

Angular2 innerHtml binding remove style attribute [duplicate]

Asked 2 years, 10 months ago Active 9 months ago Viewed 36k times



49

This question already has an answer here:

[In RC.1 some styles can't be added using binding syntax](#) 1 answer



18

My problem, that when I use innererHtml binding - angular2 remove all styles attributes. It's important for me, baccuse in my task - html is generated on server-side with all styles. Example:

```
@Component({
  selector: 'my-app',
  template: `
    <input type="text" [(ngModel)]="html">
    <div [innerHTML]="html">
    </div>
  `,
})
export class App {
  name:string;
  html: string;
  constructor() {
    this.name = 'Angular2'
    this.html = "<span style=\"color:red;\">1234</span>";
  }
}
```

But in DOM I see only 1234 and this text is not red.

<http://plnkr.co/edit/UQJOFMKI9OwMRIJ38U8D?p=preview>

Thank you!



angular

styles

edited Aug 22 '17 at 10:33

asked Sep 21 '16 at 23:27

SamProf

**Frederik Struck-Schøning****10.2k** 7 48 55**406** 1 4 6

marked as duplicate by [Günter Zöchbauer](#) angular2 Feb 6 '17 at 9:16

This question has been asked before and already has an answer. If those answers do not fully address your question, please [ask a new question](#).

4 Answers



You can leverage `DomSanitized` to avoid it.

111

The easiest way is to create custom pipe like:

```
import { DomSanitizer } from '@angular/platform-browser'
import { PipeTransform, Pipe } from "@angular/core";

@Pipe({ name: 'safeHtml' })
export class SafeHtmlPipe implements PipeTransform {
  constructor(private sanitized: DomSanitizer) {}
  transform(value) {
    return this.sanitized.bypassSecurityTrustHtml(value);
  }
}
```

So you can use it like:

```
<div [innerHTML]="html | safeHtml"></div>
```

[Plunker Example](#)

edited Oct 11 '18 at 9:29

answered Sep 22 '16 at 4:51

**yurzui****114k** 15 238 243

Very cool resolution! Thank you very much! – [SamProf](#) Sep 23 '16 at 19:08

1 Thanks ! Really useful answer. – [codelovesme](#) Jan 24 '17 at 10:32

Note that you can add a new pipe to Ionic with the command: `ionic g pipe MyPipe` – [Tyler](#) Sep 19 '17 at 6:22

Thanks so much. Very elegant and it worked for me on a different kinda problem. – [Shriram M.](#) Oct 30 '17 at 19:56

Thank you so much. Its work fine for me. – [Biby Cheriyan](#) Jan 12 '18 at 9:29

I improved the example of yurzui a bit by completing the needed imports:

31

```
import {DomSanitizer} from '@angular/platform-browser';
import {PipeTransform, Pipe} from "@angular/core";

@Pipe({ name: 'safeHtml'})
export class SafeHtmlPipe implements PipeTransform {
  constructor(private sanitized: DomSanitizer) {}
  transform(value) {
    return this.sanitized.bypassSecurityTrustHtml(value);
  }
}
```

I also had to add the class in my app.module.ts file

```
import ...
import {SafeHtmlPipe} from "../pipes/safehtml.pipe";
@NgModule({
  declarations: [
    AppComponent,
    ...,
    SafeHtmlPipe  <--
  ],
  imports: [...],
  providers: [...],
  bootstrap: [AppComponent]
})
export class AppModule {
}
```

edited Aug 22 '17 at 10:33

answered Dec 15 '16 at 11:44

[mvermand](#)



Frederik Struck-Schøning

10.2k 7 48 55



2,812 3 27 55

Note that the `sanitizer` has a few methods for *trusting* content e.g.

4

```
return this.sanitizer.bypassSecurityTrustStyle(value);  
return this.sanitizer.bypassSecurityTrustHtml(value);  
return this.sanitizer.bypassSecurityTrustXxx(value); // - see docs [1]
```

via <https://stackoverflow.com/a/41089093/142714>

So, `bypassSecurityTrustStyle` may also be what you want here, as this will show inline styles within your HTML content (`value`).

[1] docs: <https://angular.io/docs/ts/latest/api/platform-browser/index/DomSanitizer-class.html>

edited May 23 '17 at 12:26



Community ♦

1 1

answered Feb 6 '17 at 9:12



a darren

7,314 4 40 39

1 This should be the accepted answer in my opinion. only using `bypassSecurityTrustHtml()` is prone to run malicious javascript. If only styles are what you need to bypass, bypass using `bypassSecurityTrustStyle()` is better. — John Apr 10 '18 at 8:12

Angular 2 aims for a more [declarative approach](#), so directly manipulating HTML [is often discouraged](#).

2

I believe that (almost) all HTML manipulations are patched to be filtered by angular's DOM sanitization. As you can imagine `style` attributes aren't white-listed for `span` elements, in fact, [span has no allowed attributes](#) at the moment.

edited May 23 '17 at 12:18



Community ♦

1 1

answered Sep 22 '16 at 3:51



kbtzr

9,037 5 43 67

