Javascript, The Swiss Army Knife of Programming Languages

David Morcillo

30-11-2013

Introduction

We are going to do our first steps on a multiplayer game using Websockets. First, we need a web server to serve our web application (our game) and handle communications between all clients connected to the server.

Introduction

We are going to do our first steps on a multiplayer game using Websockets. First, we need a web server to serve our web application (our game) and handle communications between all clients connected to the server.

Express.js

Express is a minimal and flexible node.js web application framework, providing a robust set of features for building single and multi-page, and hybrid web applications.

Express.js: Hello World!

Installation

```
npm install express --save-dev
```

Express.js: Hello World!

Installation

```
npm install express --save-dev
```

Hello World

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

app.get('/hello.txt', function(req, res){
    var body = 'Hello World';
    res.setHeader('Content-Type', 'text/plain');
    res.setHeader('Content-Length', body.length);
    res.end(body);
});

app.listen(9000);
console.log('Listening on port 9000');
```

Express.js: Hello World!

Installation

```
npm install express --save-dev
```

Hello World

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

app.get('/hello.txt', function(req, res){
    var body = 'Hello World';
    res.setHeader('Content-Type', 'text/plain');
    res.setHeader('Content-Length', body.length);
    res.end(body);
});

app.listen(9000);
console.log('Listening on port 9000');
```

Run server

```
$ node index.js // Open browser and visit http://localhost:9000
```

Express.js: API

Serving static files with app.use

```
app.use(express.static(__dirname + '/public'));
```

Express.js: API

Serving static files with app.use

```
app.use(express.static(__dirname + '/public'));
```

Using a template engine system

Install a template engine system. For example jade:

```
$ npm install jade --save-dev
```

Use it on our Express application:

```
app.set("view engine", "jade");
app.set("views", __dirname + "/views");
```

Express.js: API

Defining routes and rendering views

```
app.get('/about', function (req, res) {
    res.render('about');
});

app.get('/credits', function (req, res) {
  res.render('credits', { name: 'test' }); // Pass parameters to the view
});

app.post('/players', function (req, res) {
    res.render('players/show');
});

app.put('/players', function (req, res) {
    res.render('players/show');
});
```

Express.js: API

Parse query parameters

```
// GET /search?q=nintendo
app.get('/search', function (req, res) {
    var q = req.query.q;
});
```

Express.js: API

Parse query parameters

```
// GET /search?q=nintendo
app.get('/search', function (req, res) {
    var q = req.query.q;
});
```

Parse body

First, we need to use `bodyParser` middleware.

```
app.use(express.bodyParser());
```

Then, we can parse body directly:

```
// POST /players player[name]=David
app.post('/players', function (req, res) {
    var playerData = req.body.player,
        playerName = playerData.name;
});
```

Express.js: Lab

- `git checkout stage_8_1`
- Install your back-end dependencies with `npm install`
- Install your front-end dependencies with `bower install`
- Start `grunt watch` for auto linting
- Find TODOs and complete the exercise.