Callback Promise JavaScript (programming language)

What's the difference between a promise and a callback in Javascript?

Answer

8 Answers

Glenn Anderson, Another hairless ape.
Updated Aug 19

If you've done any serious work in JavaScript, you have probably had to face callbacks, nested inside of callbacks, nested inside of callbacks. This is especially true of code written in node.js, since every form of i/o, such as file reads, database reads and writes, and memcache access, is asynchronous, and most code needs a more than a single i/o call. You may end up with code that looks something like this:

```
function isUserTooYoung(id, callback) {
    openDatabase(function(db) {
        getCollection(db, 'users', function(col) {
            find(col, {'id': id},function(result) {
                result.filter(function(user) {
                    callback(user.age < cutoffAge)
                })
            })
        })
    })
}
```

Pretty difficult to follow. And it can get much worse. In our current node.js codebase we sometimes do as many as ten sequential, asynchronous calls. That would be a lot of nesting. Thankfully, there's a much better way: promises.

What is a promise?

A promise is a proxy for a value not necessarily known at its creation time. With promises, rather than an asynchronous call accepting a callback, it instead returns a promise. The calling code can then wait until that promise is fulfilled before executing the next step. To do so, the promise has a method named *then,* which accepts a function that will be invoked when the promise has been fulfilled. As an example, the following is the above code rewritten using promises:

```
function isUserTooYoung(id) {
    return openDatabase()
        .then(getCollection)
        .then(find.bind(null, {'id': id}))
        .then(function(user) {
            return user.age < cutoffAge;
        });
}
```

Much easier to follow, no?

When *then* invokes the specified function, that function receives as a parameter the resolved value of the promise. So, for example, when getCollection is called, a handle to the database will be passed to it.

That's essentially it. If you have any questions, just leave them as comments.

88.6k Views · 249 Upvotes

What are the pros and cons of using promises instead of callbacks?

What are situations to use callbacks instead of promises in Javascript? Should one use promises all the time? Which will lead to more readable...

Are there any advantages to using callbacks instead of promises?

What is a promise in JavaScript?

What are callbacks and why are they so important in Javascript?

Is it worthwhile to learn and use the "promise" pattern in JavaScript while I can use the callback pattern for most problems right now? What a...

What is a callback?

What is callback hell?

Why should one use callback in JavaScript? What are its advantages?

What are the fitting situations to use promises versus callbacks in the context of JavaScript, in particular Node.js?

Related Questions

More Answers Below

- What are the pros and cons of using promises instead of callbacks?
- What are situations to use callbacks instead of promises in Javascript? Should one use promises all the time? Which will lead to more readable...
- Are there any advantages to using callbacks instead of promises?
- What is a promise in JavaScript?
- What are callbacks and why are they so important in Javascript?

John Miller, Web Deveoper, over 15 years working with client side JavaScript
Answered Aug 4 2016

ipt

ed or rejected, so the two are not fundamentally different. The main idea behind Promises is to take callbacks - especially nested callbacks where you want to perform a series of actions, each one after the resolution of the former- and “flatten out” the code required to do this. For example, the below uses JavaScript Promises to load a series of images and perform the desired action after each image load:

```
function fetchImage(url){
    return new Promise(function(resolve, reject){
        var img = new Image()
        img.onload = function(){
            resolve(url)
        }
        img.onerror = function(){
            reject(url)
        }
        img.src = url
    })
}

fetchImage('image1.png').then(function(url){
    console.log(url + ' downloaded!')
    return fetchImage('image2.png')
}).then(function(url){
    console.log(url + ' downloaded!')
    return fetchImage('image3.png')
}).then(function(url){
    console.log(url + ' downloaded!')
    return fetchImage('image4.png')
}).then(function(url){
    console.log(url + ' downloaded!')
})

//Console log:
// image1.png downloaded!
// image2.png downloaded!
// image3.png downloaded!
// image4.png downloaded!
```

Regardless of the number of images I want to load, the level of callback functions used stays at one, which is a lot more maintainable and readable then using callback functions alone. Two excellent guides on JavaScript Promises:

HTML5 rocks JavaScript Promises

Beginners’ Guide to JS Promises

36k Views · 29 Upvotes

Phips Peter, works at Asana
Answered Aug 12 2014

They, along with callbacks, streams, and generators/thunks, provide a way to deal with deferred work in your application. For example, returning the result of query to the client when it comes back from the database. The main advantage of a promise over a callback is that you have an instance of an object that represents that deferred work. Let's say for example that you wanted to do two things in parallel after some asynchronous action. With callbacks, you would have to

What are the pros and cons of using promises instead of callbacks?

What are situations to use callbacks instead of promises in Javascript? Should one use promises all the time? Which will lead to more readable...

Are there any advantages to using callbacks instead of promises?

What is a promise in JavaScript?

What are callbacks and why are they so important in Javascript?

Is it worthwhile to learn and use the "promise" pattern in JavaScript while I can use the callback pattern for most problems right now? What a...

What is callback hell?

Related Questions

Why should one use callback in JavaScript? What are its advantages?

What are the fitting situations to use promises versus callbacks in the context of JavaScript, in particular Node.js?

Given that, I would say the biggest advantage is that it can lead to cleaner simpler code which is easier to test and maintain. As another answer mentioned it's still possible to use callbacks with promises so you should have no problem using external code

42.8k Views · 15 Upvotes

Rubén Berenguel, studied at University of Barcelona
Answered Aug 11, 2014

It's somewhat tricky to answer, specially avoiding mumbo-jumbo. Also, I'm not very used to promises myself in practice, but checked them when they moved to being "relatively" available.

Promises are, in a sense, functional. A promise is a function (which as arguments, takes some more functions) and returns, well, whatever is needed.

Callbacks, aren't that much functional. You request something of a function, and tell it to, when done, follow the callback. Granted, there may be several callbacks to follow several routes.

Conceptually (specially if applied to concrete cases) they seem to work similar. But, no, they are conceptually different:

- In a callback you say "do this, then that"
- In a promise you say "compute this (using that)"

If you are not very used to functions as arguments and a little bit of functional programming this may sound weird. But well, eventually it ticks and you'll see Javascript in a different light ;)

34k Views · 18 Upvotes

Dipak I, .
Answered Apr 20, 2016

Promises are just syntactic sugar. Callback was an old way to handle asynchronous responses. Everything which you can do using callback you can do using promises - But in **more readable and nicer way.**

E.g

Using old age callback,

```
$.ajax("example.php", {
   success: function(object) {},   // callback
   error: function(object, error) {} // callback
});
```

Using promise.

```
$.ajax( "example.php") // returns a promise, which calls the methods once all t
    .done(function () {})
    .fail(function () {});
```

13.8k Views · 10 Upvotes · Answer requested by Multiple

Prashant Singla, studied at Sanatan Dharam Vidya Mandir Panipat
Answered Apr 15, 2015

Promise uses callback.
Promise is just a way of writing asynchronous code in a more synchronous fashion.
I'll try to explain with pseduo code.
Suppose the scenario is you want to get the data from server using ajax call and after when you get data you want to process the data. What you can do is -

```
//create a class http which has get method which returns instance of promise
function http(url) {
  this.get = new Promise(function(resolve, reject) {
    var client = new XMLHttpRequest();
    client.open('get', url);
    client.send();
```

What are the pros and cons of using promises instead of callbacks?

What are situations to use callbacks instead of promises in Javascript? Should one use promises all the time? Which will lead to more readable...

Are there any advantages to using callbacks instead of promises?

What is a promise in JavaScript?

What are callbacks and why are they so important in Javascript?

Is it worthwhile to learn and use the "promise" pattern in JavaScript while I can use the callback pattern for most problems right now? What a...

What is a callback?

What is callback hell?

Why should one use callback in JavaScript? What are its advantages?

What are the fitting situations to use promises versus callbacks in the context of JavaScript, in particular Node.js?

```
     } else {
       reject(this.statusText);
     }
   };
     client.onerror = function () {          reject(this.statusText);        }
  });
}

//using http module
var successCallBack = function(res) {
  console.log(res);
}
new http('/someurl').get().then(successCallBack);
```

18.2k Views · 9 Upvotes

Oodles Technologies, A Leading Web Development Company
Answered Oct 19, 2016

Due to non-blocking I/O, Node is heavy use of callbacks. All the APIs of Nodejs support callbacks.

A callback is a function called at the completion of a given task. This prevents any blocking, and allows other code to be run in the meantime.

For example - Let there are four functions functionCall, doSomeworkOne, doSomeworkTwo, doSomeworkTwo and they are performing some IO tasks.
function doSomeworkThree functionCall depends doSomeworkOne, doSomeworkOne depends doSomeworkTwo, doSomeworkTwo depends doSomeworkThree. To make these sync, callback function passed as parameter in all functions.

Visit here for the complete article : Promises VS Callback in Nodejs

Thanks,
Oodles Technologies

5.5k Views

Marek Pasieczny, works at Applandeo
Answered Oct 6, 2016

I don’t think that the clue is what is the difference. More important is what to use or what not to use. JS Developers always refer callbacks as a hell and try to avoid it in some ways like using Bluebird promises or Async.js (it was standard for handling callbacks before Promises went viral). Being honest I can recommend using of Promises and do not think about callbacks. Your life will be easier :)

As a some additional source to make comparision between callbacks and promises I can redirect you to post about avoiding callbacks:

Avoid callbacks in Node.js with Bluebird promises

4.7k Views

Related Questions

Is it worthwhile to learn and use the "promise" pattern in JavaScript while I can use the callback pattern for most problems right now? What a...

What is a callback?

What is callback hell?

Why should one use callback in JavaScript? What are its advantages?

What are the fitting situations to use promises versus callbacks in the context of JavaScript, in particular Node.js?

What is the difference between callback function and promises in JS and in which case do we have to use these?

What is a callback function in JS or Android/Windows Phone 8 programming?

What is callback code in C?

How do you avoid "callback hell" in Golang?

How do you replace "Callback Hells" with JavaScript ES6 generators?

What are the pros and cons of using promises instead of callbacks?

What are situations to use callbacks instead of promises in Javascript? Should one use promises all the time? Which will lead to more readable...

Are there any advantages to using callbacks instead of promises?

What is a promise in JavaScript?

What are callbacks and why are they so important in Javascript?

Is it worthwhile to learn and use the "promise" pattern in JavaScript while I can use the callback pattern for most problems right now? What a...

What is a callback?

What is callback hell?

Why should one use callback in JavaScript? What are its advantages?

What are the fitting situations to use promises versus callbacks in the context of JavaScript, in particular Node.js?

What are the pros and cons of using promises instead of callbacks?

What are situations to use callbacks instead of promises in Javascript? Should one use promises all the time? Which will lead to more readable...

Are there any advantages to using callbacks instead of promises?

What is a promise in JavaScript?

What are callbacks and why are they so important in Javascript?

Is it worthwhile to learn and use the "promise" pattern in JavaScript while I can use the callback pattern for most problems right now? What a...

What is a callback?

What is callback hell?

Why should one use callback in JavaScript? What are its advantages?

What are the fitting situations to use promises versus callbacks in the context of JavaScript, in particular Node.js?