

How to reset selected file with input tag file type in Angular 2?

Asked 2 years, 9 months ago Active 2 months ago Viewed 57k times



This is how my input tag looks like:

58

```
<input type="file" placeholder="File Name" name="filename" (change)="onChange($event)">
<button>Reset</button>
```



19

I want to reset the selected file in Angular 2. Help would be greatly appreciated. Let me know if you need more details.

P.S.

I could get file details from `$event` parameters and save it in a typescript variable, but this variable is not bound to the input tag.



edited Apr 13 '17 at 9:25

asked Oct 20 '16 at 22:04



akash

521 1 6 21

When you say reset, what exactly do you mean. Can you create a plnkr.co and post what problem are you facing – [abhinav](#) Oct 20 '16 at 22:24

10 Answers



You can use `ViewChild` to access the input in your component. First, you need to add `#someValue` to your input so you can read it in the component:

148

```
<input #myInput type="file" placeholder="File Name" name="filename"
(change)="onChange($event)">
```



Then in your component you need to import `ViewChild` from `@angular/core` :

```
import { ViewChild } from '@angular/core';
```

Then you use `ViewChild` to access the input from template:

```
@ViewChild('myInput')
myInputVariable: ElementRef;
```

Now you can use `myInputVariable` to reset the selected file because it's a reference to input with `#myInput`, for example create method `reset()` that will be called on `click` event of your button:

```
reset() {
  console.log(this.myInputVariable.nativeElement.files);
  this.myInputVariable.nativeElement.value = "";
  console.log(this.myInputVariable.nativeElement.files);
}
```

First `console.log` will print the file you selected, second `console.log` will print an empty array because `this.myInputVariable.nativeElement.value = ""`; deletes selected file(s) from the input. We have to use `this.myInputVariable.nativeElement.value = ""`; to reset the value of the input because input's `FileList` attribute is **readonly**, so it is impossible to just remove item from array. Here's working [Plunker](#).

edited Jul 16 '18 at 15:02

answered Oct 20 '16 at 22:28



[Stefan Svrkota](#)

29.3k 5 71 75

2 Thanks @Stefan. Your solution works! – [akash](#) Oct 20 '16 at 22:33

2 is this enough to clear value `this.myInputVariable.nativeElement.value = ""`; ?/ – [Pardeep Jain](#) Apr 7 '17 at 7:35 ✎

1 @PardeepJain Yes, this clears selected file from file input if that's what you're looking for. – [Stefan Svrkota](#) Apr 7 '17 at 7:42

1 That's a great solution with good explanation – [Tk1993](#) Sep 19 '17 at 8:51 ✎

3 `myInputVariable` is indeed of type `ElementRef` – [Blauhim](#) Nov 2 '17 at 23:09



One way to achieve it is to wrap your input in `<form>` tag and reset it.

I'm not considering attaching the form to `NgForm` or `FormControl` either.

12

```

@Component({
  selector: 'form-component',
  template: `
    <form #form>
      <input type="file" placeholder="File Name" name="filename">
    </form>
    <button (click)="reset()">Reset</button>
  `
})
class FormComponent {

  @ViewChild('form') form;

  reset() {
    this.form.nativeElement.reset()
  }
}

```

<https://plnkr.co/edit/UIqh2l093LV6GIQWKkUA?p=preview>

edited Oct 20 '16 at 22:34

answered Oct 20 '16 at 22:24



Edmar Miyake

9,033 3 23 28

Thanks @Edmar, your solution works too. – [akash](#) Oct 20 '16 at 22:33

can we use .reset() method on viewChild ?? – [Pardeep Jain](#) Apr 7 '17 at 7:32

Hello Edmar ... can you help me out with the question on this link ...stackoverflow.com/questions/48769015/... – [Heena](#) Feb 13 '18 at 15:12

Angular 5

8

html

```

<input type="file" #inputFile>

<button (click)="reset()">Reset</button>

```

template

```
@ViewChild('inputFile') myInputVariable: ElementRef;

reset() {
  this.myInputVariable.nativeElement.value = '';
}
```

Button is not required. You can reset it after change event, it is just for demonstration

edited Jun 28 '18 at 19:29

answered Mar 1 '18 at 19:16



Admir

752 7 9

I used this version and it works well for me - thanks @Admir – [user1288395](#) Mar 25 at 19:49

Short version [Plunker](#):

2

```
import { Component } from '@angular/core';

@Component({
  selector: 'my-app',
  template: `
    <input #myInput type="file" placeholder="File Name" name="filename">
    <button (click)="myInput.value = ''>Reset</button>
  `
})
export class AppComponent {

}
```

And i think more common case is to not using button but do reset automatically. Angular [Template statements](#) support chaining expressions so [Plunker](#):

```
import { Component } from '@angular/core';

@Component({
  selector: 'my-app',
```

```

template: `
  <input #myInput type="file" (change)="onChange(myInput.value, $event);
myInput.value = ''" placeholder="File Name" name="filename">
`
})
export class AppComponent {

  onChange(files, event) {
    alert( files );
    alert( event.target.files[0].name );
  }

}

```

And [interesting link](#) about why there is no recursion on value change.

edited Jun 20 '17 at 7:43

answered Jun 20 '17 at 6:15



[Oleg Mazko](#)

932 10 9

I think its simple, use #variable

2

```

<input #variable type="file" placeholder="File Name" name="filename"
(change)="onChange($event); variable.value='' ">
<button>Reset</button>

```

"variable.value = null" option also available

answered Feb 15 '18 at 12:49



[Fawad Mukhtar](#)

254 2 4

I typically reset my file input after capturing the selected files. No need to push a button, you have everything required in the \$event object that you're passing to onChange :

1

```

onChange(event) {
  // Get your files
  const files: FileList = event.target.files;

```

```
// Clear the input
event.srcElement.value = null;
}
```

Different workflow, but the OP doesn't mention pushing a button is a requirement...

answered Mar 23 '18 at 6:55



[2Toad](#)

12.1k

6

32

33

template:

0

```
<input [(ngModel)]="componentField" type="file" (change)="fileChange($event)"
placeholder="Upload file">
```

component:

```
fileChange(event) {
    alert(this.torrentFileValue);
    this.torrentFileValue = '';
}
```

answered Feb 23 at 17:54



[Hett](#)

1,199

2

15

31

If you are facing issue with ng2-file-upload,

0

HTML:

```
<input type="file" name="myfile" `**#activeFrameinputFile**` `ng2FileSelect
[uploader]="frameUploader" (change)="frameUploader.uploadAll()" />
```

component

```
import { Component, OnInit, ElementRef, ViewChild } from '@angular/core';

@ViewChild('`**activeFrameinputFile**`) `**InputFrameVariable**`: ElementRef;

this.frameUploader.onSuccessItem = (item, response, status, headers) => {
  this.`**InputFrameVariable**`.nativeElement.value = '';
};
```

edited Mar 29 at 7:00



Abhishek Ekaanth

1,411 12 26

answered Dec 4 '18 at 11:57



Anil Kumar Dyavanapalli

11 5



0



```
<input type="file" id="image_control" (change)="validateFile($event)" accept="image/gif,
image/jpeg, image/png" />
```

```
validateFile(event: any): void {
  const self = this;
  if (event.target.files.length === 1) {
    event.srcElement.value = null;
  }
}
```

answered May 9 at 8:04



Mayur Sakariya

1

1 can you provide a short description on your code explains how it works? – [Jacob Nelson](#) May 9 at 8:08



0



I have add this input tag into form tag..

```
<form id="form_data">
  <input type="file" id="file_data" name="browse"
    (change)="handleFileInput($event, dataFile, f)" />
</form>
```

I angular typescript, I have added below lines, get your form id in document forms and make that value as null.

```
for(let i=0; i<document.forms.length;i++){  
  if(document.forms[i].length > 0){  
    if(document.forms[i][0]['value']){ //document.forms[i][0] = "file_data"  
      document.forms[i][0]['value'] = "";  
    }  
  }  
}
```

Print document.forms in console and you can get idea..

answered Jun 4 at 10:04



Deepa

101 1 5