



AngularJS

Node.JS introduction

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 standard libraries

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

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

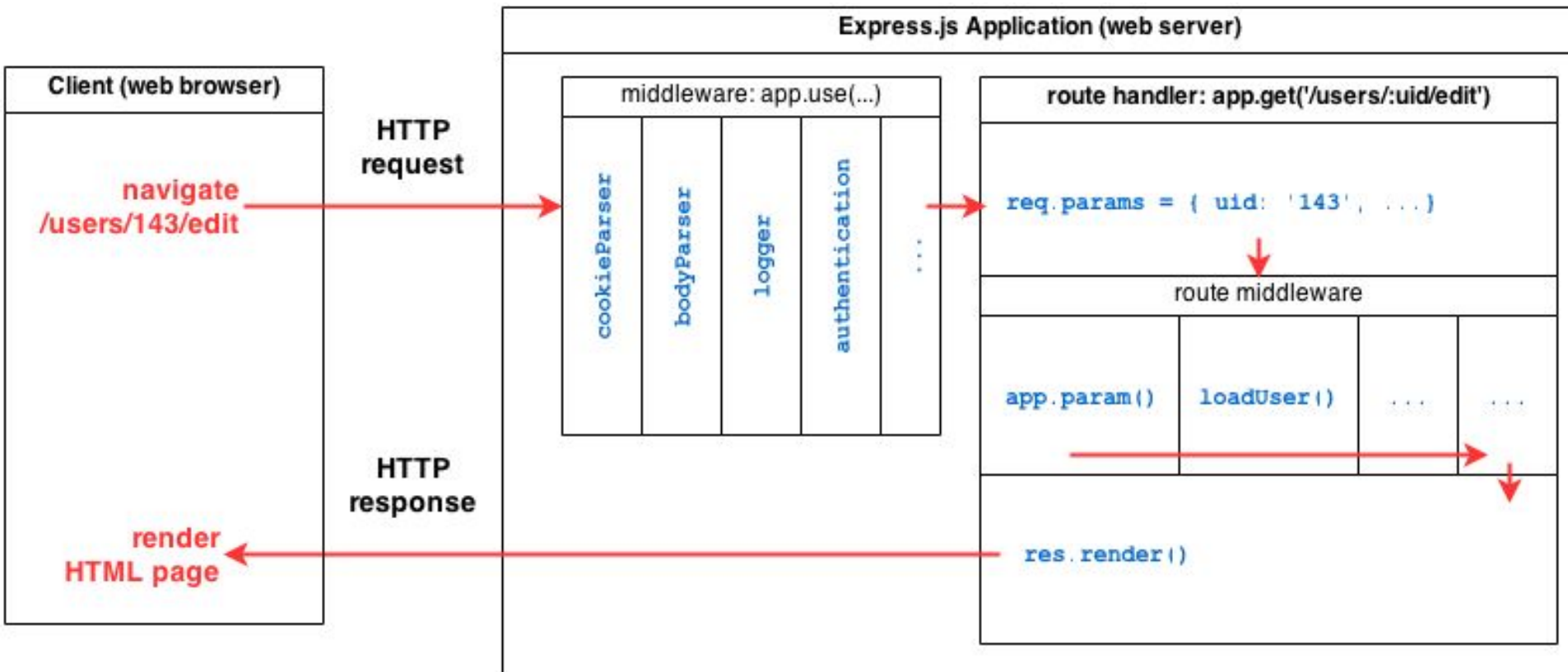
```
app.get("/greeting", function(req,res) {  
  res.send("Hello, "+req.query.name+"!");  
});
```

Express routing examples

```
app.get('/users/:id?', function(req,res) {  
  var id = req.params.id;  
  res.send( id ? 'user'+id : 'users');  
});
```

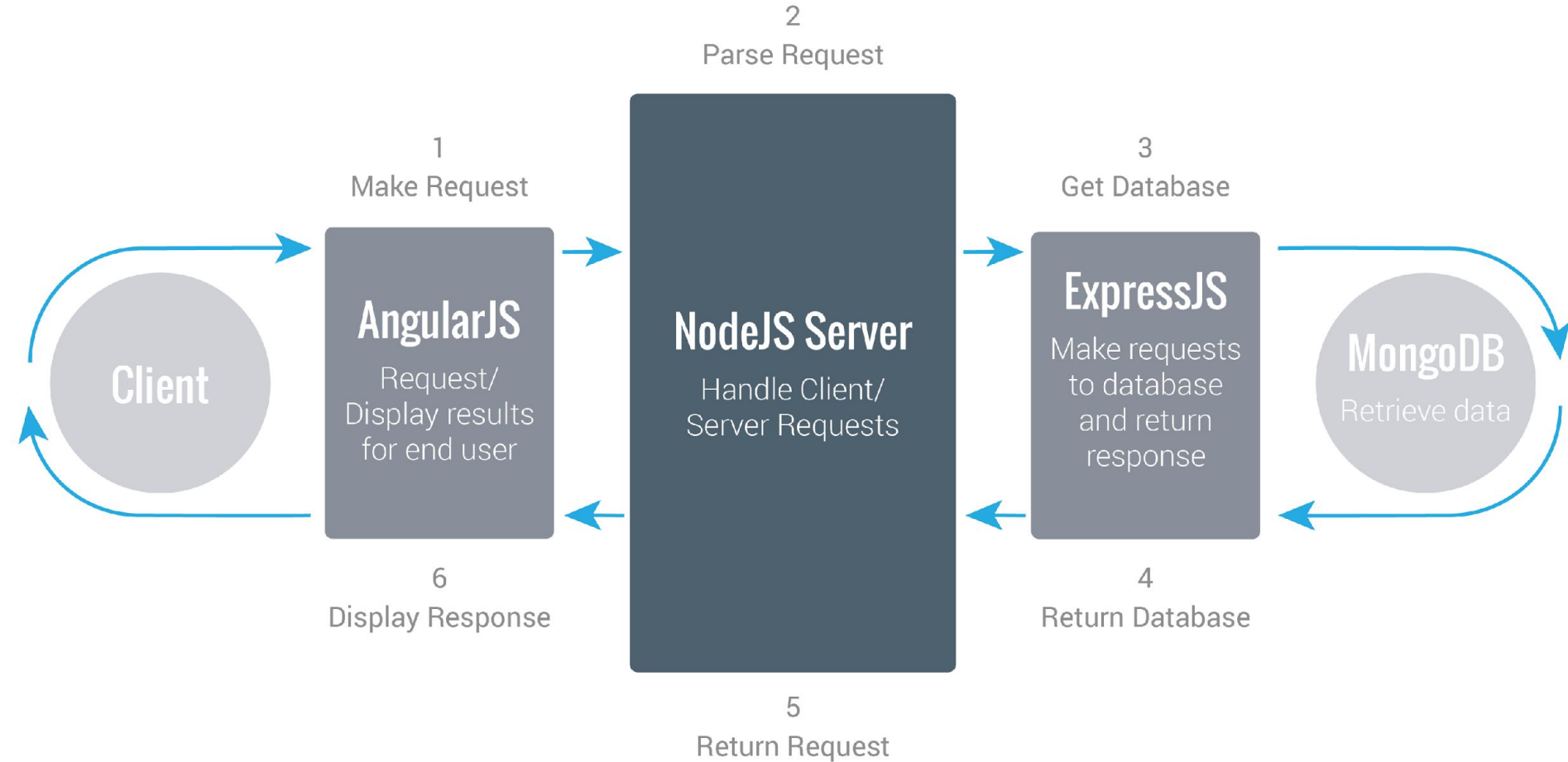
```
app.post('/users/:id.:format', function(req,res) {  
  var id = req.params.id;  
  var format = req.params.format;  
  res.send( 'user '+id+ ' format : '+format);  
});
```

NodeJS + Express application structure

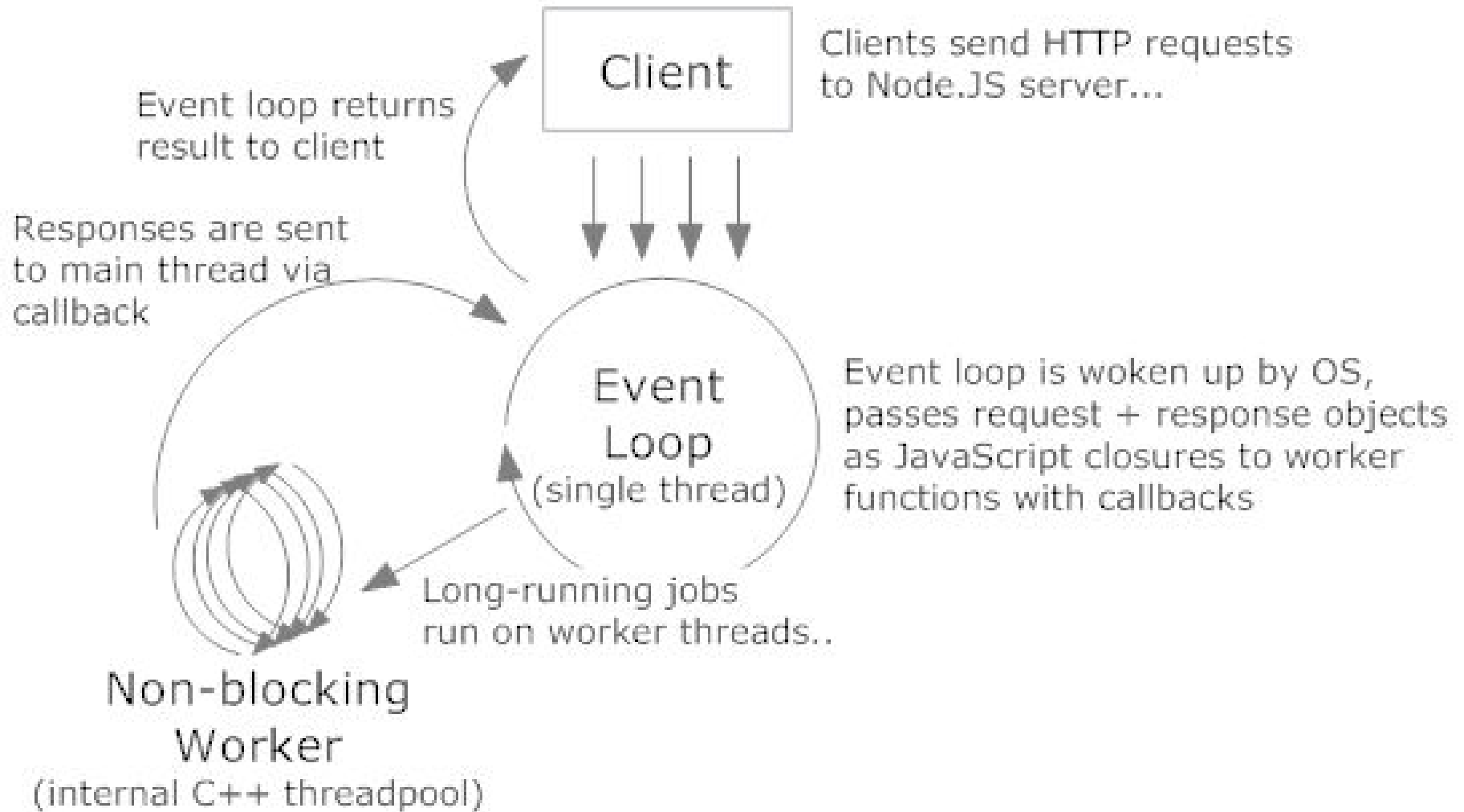


MEAN stack in work

MEAN = MongoDB+Express+AngularJS+NodeJS



NodeJS processing model



Task 2

Create the server side for the simplest application

Task 3

Notes application

Modules in NodeJS: CommonJS

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

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

exports

module.exports



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

Session support in NodeJS

```
app.use(session({
  secret: 'angular_tutorial',
  resave: true,
  saveUninitialized: true,
}));

app.get("/notes", function(req,res) {
  res.send(req.session.notes || []);
});

app.post("/notes", function(req, res) {
  if (!req.session.notes) {
    req.session.notes = [];
    req.session.last_note_id = 0;
  }
  var note = req.body;
  note.id = req.session.last_note_id;
  req.session.last_note_id++;
  req.session.notes.push(note);
  res.end();
});
```

NPM site

<https://www.npmjs.com>



find packages



[sign up](#) or [log in](#)

npm is the package manager for **javascript**.



208,319

total packages



42,263,644

downloads in the last day



732,991,513

downloads in the last week



2,635,115,824

downloads in the last month

Popular NPM modules overview



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
```

Task 4

Store notes in session

Task 5

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 6

Store notes in files