

Promise in Pipe transform

Asked 1 year, 5 months ago Active 1 year, 5 months ago Viewed 2k times

I don't want to use "| async pipe" because I don't want to get data for every component. My transform function below:

3

```
transform(key: string, ...args) {  
  this.storage.get("dictionary").then((dict) => {  
    var translated = dict[key] || "noResource";  
    console.log("key = "+key + ", value = "+translated);  
    return translated;  
  });  
}
```

I can get key and values but value does not rendered. I tried `ngZone` but not working.



ionic-framework

angular-pipe

edited Feb 6 '18 at 10:59



[martin](#)

51.3k

14

107

149

asked Feb 6 '18 at 7:43



[Berk Akkerman](#)

313

1

11

2 Answers

I don't understand why you don't want to use the only "buildin" pipe that matches your case.

5

says your pipe is :

```
@Pipe({name: 'mypipe'})  
export class MyPipe implements PipeTransform  
{  
  transform(key: string, ...args) {  
    // watch out you missed the return statement !!  
    -->return<-- this.storage.get("dictionary").then((dict) => {  
      var translated = dict[key] || "noResource";  
      console.log("key = "+key + ", value = "+translated);  
    });  
  }  
}
```

```

        return translated;
    });
}

```

the in your template you could just use

```
{{ 'mykey' | mypipe | async }}
```

If you don't want to use async pipe you will be forced to mimic is logic who is already dead simple and bug proof. No gain for you.

answered Feb 6 '18 at 8:08



Pierre Mallet

4,106 1 9 21

If you want the pipe to be asynchronous and update itself later I think the only way is by using [WrappedValue](#) .

2

For example like this where the `Promise` updates the pipe after 1s. In your use-case it's going the work exactly the same way.

See live demo: <https://stackblitz.com/edit/angular-knamur?file=src%2Fapp%2Ftest-async.ts>

```

@Pipe({name: 'testAsync', pure: false})
export class TestAsyncPipe implements PipeTransform {

    private val: number;
    private originalValue: number;

    constructor(private _ref: ChangeDetectorRef) {}

    transform(value: number): any {
        if (value === this.originalValue) {
            return this.val;
        } else {
            new Promise(resolve => {
                setTimeout(() => resolve(value * 2), 1000) ;
            }).then((v: number) => {
                this.val = v + 1;
                this._ref.markForCheck();
            });
        }

        this.originalValue = value;
    }
}

```

```
    this.val = value;

    return WrappedValue.wrap(value);
  }
}
```

Then in template use:

```
{{ 42 | testAsync }}
```

After 1s it'll update the template and show 85 .

edited Feb 6 '18 at 8:58

answered Feb 6 '18 at 8:49



[martin](#)

51.3k

14

107

149