

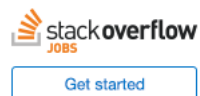
Join the Stack Overflow Community

Stack Overflow is a community of 6.9 million programmers, just like you, helping each other.
Join them; it only takes a minute:

[Sign up](#)

how to use ng-option to set default value of select element

```
36 if (dev.isBored() || job.sucks()) {
37   searchJobs({flexibleHours: true, companyCulture: 100});
38 }
39 // A career site that's by developers, for developers.
```



I've seen the documentation of the Angular select directive here: <http://docs.angularjs.org/api/ng.directive:select>. I can't figure how to set the default value. This is confusing:

select as label for value in array

Here is the object:

```
{
  "type": "select",
  "name": "Service",
  "value": "Service 3",
  "values": [ "Service 1", "Service 2", "Service 3", "Service 4" ]
}
```

The html (working):

```
<select><option ng-repeat="value in prop.values">{{value}}</option></select>
```

and then I'm trying to add an ng-option attribute inside the select element to set `prop.value` as the default option (not working).

```
ng-options="(prop.value) for v in prop.values"
```

What am i doing wrong?

angularjs

edited Sep 19 '16 at 13:10

asked Jun 26 '13 at 20:08

[francoisromain](#)
2,552 10 43 67

10 Answers

So assuming that object is in your scope:

```
<div ng-controller="MyCtrl">
  <select ng-model="prop.value" ng-options="v for v in prop.values">
  </select>
</div>
```

```
function MyCtrl($scope) {
  $scope.prop = {
    "type": "select",
    "name": "Service",
    "value": "Service 3",
    "values": [ "Service 1", "Service 2", "Service 3", "Service 4" ]
  };
}
```

Working Plunkr: <http://plnkr.co/edit/wTRXZYEPzJRizEltQ2g>

edited Dec 2 '14 at 22:36

[Jason Swett](#)
15.8k 37 132 244

answered Jun 26 '13 at 20:29

[James Kleeh](#)
6,643 3 19 43

- 2 ok i got it! the ng-option is replacing the `<option ng-repeat="value in prop.values">{{value}}</option>` thanks again – [francoisromain](#) Jun 26 '13 at 20:55
- 1 There has got to be a simpler way to do this, I find this very complicated for a simple select default value in angular js – [Edmund Rojas](#) Jul 30 '15 at 1:15
- 1 ng-option directive is somewhat complicated and confusing on its syntax. I think the flexibility to bind objects has made it a bit complicated. The way I try to remember this is that whenever I am using an object, I explicitly indicate what properties to use for the value/text attributes of the select control. For more information see this: [ozkary.com/2015/08/angularjs-ngoption-directive-simplified.html](#) – [ozkary](#) Sep 4 '15 at 14:57
- 1 if you want to build a select in the view without writing out the options and have one selected `<select ng-model="limit" value="10" ng-options="v for v in [5,10,15,20,30,50]"></select>` – [asherrard](#) May 6 '16 at 15:50
- 1 @james Kleeh ..here u are hardcoding the first option since u know them.But if we dont know the options and how can we display the first one by default? – [H Varma](#) Nov 23 '16 at 11:33



The [angular documentation for select](#)* does not answer this question explicitly, but it is there. If you look at the `script.js`, you will see this:

```
function MyCtrl($scope) {
  $scope.colors = [
    {name:'black', shade:'dark'},
    {name:'white', shade:'light'},
    {name:'red', shade:'dark'},
    {name:'blue', shade:'dark'},
    {name:'yellow', shade:'light'}
  ];
  $scope.color = $scope.colors[2]; // Default the color to red
}
```

This is the html:

```
<select ng-model="color" ng-options="c.name for c in colors"></select>
```

This seems to be a more obvious way of defaulting a selected value on an `<select>` with `ng-options`. Also it will work if you have different label/values.

* This is from Angular 1.2.7

edited Aug 19 '14 at 15:36

answered Dec 26 '13 at 19:55



Jess

7,588 6 47 86

This answer is more usefull when you are bringing data from a DB, make modifications and then persist the changes.

```
<select ng-options="opt.id as opt.name for opt in users" ng-model="selectedUser">
</select>
```

Check the example here:

<http://plnkr.co/edit/HrT5vUMJOtP9esGngbIV>

answered Jun 2 '15 at 21:22



Humberto Morera

634 5 8

```
<select name='partyid' id='partyid' class='span3'>
<option value=''>Select Party</option>
<option ng-repeat="item in partyName" value="{{item._id}}" ng-selected="obj.partyname ==
item.partyname">{{item.partyname}}
</option>
</select>
```

answered May 2 '14 at 10:06



nirav

233 2 4

- 5 Bad practice here... Have a look to the second yellow note : [docs.angularjs.org/api/ng/directive/select](#). – [M'sieur Toph](#) Oct 16 '14 at 16:13

this is not gonna work, ex. when you try to edit the record, how will the record bind with the field ? – [windmaomao](#) Feb 11 '15 at 2:02

If your array of objects are complex like:

```
$scope.friends = [{ name: 'John', uuid: 1234}, {name: 'Joe', uuid: 5678}];
```

And your current model was set to something like:

```
$scope.user.friend = {name:John, uuid: 1234};
```

It helped to use the `track by` function on uuid (or any unique field), as long as the `ng-model="user.friend"` also has a uuid:

```
<select ng-model="user.friend"
ng-options="friend as friend.name for friend in friends track by friend.uuid">
</select>
```

answered Apr 20 '15 at 18:55



BatteryAcid

4,850 2 11 26

This answer made my day! Also take a look at `track by` in this documentation:

docs.angularjs.org/api/ng/directive/ngOptions – Arashsoft Feb 1 at 19:41

I struggled with this for a couple of hours, so I would like to add some clarifications for it, all the examples noted here, refers to cases where the data is loaded from the script itself, not something coming from a service or a database, so I would like to provide my experience for anyone having the same problem as I did.

Normally you save only the id of the desired option in your database, so... let's show it

service.js

```
myApp.factory('Models', function($http) {
var models = {};
models.allModels = function(options) {
return $http.post(url_service, {options: options});
};

return models;
});
```

controller.js

```
myApp.controller('exampleController', function($scope, Models) {
$scope.mainObj={id_main: 1, id_model: 101};
$scope.selected_model = $scope.mainObj.id_model;
Models.allModels({}).success(function(data) {
$scope.models = data;
});
});
```

Finally the partial html *model.html*

```
Model: <select ng-model="selected_model"
ng-options="model.id_model as model.name for model in models" ></select>
```

basically I wanted to point that piece "*model.id_model as model.name for model in models*" the "*model.id_model*" uses the id of the model for the value so that you can match with the "*mainObj.id_model*" which is also the "*selected_model*", this is just a plain value, also "*as model.name*" is the label for the repeater, finally "*model in models*" is just the regular cycle that we all know about.

Hope this helps somebody, and if it does, please vote up :D

answered Jul 31 '14 at 22:27



Coronellx

391 5 10

1 you could just bind the select to `mainObj.id_model` directly (`ng-model="mainObj.id_model"`), rather than using a 'placeholder / plain value' variable `selected_model` – drzaus Aug 20 '14 at 13:53

yes, indeed, I just wanted to show that it could be done with a value directly within the `$scope`, but thanks for pointing that out. – Coronellx Aug 21 '14 at 5:00

Is it possible to bind to an object and not just an int? I cant get my `nGOptions` to set its selected item because it doesnt know how to bind my `{id:24}` to my `[[{id:23},{id:24},{id:25}]`. – Victorio Berra Feb 26 '15 at 17:34

you can always use `$index` to get the position, however, yes it is possible to bind to an object – Coronellx Mar 3 '15 at 17:45

An easier way to do it is to use `data-ng-init` like this:

```
<select data-ng-init="somethingHere = options[0]" data-ng-model="somethingHere" data-ng-
options="option.name for option in options"></select>
```

The main difference here is that you would need to include `data-ng-model`

answered Jul 14 '15 at 18:12



WMios

4,935 8 25 55

1 As specified in the docs: The only appropriate use of `ngInit` is for aliasing special properties of `ngRepeat`, as seen in the demo below. Besides this case, you should use controllers rather than `ngInit` to initialize values on a scope. – [wbeange](#) Jul 14 '15 at 18:18

The way I posted, with `ng-options` is much easier and more concrete. Both ways work. – [WMios](#) Jul 14 '15 at 18:20

```
<select id="itemDescFormId" name="itemDescFormId" size="1" ng-model="prop" ng-
change="update()">
  <option value="">English(EN)</option>
  <option value="23">Corsican(CO)</option>
  <option value="43">French(FR)</option>
  <option value="16">German(GR)</option>
```

Just add option with empty value. It will work.

[DEMO Plnkr](#)

answered Jun 10 '16 at 11:22



Surya R Praveen

782 5 8

1 perfectly worked. Thanks! – [Akash](#) Aug 31 '16 at 6:14

The *ng-model* attribute sets the selected option and also allows you to pipe a filter like *orderBy:orderModel.value*

index.html

```
<select ng-model="orderModel" ng-options="option.name for option in orderOptions">
</select>
```

controllers.js

```
$scope.orderOptions = [
  {"name": "Newest", "value": "age"},
  {"name": "Alphabetical", "value": "name"}
];

$scope.orderModel = $scope.orderOptions[0];
```

answered Jun 12 '15 at 21:48



David Douglas

6,464 1 35 43

If anyone is running into the default value occasionally being not populated on the page in Chrome, IE 10/11, Firefox -- try adding this attribute to your input/select field checking for the populated variable in the HTML, like so:

```
<input data-ng-model="vm.x" data-ng-if="vm.x !== '' && vm.x !== undefined && vm.x !==
null" />
```

answered Apr 16 '15 at 15:01



code_maestro

21 3