Web Development

COMP 431 / COMP 531

Lecture 24: Real-World Integration

Instructor: Mack Joyner

Department of Computer Science, Rice University

mjoyner@rice.edu

http://www.clear.rice.edu/comp431

Part II – Back End Development

COMP 531
Paper and Presentation
Presentation schedule: 11/21
Due Tuesday 11/28

Homework Assignment 7
Final WebApp
Due Thursday 11/30

COMP 531 Paper and Presentation

- Topic

  Web Development or Design

  - New technology
  - Technology comparison
  - Site design analysis
  - Enterprise in the Web
  - E-commerce
  - User experience
  - User interfaces
  - ReflectJS, Security, BigData

- Paper
  - 1000 to 2000 words
  - Proof read
  - Review and revise
  - Spelling and grammar
  - Think of it as a *blog post* that your future boss will read

- Presentation
  - Not more than 5 minute talk
  - slides, web sites, demos, props, etc...

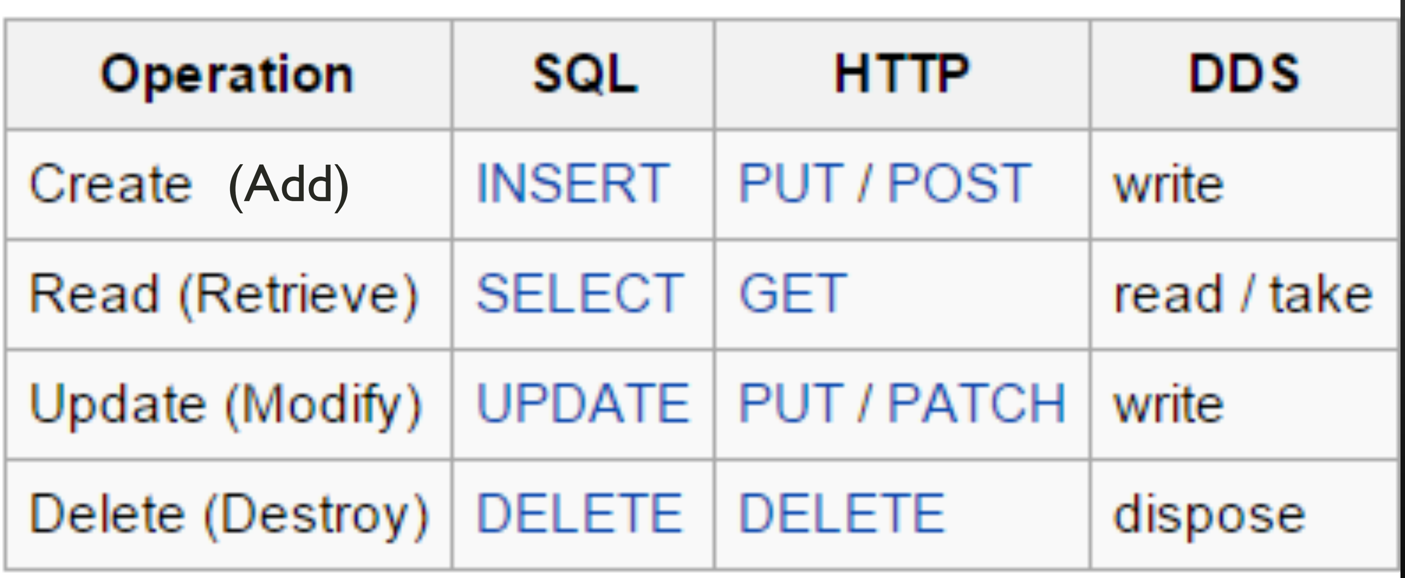
We *will* have an in-class submissions

Representational State Transfer (ReST)

- First proposed by Roy Thomas Fielding in his 2000 PhD dissertation
- Client-server separation of concerns to simplify component implementation, reduction of complexity, increase scalability
- Standard HTTP method verbs
- Typically JSON messages
- Hypertext link to state
- REST is stateless and cacheable

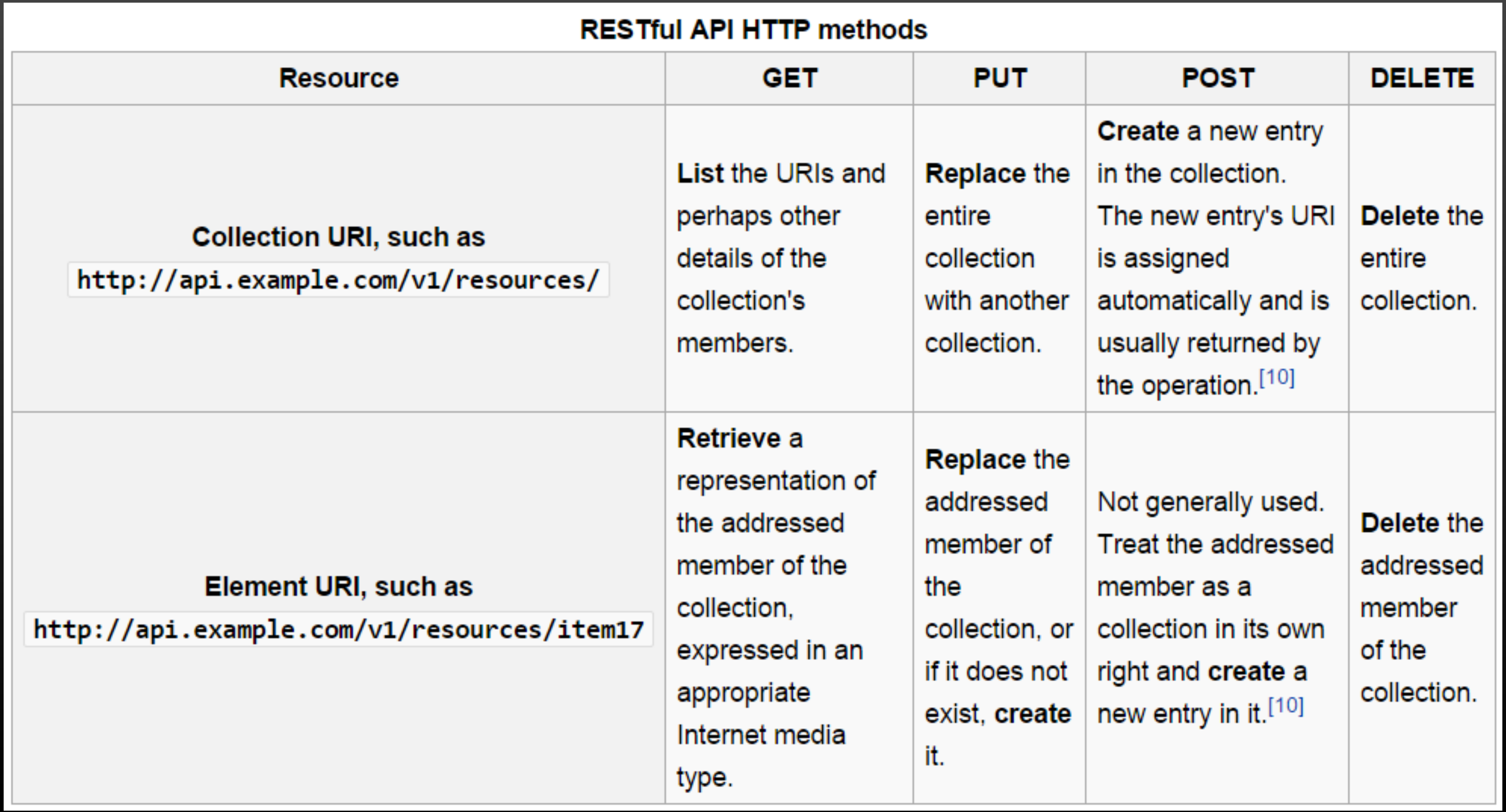
RESTful CRUD

- Data access and manipulation through CRUD operations



Operation	SQL	HTTP	DDS
Create (Add)	INSERT	PUT / POST	write
Read (Retrieve)	SELECT	GET	read / take
Update (Modify)	UPDATE	PUT / PATCH	write
Delete (Destroy)	DELETE	DELETE	dispose

Representational State Transfer Example

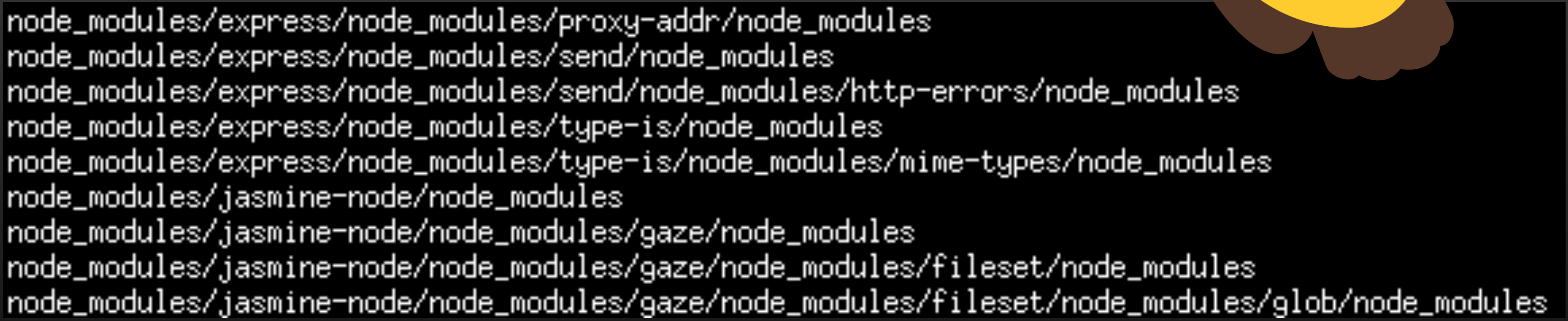


RESTful API HTTP methods

Resource	GET	PUT	POST	DELETE
Collection URI, such as `http://api.example.com/v1/resources/`	**List** the URIs and perhaps other details of the collection's members.	**Replace** the entire collection with another collection.	**Create** a new entry in the collection. The new entry's URI is assigned automatically and is usually returned by the operation.[10]	**Delete** the entire collection.
Element URI, such as `http://api.example.com/v1/resources/item17`	**Retrieve** a representation of the addressed member of the collection, expressed in an appropriate Internet media type.	**Replace** the addressed member of the collection, or if it does not exist, **create** it.	Not generally used. Treat the addressed member as a collection in its own right and **create** a new entry in it.[10]	**Delete** the addressed member of the collection.

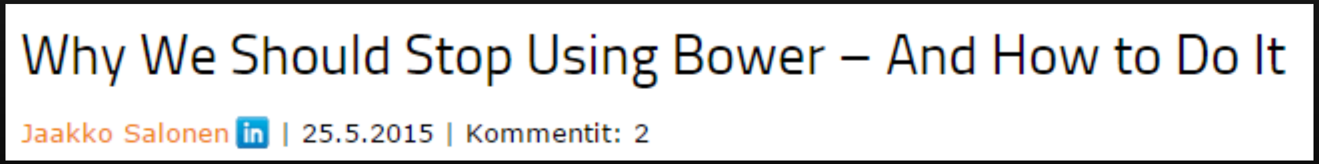
Making Things Better? *Bower*

- Bower is a *front end* package manager



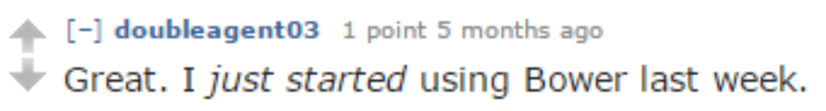
```
node_modules/express/node_modules/proxy-addr/node_modules
node_modules/express/node_modules/send/node_modules
node_modules/express/node_modules/send/node_modules/http-errors/node_modules
node_modules/express/node_modules/type-is/node_modules
node_modules/express/node_modules/type-is/node_modules/mime-types/node_modules
node_modules/jasmine-node/node_modules
node_modules/jasmine-node/node_modules/gaze/node_modules
node_modules/jasmine-node/node_modules/gaze/node_modules/fileset/node_modules
node_modules/jasmine-node/node_modules/gaze/node_modules/fileset/node_modules/glob/node_modules
```

```
npm install bower -g
bower init
bower install mocha
```



Why We Should Stop Using Bower – And How to Do It

Jaakko Salonen in | 25.5.2015 | Kommentit: 2

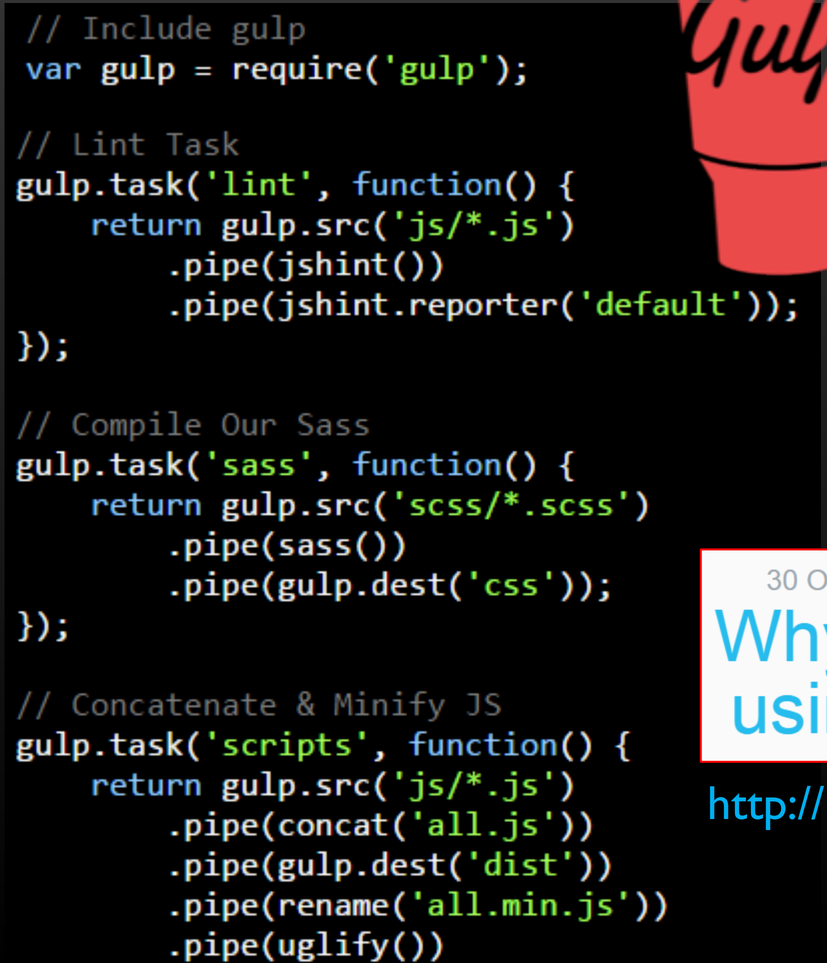


[–] doubleagent03 1 point 5 months ago

Great. I *just started* using Bower last week.

Making Things Better? *Grunt / Gulp*

- Task runners

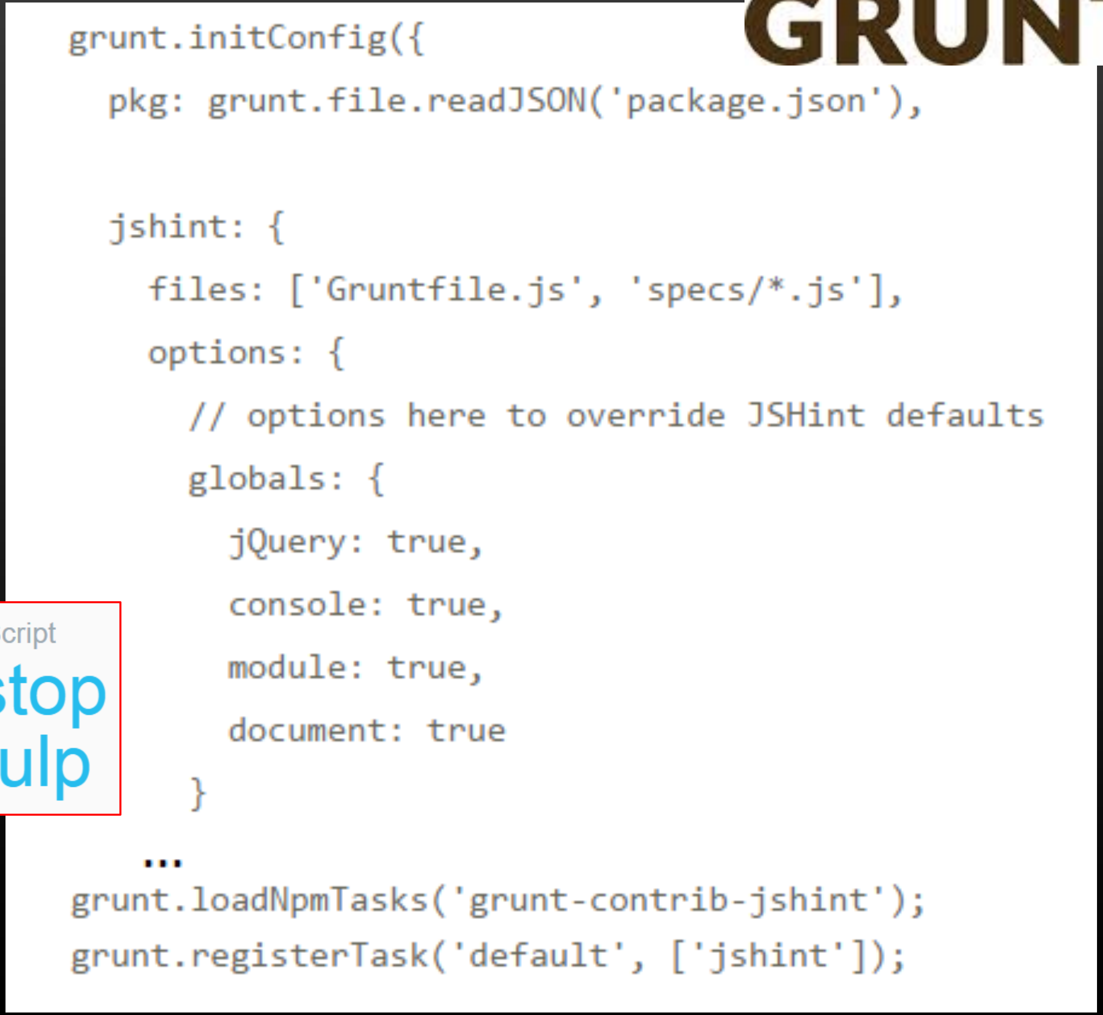


```
// Include gulp
var gulp = require('gulp');

// Lint Task
gulp.task('lint', function() {
    return gulp.src('js/*.js')
        .pipe(jshint())
        .pipe(jshint.reporter('default'));
});

// Compile Our Sass
gulp.task('sass', function() {
    return gulp.src('scss/*.scss')
        .pipe(sass())
        .pipe(gulp.dest('css'));
});

// Concatenate & Minify JS
gulp.task('scripts', function() {
    return gulp.src('js/*.js')
        .pipe(concat('all.js'))
        .pipe(gulp.dest('dist'))
        .pipe(rename('all.min.js'))
        .pipe(uglify())
```

30 Oct 2014 on Node.js | npm | JavaScript

Why we should stop using Grunt & Gulp

http://blog.keithcirkel.co.uk

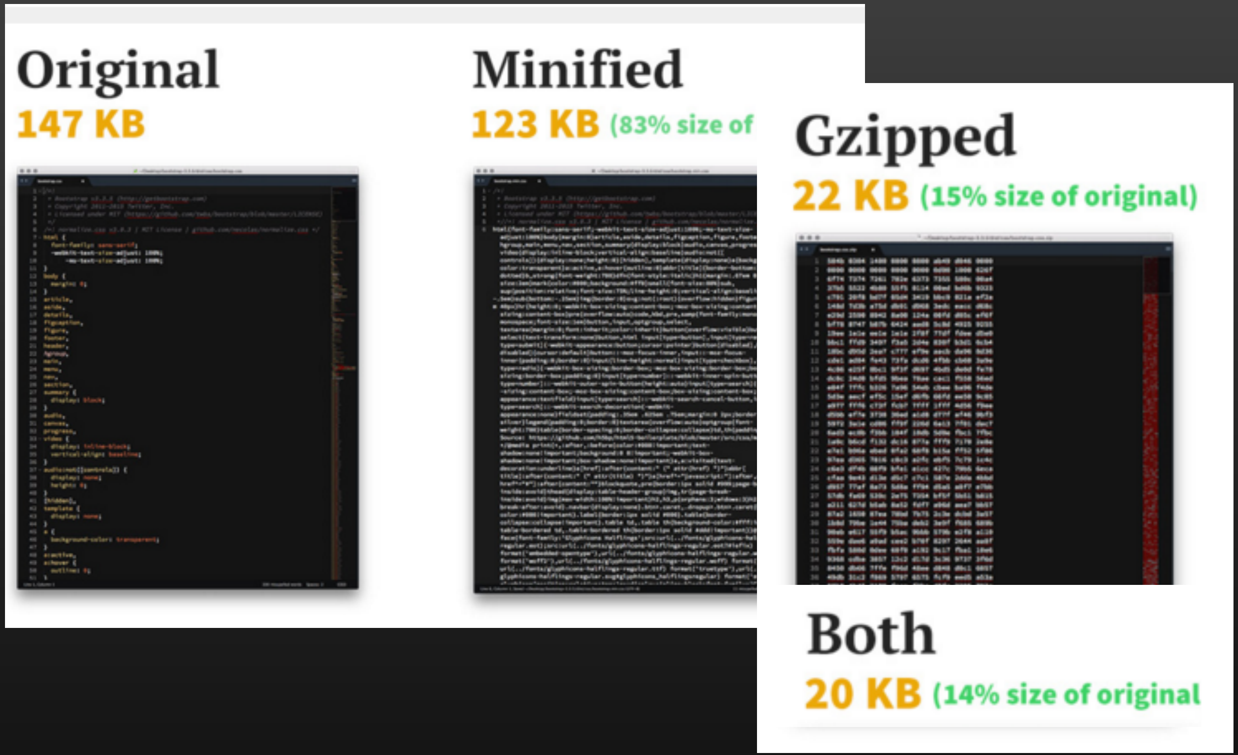


```
grunt.initConfig({
  pkg: grunt.file.readJSON('package.json'),

  jshint: {
    files: ['Gruntfile.js', 'specs/*.js'],
    options: {
      // options here to override JSHint defaults
      globals: {
        jQuery: true,
        console: true,
        module: true,
        document: true
      }
    ...
grunt.loadNpmTasks('grunt-contrib-jshint');
grunt.registerTask('default', ['jshint']);
```

Making Things Better: *Uglify = Minification*

- Good for front-end use
- Two main features:
  - Concatenates files together
  - Removes whitespace

- Big win is compression
  - Concatenate your files then gzip



Original
147 KB

Minified
123 KB (83% size of

Gzipped
22 KB (15% size of original)

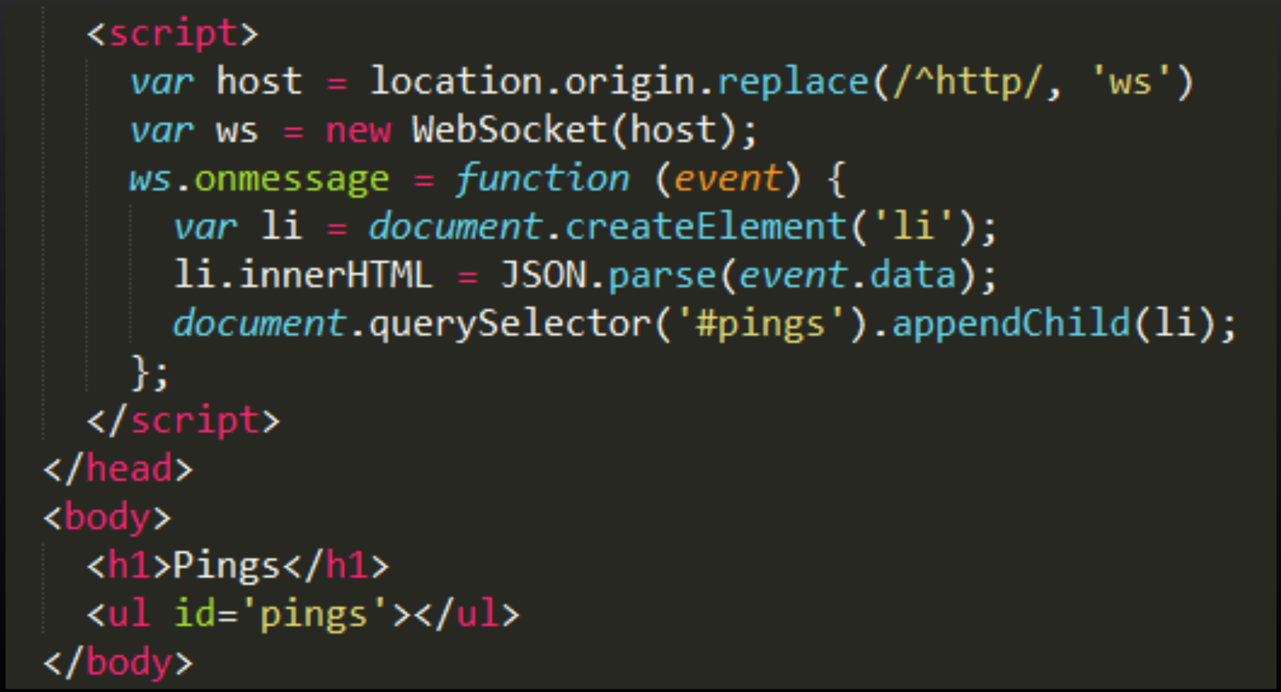
Both
20 KB (14% size of original

```
find app -type f | grep -E .js$ |\
  grep -v spec.js | xargs cat | gzip > app.js.gz
```

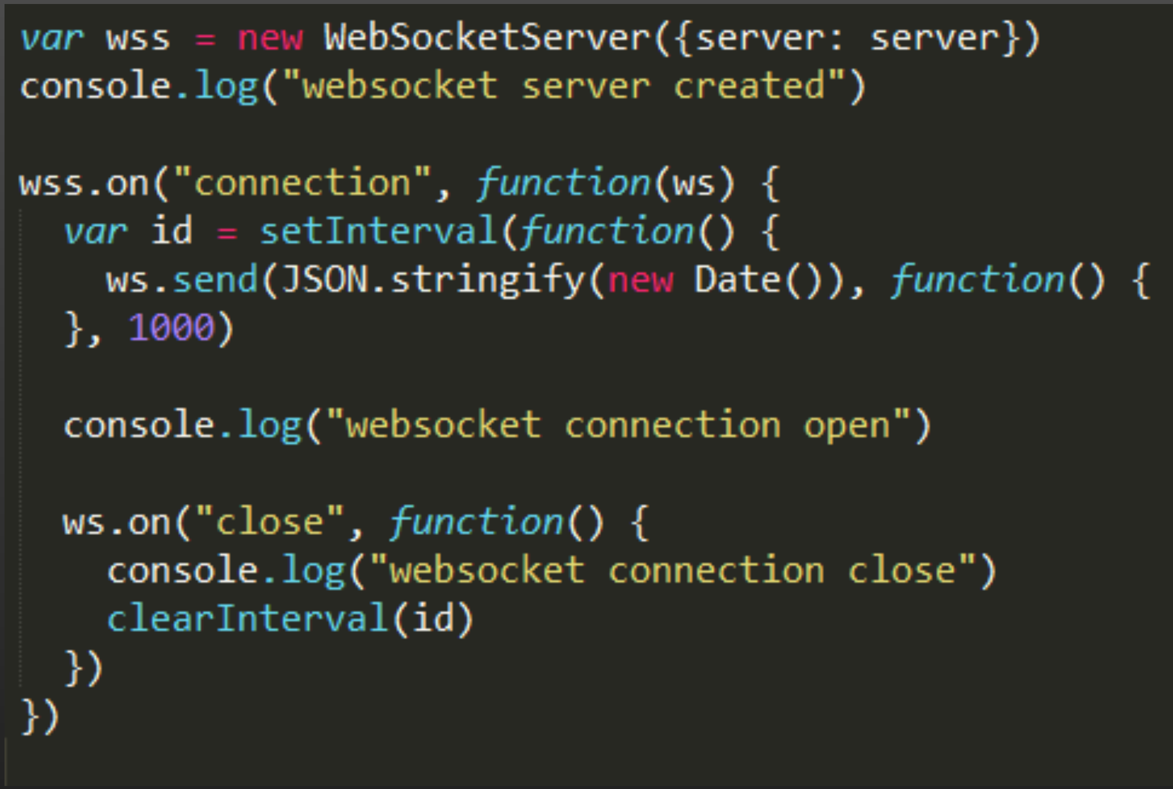
webpack can do this for us!

Web Sockets

- http requests are initiated by browser
  - One-way communication
  - Requires polling by client for server updates

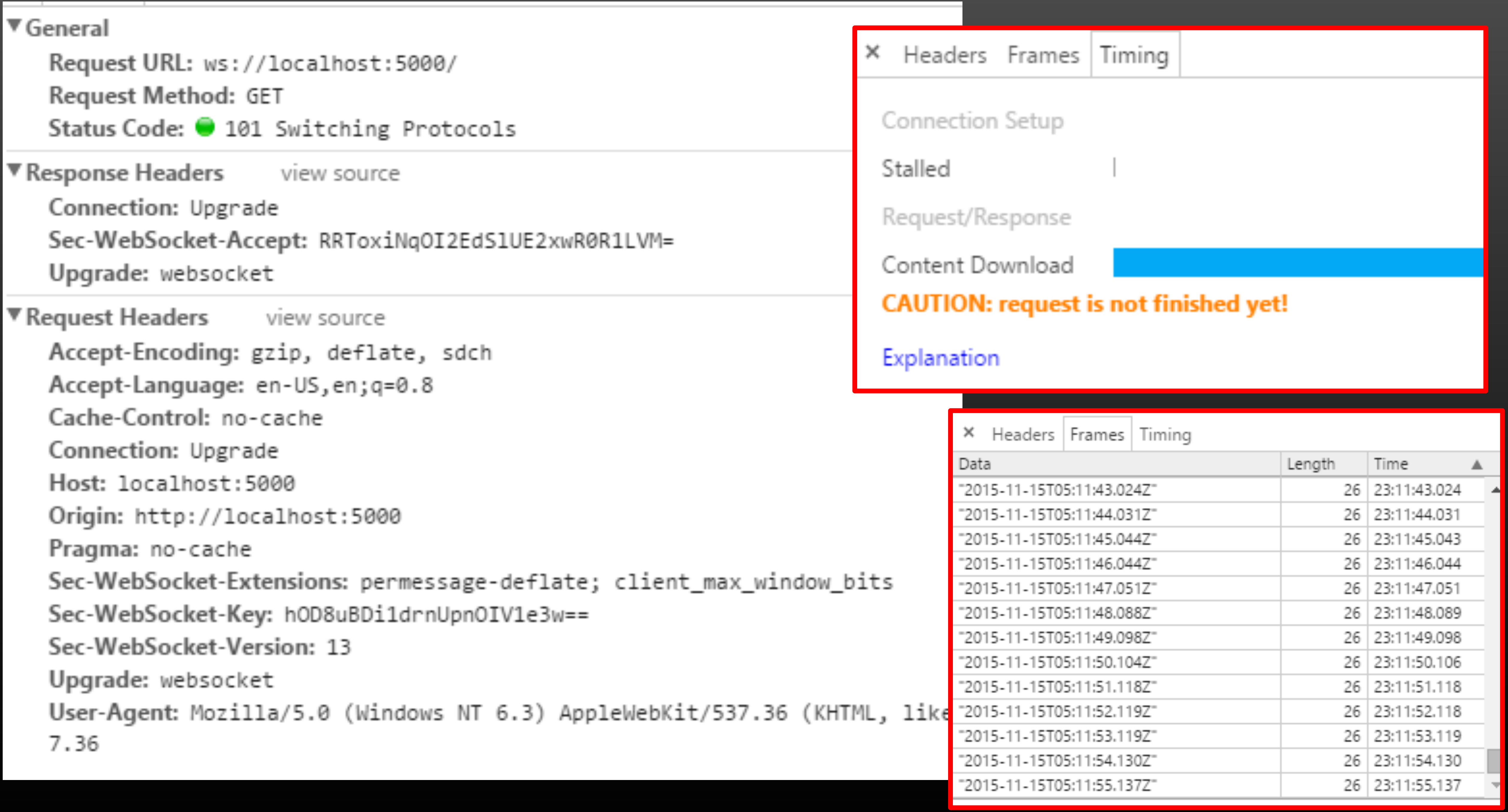


```
  <script>
    var host = location.origin.replace(/^http/, 'ws')
    var ws = new WebSocket(host);
    ws.onmessage = function (event) {
      var li = document.createElement('li');
      li.innerHTML = JSON.parse(event.data);
      document.querySelector('#pings').appendChild(li);
    };
  </script>
</head>
<body>
  <h1>Pings</h1>
  <ul id='pings'></ul>
</body>
```



```
var wss = new WebSocketServer({server: server})
console.log("websocket server created")

wss.on("connection", function(ws) {
  var id = setInterval(function() {
    ws.send(JSON.stringify(new Date()), function() {
  }, 1000)

  console.log("websocket connection open")

  ws.on("close", function() {
    console.log("websocket connection close")
    clearInterval(id)
  })
})
```

- WebSockets are two way pipes
  - Ideal for chatty communication



▼ General

Request URL: ws://localhost:5000/
Request Method: GET
Status Code: ● 101 Switching Protocols

▼ Response Headers   view source

```
Connection: Upgrade
Sec-WebSocket-Accept: RRToxiNqOI2EdS1UE2xwR0R1LVM=
Upgrade: websocket
```

▼ Request Headers   view source

```
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
Cache-Control: no-cache
Connection: Upgrade
Host: localhost:5000
Origin: http://localhost:5000
Pragma: no-cache
Sec-WebSocket-Extensions: permessage-deflate; client_max_window_bits
Sec-WebSocket-Key: hOD8uBDi1drnUpnOIV1e3w==
Sec-WebSocket-Version: 13
Upgrade: websocket
User-Agent: Mozilla/5.0 (Windows NT 6.3) AppleWebKit/537.36 (KHTML, like
7.36
```

Connection Setup
Stalled
Request/Response
Content Download

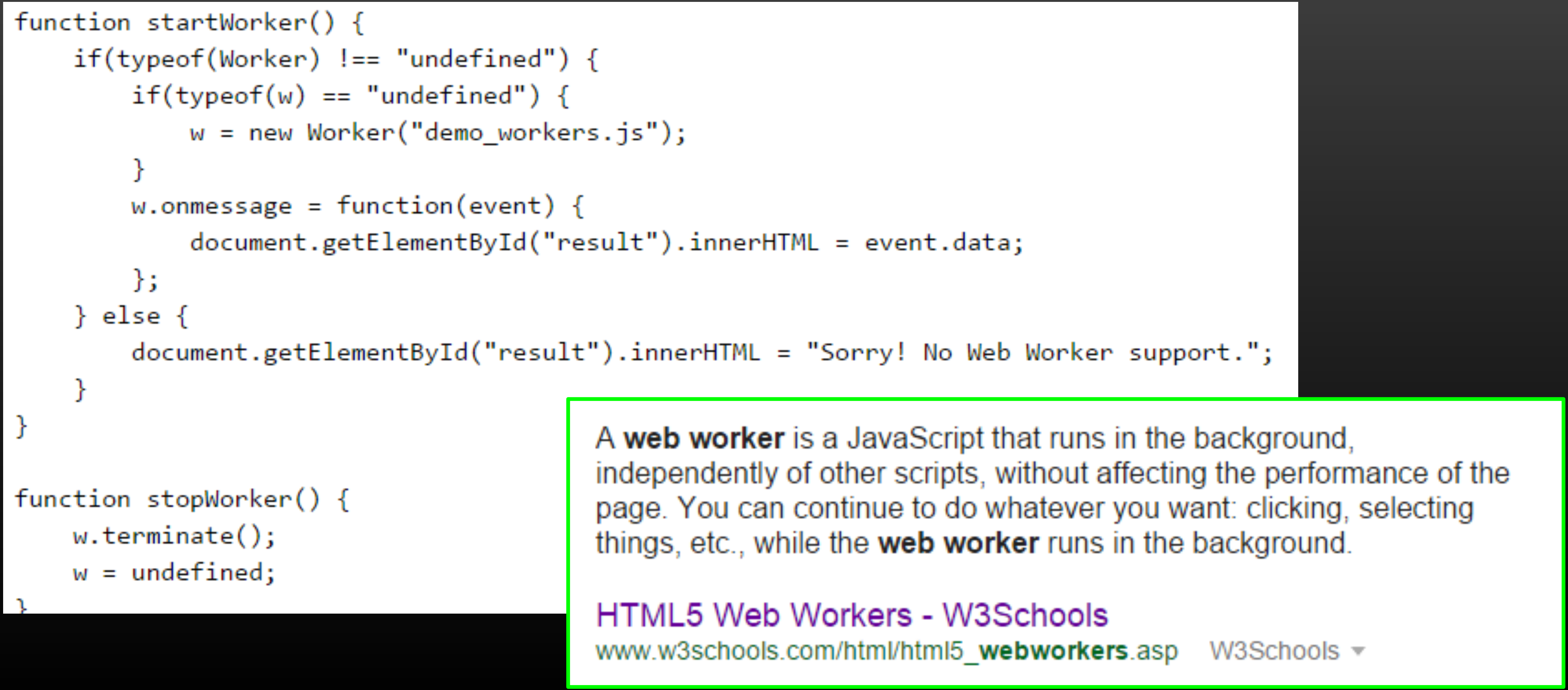
CAUTION: request is not finished yet!

Explanation

Data	Length	Time
"2015-11-15T05:11:43.024Z"	26	23:11:43.024
"2015-11-15T05:11:44.031Z"	26	23:11:44.031
"2015-11-15T05:11:45.044Z"	26	23:11:45.043
"2015-11-15T05:11:46.044Z"	26	23:11:46.044
"2015-11-15T05:11:47.051Z"	26	23:11:47.051
"2015-11-15T05:11:48.088Z"	26	23:11:48.089
"2015-11-15T05:11:49.098Z"	26	23:11:49.098
"2015-11-15T05:11:50.104Z"	26	23:11:50.106
"2015-11-15T05:11:51.118Z"	26	23:11:51.118
"2015-11-15T05:11:52.119Z"	26	23:11:52.118
"2015-11-15T05:11:53.119Z"	26	23:11:53.119
"2015-11-15T05:11:54.130Z"	26	23:11:54.130
"2015-11-15T05:11:55.137Z"	26	23:11:55.137

Queues and Workers

Web Worker = Front End



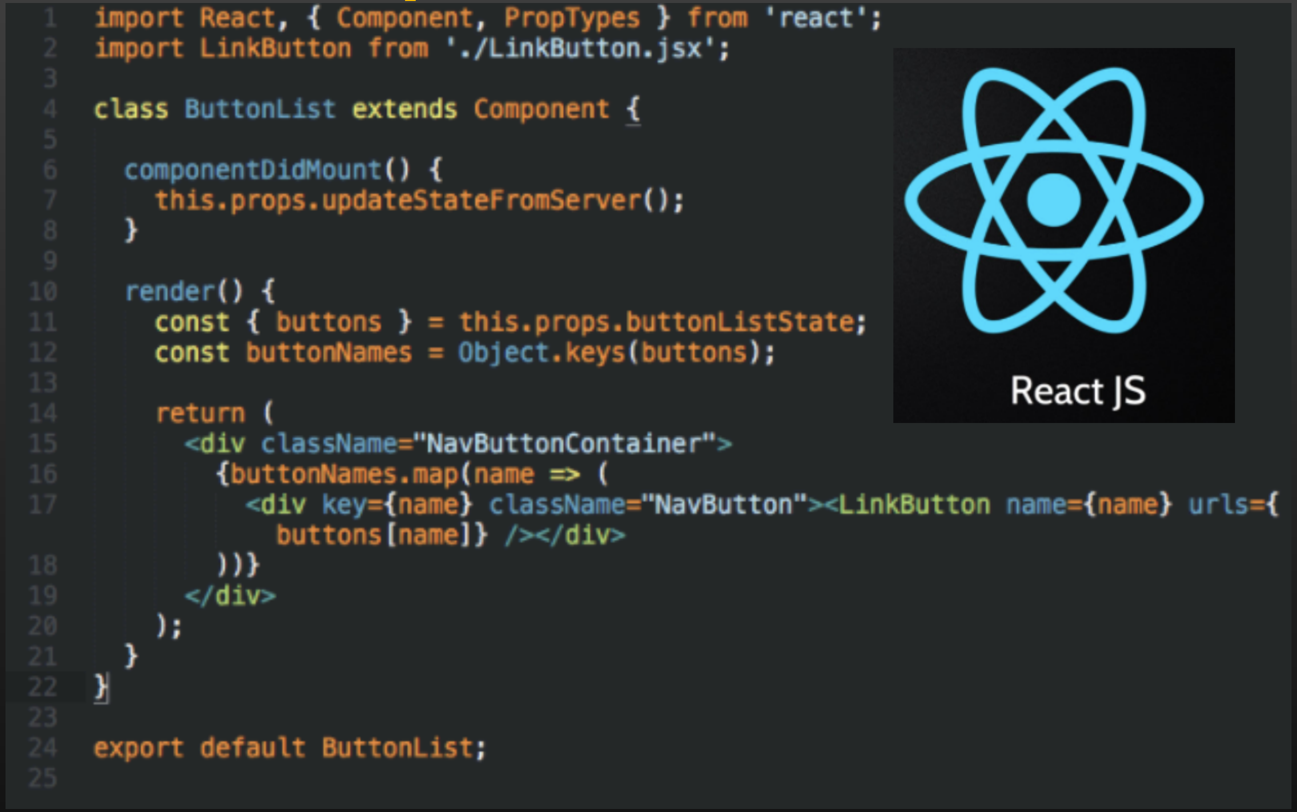
```
function startWorker() {
    if(typeof(Worker) !== "undefined") {
        if(typeof(w) == "undefined") {
            w = new Worker("demo_workers.js");
        }
        w.onmessage = function(event) {
            document.getElementById("result").innerHTML = event.data;
        };
    } else {
        document.getElementById("result").innerHTML = "Sorry! No Web Worker support.";
    }
}

function stopWorker() {
    w.terminate();
    w = undefined;
}
```

A **web worker** is a JavaScript that runs in the background, independently of other scripts, without affecting the performance of the page. You can continue to do whatever you want: clicking, selecting things, etc., while the **web worker** runs in the background.

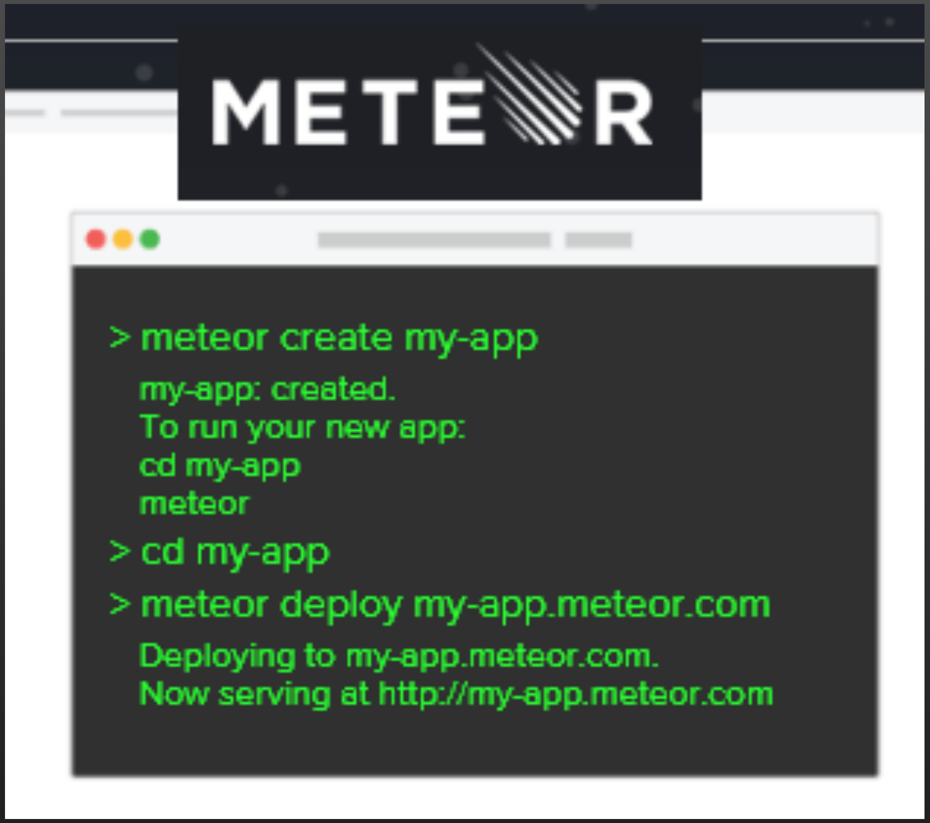
HTML5 Web Workers - W3Schools

www.w3schools.com/html/html5_**webworkers**.asp W3Schools ▾

Competitors



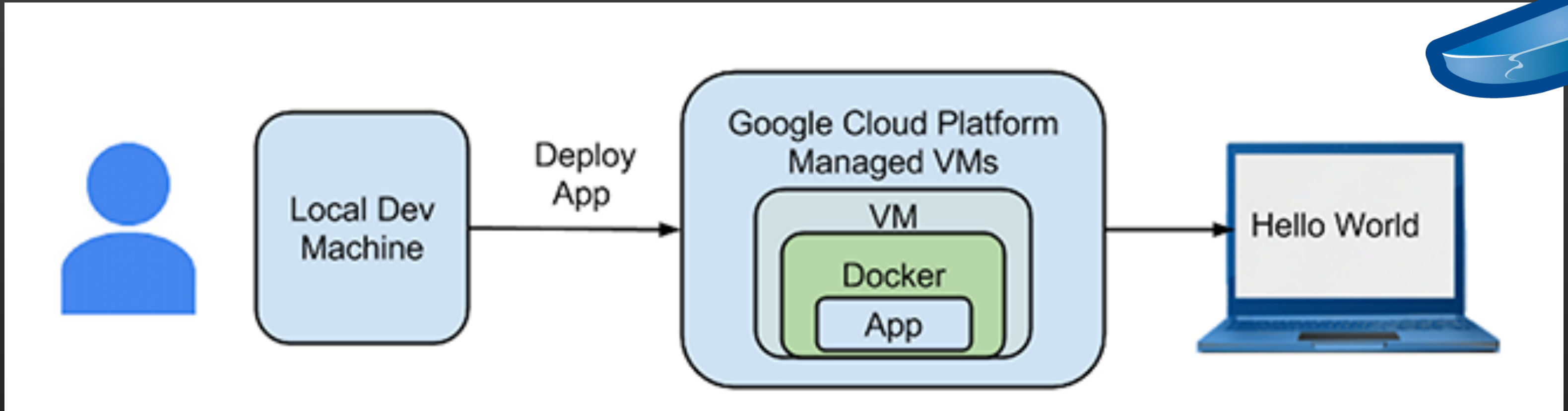
```
import React, { Component, PropTypes } from 'react';
import LinkButton from './LinkButton.jsx';

class ButtonList extends Component {

  componentDidMount() {
    this.props.updateStateFromServer();
  }

  render() {
    const { buttons } = this.props.buttonListState;
    const buttonNames = Object.keys(buttons);

    return (
      <div className="NavButtonContainer">
        {buttonNames.map(name => (
          <div key={name} className="NavButton"><LinkButton name={name} urls={
            buttons[name]} /></div>
        ))}
      </div>
    );
  }
}

export default ButtonList;
```



```
> meteor create my-app
  my-app: created.
  To run your new app:
  cd my-app
  meteor
> cd my-app
> meteor deploy my-app.meteor.com
  Deploying to my-app.meteor.com.
  Now serving at http://my-app.meteor.com
```

Sails is built on Node.js, Connect, Express, and Socket.io.

Google App Engine



Local Dev Machine

Deploy App

Google Cloud Platform Managed VMs

VM

Docker

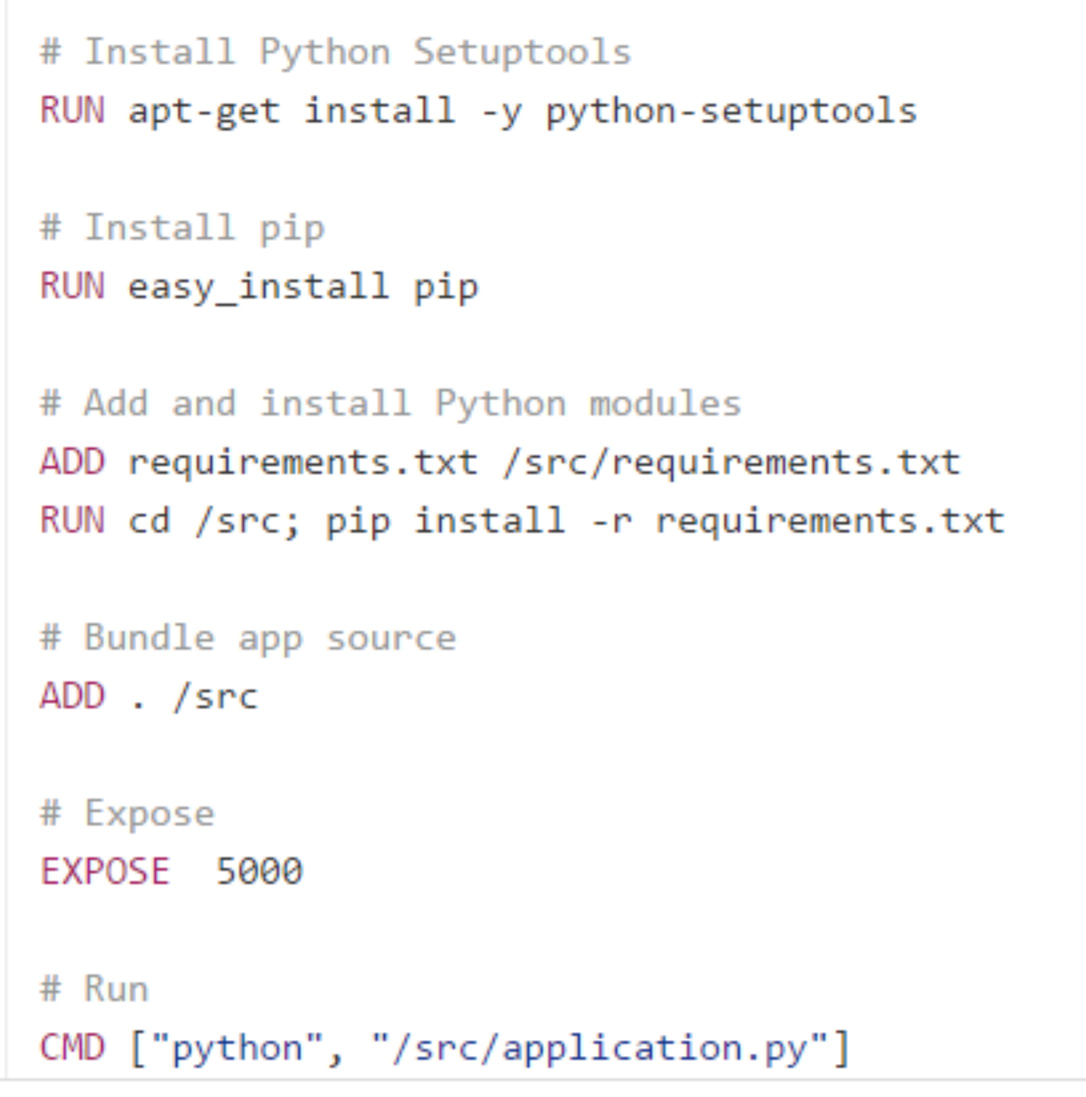
App

Hello World

- Distributed Datastore
- Built-in Memcache
- Integrated with Google Services
- Automatic CDN
- Auto scaling
- $(GAE) > $(AWS) for constant load

AWS with Docker

- Service deployment when the service is more than a simple web service
- Dockerfile is used to create an image
- Each image “boots” and executes the desired program

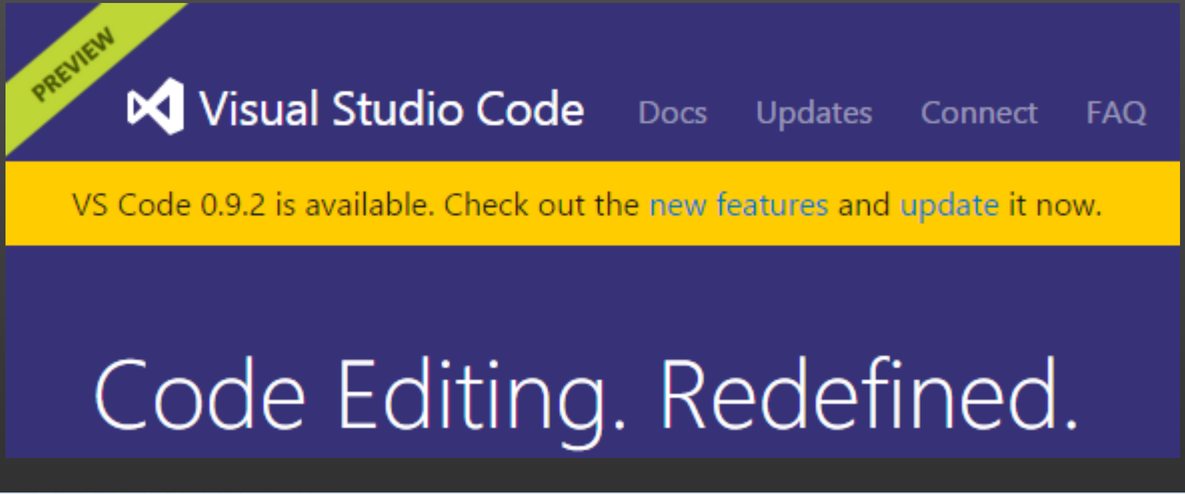


```
# Install Python Setuptools
RUN apt-get install -y python-setuptools

# Install pip
RUN easy_install pip

# Add and install Python modules
ADD requirements.txt /src/requirements.txt
RUN cd /src; pip install -r requirements.txt

# Bundle app source
ADD . /src

# Expose
EXPOSE  5000

# Run
CMD ["python", "/src/application.py"]
```

Search Engine Optimization

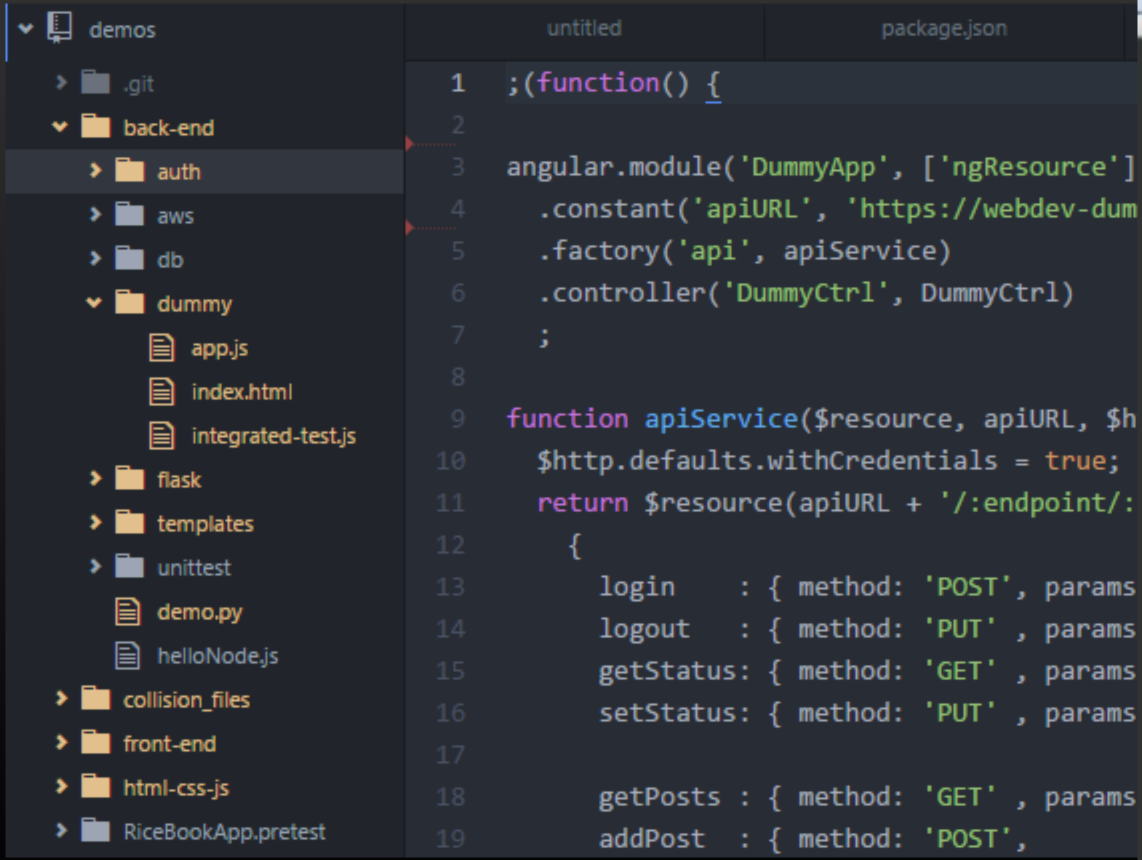
- Google bot crawls the web scraping content
- But the bot can't get AJAX
  - it doesn't follow fragments and has no state

- sitemap.xml – list of sites you want Google to index
- prerender pages served to Google – provide pre-determined content
  - can also help for user page rendering, specifically visitors (i.e., not logged in users)

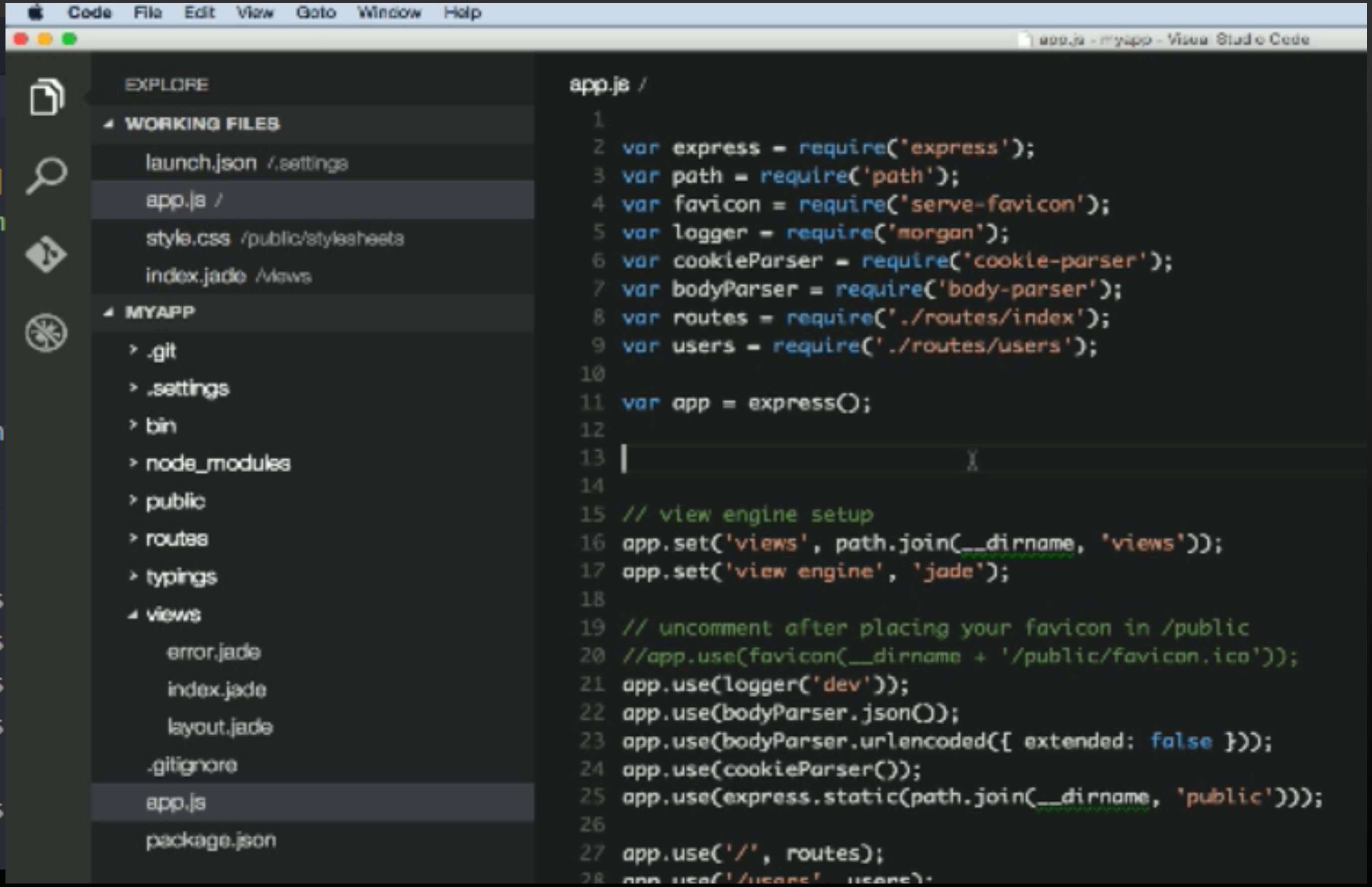


VS Code 0.9.2 is available. Check out the new features and update it now.

Code Editing. Redefined.



```
;(function() {

angular.module('DummyApp', ['ngResource']
  .constant('apiURL', 'https://webdev-dum
  .factory('api', apiService)
  .controller('DummyCtrl', DummyCtrl)
  ;

function apiService($resource, apiURL, $h
  $http.defaults.withCredentials = true;
  return $resource(apiURL + '/:endpoint/:
    {
      login     : { method: 'POST', params
      logout    : { method: 'PUT' , params
      getStatus: { method: 'GET' , params
      setStatus: { method: 'PUT' , params

      getPosts : { method: 'GET' , params
      addPost  : { method: 'POST',
```



```

var express = require('express');
var path = require('path');
var favicon = require('serve-favicon');
var logger = require('morgan');
var cookieParser = require('cookie-parser');
var bodyParser = require('body-parser');
var routes = require('./routes/index');
var users = require('./routes/users');

var app = express();



// view engine setup
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'jade');

// uncomment after placing your favicon in /public
//app.use(favicon(__dirname + '/public/favicon.ico'));
app.use(logger('dev'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: false }));
app.use(cookieParser());
app.use(express.static(path.join(__dirname, 'public')));

app.use('/', routes);
app.use('/users', users);
```