

Angular

ETENDRE L'HTML

Directive

Présentation

Permet d'enrichir le HTML

Permet d'attacher des évènements aux éléments

Permet de modifier le DOM

Les formes

Une directive peut prendre 4 formes

Élément (E) : <hello></hello>

Attribut (A) : <p hello></p>

Class (C) : <p class="hello"></p>

Commentaire (M) : <!-- directive:hello -->

Créer une directive

Utiliser la fonction `.directive(nom, fonction)` sur un module

Doit retourner un objet

```
var app = angular.module('app', []);  
  
app.directive('hello', function () {  
  return {  
    restrict: 'AE',  
    replace: 'true',  
    template: '<h1>Hello</h1>'  
  };  
});
```

Est-ce que la directive doit être remplacée

Comment la directive doit être utilisée

Template de remplacement

Scope

Par défaut une directive ne possède pas de scope enfant

Elle hérite du scope de son parent

Pour profiter du scope, il faut utiliser la fonction *link*

Exemple

jQuery a été inclut dans le fichier index.html pour cet exemple

```
app.directive('hello', function () {
  return {
    restrict: 'AE',
    replace: 'true',
    template: '<div><h1>Hello</h1><p>Couleur : {{ color }}</p></div>',
    link: function (scope, elem, attrs) {
      elem.on('click', 'h1', function () {
        $(this).css('color', scope.color);
      });
    }
  };
});

<input type="text" ng-model="color" placeholder="Entrer une couleur" />
<hello></hello>
```

Autres propriétés

Il est possible de lier un fichier contenant le template plutôt qu'une string

- Utiliser : `templateUrl : 'nomfichier.html'`

Il est possible de changer le scope de la directive

- Pour créer un scope isolé, utiliser : `scope : {}`

Il est possible de déplacer du contenu

- Utiliser : `transclude : true`

Pour voir toutes les options, la [documentation des directives](#)

Composant

Présentation

Permet de créer un contenu réutilisable

Possède un scope isolé (qui n'a pas de lien avec le monde extérieur)

Créer un composant

Utiliser la fonction `.component('nom', objet)` sur un module

Sucre syntaxique d'une directive

```
app.component('hello', {  
  template: '<input type="text" ng-model="$ctrl.user" />Bonjour, {{$ctrl.user}}!',  
  controller: function() {  
    this.user = 'John Doe';  
  }  
});
```

```
<hello></hello>
```

Best practices

Communauté

Il existe un [guide de style](#) fait par la communauté Angular

Explique les pour/contre de chaque écriture