

the
NODE FIRM

Copyright© 2013 The Node Firm. All Rights Reserved.

HTTP SERVER

In Node, HTTP server is a first-class citizen:

01_hello_world.js:

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

var server = http.createServer();

server.on('request', function(req, res) {
  res.writeHead(200, {'content-type': 'text/plain'});
  res.write('Hello ');
  res.write('World\n');
  res.end();
});

server.once('listening', function() {
  console.log('server is listening at http://localhost:8081');
});

server.listen(8081);
```

Here is an alternative and more terse version of this server:

02_hello_world_terse.js:

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

http.createServer(function(req, res) {
  res.setHeader('content-type', 'text/plain');
  res.end('Hello World\n');
}).listen(8081, function() {
  console.log('server is listening at http://localhost:8081');
});
```

REQUEST: INCOMINGMESSAGE

03_reflect_request.js:

```
var inspect = require('util').inspect;

require('http').createServer(function(req, res) {

  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');

  var inspected = {
    'url': req.url,
    'method': req.method,
    'headers': req.headers
  };

  res.end(inspect(inspected));
}).listen(8081, function() {
  console.log('server is listening at http://localhost:8081');
});
```

STREAMS

THE REQUEST STREAM

The request object is a readable stream, you can pipe it into a writable stream.

04_pipe_request_file.js:

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

var server = http.createServer(function(req, res) {
  var ws = createWriteStream();
  ws.write(req.method + ' ' + req.url + '\n');
  ws.write(JSON.stringify(req.headers) + '\n\n');
  req.pipe(ws);
  res.setHeader('content-type', 'text/plain');
  res.end(ws._file);
});

server.listen(8081, function() {
  console.log('server is listening at http://localhost:8081');
});

function createWriteStream() {
  var file = Math.floor(Math.random() * 999999) + '.txt';
  var ws = fs.createWriteStream(__dirname + '/' + file);
  ws._file = file;
  return ws;
}
```

The server responds with the filename where it saved the data

```
$ curl http://localhost:8081/abcdef -X POST --data "some data" -i
HTTP/1.1 200 OK
content-type: text/plain
Date: Wed, 03 Apr 2013 17:37:41 GMT
Connection: keep-alive
Transfer-Encoding: chunked

50861.txt
```


THE RESPONSE STREAM

The response object is a writable stream

This example pipes an HTTP request to the HTTP response:

05_pipe_to_response.js:

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

http.createServer(function(req, res) {
  res.setHeader('content-type', 'application/json');

  http.get('http://registry.npmjs.org/-/all', function(npmres) {
    npmres.pipe(res);
    npmres.on('data', function(d) {
      console.log('chunk with %d bytes', d.length);
    });
  });
}).listen(8081);
```

Pipe the entire NPM registry into the server response.

```
$ curl -i http://localhost:8081
```

SUMMARY

- HTTP is a fundamental part of node
- The request object is a readable stream
- The response object is a writable stream
- Headers are set with `response.setHeader(name, value)`
- Status code is set with `response.statusCode = x`