
Ajax IM

What is Ajax IM?

Ajax IM (“Ajax Instant Messenger”) is an instant messenger for the web. It is intended to be integrated with existing websites to provide a more interactive community experience.

Features

- Instant communication between connected users
- Robust backend server built on Node.js
- Can be integrated with virtually any existing website
- Seamless authentication
- Works across an entire website, unobtrusively
- Automatic state management between page reloads
- User state and statuses
- and *much* more!

Installation

Install `Node.js`:

```
1 wget http://nodejs.org/dist/v0.8.9/node-v0.8.9-linux-x86.tar.gz
2 tar xzf node-v0.8.9-linux-x86.tar.gz
3 cd node-v0.8.9
4 ./configure
5 make
6 make install
```

Install Node Package Manager (`npm`):

```
1 See instructions at http://github.com/isaacs/npm.
```

Install `Express.js`, `Connect` and `Cookie`:

```
1 npm install express
2 npm install connect
3 npm install cookie
```

Compile the daemon add-on if you plan on letting the server daemonize itself:

```
1 cd server/libs/daemon
2 node-waf configure build
```

```
3 cp build/default/daemon.node .
4 rm -rf build
```

Installation for Development

If you want to test Ajax IM as a standalone app for development, you will need to install [Jade](#) as well.

To install [Jade](#):

```
1 npm install jade
```

Starting up the server

Starting the server in *development* mode is as simple as:

```
1 node server/app.js
```

To start the server in *production* mode:

```
1 EXPRESS_ENV=production node server/app.js
```

To start the server as a *daemon* in *production* mode:

```
1 node server/app.js start production
```

Testing it out

Once the server is up and running in *development* mode, you can test it out through the included development testing app. The below instructions are assuming that you have left all default configuration options. If not, please replace the host/port values with the correct ones.

To get started, first initialize a session cookie by going to:

```
1 http://localhost:8000/dev/cookie
```

Then head over to the development page that will initialize the client:

```
1 http://localhost:8000/dev/
```

That's it!

More Information

- Follow endtwist on twitter for updates
- Google Group for discussion
- GitHub Wiki for guidance

Node Compatibility

The `master` branch of Ajax IM is compatible with node –version:

```
1  v0.8.9
```

Contributing

Pull requests are being accepted! If you would like to contribute, simply fork the project and make your changes.

Style Guide

If you intend on contributing, please follow this style guide when submitting patches or commits. Submissions that do not follow these guidelines will not be accepted.

- Use 4 space indents (not tabs!)
- No trailing whitespace
- Blank line at the end of files
- Semi-colons at the ends of lines, where appropriate
- Keep lines to 80 characters or less
- Never bump the version

No whitespace between keys and values:

```
1  {foo: 'bar'}
2  // good
3
4  {foo : 'bar'}
5  // bad
```

Hash formatting:

```
1  {foo: 'bar', baz: 'taz'}
2  // good
3
```

```
4 {
5     foo: 'bar',
6     baz: 'taz',
7     moo: 'cow'
8 }
9 // good
10
11 { foo: 'bar', baz: 'taz' }
12 // bad
13
14 {foo: 'bar',
15  baz: 'taz',
16  moo: 'cow'}
17 // bad
```

Chained methods:

```
1 str
2   .strip
3   .replace(...)
4   .replace(...)
5 // good
6
7 str
8 .strip
9 .replace(...)
10 .replace(...)
11 // bad
12
13 str.
14     strip.
15     replace(...).
16     replace(...)
17 // bad
```

Single quotes over double quotes, unless double quotes make sense:

```
1 'hello'
2 // good
3
4 "what's up?"
5 // good
6
7 "hello"
8 // bad
```

Ternary expressions are fine, but cannot be nested and must be formatted as:

```
1 foo = (a ? b : c);
2 // good
3
```

```
4 foo = (something.hasAPropertyLikeThis == 'some other value'
5       ? 'result one'
6       : 'result two'
7       );
8 // good
9
10 foo = something.hasAPropertyLikeThis === 'some other value' ?
11     'result one' :
12     'result two'
13 // bad
```

Use braces for conditionals, unless conditionals are single statements:

```
1 if(foo) {
2     bar();
3     baz();
4 }
5 // good
6
7 if(foo) bar();
8 else baz();
9 // good
10
11 if(foo)
12     bar(),
13     baz()
14 // bad
```

Closures:

```
1 function() {
2 }
3 // good
4
5 function () {
6 }
7 // bad
8
9 function()
10 {
11 }
12 // bad
```

Methods:

```
1 foo.bar = function() {
2 }
3 // good
4
5 foo.bar = function (){
6 }
```

```
7 // bad
8
9 foo.bar = function()
10 {
11 }
12 // bad
```

License

(The MIT License)

Copyright (c) 2012 [Daniel Howard] (<http://www.svexpertise.com>)

Copyright (c) 2010 Joshua Gross

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.