

Server-Side JS using Node.js



Non-blocking IO

Example 1 – Building non-blocks

```
/* Blocking/synchronous web servers (pseudo-code) */
roomName = request.roomName
roomInfo = request.roomInfo
if roomExists(roomName): //roomExists checks the rooms DB
    makeRoom(roomName, roomInfo) //writes to the DB
    return "{\"room\": \"" + roomName + "\"}"
else:
    return "{\"error\": \"room exists\"}"
```

```
/* Node.js: non-blocking/asynchronous web server */
var roomName = request.roomName;
roomExists(roomName, function(exists) {
    if(exists) {
        response.send({ error: "room exists" });
    } else {
        makeRoom(roomName, function(room) {
            room.roomInfo = request.roomInfo;
        });
        //response is sent before the room is fully created
        response.send({ room: roomName });
    }
});
```

Ping-pong server

Example 2 – Ping pong

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

//Makes a 6 second long page loop
http.createServer(function (req, res) {
  if(req.url === "ping") {
    res.writeHead(200, {'Refresh': '3; url=/pong.html'});
    res.end("Ping"); //Sends the text "Ping"
  } else if(req.url === "pong") {
    res.writeHead(200, {'Refresh': '3; url=/ping.html'});
    res.write("Pong"); //Writes "Pong"
    res.end(); //Sends the response
  } else {
    res.writeHead(200, {'Location': '1; url=/ping.html'});
    res.end();
  }
}).listen(4242, '127.0.0.1');
console.log('Navigate to http://127.0.0.1:4242/');
```

A slightly less boring “Hello world”

Example 3 – GET me a variable

```
var express = require("express");
var app = express(); //Creates a new Express app

//These includes are run when a request is received
//These includes are known as "middleware"
var __dirname = "/home/nhynes"
app.configure(function () {
  app.use(express.static(__dirname + '/static'));
  //serves resources not matched by a route
});

app.get("/", function(req, res) {
  var name = req.param("name"); //like /?name=Nick
  res.send("Hello " + name); //headers are auto-set
});

app.listen(4242); //starts the server
```

Routing using Express

```
//In general,  
app.VERB("route", function(request, response) { });  
//if the HTTP method is VERB and the request URL  
//matches route, then the function will be called
```

Example 4 – GET me a variable

```
var express = require("express");  
var app = express(); //Creates a new Express app  
  
app.get("/:clothingArticle/:type", function(req, res) {  
  var article = req.params.clothingArticle;  
  var articleType = req.params.type;  
  res.send("Your " + articleType + " " + article + " " +  
    "looks ridiculous.");  
});  
  
app.listen(4242); //starts the server
```

Informal form input

Example 5 – Forming a form of form processing

```
var express = require("express");
var app = express(); //Creates a new Express app

var clothes = {};

app.post("/:clothingArticle", function(req, res) {
  var article = req.params.clothingArticle;
  if(clothes[article] === undefined) {
    clothes[article] = [];
  }

  var articleType = req.param("type");
  var clothes[article].append(articleType);

  res.status(200).end(); //Shorthand way to respond
});

app.listen(4242); //starts the server
```

New clothes by POST (client-side)

Example 6 – Clothes by POST

```
//This is called when a form containing an input for  
//article of clothing and the article type is submitted  
function makeClothing() {  
    var article = $("#newClothingArticle").val();  
    var articleType = $("#newClothingArticleType").val();  
  
    $.post("/"+article, "type="+articleType)  
        .done(function() {  
            console.log("Enjoy your new clothes!");  
        })  
        .fail(function() {  
            console.log("Unable to make your clothing.  
                        This is probably for the best.");  
        });  
}
```

Example 7 – Putting it all together

```
var app = require("express")(); //Make a new Express app

var clothes = {};

app.get("/:clothingArticle", function(req, res) {
  //return a JSON Array of current types of this article
  var article = req.params.clothingArticle || [];
  res.send(clothes[article]);
});

app.get("/:clothingArticle/:type", function(req, res) {
  //Given both article and type, print a silly message
  var article = req.params.clothingArticle;
  var type = req.params.type;

  if(clothes[article] === undefined) {
    res.status(400).end("No such article of clothing");
  } else if(clothes[article].indexOf(type) === -1) {
    res.status(400).end("No such type of "+article);
  } else {
    res.send("Your " + type + " " + article +
      " looks ridiculous.");
  }
});

app.post("/:clothingArticle", function(req, res) {
  var article = req.params.clothingArticle;
  if(clothes[article] === undefined) {
    clothes[article] = [];
  }

  clothes[article].append(req.param("type"));
  res.status(200).end(); //Shorthand way to respond
});

app.listen(4242); //starts the server
```