

Node.js in real life

Show&Tell Presentation

By Tanya Nam

August 8th, 2012

What is node.js

- Node.js is a platform for building fast, scalable network applications
- Uses event-driven, asynchronous I/O to minimize overhead and maximize scalability
- Great for data-intensive, real-time applications
- Developed by Joyent in 2009

Runs on V8



Actually, Google V8

Client: Chrome

Server: node.js



Event-oriented vs threaded

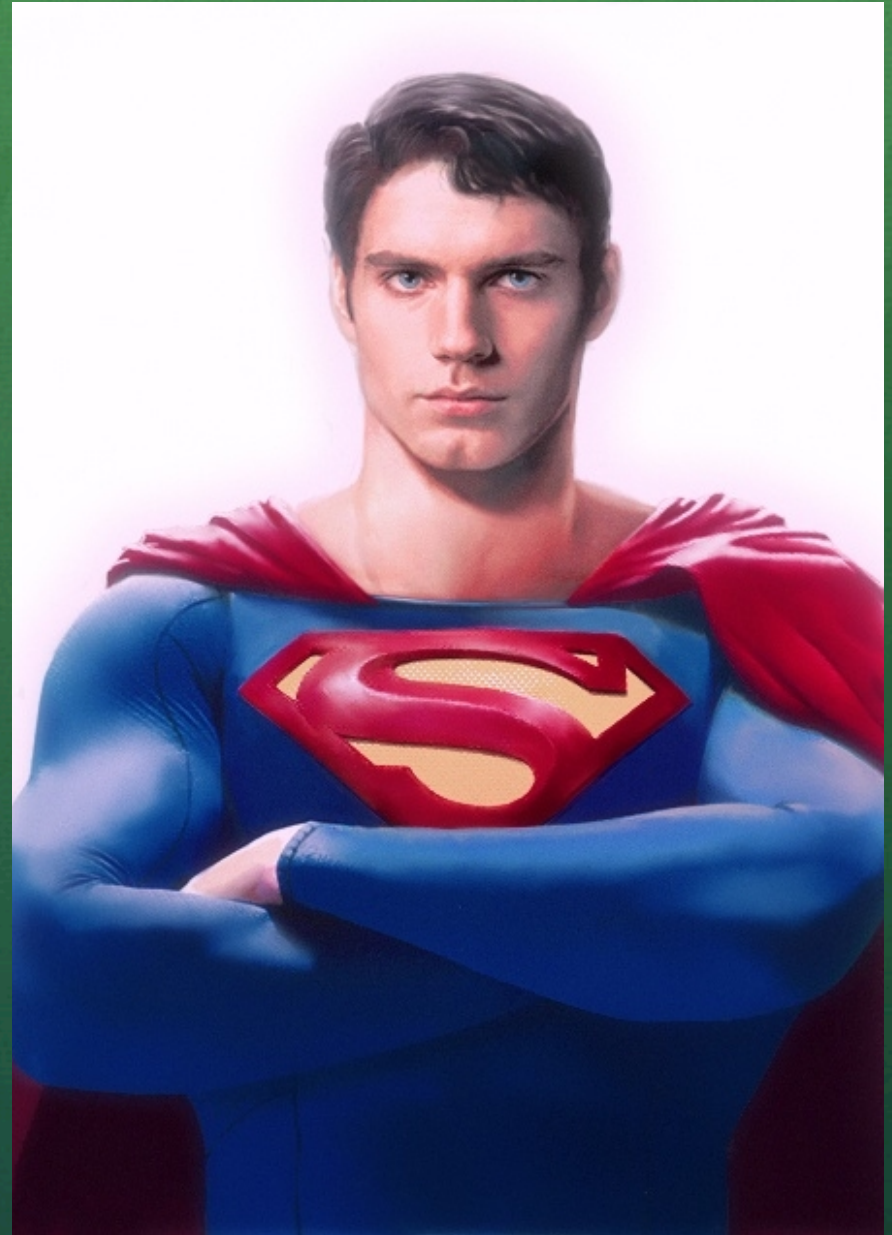
Explained with bunnies

```
// blocking I/O + threads
var urls = db.query("select * from urls"); // wait
urls.each(function (url) {
  var page = http.get(url); // wait
  save(page); // wait
});

// non-blocking I/O + event loop
db.query("select * from urls", function (urls) {
  urls.each(function (url) {
    http.get(url, function (page) {
      save(page);
    });
  });
});
});
```

It's cool because:

- Can handle lots of requests at the same time
- JavaScript at its core
- Vibrant community- lots of modules already written
- NPM - makes it easy to install modules



Who uses it

- Walmart mobile app
- Mockingbird – web app for creating wireframes
- Klout
- Storify
- Rdio: streaming audio
- Yammer
- Scrabbly: multiplayer online Scrabble game

Simple web server

```
var http = require("http");

http.createServer(function(request, response)
{
    response.writeHead(200, {"Content-Type":
"text/plain"});
    response.write("Hello World");
    response.end();
}).listen(3000);
```

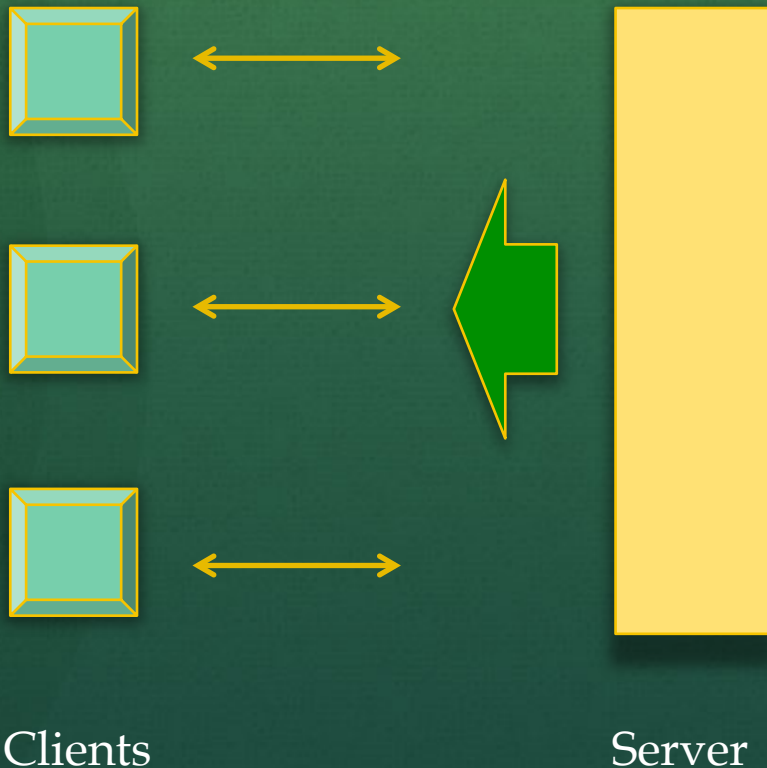

Typical architecture



- Additional modules:
 - mongodb
 - jade
 - oauth
 - socket.IO
- App framework:
 - routing
 - request handling
- Middleware:
 - session handling
 - logging
 - error reporting
 - form data processing
- HTTP server

Chat app using socket.IO

WebSocket is a web technology providing for bi-directional, full-duplex communications channels over a single TCP connection.



When client connects:

- Add to list of users
- Notify all that user entered

When client sends message:

- Display message for all users

When client disconnects:

- Remove from list of users
- Notify all that user disconnected

Examples

Example: multi-user chat

Server: node.js

Modules: express, socket.IO

Repo on git: <https://github.com/ntanya/chat-show-tell.git>

Deployed on Heroku: <http://chat-show-tell.herokuapp.com/>

Example: Twitter trend monitor

Server: node.js

Database: MongoDB hosted on MongoLab

Modules: connect, express, mongodb, jade, cron, oauth

Repo on git: <https://github.com/ntanya/datacoll> (private repo)

Deployed on Heroku: instantag.com

Additional resources

<http://nodejs.org/jsconf.pdf>

<http://mashable.com/2011/03/10/node-js/>

<http://expressjs.com/>

<http://howtonode.org/>

<http://www.nodebeginner.org/>

<http://dailyjs.com/>

Hosting: Heroku, Nodejitsu