

# AngularJS String with newlines, shown without breaks

Asked 5 years, 7 months ago   Active 3 years, 5 months ago   Viewed 46k times



In the database I'm saving input from a textarea, where you can add breaks. But how to show them in a Angular view? For PHP this is `n12br()` .

36

Like `<div>{{ item.foobar }}</div>` , with breaks shown that are saved like that in the database.



javascript

angularjs



5

asked Feb 16 '14 at 16:18



[user1469734](#)

755 7 28 46

This could also be used to ask/answer the same question about with line breaks(`\n`) instead of just html breaks(`<br>`). – [discodane](#) Mar 24 '17 at 19:55

## 4 Answers



Binding an HTML seems a bit unsafe. The directive suggested by CuriousGuy has a bit of extra engineering to correctly escape the HTML.

91

I find it easier to use a styling rule `white-space: pre-line` .



Example at [JSFiddle](#).



See also:

- [Mozilla Developer Network documentation for `white-space`](#)
- [Internet Explorer documentation for `white-space`](#)
- [W3 documentation for `white-space`](#)

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



Thiago Negri

3,653 1 20 37

6 This answer should be accepted – [Oleg](#) Jan 21 '15 at 8:22

1 pre-line! you're THE boss. – [Luis Masuelli](#) Mar 30 '16 at 16:27

What is the browser compatibility on this one? – [Eduard Luca](#) Apr 25 '16 at 12:13

Yes, it works. Thank you. Very nice approach – [saike](#) May 26 '16 at 9:40

Super ! very nice, easy and clean. – [Al-Mothafar](#) Dec 4 '16 at 11:29

Best solution:

11

```
<div class="multi_lines_text">{{ item.foobar }}</div>
```

```
<style>
```

```
.multi_lines_text { white-space: pre-line; }
```

```
</style>
```

ng-bind-html - it's not safe way and display error in console.

answered Mar 30 '16 at 16:24



Sergii Semesko

111 1 4

If you want to add line breaks only and all the rest text to show as is you can use the following filter:

4

```
app.filter('nl2br', function() {
  var span = document.createElement('span');
  return function(input) {
    if (!input) return input;
    var lines = input.split('\n');

    for (var i = 0; i < lines.length; i++) {
```

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

```

    }
    return lines.join('<br />');
  }
});

```

then use it in view like this:

```
<div ng-bind-html="someVar | nl2br"></div>
```

In this case the only HTML tag that is present in the result string is `<br />`. All other tags will be escaped. And don't forget to load `ngSanitize` module (`angular-sanitize.js` script).

The full working example you can find [on jsFiddle](#).

edited Feb 16 '14 at 20:21

answered Feb 16 '14 at 17:29



Oleg

15k 7 50 74

Any particular reason you don't just do something like `return (input || '').replace(/\n/g, '<br>');` ? – CWSpear Jul 15 '14 at 22:06

@CWSpear The idea is to make `someVar` be shown as simple text, without rendering HTML tags (except `<br>` on the line breaks). You may compare these two jsFiddles: [first](#) and [second](#) – Oleg Jul 16 '14 at 6:08

1 @CWSpear my apologies, here are corrected jsFiddles: [first](#) and [second](#). (StackOverflow doesn't allow editing comments after 5 mins since they were posted) – Oleg Jul 16 '14 at 9:51

Googling angularjs nl2br yielded this result:

1

[AngularJS - Remove \n from data](#)

Namely the last answer which demonstrates how to replace new lines with `<br />` using a filter. Combine that with `ng-bind-html`, as the unsafe version might not be currently available. You should probably convert the line breaks to `<br />` when inserting them to database(makes more sense to me).

edited May 23 '17 at 11:54

answered Feb 16 '14 at 16:41



Erti-Chris Eelmaa

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

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