AngularJS

Node.JS introduction

LUXOFT

LXFT LISTED NYSE

What is Node.JS?

- v8 JavaScript runtime
- Event driven
- Non-blocking standard libraries
- Most APIs speak streams
- Provides a package manager and module system

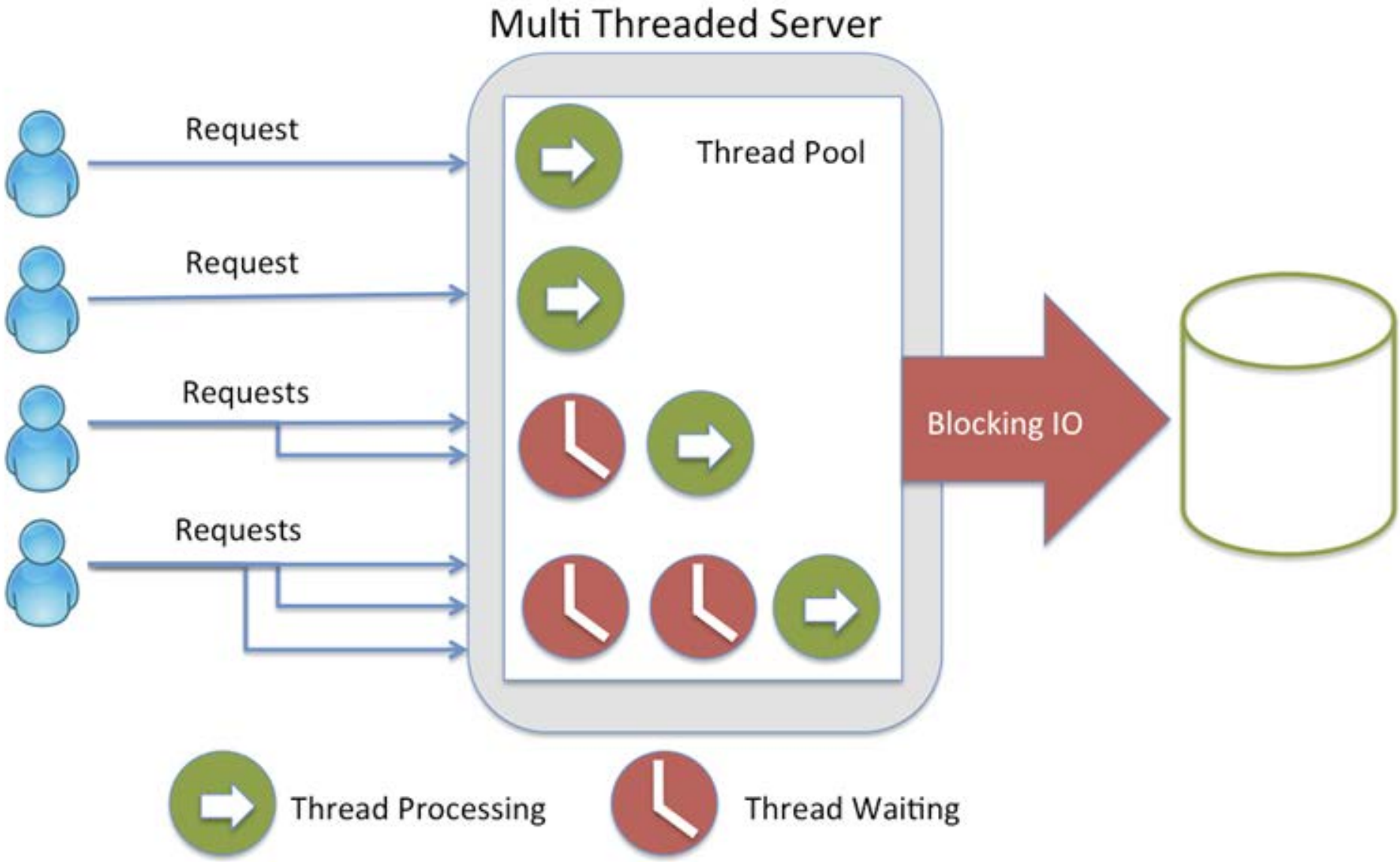
JavaScript everywhere

- Code reuse
- Same programming culture on client and server
- Lots of JavaScript programmers

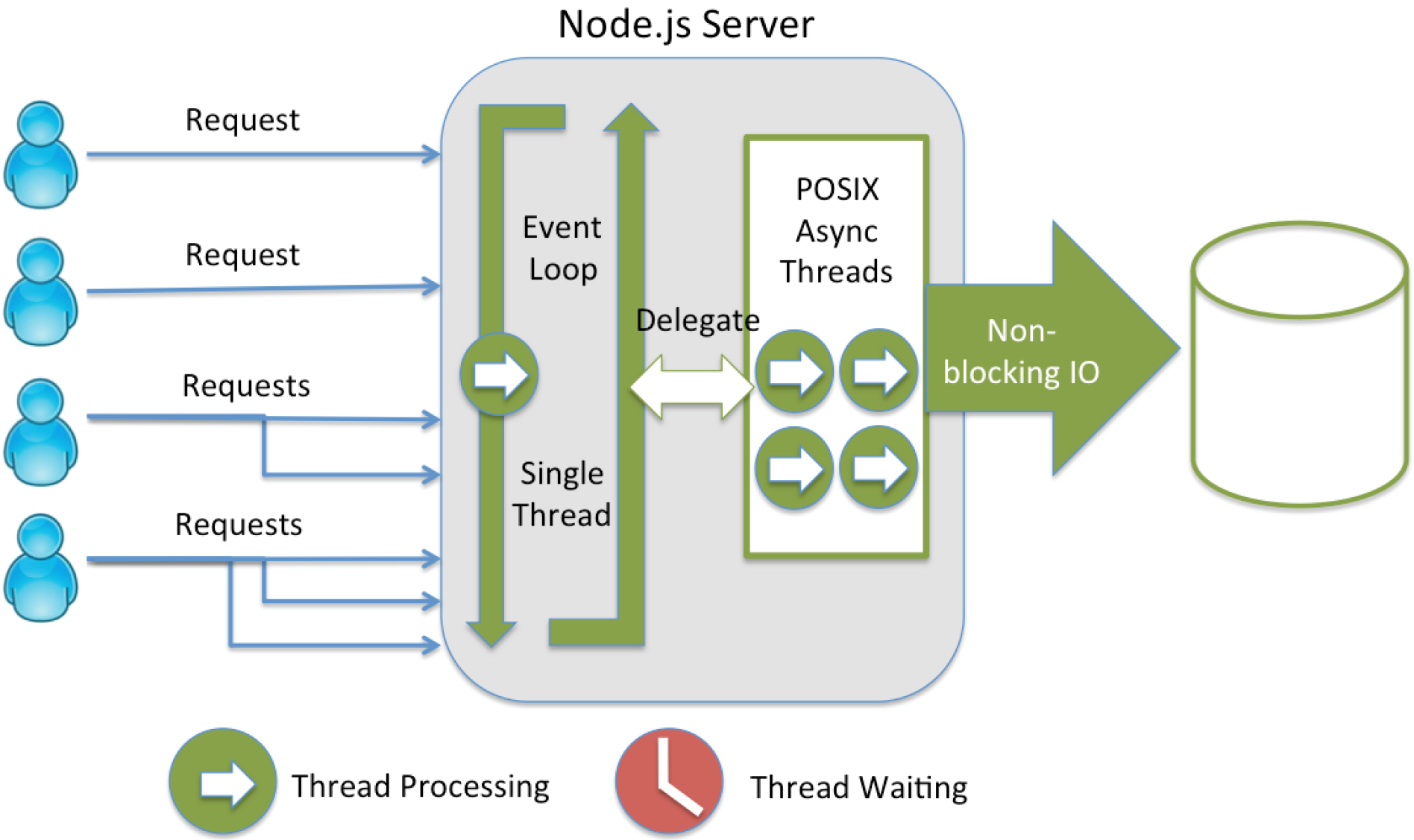
Node.JS standrard libraries

- Asynchronous by default
- Low level
- HTTP is first class citizen

Multithreaded server request processing



Node.js request processing with event loop



NPM – Node Package Manager

- Handles and resolves dependencies
- More than 300 000 modules available
- package.json – dependencies definition

Routes with Express

- Respond to HTTP requests with a callback
- Supports variable placement in routes
- Easy to serve JSON

Example

```
app.get("/greeting", function(req,res) {
        res.send("Hello, "+req.query.name+"!");
});
```

get and post request

```
var express = require('express');
var app = express();

app.get('/hello', function(req, res){
    res.send("Hello World!");
});

app.post('/hello', function(req, res){
    res.send("You just called the post method at '/hello'!\n");
});

app.all('/test', function(req, res){
    res.send("HTTP method doesn't have any effect on this route!");
});

app.listen(3000);
```

HTTP methods:

- GET
- POST
- PUT
- DELETE

parameters in request url

```
var express = require('express');
var app = express();

app.get('/:id', function(req, res){
    res.send('The id you specified is ' + req.params.id);
});
app.listen(3000);
```

http://localhost:3000/123

The id you specified is 123

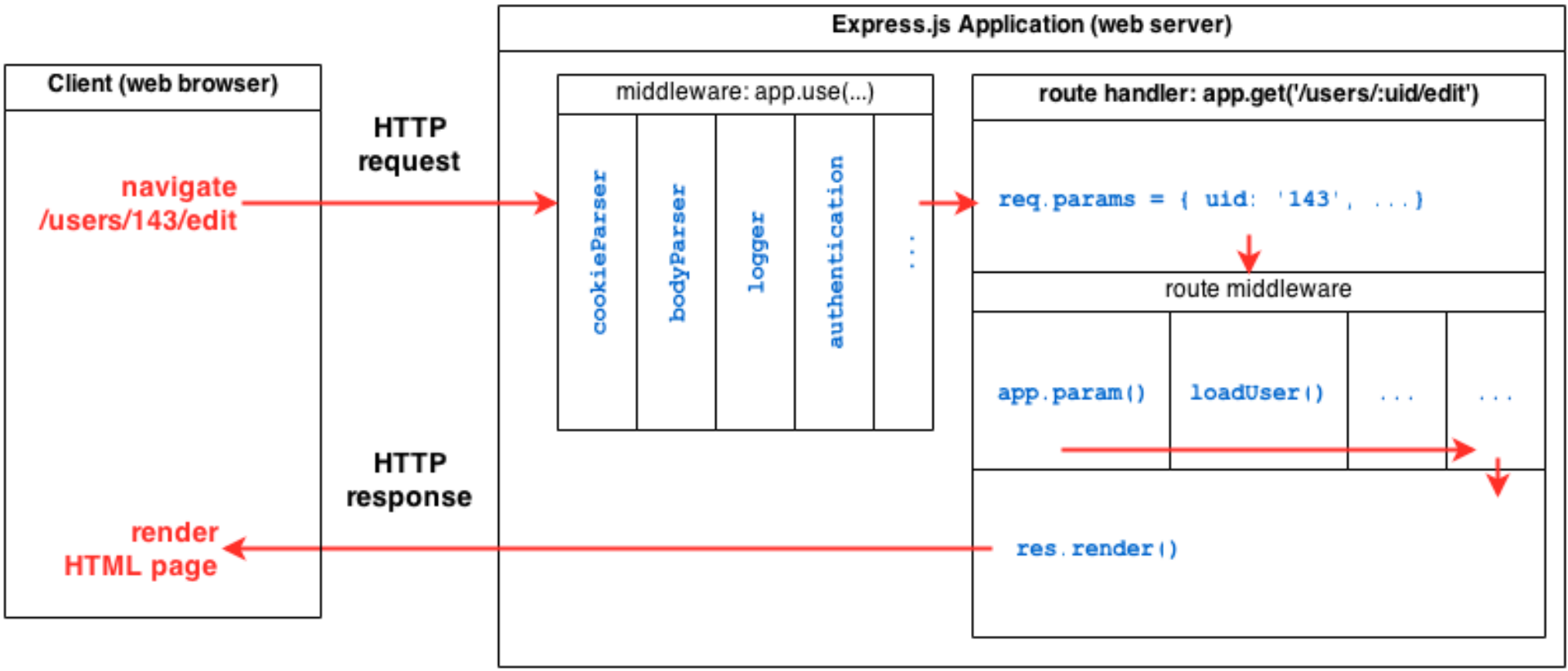
request query

```
// GET /search?q=paul+mccartney
req.query.q
// => "paul mccartney"

// GET /shoes?order=desc&shoe[color]=blue&shoe[type]=converse
req.query.order
// => "desc"

req.query.shoe.color
// => "blue"

req.query.shoe.type
// => "converse"
```

response methods

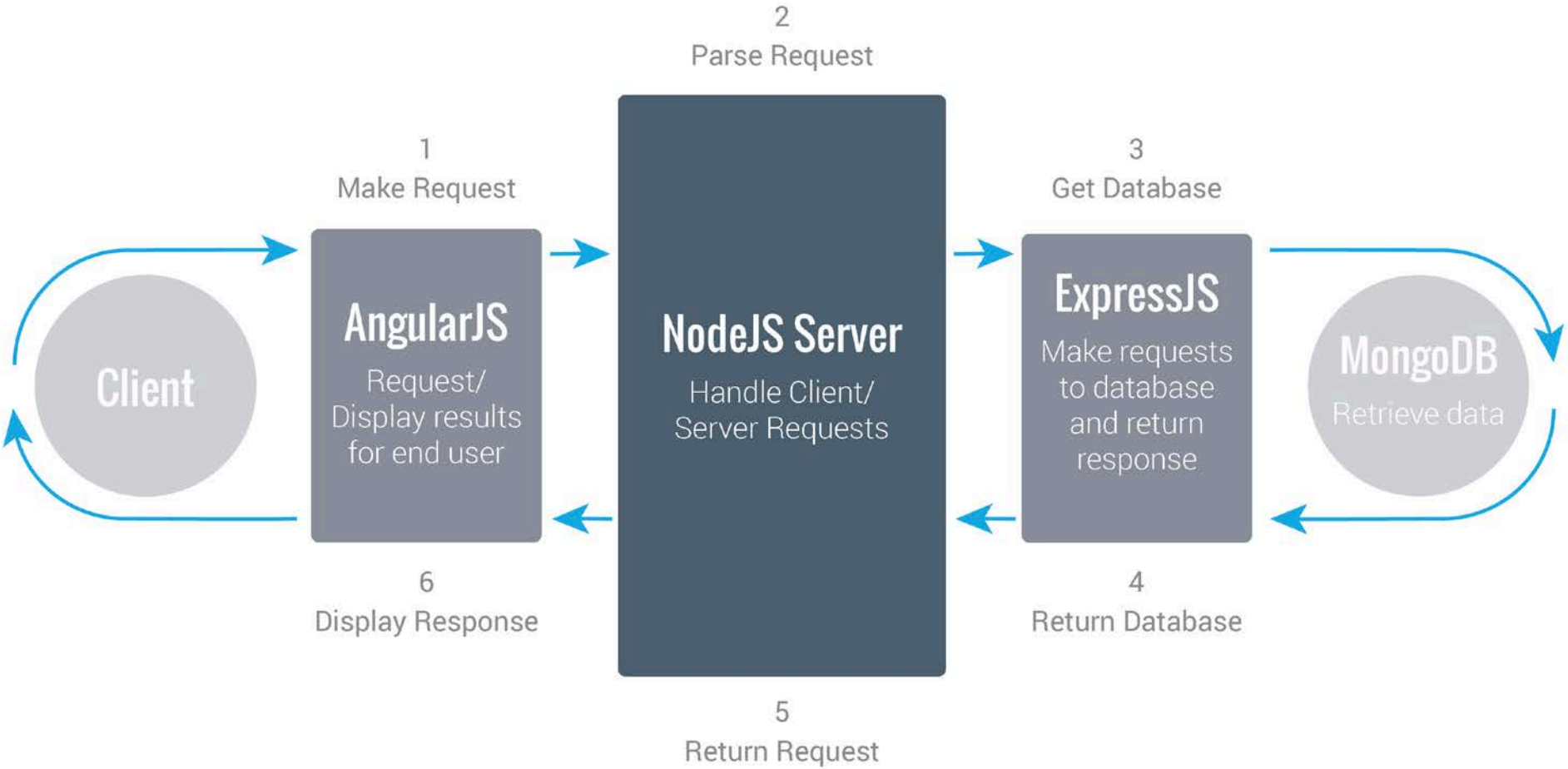
Method	Description
res.download()	Prompt a file to be downloaded: res.download('/report-12345.pdf');
res.end()	End the response process.
res.json()	Send a JSON response.
res.jsonp()	Send a JSON response with JSONP support.
res.redirect()	Redirect a request.
res.render()	Render a view template: res.render('user', { name: 'Tobi' }, function(err, html) { … });
res.send()	Send a response of various types.
res.sendFile()	Send a file as an octet stream.
res.sendStatus()	Set the response status code and send its string representation as the response body.

NodeJS + Express application structure

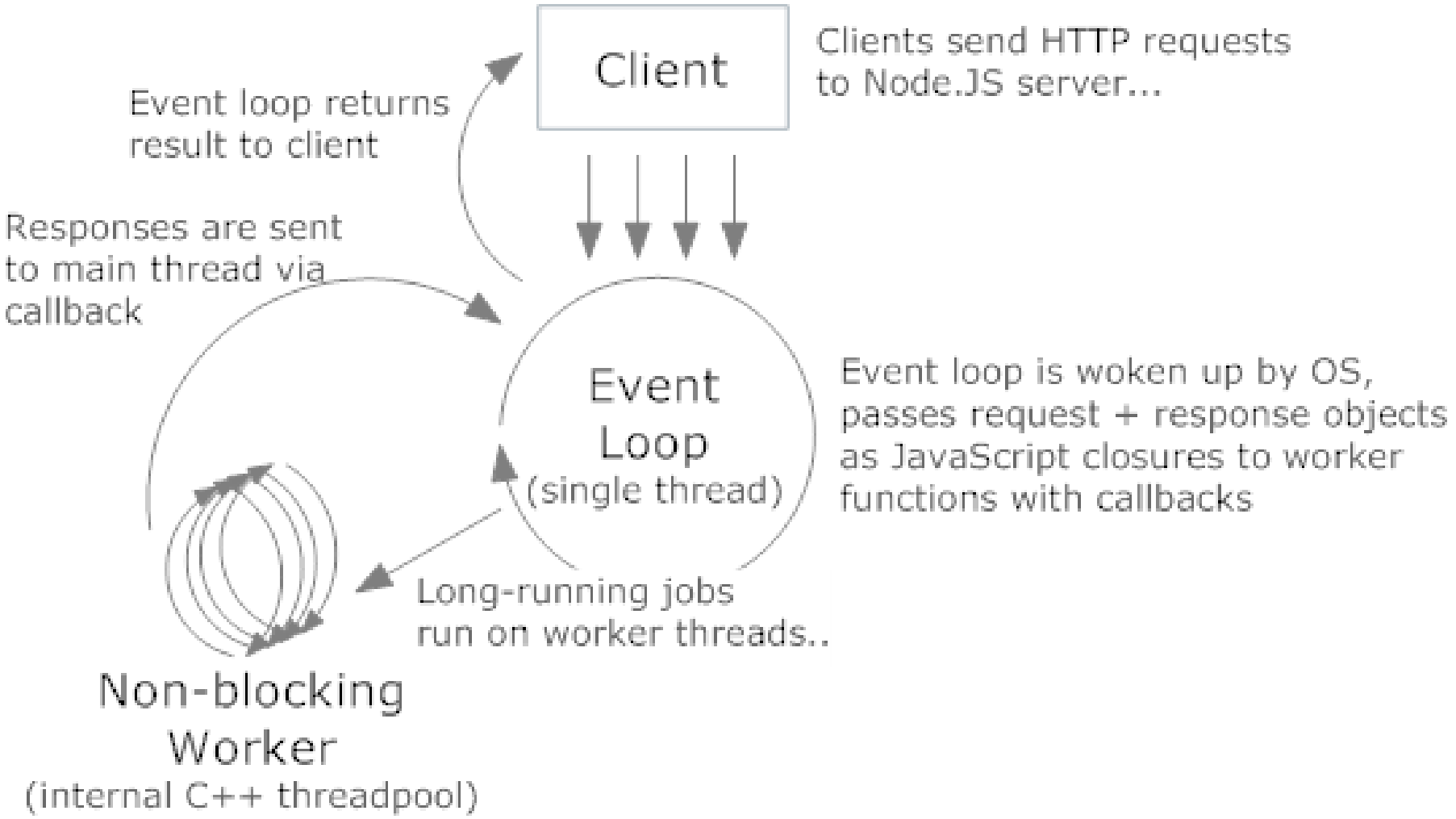


MEAN stack in work

MEAN = MongoDb+Express+AngularJS+NodeJS



NodeJS processing model



Task 2

Create the server side for the simplest application

Task 3

Notes application

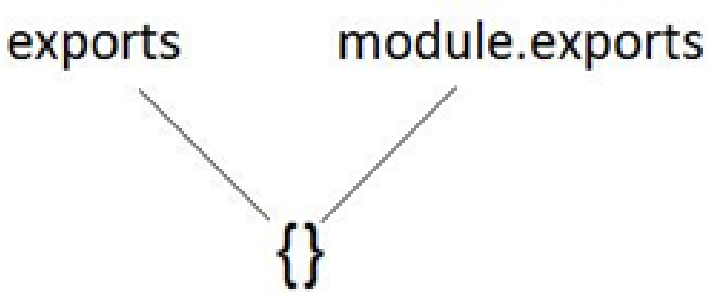
body-parser middleware

```
var bodyParser = require('body-parser');
//To parse URL encoded data
app.use(bodyParser.urlencoded({ extended: false }))
//To parse json data
app.use(bodyParser.json())

app.post("/", function (req, res) {
    console.log(req.body.name);
    console.log(req.body.email);
});
```

```
<form method="post" action="/">
    <input type="text" name="name">
    <input type="text" name="email">
    <input type="submit" value="Submit">
</form>
```

Modules in NodeJS: CommonJS

```
// module.js
exports.name = function() {
        console.log('My name is John');
};

var module= require('./module.js');
module.name(); //
```



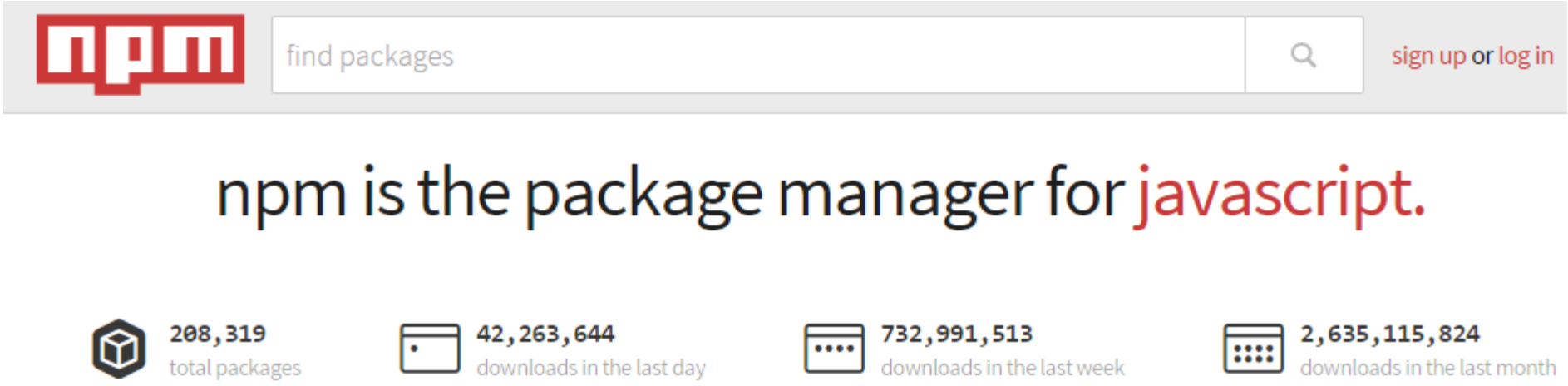
Both of them are references to the same (empty) object at the beggining. (But only module.exports will be returned!)

Session support in NodeJS

```
var session = require('express-session');
app.use(session({secret: "Shh, its a secret!"}));
app.get('/', function(req, res){
    if(req.session.page_views){
        req.session.page_views++;
        res.send("You visited this page " +
                                    req.session.page_views + " times");
    } else {
        req.session.page_views = 1;
        res.send("Welcome to this page for the first time!");
    }
});
```

NPM site

https://www.npmjs.com



Popular NPM modules overview

browserify

browser-side require() the node way

10.2.6 published 4 months ago by subst...

express

express

Fast, unopinionated, minimalist web...

4.13.1 published 5 months ago by doug...

grunt-cli

The grunt command line interface.

0.1.13 published 2 years ago by tkellen

npm

a package manager for JavaScript

2.13.0 published 5 months ago by zkat

bower

The browser package manager

1.4.1 published 8 months ago by sheerun

cordova

Cordova command line interface tool

5.1.1 published 5 months ago by stevegill

gulp

The streaming build system

3.9.0 published 6 months ago by phated

forever

A simple CLI tool for ensuring that a g...

0.14.2 published 5 months ago by index...

yo

CLI tool for running Yeoman generat...

1.4.7 published 6 months ago by sindres...

karma

Spectacular Test Runner for JavaScri...

0.13.1 published 4 months ago by dignifi...

Forever module

Installation:

```
sudo npm install -g forever
```

Use forever -w, instead of node to start your app:

```
$ forever -w app.js
```

Task 3

Store notes in session

Task 4

Store session in mongodb

Reading from files

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

//Asynchronous read
fs.readFile('input.txt', function (err, data) {
      if (err) {
          return console.error(err);
      }
      console.log("Asynchronous read: " + data.toString());
});

//Synchronous read
var data = fs.readFileSync('input.txt');
console.log("Synchronous read: " + data.toString());

console.log("Program Ended");
```

Writing to files

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

console.log("Going to write into existing file");
fs.writeFile('input.txt', 'Simply Easy Learning!',              function(err) {
        if (err) {
            return console.error(err);
        }
        console.log("Data written successfully!");
        console.log("Let's read newly written data");
        fs.readFile('input.txt', function (err, data) {
           if (err) {
              return console.error(err);
           }
           console.log("Asynchronous read: " + data.toString());
        });
    });
```

Task 5

Store notes in files