NODE.JS

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 FEATURES

Node.js = Runtime Environment + JavaScript Library

- Asynchronous and Event Driven
- Very Fast
- Single Threaded but Highly Scalable

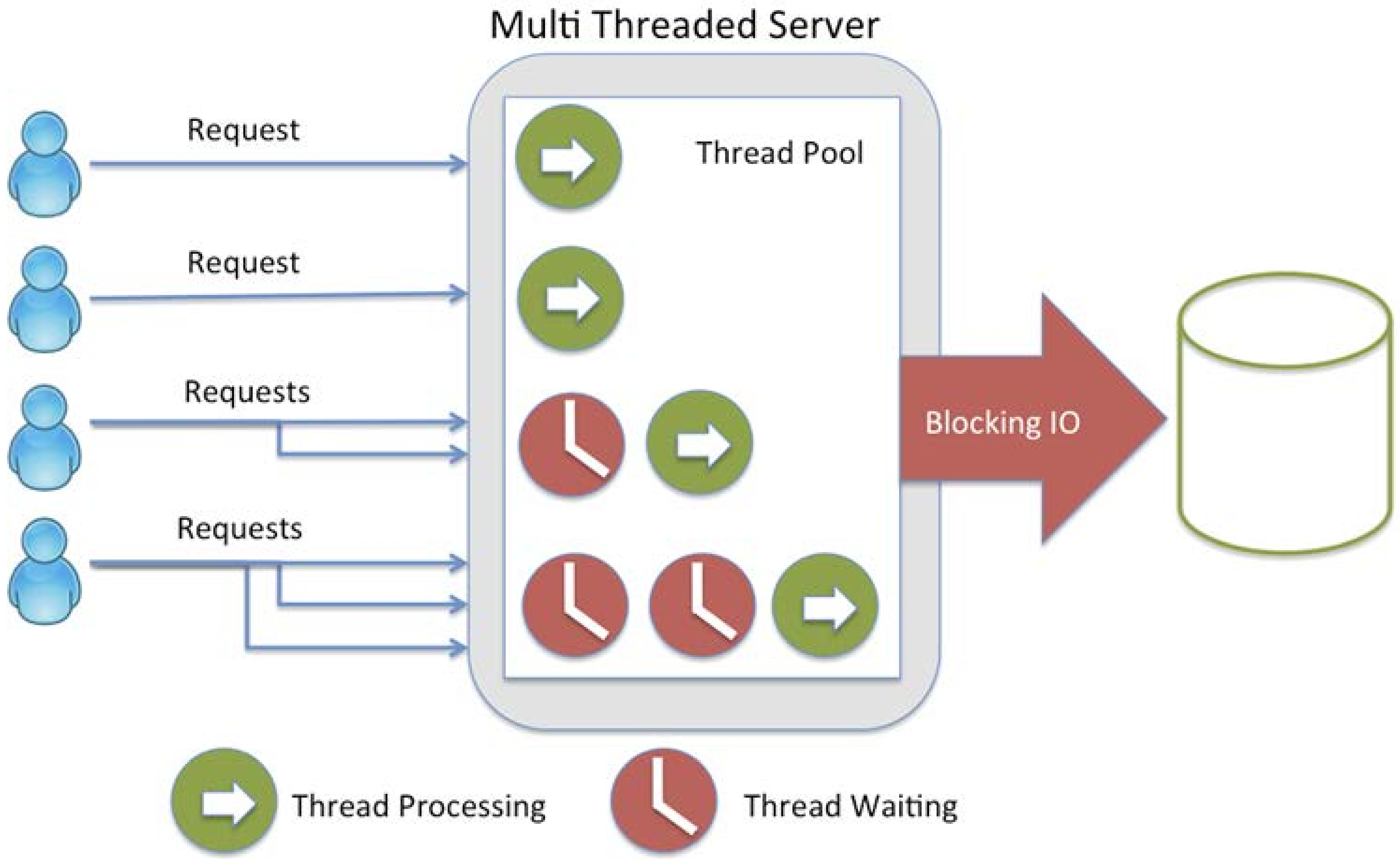
Where to Use Node.js?

- I/O bound Applications
- Data Streaming Applications
- Data Intensive Real-time Applications (DIRT)
- JSON APIs based Applications
- Single Page Applications

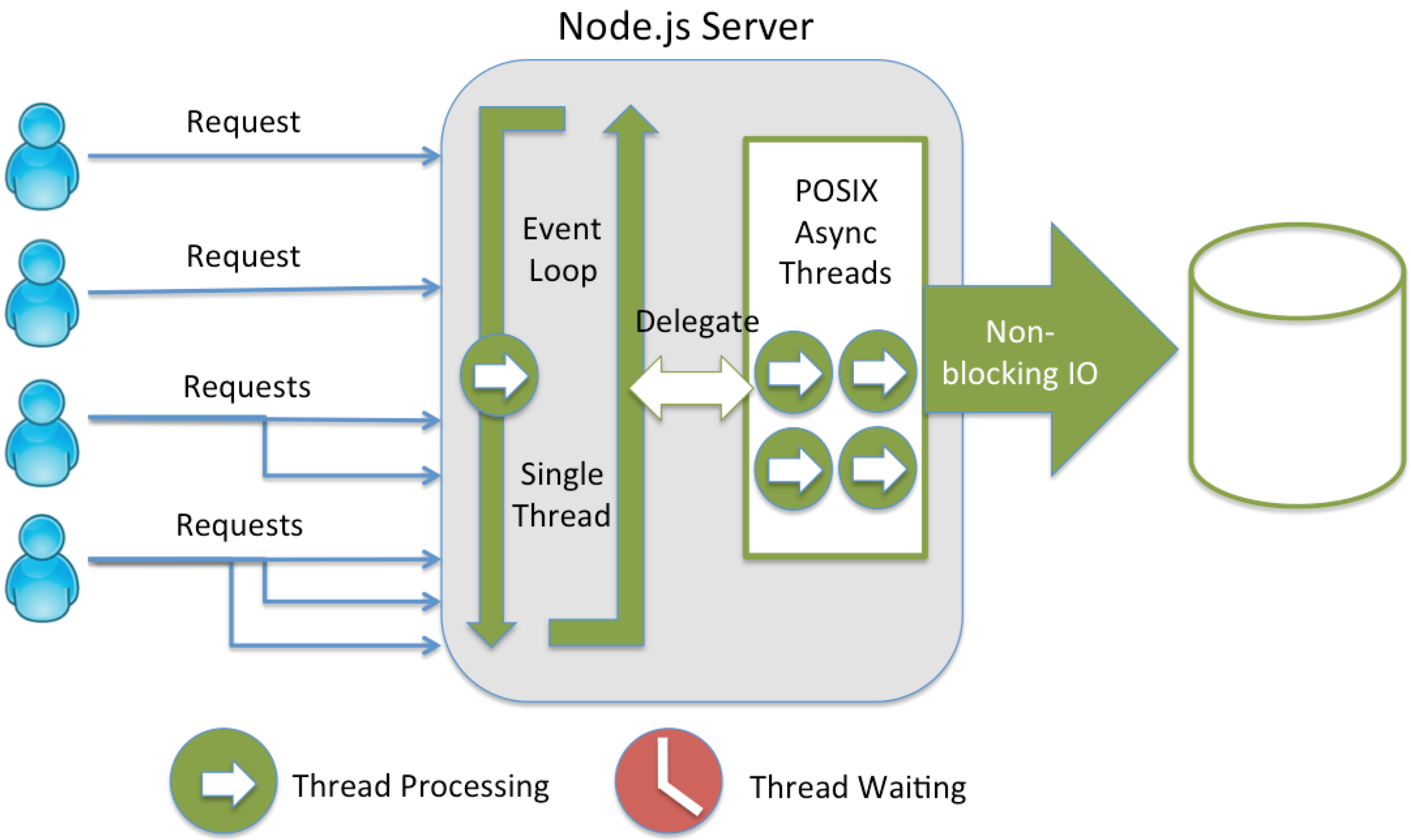
Where Not to Use Node.js?

- CPU intensive applications

MULTITHREADED SERVER REQUEST PROCESSING



NODE.JS REQUEST PROCESSING WITH EVENT LOOP



NPM

NPM

Node Package Manager (NPM) provides two main functionalities:

- Online repositories for node.js packages/modules
- Command line utility to install Node.js packages, do version management and dependency management

PACKAGE.JSON

```
{
  "name": "async-lib",
  "version": "1.1.2",
  "description": ”Async library",
  "main": "index.js",
  "scripts": {
    "test": "mocha test.js"
  },
  "author": "Vladimir Sonkin",
  "license": "ISC",
  “keywords": “async",
  "dependencies": {
    "bluebird": "^3.5.0"
  },
  "devDependencies": {
    "mocha": "~2.1.0"
  }
}
```

Commands:

- npm init
- npm install express
- npm uninstall express
- npm install express@3.2.1
- npm search express
- npm update express
- npm install webpack -g
- npm adduser
- npm publish

NPM FOLDERS

Local install (default): puts stuff in **./node_modules** of the current package root.

Global install (with -g): puts stuff in **/usr/local** or wherever node is installed.

Install it locally if you're going to require() it.

Install it globally if you're going to run it on the command line.

If you need both, then install it in both places, or use npm link.

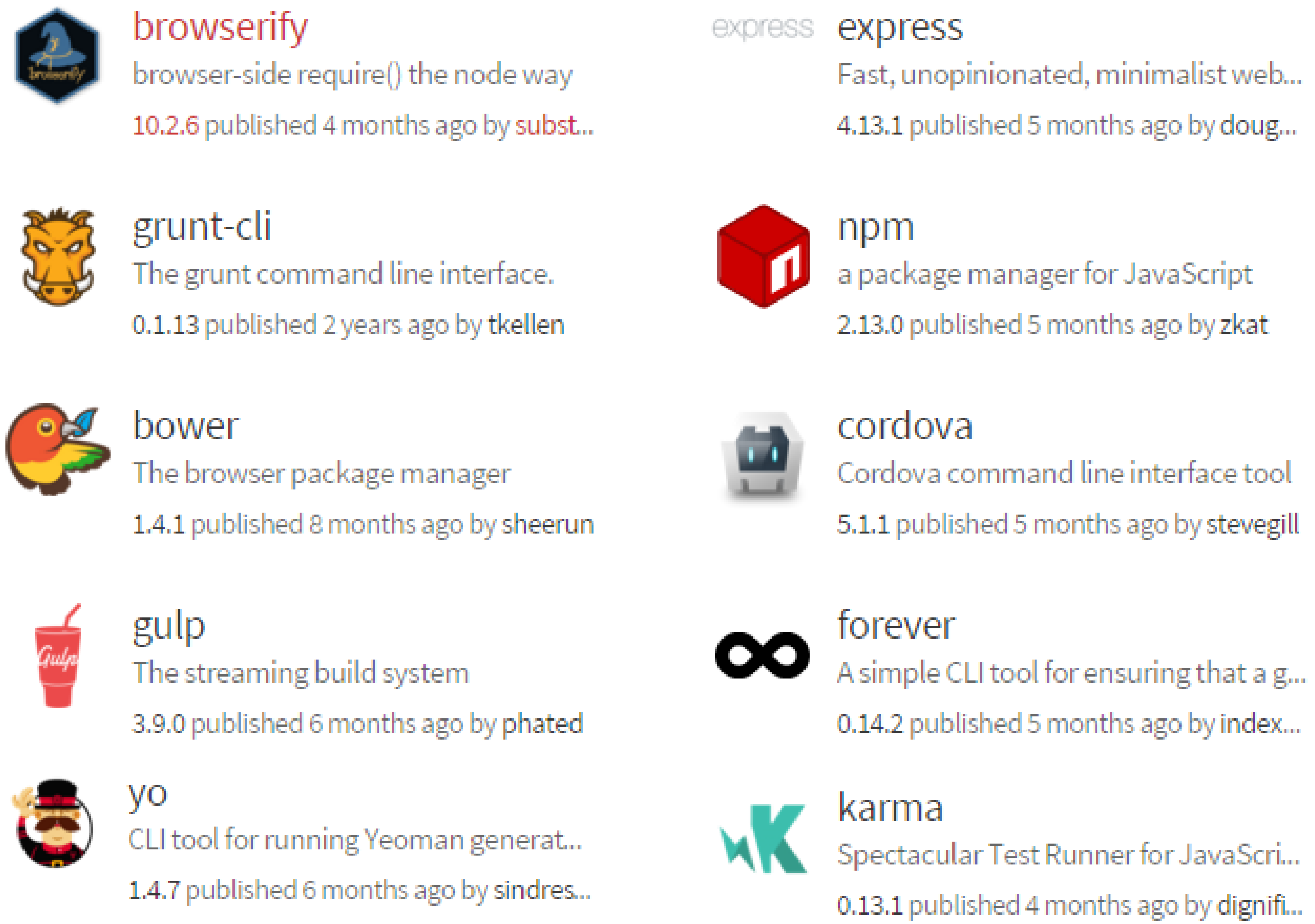
PACKAGE-LOCK.JSON

package-lock.json is automatically generated for any operations where npm modifies either the node_modules tree, or package.json

This file is intended to be committed into source repositories, and serves various purposes:

- Describe a **single representation** of a dependency tree
- Provide a facility for users to "**time-travel**" to previous states of node_modules without having to commit the directory itself.
- To facilitate greater visibility of tree changes through readable source control **diffs**.

POPULAR NPM MODULES



browserify
browser-side require() the node way
10.2.6 published 4 months ago by subst...

grunt-cli
The grunt command line interface.
0.1.13 published 2 years ago by tkellen

bower
The browser package manager
1.4.1 published 8 months ago by sheerun

gulp
The streaming build system
3.9.0 published 6 months ago by phated

yo
CLI tool for running Yeoman generat...
1.4.7 published 6 months ago by sindres...

express
Fast, unopinionated, minimalist web...
4.13.1 published 5 months ago by doug...

npm
a package manager for JavaScript
2.13.0 published 5 months ago by zkat

cordova
Cordova command line interface tool
5.1.1 published 5 months ago by stevegill

forever
A simple CLI tool for ensuring that a g...
0.14.2 published 5 months ago by index...

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
```

Alternates:

- nodemon
- supervisor

COMMON.JS MODULE SYSTEM

COMMONJS MODULE

```
function myModule() {
    this.hello = function() {
        return 'hello!';
    }

    this.goodbye = function() {
        return 'goodbye!';
    }
}

module.exports = myModule;
```

COMMONJS CLIENT

```
var myModule = require('myModule');

var myModuleInstance = new myModule();
myModuleInstance.hello(); // 'hello!'
myModuleInstance.goodbye(); // 'goodbye!'
```

EXPRESS FRAMEWORK

MINIMAL EXPRESS APP

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

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

app.listen(3000);
```

```
npm init
npm install express
npm install -g nodemon
```

```
> nodemon server.js
```

EXPRESS ROUTING

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"
```

PATTERN MATCHED ROUTE

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

app.get('/things/:id([0-9]{5})', function(req, res){
    res.send('id: ' + req.params.id);
});

// Other routes here
app.get('*', function(req, res){
    res.send('Sorry, this is an invalid URL.');
});

app.listen(3000);
```

this will only match the requests that have a 5-digit long id

http://localhost:3000/things/12345

id:12345

http://localhost:3000/things/123

Sorry, this is an invalid URL

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.

MIDDLEWARE

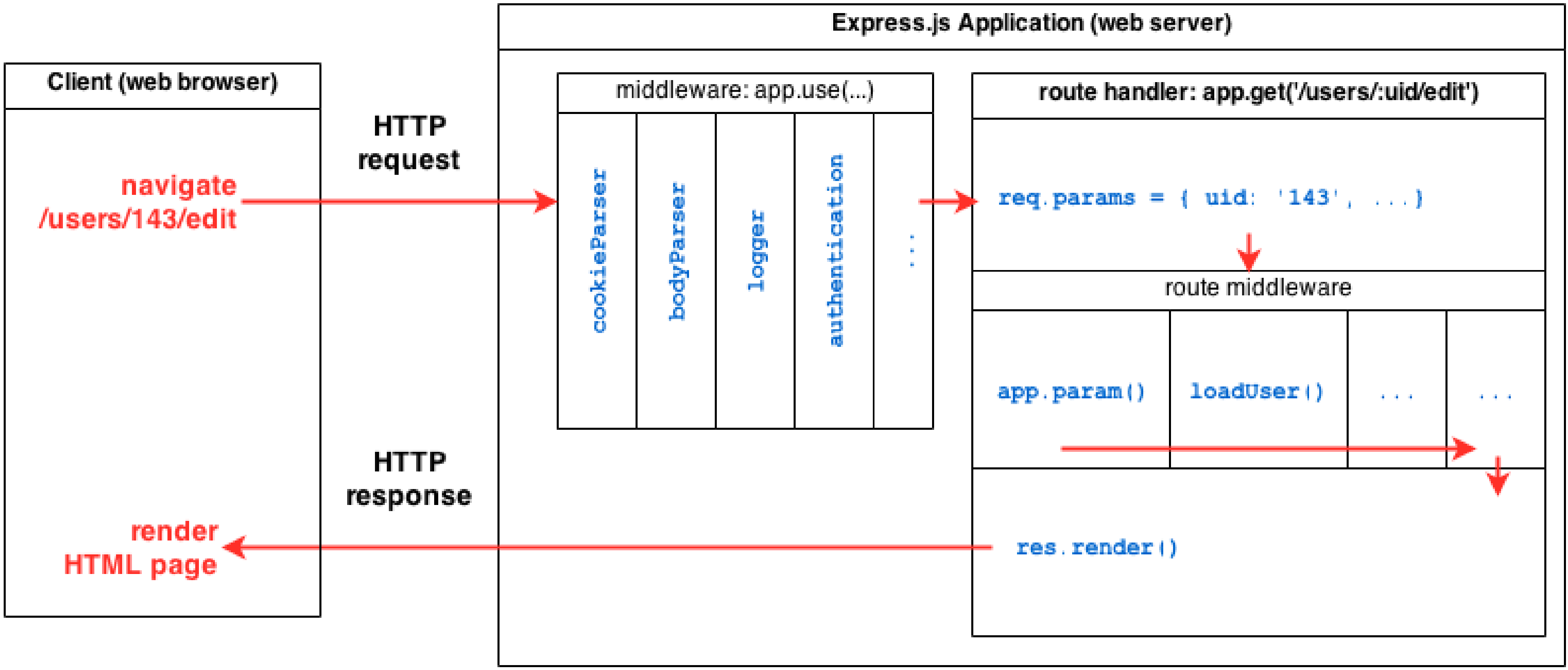
MIDDLEWARE

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

var timeLogger = function (req, res, next) {
    console.log("A new request received at " + Date.now());
    next();
};
app.use(timeLogger);

app.get('/', function (req, res) {
    res.send('Hello World!')
});
app.listen(3000);
```

EXPRESS STRUCTURE



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>
```

SESSION MIDDLEWARE

```
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!");

    }

});
```

SERVING STATIC FILES

```
app.use(express.static('public'))
```

Now, you can load the files that are in the public directory:

- http://localhost:3000/images/kitten.jpg
- http://localhost:3000/css/style.css
- http://localhost:3000/js/app.js
- http://localhost:3000/images/bg.png
- http://localhost:3000/hello.html
- http://localhost:3000/hello - not static, will be processed by the server

The path that you provide to the express.static function is **relative** to the directory from where you launch your node process. If you run the express app from another directory, it's safer to use the **absolute path**:

```
app.use(express.static(path.join(__dirname, 'public')))
```

USE EXPRESS TO CREATE REST SERVICE

EXAMPLE: REST SERVICE - GET

```
var express = require('express');
var app = express();
app.use(express.static('public'));

var session = require('express-session');
app.use(session({secret: "notes app",resave:true,saveUninitialized:true}));

app.get('/', function(req, res){
   if (!req.session.notes) req.session.notes = [];
   res.send({notes:req.session.notes});
});
```

EXAMPLE: REST SERVICE - POST

```
var bodyParser = require('body-parser');
app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());

app.post('/', function(req, res){
    let note = req.body.note;
    if (!req.session.notes) req.session.notes = [];
    req.session.notes.push(note);
    res.send({notes:req.session.notes});
});
```

THANK YOU!