

APPMOB - AngularJS Scopes

Olivier Liechti & Simon Oulevay
COMEM Applications Mobiles

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

What is a Scope?

The **scope** is the glue between application controller and the view.

Every time you use a directive like **ng-app**, **ng-controller** or **ng-repeat**, a new scope is created.

```
<div ng-controller="GreetCtrl">  
  Hello {{ name }}!  
</div>
```

This first scope is **limited** to this **div** tag.

```
.controller("GreetCtrl", function($scope) {  
  $scope.name = "World";  
})
```

```
<div ng-controller="ListCtrl">  
  <ol>  
    <li ng-repeat="name in names">  
      {{ name }}  
    </li>  
  </ol>  
</div>
```

```
.controller("ListCtrl", function($scope) {  
  $scope.names = ["Igor", "Misko", "Vojta"];  
})
```

With this **ng-repeat**, Angular will automatically create 3 new scopes, **one for each line**. Each of these scopes will have the **name** property set to the next name in the list.

Scope Hierarchy

Scopes form a hierarchy. Each new scope is added as a **child scope** of their **parent scope**.

```
<!DOCTYPE html>
<html ng-app class="ng-scope">
  <head>...</head>
  <body>
    <div>
      <div ng-controller="GreetCtrl" class="ng-scope ng-binding">
        Hello World!
      </div>
      <div ng-controller="ListCtrl" class="ng-scope">
        <ol>
          <!-- ngRepeat: name in names -->
          <li ng-repeat="name in names" class="ng-scope ng-binding">
            Igor
          </li>
          <li ng-repeat="name in names" class="ng-scope ng-binding">
            Misko
          </li>
          <li ng-repeat="name in names" class="ng-scope ng-binding">
            Vojta
          </li>
        </ol>
      </div>
    </div>
  </body>
</html>
```



The diagram illustrates the scope hierarchy for the provided HTML code. It shows a root scope (ng-scope) containing two child scopes. The first child scope (GreetCtrl) contains the text 'Hello World!'. The second child scope (ListCtrl) contains an ordered list of three items: 'Igor', 'Misko', and 'Vojta'. Each item is contained within its own scope (ng-scope ng-binding). The hierarchy is visualized by nested boxes and a tree view on the right side of the code editor.

Scope	Controller	Content
Root Scope	ng-scope	<html> <head>...</head> <body> <div>
Child Scope 1	GreetCtrl	Hello World!
Child Scope 2	ListCtrl	 <!-- ngRepeat: name in names -->
Child Scope 3	ng-scope ng-binding	Igor
Child Scope 4	ng-scope ng-binding	Misko
Child Scope 5	ng-scope ng-binding	Vojta

Scope Prototypical Inheritance

heig-vd

Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Child scopes prototypically inherit from their parent scope.

The **ng-app** directive creates the special **\$rootScope**, the top-level scope of your whole application.

```
<html ng-app="scope-demo">
  <head>...</head>
  <body>
    <div ng-controller="GreetCtrl">
      Hello {{ name }}!
    </div>
```

```
angular.module("scope-demo", [])
```

You can inject the **\$rootScope** in any controller or service.

```
.controller("GreetCtrl", function($scope, $rootScope) {
  $scope.name = "World";
  $rootScope.department = "Angular";
})
```

Anything you add on the **\$rootScope** is available on all scopes.

```
<div ng-controller="ListCtrl">
  <ol>
    <li ng-repeat="name in names">
      {{ name }} from {{ department }}
    </li>
  </ol>
</div>
</body>
</html>
```

```
.controller("ListCtrl", function($log, $scope) {
  $scope.names = ["Igor", "Misko", "Vojta"];
  $log.debug($scope.department); // "Angular"
});
```

The **department** property was not defined on the **ng-repeat scope**. It is available because it is **prototypically inherited** through the scope created by **ListCtrl** and the **\$rootScope**.

Scopes can propagate **events** in similar fashion to DOM events. Events can be **broadcasted to the scope children** or **emitted to scope parents**.

```
.controller("AnyCtrl", function($scope) {  
    // Emit an event to all parent scopes.  
    $scope.$emit("somethingHappened", { data: "foo" });  
  
    // Broadcast an event to all child scopes.  
    $scope.$broadcast("somethingElseHappened", 1, 2, 3);  
  
    // Listen for an event.  
    $scope.$on("eventName", function(event, arg1, arg2) {  
        // The callback function will be called with an event object  
        // and the arguments passed to $emit or $broadcast.  
    });  
})
```

<http://codepen.io/AlphaHydrae/pen/emQPzY/>

More Information

<https://docs.angularjs.org/guide/scope/>

[https://github.com/angular/angular.js/wiki/
Understanding-Scopes/](https://github.com/angular/angular.js/wiki/Understanding-Scopes/)