

## Angular 2: Get Values of Multiple Checked Checkboxes

[Ask Question](#)

My problem is really simple: I have a list of checkboxes like this:

50

```
<div class="form-group">
  <label for="options">Options :</label>
  <label *ngFor="#option of options" class="form-control">
    <input type="checkbox" name="options" value="option" /> {{option}}
  </label>
</div>
```



16

And I would like to send an array of the selected options, something like:  
[option1, option5, option8] if options 1, 5 and 8 are selected. This array is part of a JSON that I would like to send via an HTTP PUT request.

Thanks for your help!

[json](#)
[forms](#)
[checkbox](#)
[angular](#)
  
[angular2-forms](#)

asked Jan 25 '16 at 16:11



**Yannick Morel**

692 2 7 18

### 10 Answers

Here's a simple way using `ngModel` (final Angular 2)

96

```
<!-- my.component.html -->

<div class="form-group">
  <label for="options">Options:</label>
  <div *ngFor="let option of options"
    <label>
      <input type="checkbox"
        name="options"
        value="{{option.value}} [ (ngModel) ]"
        [(ngModel)]="option" />
    </label>
  </div>
</div>
```

```
// my.component.ts
```

```
@Component({ moduleId: module.id, template: `<div class="form-group">
  <label for="options">Options:</label>
  <div *ngFor="let option of options"
    <label>
      <input type="checkbox"
        name="options"
        value="{{option.value}} [ (ngModel) ]"
        [(ngModel)]="option" />
    </label>
  </div>
</div>`
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

```

    {name:'OptionC', value:'3', check
  ]

  get selectedOptions() { // right no
    return this.options
      .filter(opt => opt.chec
      .map(opt => opt.value)
  }
}

```

edited Oct 21 '16 at 0:43

answered Oct 21 '16 at 0:32



[ccwasden](#)

5,213 2 24 30

6 How would you access the selected options? I'm not familiar with the `get` function declaration – [Jordash](#) Dec 30 '16 at 3:02

3 just leave off the `()` (ie. `this.selectedOptions` instead of `this.selectedOptions()`) – [ccwasden](#) Jan 2 '17 at 19:04



I have find a solution thanks to Gunter! Here is my whole code if it could help anyone:

27



```

<div class="form-group">
  <label for="options">Opt:
  <div *ngFor="#option of (
    <label>
      <input type="che
        name="opt:
        value="{{(
          [checked]:
            (change)='
          {{option}}
        </label>
      </div>
    </div>
  </div>

```

Here are the 3 objects I'm using:

```

options = ['OptionA', 'OptionB', 'Opt
optionsMap = {
  OptionA: false,
  OptionB: false,
  OptionC: false,
};
optionsChecked = [];

```

And there are 3 useful methods:

1. To initiate `optionsMap` :

```

initOptionsMap() {
  for (var x = 0; x<this.order.opt:
    this.optionsMap[this.options|
  ,

```

```
updateCheckedOptions(option, event) {
  this.optionsMap[option] = event.t
}
```

3. to convert optionsMap into optionsChecked and store it in options before sending the POST request:

```
updateOptions() {
  for(var x in this.optionsMap) {
    if(this.optionsMap[x]) {
      this.optionsChecked.push(x)
    }
  }
  this.options = this.optionsChecked;
  this.optionsChecked = [];
}
```

edited Mar 29 '16 at 12:52



[Joel Almeida](#)

6,129 3 19 49

answered Feb 24 '16 at 17:08



[Yannick Morel](#)

692 2 7 18

i think according to your solution its must to know the all values of array of checkboxes is't it ? – [Pardeep Jain](#) Feb 25 '16 at 5:52

Yes exactly, I'm not sure to understand how it could works otherwise actually... – [Yannick Morel](#) Feb 25 '16 at 12:06

see my answer. hope it may help you. where if you dont know the length of array or even don't know about the all entries, code run without error. try Plnkr given – [Pardeep Jain](#) Feb 25 '16 at 12:10

this shouldnt be that difficult, why doesnt the ngModel just take care of it :( – [webmaster](#) Oct 9 '16 at 22:35

I believe my answer here: [stackoverflow.com/a/41027266/4496240](https://stackoverflow.com/a/41027266/4496240), would be better suited for this as it's requires much less work and is done in a more "angular" centric way. It's also updated for the new forms library. – [Austin Shoecraft](#) Dec 7 '16 at 21:04



create a list like :-

18

```
this.xyzlist = [
  {
    id: 1,
    value: 'option1'
  },
  ...
]
```

```
    }
  ];

```

Html :-

```
<div class="checkbox" *ngFor="let lis
  <label>
    <input formControlName=
      {{list.id}} (change)="onCheckboxChan
    </div>

```

then in it's component ts :-

```
onCheckboxChange(option, event) {
  if(event.target.checked) {
    this.checkedList.push(option.i
  } else {
    for(var i=0 ; i < this.xyzlist.l
      if(this.checkedList[i] == opti
        this.checkedList.splice(i,1)
    }
  }
}
console.log(this.checkedList);
}

```

edited Feb 25 at 9:55



priya\_21

61 1 12

answered Jan 17 '18 at 21:57



Alok Kamboj

497 6 9

- 1 Stuck for many hours figuring out the issue. This solution worked like charm! – Pavan Nov 14 '18 at 5:56



4



```
<input type="checkbox" name="options"
  $event)" />

```

```
export class MyComponent {
  checked: boolean[] = [];
  updateChecked(option, event) {
    this.checked[option]=event; // or
  }
}

```

answered Jan 25 '16 at 16:15



Günter Zöchbauer

344k 77 1065 975

this.checked return [undefined: Event] after checked or uncheck why ? – Pardeep Jain Feb 25 '16 at 5:38

I don't know what this.checked should refer to and at what time. Can

you create a Plunker? – Günter Zöchbauer Feb 25 '16 at 5:40

Sorry, I only looked at the source of the question, not of my answer ;-p. `option` is not defined. If you add a field with this name it's no longer undefined (see [plnkr.co/edit/MsJDIEimeeeV4Wk3NNk5?p=preview](https://plnkr.co/edit/MsJDIEimeeeV4Wk3NNk5?p=preview)). In the question `option` is the value variable of `*ngFor="#option of xxx"`. It just needs to exist in the scope. – [Günter Zöchbauer](#) Feb 25 '16 at 6:14

- 1 An example [plnkr.co/edit/N9NXBYcwhon6lTr8RP5y?p=preview](https://plnkr.co/edit/N9NXBYcwhon6lTr8RP5y?p=preview) There are different ways to store the checked state (with the `option`, or store the index instead of the value, ...) – [Günter Zöchbauer](#) Feb 25 '16 at 6:45



▲ I hope this would help someone who has the same problem.

3

templet.html

```
<form [formGroup] = "myForm" (ngSubmi
  <ng-template ngFor [ngForOf]="fligh
    <input type="checkbox" [value]="
      (change)="flightids[i]=
    [$event.target.checked,$event.target.
  </ng-template>
</form>
```

component.ts

flightids array will have another arrays like this `[ [ true, 'id_1'], [ false, 'id_2'], [ true, 'id_3']...]` here true means user checked it, false means user checked then unchecked it. The items that user have never checked will not be inserted to the array.

```
flightids = [];
confirmFlights(value){
  //console.log(this.flightids);

  let confirmList = [];
  this.flightids.forEach(id => {
    if(id[0]) // here, true means t
      confirmList.push(this.flightL
  });
  //console.log(confirmList);
}
```

answered Jun 29 '17 at 18:23



[James Sung](#)

31 3



```

<div class="form-group">
  <label for="options">Options:
  <div *ngFor="let option of options">
    <label>
      <input type="checkbox"
        name="options-{{option.Id}}"
        value="{{option.Id}}"/>
      {{option.Name}}
    </label>
  </div>
  <button type="submit">Submit
</div>

```

**\*\* XYZ.Component.ts \*\***

2. create a list -- xyzlist.
3. assign values, I am passing values from Java in this list.
4. Values are Int-Id, boolean - checked (Can Pass in Component.ts).
5. Now to get value in Component.ts.

```

xyzlist; // Just created a List
onClicked(option, event) {
  console.log("event " + this.option);
  console.log("event checked " + this.option.checked);
  console.log("event checked " + this.option.checked);
  for (var i = 0; i < this.xyzlist.length; i++) {
    console.log("test --- " + this.xyzlist[i].Id);
    if (this.xyzlist[i].Id == this.option.Id) {
      this.xyzlist[i].checked = true;
    }
  }
  console.log("after update");
}

```

edited Feb 20 '17 at 9:20

answered Dec 17 '16 at 9:47



Gourav Bhatia

146 1 11

2

I just faced this issue, and decided to make everything work with as less variables as i can, to keep workspace clean. Here is example of my code

```

<input type="checkbox" (change)="changeModel($event, list, val)"
[checked]="modelArr.includes(option.Id)"

```

Method, which called on change is pushing value in model, or removing it.

```

public changeModel(ev, list, val) {

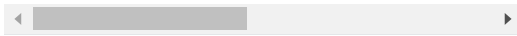
```

```
list.splice(i, 1);
}
}
```

answered Oct 27 '17 at 8:14



tv1st  
29 3



1 I have encountered the same problem and now I have an answer I like more (may be you too). I have bounded each checkbox to an array index.

First I defined an Object like this:

```
SelectionStatusOfMutants: any = {};
```

Then the checkboxes are like this:

```
<input *ngFor="let Mutant of Mutants"
[(ngModel)]="SelectionStatusOfMutants
```

As you know objects in JS are arrays with arbitrary indices. So the result are being fetched so simple:

Count selected ones like this:

```
let count = 0;
Object.keys(SelectionStatusOfMuta
  if (SelectionStatusOfMutants[
    count++;
  });
```

And similar to that fetch selected ones like this:

```
let selecteds = Object.keys(Selection
  return SelectionStatusOfMutan
  });
```

You see?! Very simple very beautiful. TG.

answered Mar 27 '17 at 9:27



ConductedClever  
1,574 13 38



1 @ccwasden solution above works for me with a small change, each checkbox must have a unique name otherwise binding wont work



```

        name="options_{{i}}
        value="{{option.va
        [(ngModel)]="optio
        {{option.name}}
      </label>
    </div>
  </div>

// my.component.ts

@Component({ moduleId:module.id, temp

export class MyComponent {
  options = [
    {name:'OptionA', value:'1', check
    {name:'OptionB', value:'2', check
    {name:'OptionC', value:'3', check
  ]

  get selectedOptions() { // right no
    return this.options
      .filter(opt => opt.chec
      .map(opt => opt.value)
  }
}

```

and also make sur to import  
FormsModule in your main module

```

import { FormsModule } from '@angular

imports: [
  FormsModule
],

```

answered Nov 6 '18 at 6:10



Syed Waqas

618 1 8 27



0 Since I spent a long time solving a similar problem, I'm answering to share my experience. My problem was the same, to know, getting many checkboxes value after a specified event has been triggered. I tried a lot of solutions but for me the sexiest is using **ViewChildren**.

```

import { ViewChildren, QueryList } fr

/** Get handle on cmp tags in the tem
@ViewChildren('cmp') components: Quer

ngAfterViewInit(){
  // print array of CustomComponent
  console.log(this.components.toArr
}

```

Found here:

<https://stackoverflow.com/a/40165639/4775727>

Potential other solutions for ref, there



- [Angular 6: How to build a simple multiple checkbox to be checked/unchecked by the user?](#)
- [Angular 6 How To Get Values From Multiple Checkboxes and Send in From](#)
- [Angular how to get the multiple checkbox value?](#)
- [Angular 2: Get Values of Multiple Checked Checkboxes](#)
- <https://medium.com/@vladguleaev/reusable-angular-create-multiple-checkbox-group-component-84f0e4727677>
- <https://netbasal.com/handling-multiple-checkboxes-in-angular-forms-57eb8e846d21>
- <https://medium.com/@shlomiassaf/the-angular-template-variable-you-are-missing-return-of-the-var-6b573ec9fdc>
- <https://www.bennadel.com/blog/3205-using-form-controls-without-formsmodule-or-ngmodel-in-angular-2-4-1.htm>
- [How to get checkbox value in angular 5 app](#)
- [Angular 2: Get Values of Multiple Checked Checkboxes](#)
- [Filter by checkbox in angular 5](#)
- [How to access multiple checkbox values in Angular 4/5](#)

answered Mar 23 at 13:11



4x10m

192 2 9

