

# How do I use and apply JavaScript decorators?

Asked 6 years, 11 months ago   Modified 6 years, 11 months ago   Viewed 30k times



21

I am trying to understand how to use decorators in a very simple piece of code, so I can apply this concept to my bigger project. Taking cue from Addy Osmani's article [here](#), I created a simple piece of code as below.



Say, I have a class called `Cat`, with a `meow()` method, I want to decorate it with some logging, as below.



```
class Cat {
  @logger
  meow() { console.log( ' Meeeoow! ' ) }
};

function logger(target, key, descriptor) {
  console.log("Cat snarling...");
  return descriptor;
}

const cat = new Cat();
cat.meow();
```

When I try to execute this against the Node.js interpreter (version 9.1.0), I get the following error.

```
/Users/ravindranath/projects/decorators/index.js:2    @logger    ^

SyntaxError: Invalid or unexpected token
    at createScript (vm.js:80:10)
    at Object.runInThisContext (vm.js:152:10)
    at Module._compile (module.js:605:28)
    at Object.Module._extensions..js (module.js:652:10)
    at Module.load (module.js:560:32)
    at tryModuleLoad (module.js:503:12)
    at Function.Module._load (module.js:495:3)
    at Function.Module.runMain (module.js:682:10)
    at startup (bootstrap_node.js:191:16)
    at bootstrap_node.js:613:3
```

So, my questions are:

1. Does Node.js 9.x support decorator syntax? Or is it coming up in some future version?

2. I see some `express-js` based decorators on GitHub, but I am unable to figure out how to create my own decorator. Can someone provide a simple basic example of creating a custom decorator with Node.js?

`node.js` `decorator` `ecmascript-next`

Share

Improve this question

Follow

edited Jan 2, 2018 at 17:15



Andrew Li

57.8k ● 14 ● 134 ● 148

asked Jan 2, 2018 at 4:58



Mopparthy Ravindranath

3,298 ● 8 ● 48 ● 86

3 you can check here [node.green](#) – Manjeet Singh Jan 2, 2018 at 5:38

I've written a short article on how to quickly get started with this - [dev.to/dpkshrma/...](#)  
– Deepak Jun 5, 2020 at 22:38

## 1 Answer

Sorted by: Highest score (default) ▾



23



Decorators are *not* part of ECMAScript 2016 (aka 7). Decorators are currently in [Stage 2 Draft](#) out of the total 4 stages a feature goes through before being finalized and becoming part of the language. They'll probably be integrated into the language in the near future, but its features and specifics are subject to change. Because of this, you'll have to use a transpiler such as Babel to transform the decorators into code the Node runtime can understand (ECMAScript 2016) by installing the [transform-decorators](#) Babel plugin.

As for creating decorators, you've already done so. Each decorator is just a function that wraps another, that is provided with arguments based on the usecase, in your case `target`, `key`, and `descriptor`. Your `logger` function:

```
function logger(target, key, descriptor) {  
  console.log("Cat snarling...");  
  return descriptor;  
}
```

Is already a decorator. For class *properties and methods*, `target` refers to the class of the property, `key` is the property name, and `descriptor` is the descriptor of the property. The decorator is then called and the property of the class is defined via [Object.defineProperty](#) once desugared. Your example can be boiled down to this:

```
class Cat { }  
  
let meowDescriptor = {
```

```

    type: 'method',
    initializer: () => () => {
      console.log(' Meeeoow! ');
    },
    enumerable: false,
    configurable: true,
    writable: true
  }

  function logger(target, key, descriptor) {
    console.log("Cat snarling...");
    return descriptor;
  }

  meowDescriptor = logger(Cat.prototype, 'meow', meowDescriptor);
  Object.defineProperty(Cat.prototype, 'meow', {
    ...meowDescriptor,
    value: meowDescriptor.initializer()
  });

```

For classes themselves, decorators take one argument, `target` which describes the decorated class. I suggest reading [some documentation](#) on the subject to become acquainted with it.

Share

edited Jan 2, 2018 at 5:31

answered Jan 2, 2018 at 5:02

Improve this answer



**Andrew Li**

57.8k ● 14 ● 134 ● 148

Follow

---

9 Three years later, and it's still in stage 2 draft :( I'm writing Node services - I thought here I'd be safe and not need to use a transpiler. – [Snake Verde](#) Apr 12, 2021 at 23:25 ✎

---