

How can I set the value of a native element in Angular, using Renderer2?

▲ I'd like to set the innerText/innerHTML/textContent of a nativeElement?

10 `this.render.setValue(this.nativeElement.querySelector('.down .inn'), timeVal);`

▼ where **timeVal** is a string

★ the element is correctly selected, but **setValue** seems not working at all

2

javascript angular dom renderer

asked Apr 4 '18 at 15:21



Donovan

1,184 4 26 49

How do you create `nativeCloneLi` ? – ForestG Apr 4 '18 at 15:25

Using an ElementRef `instance.nativeElement` – Donovan Apr 4 '18 at 23:52

1 Answer

▲ You need to use `renderer.setProperty()` instead of `renderer.setValue()` .

19 `import { Component, Renderer2, AfterViewInit, ElementRef, ViewChild } from '@angular/core';`

▼ `@Component({`

Join Stack Overflow to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google



```
export class AppComponent implements AfterViewInit {  
  @ViewChild('el') el: ElementRef;  
  
  constructor(private renderer: Renderer2) {}  
  
  ngAfterViewInit() {  
    this.renderer.setProperty(this.el.nativeElement, 'innerHTML', '<h1>Hello  
world</h1>');  
  }  
}
```

[Live demo](#)

edited Apr 6 '18 at 13:42

answered Apr 4 '18 at 15:33



Tomasz Kula

8,273 20 43

Should we also use **renderer** if we just want to read a value on the native element, for instance `const top = this._elementRef.nativeElement.getBoundingClientRect().top`; – [abyrne85](#) Oct 17 '18 at 11:09

@abyrne85 I'm also curious, should we use the **renderer** if we just want to read the element. – [calbear47](#) Nov 29 '18 at 14:39

- 1 @abyrne85, @calbear47 No, there is no need to. **Renderer2** provides a API to manipulate native elements in platform agnostic manner. Meaning it's releasing you from direct DOM access and allowing you operate on it in abstract way. But as long as you only read and not changing anything you can use native element directly (if access to them is supported) or provide custom **Renderer2** implementation with features you are missing. Actually Angular doesn't provide any support (except `ElementRef` and [events](#)) to get anything from DOM. – [Bardr](#) Feb 7 at 15:02

Got a question that you can't ask on public Stack Overflow? [Learn more](#) about sharing private information with Stack Overflow for Teams.



Join **Stack Overflow** to learn, share knowledge, and build your career.

Email Sign Up

OR SIGN IN WITH



Google

Facebook