Modules/1.0

From CommonJS Spec Wiki
< Modules

STATUS: SUPERSEDED BY 1.1

Implementations
Flusspferd, GLUEscript, GPSEE, JSBuild, Narwhal (0.1), Persevere, RingoJS, SproutCore 1.1/Tiki, node.js, TeaJS (formerly v8cgi), CouchDB, Smart Platform, Yabble, Wakanda, XULJet

This specification addresses how modules should be written in order to be interoperable among a class of module systems that can be both client and server side, secure or insecure, implemented today or supported by future systems with syntax extensions. These modules are offered privacy of their top scope, facility for importing singleton objects from other modules, and exporting their own API. By implication, this specification defines the minimum features that a module system must provide in order to support interoperable modules.

Contents

Contract

Module Context

1. In a module, there is a free variable "require", that is a function.
   1. The "require" function accepts a module identifier.
   2. "require" returns the exported API of the foreign module.
   3. If there is a dependency cycle, the foreign module may not have finished executing at the time it is required by one of its transitive dependencies; in this case, the object returned by "require" must contain at least the exports that the foreign module has prepared before the call to require that led to the current module's execution.
   4. If the requested module cannot be returned, "require" must throw an error.

2. In a module, there is a free variable called "exports", that is an object that the module may add its API to as it executes.
3. modules must use the "exports" object as the only means of exporting.

Module Identifiers

1. A module identifier is a String of "terms" delimited by forward slashes.
2. A term must be a camelCase identifier, ".", or "..".
3. Module identifiers may not have file-name extensions like ".js".
4. Module identifiers may be "relative" or "top-level". A module identifier is "relative" if the first term is "." or "..".
5. Top-level identifiers are resolved off the conceptual module name space root.
6. Relative identifiers are resolved relative to the identifier of the module in which "require" is written and called.

Unspecified

This specification leaves the following important points of interoperability unspecified:

1. Whether modules are stored with a database, file system, or factory functions, or are interchangeable with link libraries.
2. Whether a PATH is supported by the module loader for resolving module identifiers.

Unit Tests

- Unit Tests at Google Code (http://code.google.com/p/interoperablejs/) by Kris Kowal
- Unit Tests Git Mirror (http://github.com/ashb/interoperablejs/tree/master) by Ash Berlin

Sample Code

math.js

```
exports.add = function() {
    var sum = 0, i = 0, args = arguments, l = args.length;
    while (i < l) {
        sum += args[i++];
    }
    return sum;
};
```

increment.js

```
var add = require('math').add;
exports.increment = function(val) {
    return add(val, 1);
};
```

program.js

```
var inc = require('increment').increment;
var a = 1;
inc(a); // 2
```

Notes

- Secure Modules
- Notes on compiling modules for browsers
- On writing modules that also work as <script>s

Amendment Proposals

1. Module Loaders
2. Module Metaobject

Implementations

Passing all compliance tests:

- Flusspferd (Spidermonkey / C++) http://flusspferd.org/
- Ondrej Zara's v8cgi for V8 http://code.google.com/p/v8cgi/
- Tom Robinson's narwhal for Rhino http://github.com/tlrobinson/narwhal/
- Wes Garland's GPSEE for Spidermonkey http://code.google.com/p/gpsee/
- Kris Kowal's chiron for browsers http://github.com/kriskowal/chiron/
- Hannes Wallnoefer's RingoJS for Rhino at http://github.com/ringo/ringojs
- Ben Collver's Trogl for v8 at http://bencollver.googlepages.com/trogl-1.zip
- Atul Varma's Pydertron for Python/Spidermonkey at http://hg.toolness.com/pydertron/raw-file/tip/docs.html
- Daniel Friesen's MonkeyScript Lite (http://lite.monkeyscript.org) for Rhino
- Irakli Gozalishvili's narwhal for XULRunner http://github.com/gozala/narwhal-xulrunner/
- James Brantly's Yabble for browsers http://github.com/jbrantly/yabble
- 4D's Wakanda (JavaScriptCore/SquirrelFish) http://www.wakandasoft.com

Fully implemented:

- Kris Zyp's Persevere for Rhino at http://www.persvr.org/

In development:

- nathan smith is implementing this for JScript and ASP at http://github.com/smith/interoperablejscript/tree/master
- mob is implementing this in Ejscript http://www.ejscript.org
- pmuellr posted a sample loader http://wiki.github.com/pmuellr/modjewel

Related Documents

- Proposal to ECMA TC39: Module System for ES-Harmony (http://docs.google.com/Doc?

id=dfgxb7gk_34gpk37z9v&hl=en)
- Presentation to ECMA TC39: Modules (http://docs.google.com/Presentation?docid=dcd8d5dk_0cs639jg8&hl=en)

Retrieved from "http://wiki.commonjs.org/index.php?title=Modules/1.0&oldid=5872"
Category: Spec

- This page was last modified on 17 July 2012, at 23:15.
- This page has been accessed 35,827 times.