

- Create a WCF DataService in Visual Studio 2015
- Story of Equality in .Net - Part 6



CODE PROJECT®
For those who code

articles quick answers discussions community help

Articles » Web Development » Client side scripting » General



Make Me a Promise



Keyhole Software, 20 Oct 2014

CPOL

Rate: 

★★★★★ 5.00 (1 vote)

Make me a promise



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](#).

If you haven't worked with JavaScript in over a decade, you might be surprised at the advances that have been made in this language. At one time, JavaScript was used mainly for swapping "one image for another in response to user-generated mouse events" [1], and now, it is a full-fledged language used for both front end and back end work.

Since JavaScript is most often used on the web, we have to deal with asynchronous patterns often in daily use. Traditionally, if you have a bunch of asynchronous calls that need to be processed in order, you'd get something that looked like this:

[Hide](#) [Copy Code](#)

```
step1(function(value1) {
  step2(value1, function(value2) {
    step3(value2, function(value3) {
      step4(value3, function(value4) {
        // Do something with value4
      });
    });
  });
});
```

This is called the "Pyramid of Doom" by some, a situation where your code marches to the right faster than it marches forward.[2]

So, how do we harness the power of callbacks without the confusing mess of nested functions? Promises.

What Are Promises?

A promise represents the eventual result (either a value or a thrown exception) of an asynchronous operation.

The main way in which you interact with a promise is through its **then** method, which registers a callback to receive either the promise's eventual value or the reason why the promise cannot be fulfilled [3].

There are a number of different implementations of a promise framework in JavaScript. Some of the more well-known ones are jQuery's Deferred Object [4] and Q by Kris Kowal [2]. Limited support for promises is included in ECMAScript 6 (still a work in progress) and full support for them is expected in ECMAScript 7 (also, a work in progress) [5].

Tutorial

I'm most familiar with the AngularJS implementation of promises, which is a subset of the functionality included with Q [6], so I'll give some examples on how you might use promises in your AngularJS project.

[Hide](#) [Copy Code](#)

```
// for the purpose of this example let's assume that variable `$q`  
// is available in the current lexical scope (it could have been injected or passed in).  
function step1() {  
    var deferred = $q.defer();  
    longRunningCallWithCallback(function(error, response) {  
        if (error) {  
            deferred.reject(error);  
        } else {  
            deferred.resolve(response);  
        }  
    });  
    return deferred.promise;  
}  
  
var resolvedValue;  
step1().then(function(value) {  
    resolvedValue = value;  
}, function(reason) {  
    $log.error(reason)  
});
```

The above example shows how you might convert an existing function you rely on that has a callback function into a promise returning function.

First, we create a **Deferred** object that represents the task that will finish in the future. Then, inside our existing function's callback method, we either reject the promise when the callback returns an error or we resolve the promise with the returned value. In the last line of the function, we return the promise so it can be used by the callee to get access to the result of the deferred task when it completes.

[Hide](#) [Copy Code](#)

```
function eachItemInALoopCallsALongProcess() {  
    var promises = [];  
    angular.forEach(items, function(item) {  
        var deferred = $q.defer();  
        promises.push(deferred.promise);  
        longRunningFunction(item, function(error, response) {  
            if (error) {  
                deferred.reject(error);  
            } else {  
                deferred.resolve(response);  
            }  
        });  
    });  
    return $q.all(promises);  
}
```

This example shows a case where we have a bunch of items that we want to process that don't rely on each other, but we do want them to all complete before continuing. The **\$q.all** is the magic piece. It combines all of the promises in the array into a single promise that doesn't resolve until all of the promises in the array are resolved. If any of the promises are rejected, then the combined promise will also be rejected with that same rejection reason.

Conclusion

Hopefully I've given you a glimpse into the world of JavaScript promises and that you'll give them a shot when you come across a mess of callback in the future.

CodeProject

In addition to the References listed below, there are also some links in the Resources section with more detailed information and examples of using promises across a wide variety of situations.

References

1. http://www.oreillynet.com/pub/a/javascript/2001/04/06/js_history.html
2. <http://documentup.com/kriskowal/q>
3. <https://promisesaplus.com/>

4. <https://api.jquery.com/category/deferred-object/>
5. <https://en.wikipedia.org/wiki/ECMAScript>
6. [https://docs.angularjs.org/api/ng/service/\\$q](https://docs.angularjs.org/api/ng/service/$q)

Resources

- <http://keyholesoftware.com/2014/07/23/javascript-promises-are-cool/>
- <http://blogs.msdn.com/b/ie/archive/2011/09/11/asynchronous-programming-in-javascript-with-promises.aspx>
- https://en.wikipedia.org/wiki/Futures_and_promises
- <https://github.com/kriskowal/q/wiki/General-Promise-Resources>
- <http://wiki.commonjs.org/wiki/Promises>

License

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

Share



About the Author



Keyhole Software



Keyhole Software
United States

Keyhole is a software development and consulting firm with a tight-knit technical team. We work primarily with Java, .NET, and Mobile technologies, specializing in application development. We love the challenge that comes in consulting and blog often regarding some of the technical situations and technologies we face. Kansas City, St. Louis and Chicago.

Group type: Organisation

3 members



Comments and Discussions

Add a Comment or Question

Search Comments

Profile popups Spacing Relaxed ▾ Layout Open All ▾ Per page 25 ▾ Update

-- There are no messages in this forum --

