

Angular2 table rows as component

Asked 3 years, 8 months ago Active 1 year, 6 months ago Viewed 53k times



I am experimenting with angular2 2.0.0-beta.0

83

I have a table and the line content is generated by angular2 this way:



15

```
<table>
  <tr *ngFor="#line of data">
    .... content ....
  </tr>
</table>
```

Now this works and I want to encapsulate the content into a component "table-line".

```
<table>
  <table-line *ngFor="#line of data" [data]="line">
  </table-line>
</table>
```

And in the component, the template has the <tr><td> content.

But now the table does no more work. Which means, the content is no longer shown in columns. In the browser, the inspector shows me that the DOM elements look like this:

```
<table>
  <table-line ...>
    <tbody>
      <tr> ....
```

How can I make this work?



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



2,139 1 18 21



1,113 2 14 22

4 Answers

**use existing table elements as selector**

93

The table element doesn't allow `<table-line>` elements as children and the browser just removes them when it finds them. You can wrap it in a component and still use the allowed `<tr>` tag. Just use `"tr"` as selector.

**using `<template>`**

`<template>` should be allowed as well but doesn't yet work in all browsers. Angular2 actually never adds a `<template>` element to the DOM, but only processes them internally, therefore this can be used in all browsers with Angular2 as well.

Attribute selectors

Another way is to use attribute selectors

```
@Component({
  selector: '[my-tr]',
  ...
})
```

to be used like

```
<tr my-tr>
```

edited Feb 14 '17 at 10:28

answered Jan 1 '16 at 13:21

**Günter Zöchbauer**

367k 83 1171 1062

So I would create my `TableItemComponent` with a selector = `'tr'`. But how can use `'tr'` in other places for different issues? – [fbenoit](#) Jan 1 '16 at 13:24

- 6 If the parent component includes your custom `tr` tag it will be used, otherwise it's default browser behavior takes place. You can also add attributes or classes to your component selector like `<tr line-item>` with a component selector `"tr[line-item]"` or `<tr class="line-item">` with

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

- 5 Is there any real drawback to this? It works fine, however the default settings for `tslint` direct me to the [style guide](#) recommending *against* this. I dislike turning off recommended linting rules, but as far as I can tell, it seems a fairly arbitrarily style rule and unavoidable in some situations (such as OP's problem) – [Rob ♦](#) May 31 '17 at 8:18
- 1 I'm pretty sure there is no drawback, except that element selectors are way more common and therefore should be the default. If there is a valid case like table elements or `` or similar, I don't see any reason why not to use it. – [Günter Zöchbauer](#) May 31 '17 at 17:18
- 2 Making my component selector use `tr[line-item]` worked and conforms to Angular Style guide rule STYLE 05-03 that `tslint` complains against. – [C.M.](#) Jun 7 '17 at 16:25

I found the example very usefull but it didn't work in the 2,2.3 build, so after much head scratching made it work again with a few small changes.

25

```
import {Component, Input} from '@angular/core'

@Component({
  selector: "[my-tr]",
  template: `<td *ngFor='let item of row'>{{item}}</td>`
})
export class MyTrComponent {
  @Input("line") row:any;
}

@Component({
  selector: "my-app",
  template: `<h1>{{title}}</h1>
<table>
  <tr *ngFor="let line of data" my-tr [line]="line"></tr>
</table>`
})
export class AppComponent {

  title = "Angular 2 - tr attribute selector!";
  data = [ [1,2,3], [11, 12, 13] ];
  constructor() { console.clear(); }
}
```

answered Dec 22 '16 at 16:04

 [DudeOfDoom](#)

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

1 Great answer. I just had to add the `MyTrComponent` to the `app.module.ts` and works fine. – [Tito Leiva](#) Nov 12 '18 at 11:49

Here's an example using a component with an attribute selector:

24

```
import {Component, Input} from '@angular/core';
@Component({
  selector: '[myTr]',
  template: `<td *ngFor="let item of row">{{item}}</td>`
})
export class MyTrComponent {
  @Input('myTr') row;
}
@Component({
  selector: 'my-app',
  template: `{{title}}
<table>
  <tr *ngFor="let line of data" [myTr]="line"></tr>
</table>
`
})
export class AppComponent {
  title = "Angular 2 - tr attribute selector";
  data = [ [1,2,3], [11, 12, 13] ];
}
```

Output:

```
1   2   3
11  12  13
```

Of course the template in the `MyTrComponent` would be more involved, but you get the idea.

Old (beta.0) [plunker](#).

edited Oct 6 '16 at 15:10

answered Jan 1 '16 at 17:14



[Mark Rajcok](#)

306k 95 447 467

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

@PashaPasha, see stackoverflow.com/questions/34280475/... – Mark Rajcok Apr 11 '16 at 14:20 

- 11 This doesn't work at all(tried with angular 2.3). Error thrown: Error: Uncaught (in promise): Error: Template parse errors: Can't bind to 'myTr' since it isn't a known property of 'tr'. – Vukasin Jan 4 '17 at 9:03

try this

-3

```
@Component({
  selector: 'parent-selector',
  template: '<table><body><tr></tr></body></table>'
  styles: 'tr{ display:table-row; box-sizing:inherit; }'
})
export class ParentComponent{
}

@Component({
  selector: 'parent-selector',
  template: '<td>Name</td>Date<td></td><td>Stackoverflow</td>'
})
export class ChildComponent{}
```

answered Mar 15 '18 at 17:29



Matsteel

55 1 2