

GitHub

This repository [Explore](#) [Features](#) [Enterprise](#) [Pricing](#)[Sign up](#)[Sign in](#)[rwaldron](#) / [jquery-hive](#)[Watch](#) 14[★ Star](#) 196[Fork](#) 18[Code](#)[Issues](#) 5[Pull requests](#) 1[Pulse](#)[Graphs](#)

jQuery Plugin for creating and managing web workers across implementations. Includes Hive.Pollen.js - the thread-safe utility library for connecting worker threads to the Hive. <http://weblog.bocoup.com/javascript-web-workers-from-basics-to-jquery-hive-part-i>

[39 commits](#)[3 branches](#)[0 releases](#)[1 contributor](#)[Branch: master](#)[New pull request](#)[New file](#)[Find file](#)[HTTPS](#)<https://github.com/rwaldron/>[Download ZIP](#)[rwaldron](#) Merge pull request #10 from rwdrn/9 ...

Latest commit 375d861 on 16 Dec 2011

[demo](#)

Added: demos, unit tests; Updated: native complex object messaging logic

6 years ago

[kit](#)

Added: demos, unit tests; Updated: native complex object messaging logic

6 years ago

[qunit](#)

Added: demos, unit tests; Updated: native complex object messaging logic

6 years ago

[test](#)

Adds support for interchangeable args to inArray. Fixes #4

5 years ago

[README.markdown](#)

Readme updates

6 years ago

[changelog](#)

Adding changelog; marks v0.2.0

5 years ago

[jquery.hive.js](#)

Explicit thread.terminate() when worker objects are destroyed. Fixes #2

5 years ago

[jquery.hive.pollen.js](#)

Fixes #9. Removes 2 year old browser sniffing hack

4 years ago

[README.markdown](#)

Enter The jQuery.Hive

- Simplify the client/main page worker setup API
- Wrap Worker constructor and functions in syntax that jQuery developers are familiar with
- Normalize cross-implementation inconsistencies; message serialization/deserialization
- Worker-to-Worker Direct Messaging
- Worker memoization

jQuery.Hive.js API

Feed Workers With Pollen

- Provides a light weight, reusable library for:
- AJAX, Worker-to-Worker Direct Messaging, Worker memoization
- Object, Array and String Manipulation
- Query JSON objects with JSONPath
- Variable evaluation and logic control flow utilities
- Syntax that jQuery developers will recognize and understand

jQuery.Hive.Pollen.js API

Basic Client Setup

Assumes \geq jQuery 1.4 and jQuery.Hive.js are loaded

```

$(function () {

    $.Hive.create({

        worker: 'worker.js',
        receive: function (data) {

            console.group('RECEIVED MESSAGE - WORKER: #' + data._from_);
            console.log( data );
            console.groupEnd();

            /*
            -----
            Possible uses:

            Populate a massive data table...

            Update a browser based IM client

            Update a feed reader app ( 1-to-1 worker to feed?)

            Handle audio data from typed array

            Handle pixel array
            -----
            */

        },
        created: function (hive) {

            /*
            -----
            Possible uses:

            Impress the hell out of your friends by
            executing code after all the workers are created
            -----
            */

        }
    });

    /*
    -----
    This contradicts what I noted above, but it's for
    illustration purposes, so I'm ok with that.
    -----
    */

    $( $.Hive.get(1) ).send({

        "message" : {
            "a" : "a-value",
            "b" : "b-value",
            "c" : "c-value"
        }
    });

    /*
    -----
    Alternative syntax
    -----
    */
    $.Hive.get(1).send({

        "message" : {
            "a" : "a-value",
            "b" : "b-value",
            "c" : "c-value"
        }
    });

```

```

    });

    /**
     * -----
     * Specify an additional callback
     * -----
     */
    $.Hive.get(1).send({

        "message" : {
            "a" : "a-value",
            "b" : "b-value",
            "c" : "c-value"
        }

    }, function (data) {

        console.log('This is from a task specific message receipt callback');

    });

});

```

Basic Worker Example

```

importScripts('jquery.hive.pollen.js');

$(function (data) {

    // `this` equals WorkerGlobalScope

    $.ajax.get({
        url: 'get-data-from-the-server.php',
        dataType: 'json',
        data: $.param(data.message),
        success: function (jsonObj) {

            // Assume its a list of companies with some contact data.

            $.send(
                $.unique(
                    $.filter(jsonObj, function (obj) {

                        // If not passed in the data property above, we could filter here.
                        // Not the most efficient way, the example is really to illustrate Pollen's syntax

                        if ( $.inStr(obj.company, data.company) ) {
                            return true;
                        }
                        return false;
                    })
                )
            );

            // OR...

            $.send(
                $.query(

                    // Get filtered data with a JSONPath query
                    "?company='"+data.company+"'",
                    jsonObj

                )
            );
        }
    });
});

```

```
});
```

