## Services

Les services AngularJS permettent de partager du code dans une application, via l'injection de dépendances.

AngularJS propose un grand nombre de services (et providers) prêts à l'emploi. Ils sont précédés du symbole \$ (\$scope, \$http,\$timeout...).

Les services sont :

- Lazily instantiated, c'est-à-dire instanciés uniquement si on en a besoin
- Des singletons, c'est-à-dire qu'ils ne sont instanciés qu'une seule fois

Exemples d'utilisation de services :

- Récupération de données via Ajax (au lieu de le faire un peu partout dans les contrôleurs)
- Regroupement de fonctions « utilities » utilisées dans une application
- Définition de paramètres (configuration) d'une application

### -- Quelques services AngularJs

Nom	Rôle
<b>\$animate</b>	permet d'accéder à des fonctions de manipulation des éléments du DOM (ajout, modification, insertion, déplacement...)
<b>ngCookies \$cookies</b>	gestion des cookies
<b>\$document</b>	wrapper vers window.document
<b>\$http</b>	fourni des méthodes permettant d'effectuer des requêtes ajax vers un serveur (get, post, put, delete...)
<b>\$interval</b>	wrapper pour la fonction setInterval de js
<b>\$q</b>	permet d'agir sur l'exécution de fonctions asynchrones et de récupérer leur valeur de retour en fin d'exécution
<b>\$rootScope</b>	Retourne le root scope de l'application
<b>\$timeout</b>	wrapper pour la fonction setTimeout de js
<b>\$window</b>	wrapper vers window

### -- Injection d'un service

Exemple d'injection du service **\$window** dans un contrôleur pour permettre une redirection de la page vers une autre URL.

voir [injection\\_de\\_dependance\\_dependency\\_injection](#)

```
angular.module('myApp').controller('GotoController', [
  '$scope', '$window', function($scope, win) {
    $scope.location="http://www.google.fr";
    $scope.go=function(){win.location=$scope.location;};
  }
]);
```

```
} ]);
```

## -- \$http service

Le service \$http (ng module) permet de faire des requêtes HTTP vers un serveur, via l'objet [XmlHttpRequest](#) du navigateur ou via [JSONP](#).

### -- Usage classique

```
// Exemple de get :
$http.get('/someUrl').
  success(function(data, status, headers, config) {
    // Fonction de rappel appelée de manière asynchrone
    // quand la réponse est reçue
  }).
  error(function(data, status, headers, config) {
    // Appelé de manière asynchrone en cas d'erreur
    // ou si le serveur retourne une réponse avec un statut d'erreur.
  });
```

```
// Exemple de post avec passage de données :
$http.post('/someUrl', {msg:'hello word!'}).
  success(function(data, status, headers, config) {
    // Fonction de rappel appelée de manière asynchrone
    // quand la réponse est reçue
  }).
  error(function(data, status, headers, config) {
    // Appelé de manière asynchrone en cas d'erreur
    // ou si le serveur retourne une réponse avec un statut d'erreur.
  });
```

### -- Raccourcis

- [\\$http.get](#)
- [\\$http.head](#)
- [\\$http.post](#)
- [\\$http.put](#)
- [\\$http.delete](#)
- [\\$http.jsonp](#)
- [\\$http.patch](#)

```
$http.get('/someUrl').success(successCallback);
$http.post('/someUrl', data).success(successCallback);
```

## -- \$q

Le service **\$q** (ng module) permet d'exécuter des fonctions de manière asynchrone et utilise leurs valeurs de retour ou d'erreur quand l'exécution est terminée.

From:

<http://slamwiki2.kobject.net/> - **Broken SlamWiki 2.0**

Permanent link:

<http://slamwiki2.kobject.net/slam4/richclient/angularjs/services?rev=1422066214>

Last update: **2019/08/31 14:40**

