

Not quite what you are looking for? You may want to try:

- [Securing Your IoT Deployment](#)
- [Wpf image buttons 100% in xaml](#)



[highlights off](#)

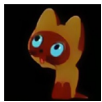
12,505,335 members (64,958 online)

Devendra Katuke ▼ 355 Sign out



[articles](#) [Q&A](#) [forums](#) [lounge](#)

Dotcall - A Callback Hell Remedy



_exebook, 25 Nov 2014

CPOL

Rate:

★★★★★ 5.00 (2 votes)

A syntax sugaring module that will silently convert sync-style dot preceded calls into an async ladder.



Is your email address OK? You are signed up for our newsletters but your email address is either unconfirmed, or has not been reconfirmed in a long time. Please [click here to have a confirmation email sent](#) so we can confirm your email address and start sending you newsletters again. Alternatively, you can [update your subscriptions](#).

Introduction

Ever been to callback hell? Here, I propose another solution. A syntax sugaring called Dotcall. I developed it while working on my various *Node.js* projects because I realized I spend too much time on very simple tasks only because callback syntax is so unfriendly. It is not rocket science, it is just time consuming to get all the curly and less curly braces in place. I post it here because it could be used by someone, maybe you will report issues or even help make it better with your forks or commits. The source is here <https://github.com/exebook/dotcall>.

Theory

The theory behind it is this:

- Most of the time, the result of an asynchronous function is not used, and after it was called, the upper functions also exit immediately. Although sometimes the upper function can do something after it called the first **async(callback)**, usually it does nothing and returns. Sometimes, the asynchronous function could return something useful, but most of the time, it just returns **undefined** and the real stuff is returned with **callback(result)**.
- When the above situation is true, the special syntax sugar can be applied.

There is some discussion of this theory on StackOverflow: <http://stackoverflow.com/questions/27102432/automatic-callback-ladder-creator>

Syntax

The syntax of dotcall is very simple, when the call's brace is preceeded with a dot .(), then the call is converted (hence the name):

Hide Copy Code

```
var a = f.()
console.log(a)
```

Is replaced with:

```
f(function(DOTCALL1) { var a = DOTCALL1
  console.log(a)
})
```

In case your callback uses more than one parameter, there is an extended syntax, called double colon syntax.

```
redis.set('a', 12345)
redis.get('a' :: err, data)
if (err) console.log(err)
console.log(data) // outputs 12345
```

Another case of **double colon** notation is to specify the useful argument number like this:

```
redis.set('a', 12345)
console.log(redis.get('a' :: 2))
```

The above outputs **12345**, because Redis function **get(err, data)** uses **data** as a second parameter.

Implementation

Instead of using complex JavaScript parser, I decided to use a naive approach and simple string manipulation with **.IndexOf**, **.replace**, **.substr** and **RegExp**. So the module is fast and small. The whole source code is under 300 lines, and in fact most of that is comments. Since the approach is naive, of course it will only handle most stright forward situations, but that could cover almost everything I need to talk to DB or do some async file or network operations.

install with npm

```
npm install dotcall
```

About Me

My name is Yakov, I am from Russia. I have been working for many years on Exebook Self-Publisher, which is closed now. Later, I created Deodar file manager and text editor in Node.js+OpenGL. You can find it at <https://github.com/exebook/deodar> under Unlicense license, but it is not localized to English. Now I am working for uteeter.com and ksana.tw. I program mostly in JavaScript.

License

This article, along with any associated source code and files, is licensed under [The Code Project Open License \(CPOL\)](#)

Share

[EMAIL](#)
[TWITTER](#)

About the Author




_exebook

No Biography provided

uTeeter.com

Russian Federation 

Comments and Discussions



Search Comments

-- There are no messages in this forum --

[Permalink](#) | [Advertise](#) | [Privacy](#) | [Terms of Use](#) | Mobile
Web01 | 2.8.160927.1 | Last Updated 25 Nov 2014

Select Language | ▼

Layout: [fixed](#) | [fluid](#)

Article Copyright 2014 by _exebook
Everything else Copyright © [CodeProject](#), 1999-2016