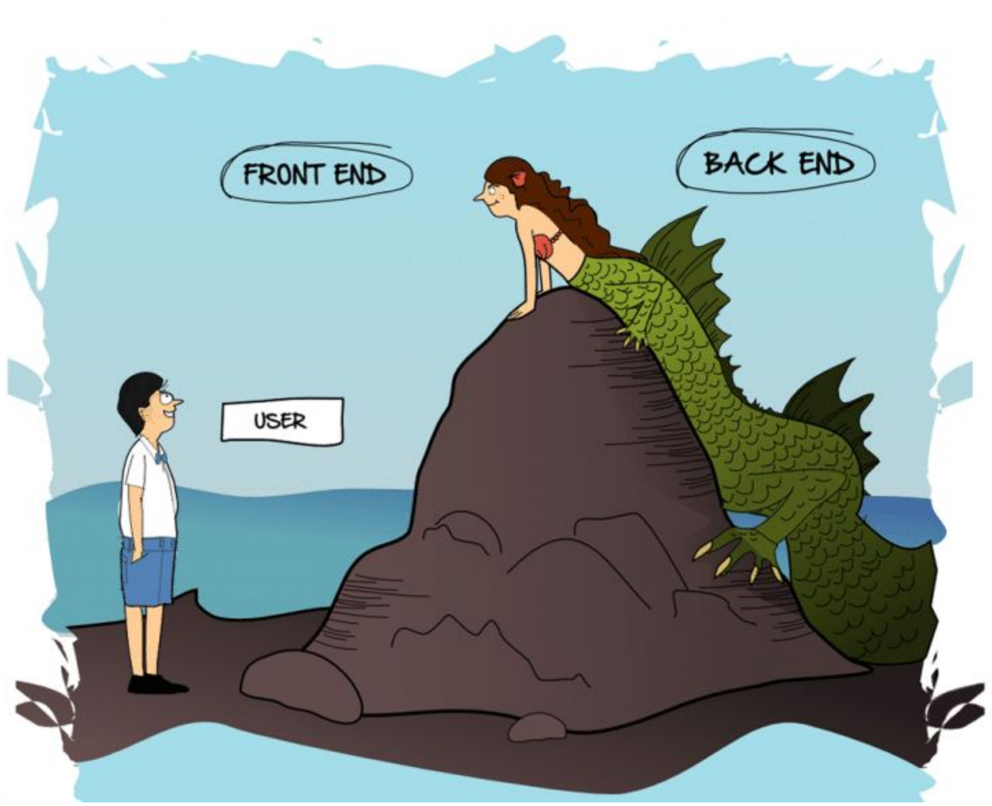
node

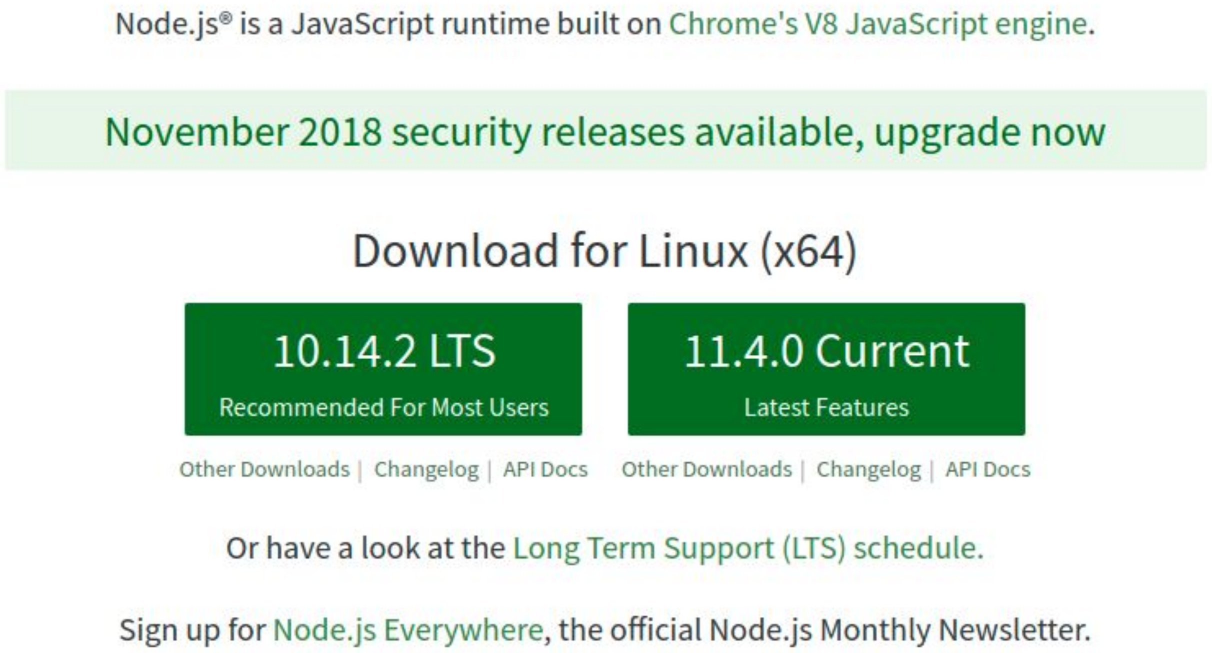
JS®

Год выпуска: **2009**
Движок: **V8**
Написан на: **C, C++**

FRONT END: HTML, CSS, JavaScript
BACK END: PHP, PYTHON, JAVA



https://nodejs.org/en/



Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine.

November 2018 security releases available, upgrade now

Download for Linux (x64)

10.14.2 LTS
Recommended For Most Users

11.4.0 Current
Latest Features

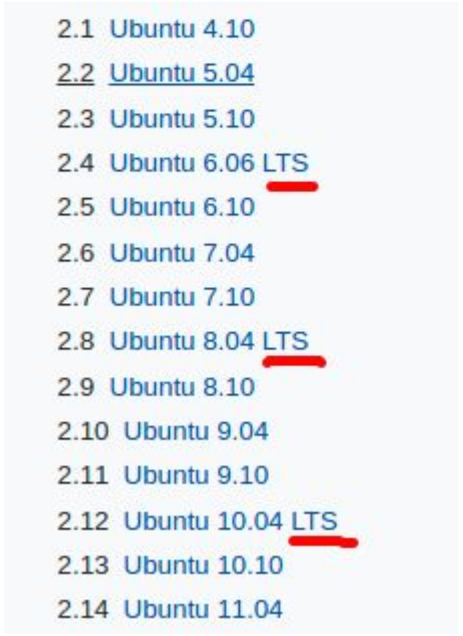
Other Downloads | Changelog | API Docs

Other Downloads | Changelog | API Docs

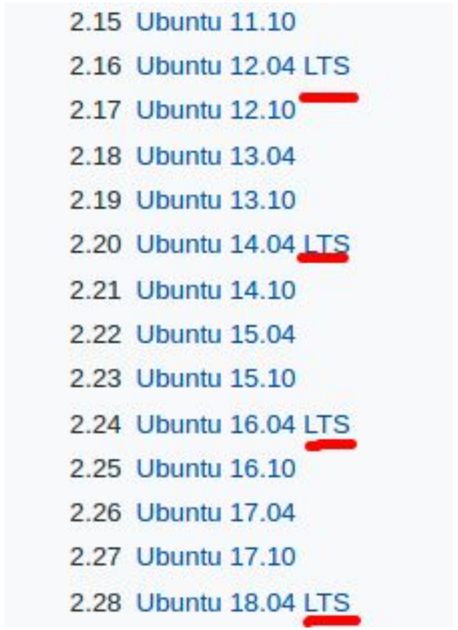
Or have a look at the Long Term Support (LTS) schedule.

Sign up for Node.js Everywhere, the official Node.js Monthly Newsletter.

LTS (англ. Long Term Support;



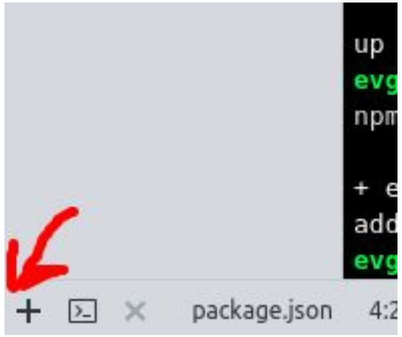
2.1 Ubuntu 4.10
2.2 Ubuntu 5.04
2.3 Ubuntu 5.10
2.4 Ubuntu 6.06 LTS
2.5 Ubuntu 6.10
2.6 Ubuntu 7.04
2.7 Ubuntu 7.10
2.8 Ubuntu 8.04 LTS
2.9 Ubuntu 8.10
2.10 Ubuntu 9.04
2.11 Ubuntu 9.10
2.12 Ubuntu 10.04 LTS
2.13 Ubuntu 10.10
2.14 Ubuntu 11.04



2.15 Ubuntu 11.10
2.16 Ubuntu 12.04 LTS
2.17 Ubuntu 12.10
2.18 Ubuntu 13.04
2.19 Ubuntu 13.10
2.20 Ubuntu 14.04 LTS
2.21 Ubuntu 14.10
2.22 Ubuntu 15.04
2.23 Ubuntu 15.10
2.24 Ubuntu 16.04 LTS
2.25 Ubuntu 16.10
2.26 Ubuntu 17.04
2.27 Ubuntu 17.10
2.28 Ubuntu 18.04 LTS

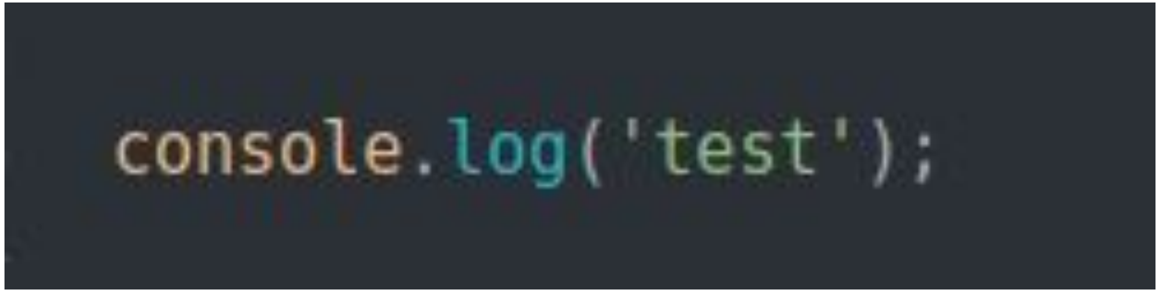
NPM (node packet manager)

https://atom.io/



Пакет platformio-ide-terminal

index.js



```
console.log('test');
```

in console:

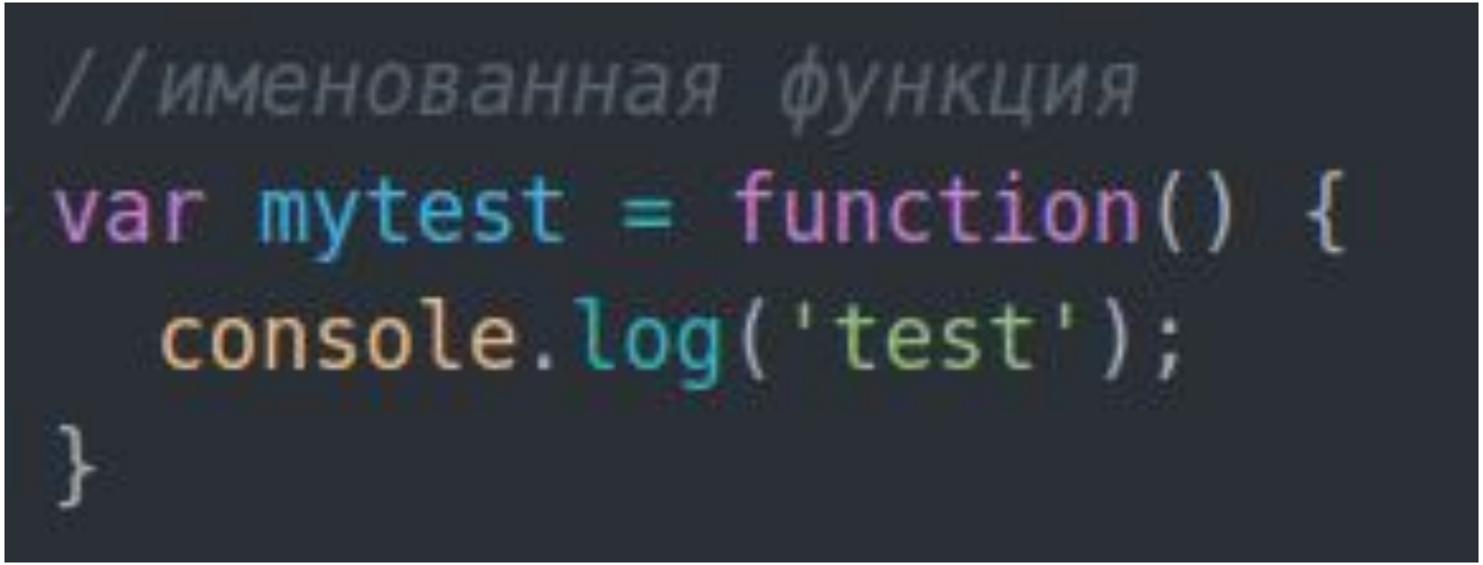
node file.js

__dirname

__filename

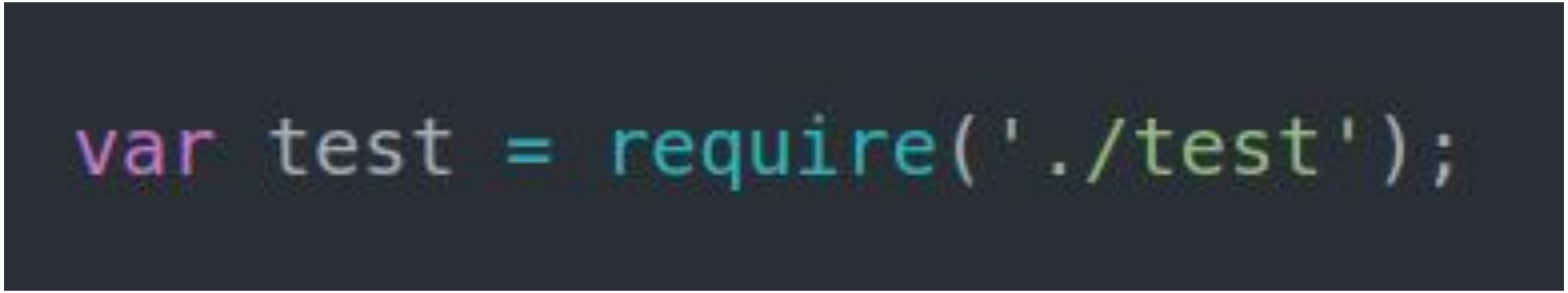
```
var timerId =
setInterval(function() {
  console.log( "тик" );
}, 2000);
```

Именованная функция

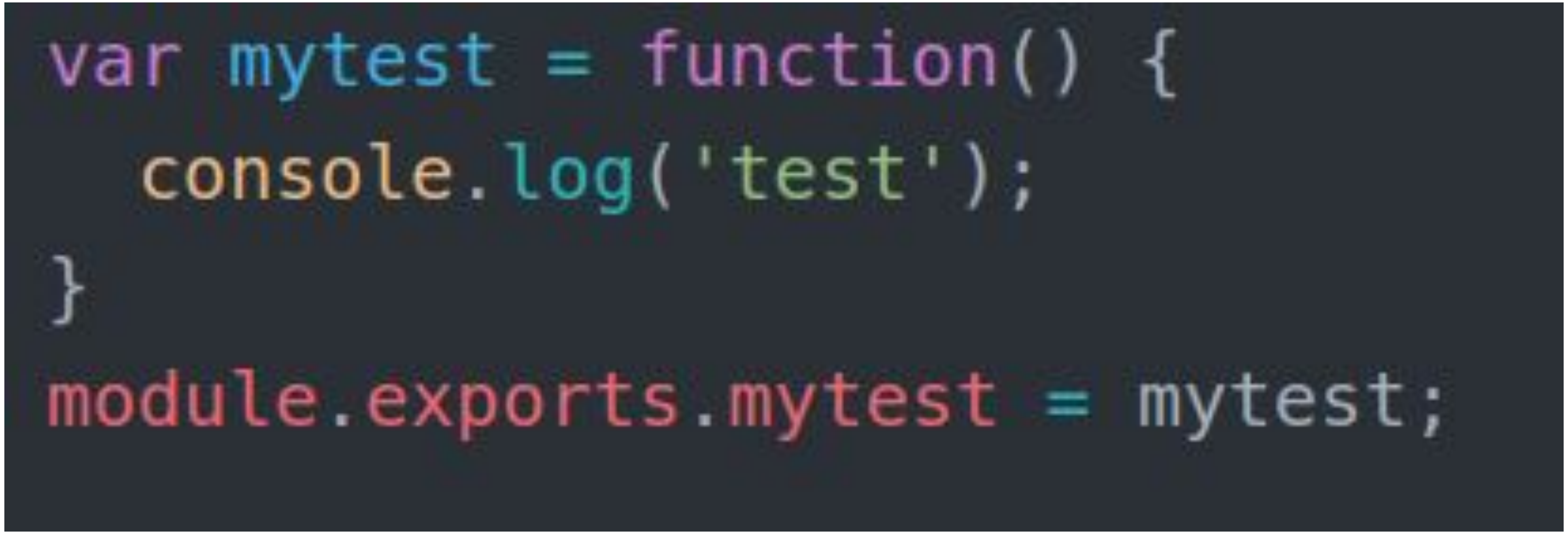


```
//именованная функция
var mytest = function() {
  console.log('test');
}
```

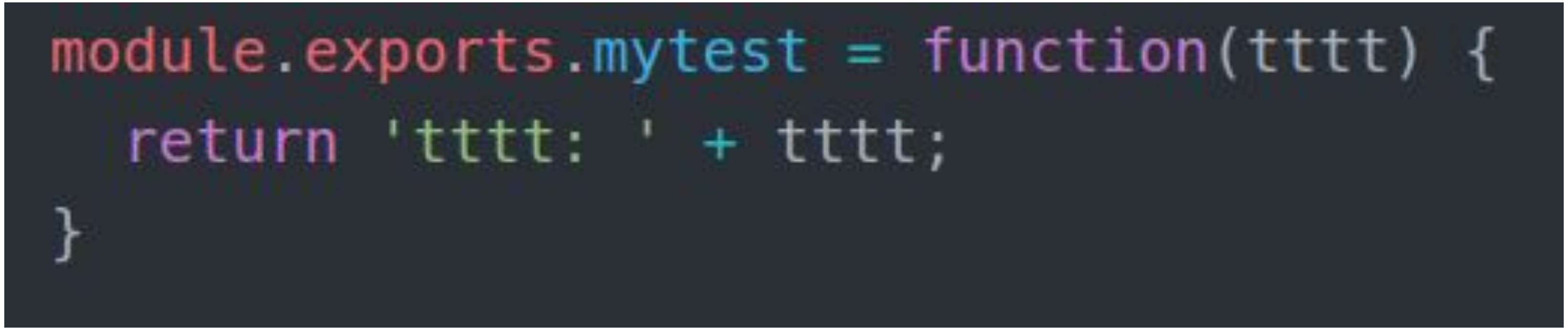
Модули



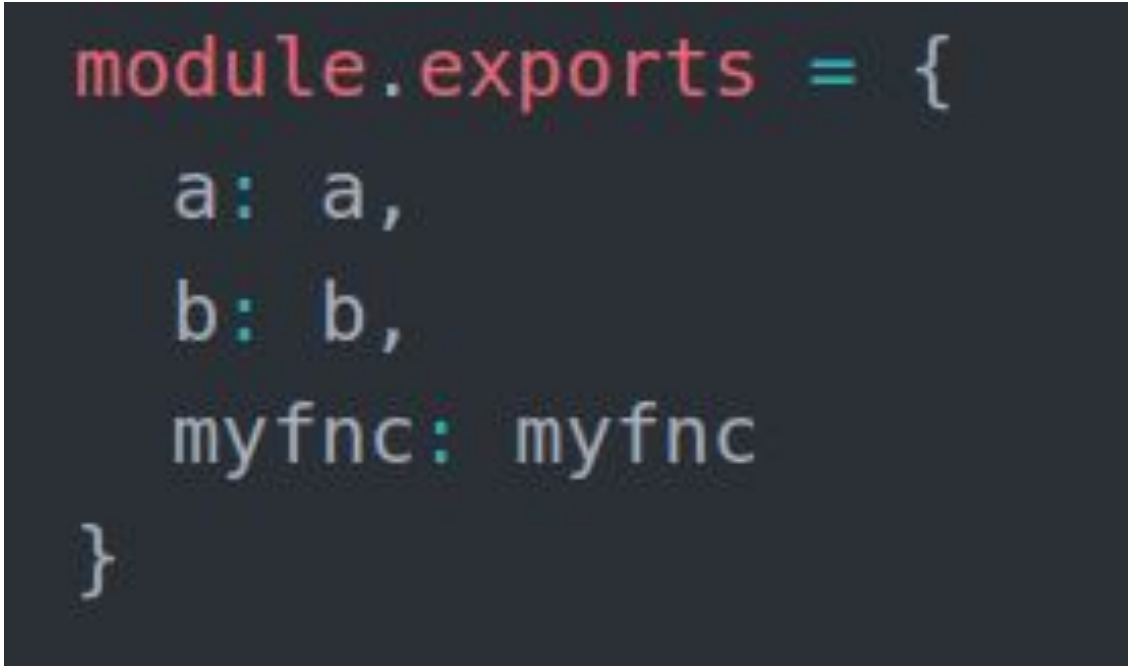
```
var test = require('./test');
```



```
var mytest = function() {
  console.log('test');
}
module.exports.mytest = mytest;
```

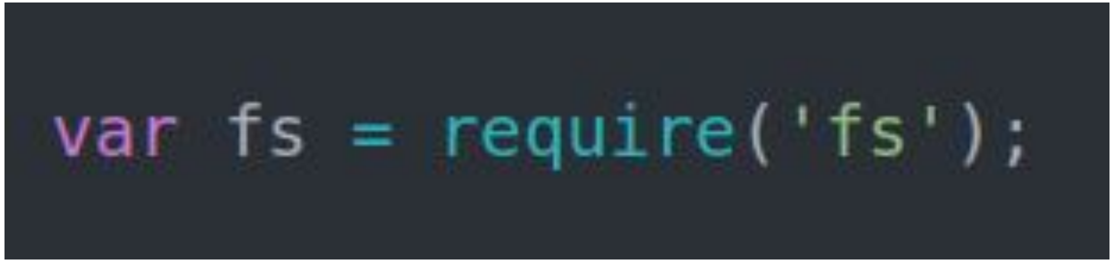


```
module.exports.mytest = function(tttt) {
  return 'tttt: ' + tttt;
}
```



```
module.exports = {
  a: a,
  b: b,
  myfnc: myfnc
}
```

Работа с файлами



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

```
//синхронно читаем из файла
var myfiletext = fs.readFileSync('text.txt', 'utf8');
console.log(myfiletext);
```

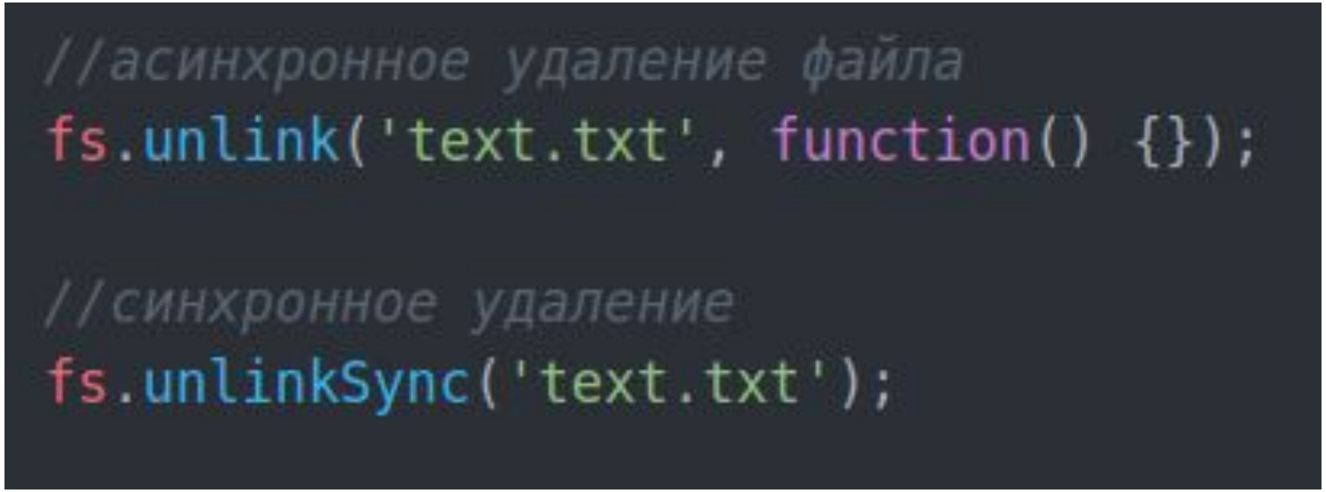
```
//синхронно пишем в файл
fs.writeFileSync('ololo.txt', "олололол");
```

Async

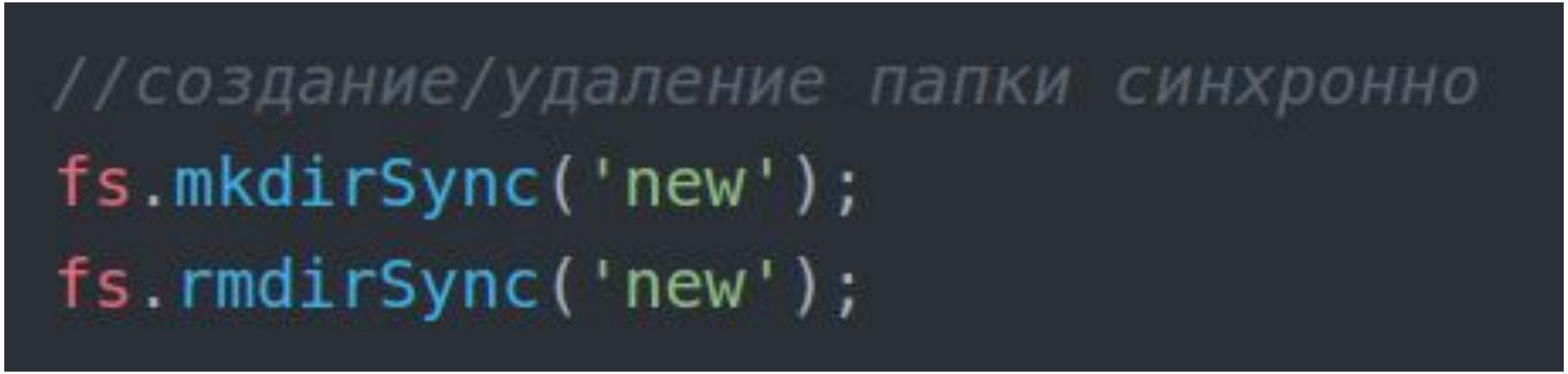
```
fs.readFile('text.txt', 'utf8', function(err, data) {
  console.log(data);
});

console.log('test');
```

```
fs.writeFile('text.txt', 'my text', function(err, data) {

});
```

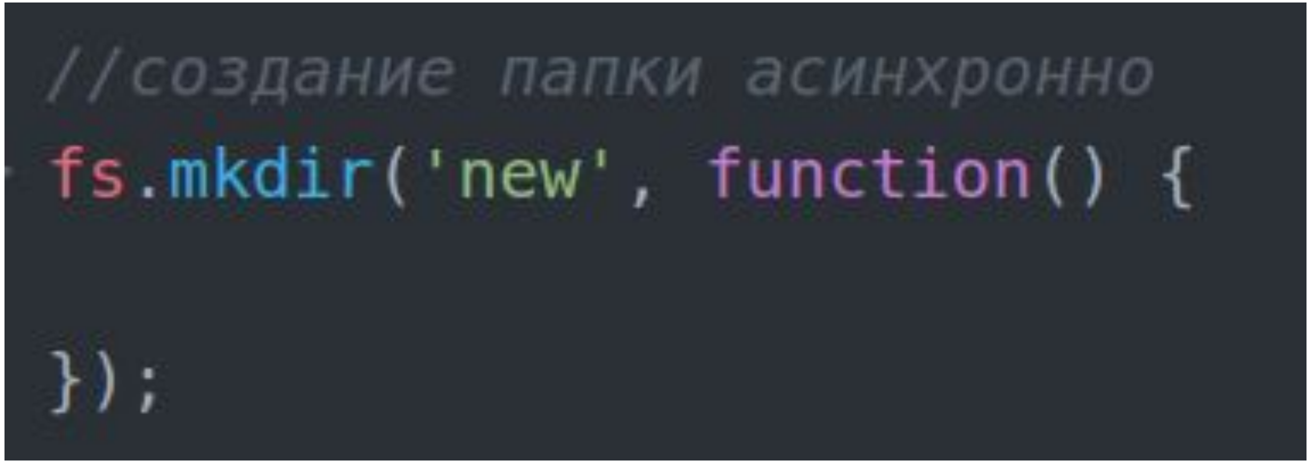
Работы с файлами



```
//асинхронное удаление файла
fs.unlink('text.txt', function() {});

//синхронное удаление
fs.unlinkSync('text.txt');
```



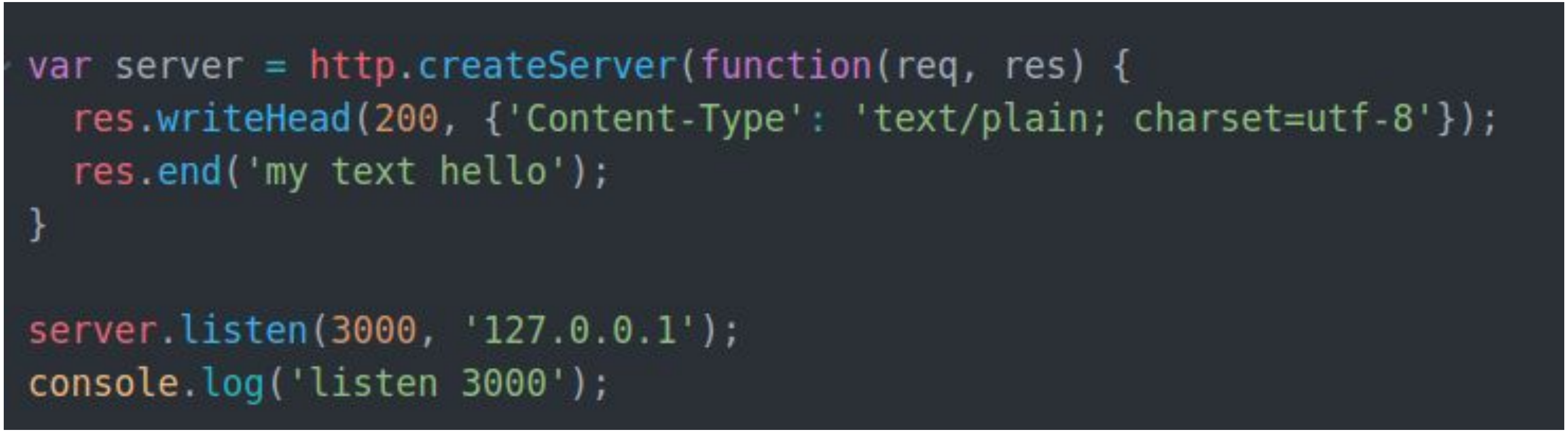
```
//создание/удаление папки синхронно
fs.mkdirSync('new');
fs.rmdirSync('new');
```



```
//создание папки асинхронно
fs.mkdir('new', function() {


});
```

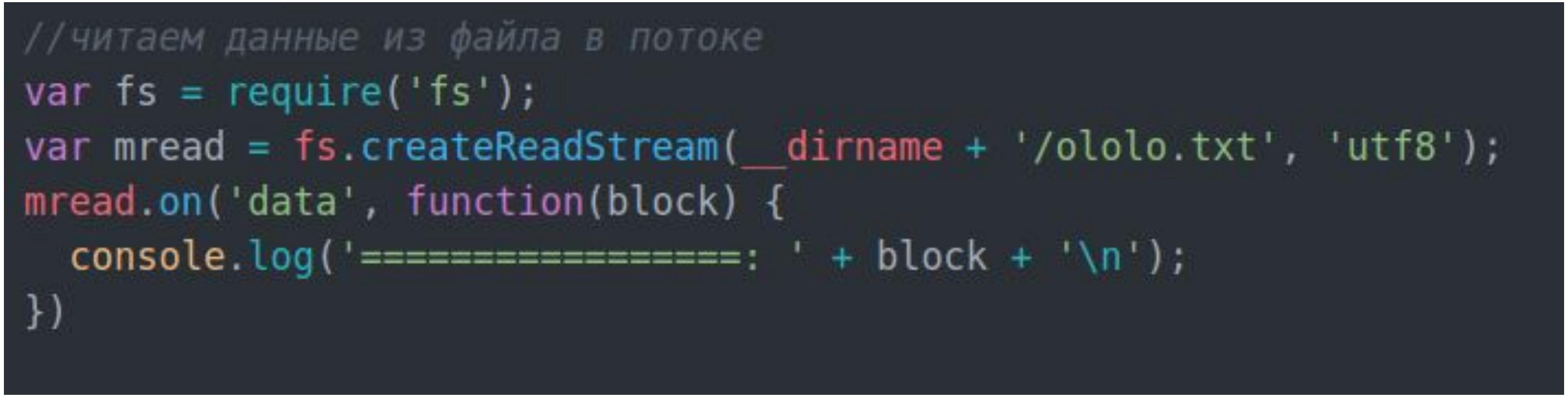
SERVER

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

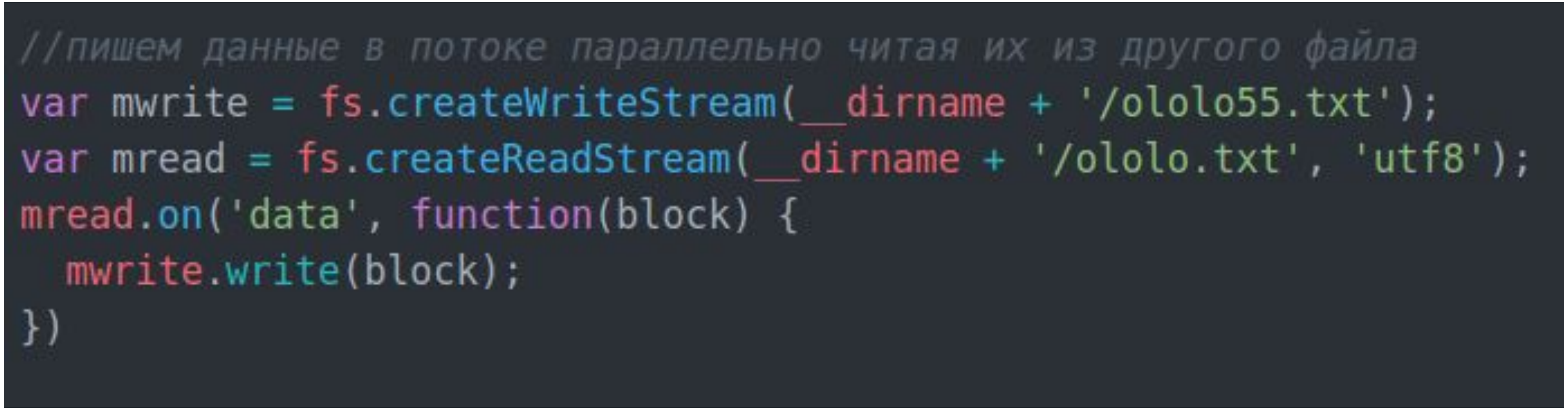


```
var server = http.createServer(function(req, res) {
  res.writeHead(200, {'Content-Type': 'text/plain; charset=utf-8'});
  res.end('my text hello');
}

server.listen(3000, '127.0.0.1');
console.log('listen 3000');
console.log('URL' + req.url);
```

Потоки

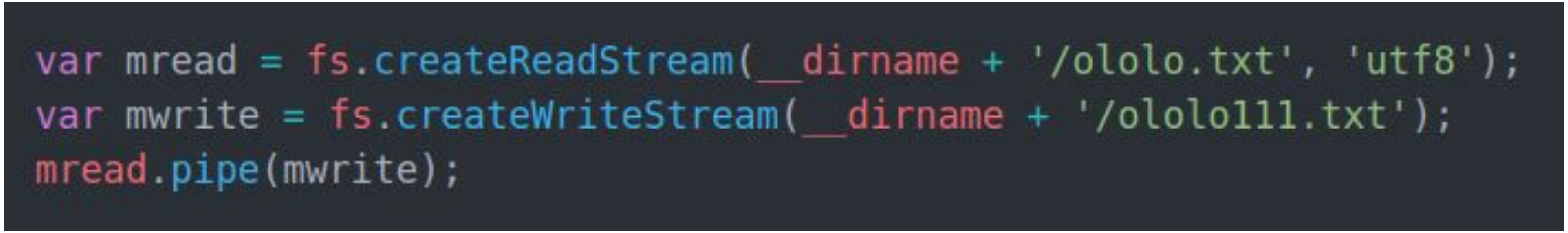


```
//читаем данные из файла в потоке
var fs = require('fs');
var mread = fs.createReadStream(__dirname + '/ololo.txt', 'utf8');
mread.on('data', function(block) {
  console.log('===================: ' + block + '\n');
})
```

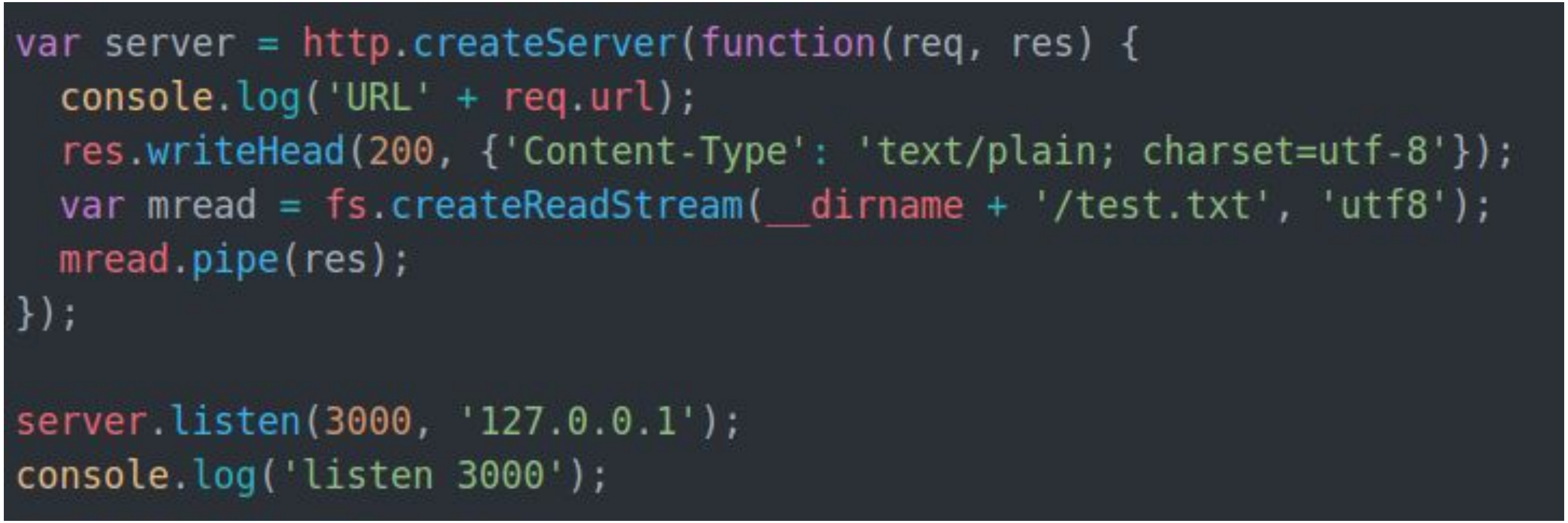


```
//пишем данные в потоке параллельно читая их из другого файла
var mwrite = fs.createWriteStream(__dirname + '/ololo55.txt');
var mread = fs.createReadStream(__dirname + '/ololo.txt', 'utf8');
mread.on('data', function(block) {
  mwrite.write(block);
})
```

PIPE



```
var mread = fs.createReadStream(__dirname + '/ololo.txt', 'utf8');
var mwrite = fs.createWriteStream(__dirname + '/ololo111.txt');
mread.pipe(mwrite);
```

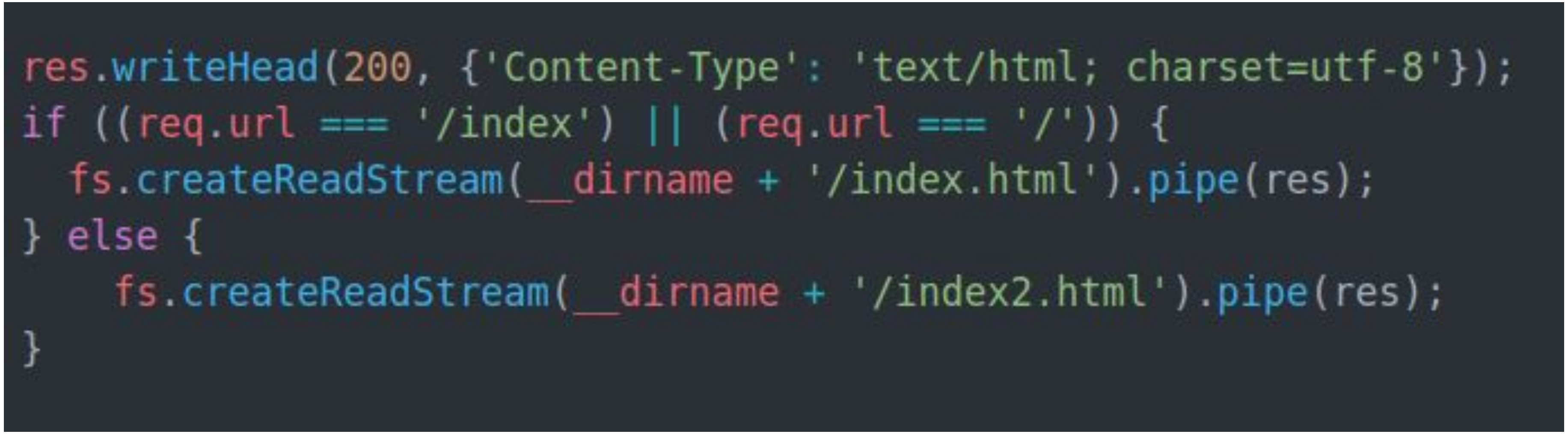


```
var server = http.createServer(function(req, res) {
  console.log('URL' + req.url);
  res.writeHead(200, {'Content-Type': 'text/plain; charset=utf-8'});
  var mread = fs.createReadStream(__dirname + '/test.txt', 'utf8');
  mread.pipe(res);
});

server.listen(3000, '127.0.0.1');
console.log('listen 3000');
```

```
res.writeHead(200, {'Content-Type': 'text/html; charset=utf-8'});
```

```
res.writeHead(200, {'Content-Type': 'application/json; charset=utf-8'});
var obj = {
  a: 'AAAAA',
  b: 'BBBBB',
  c: 'CCCCC'
}
res.end(JSON.stringify(obj));
```

Маршрутизация

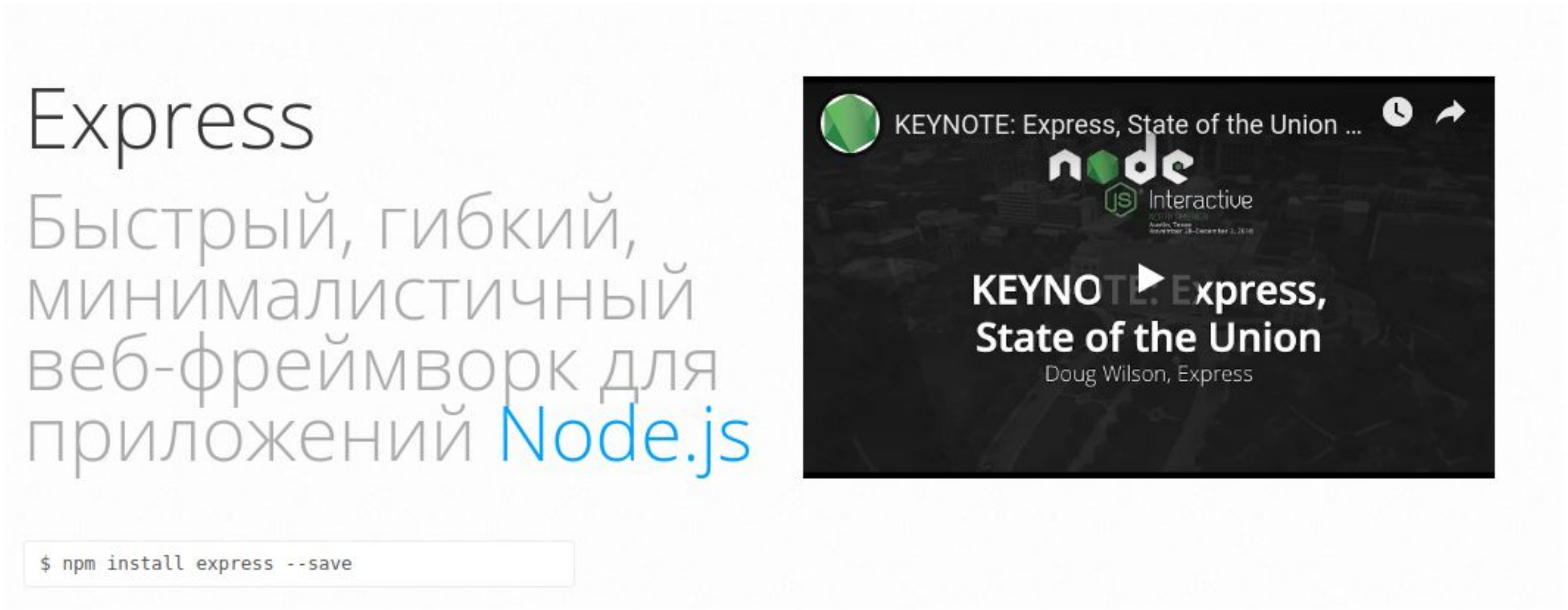


```
res.writeHead(200, {'Content-Type': 'text/html; charset=utf-8'});
if ((req.url === '/index') || (req.url === '/')) {
  fs.createReadStream(__dirname + '/index.html').pipe(res);
} else {
    fs.createReadStream(__dirname + '/index2.html').pipe(res);
}
```

```
npm init

npm install packet_name
npm uninstall packet_name

npm install packet_name -save


npm install
```

package.json

https://expressjs.com/ru/

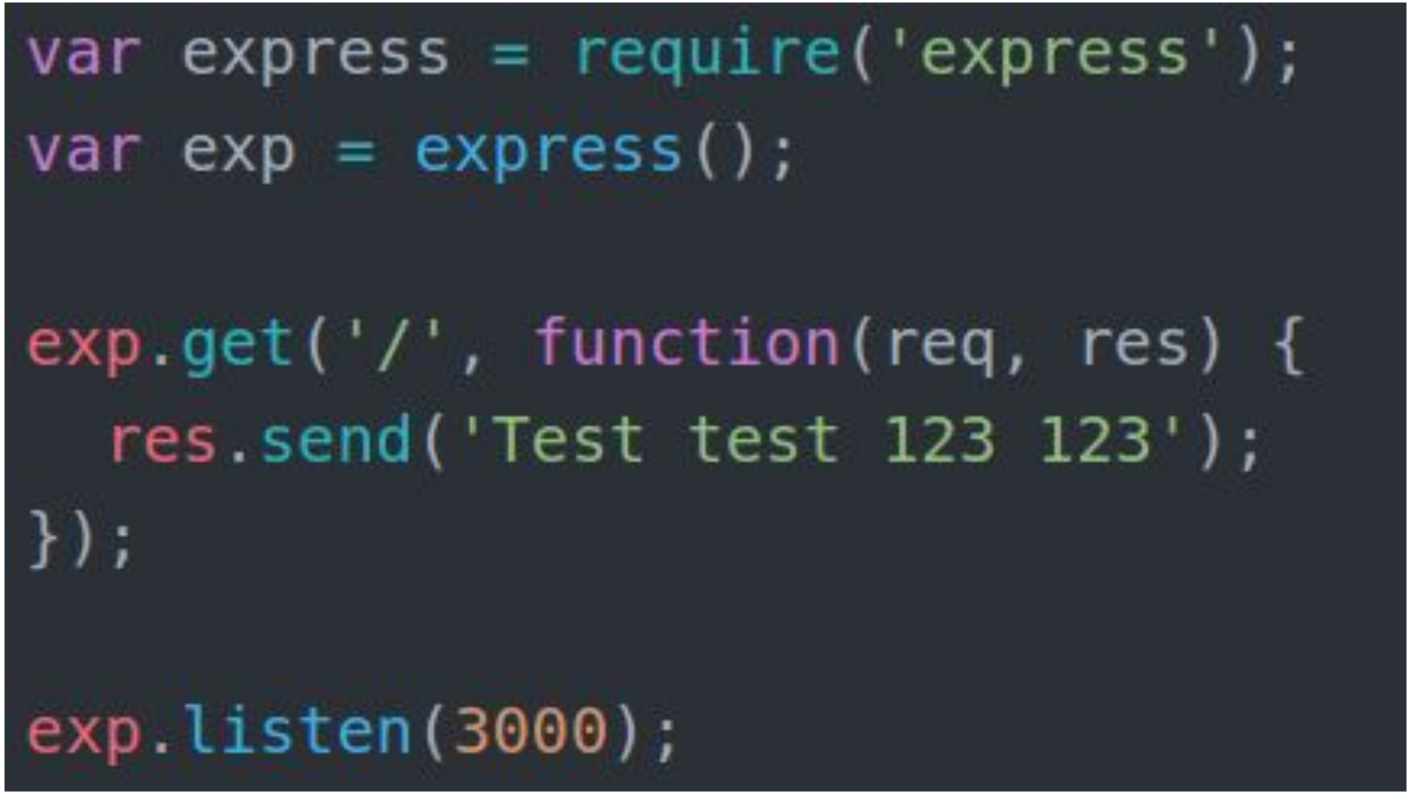


Express

Быстрый, гибкий, минималистичный веб-фреймворк для приложений Node.js

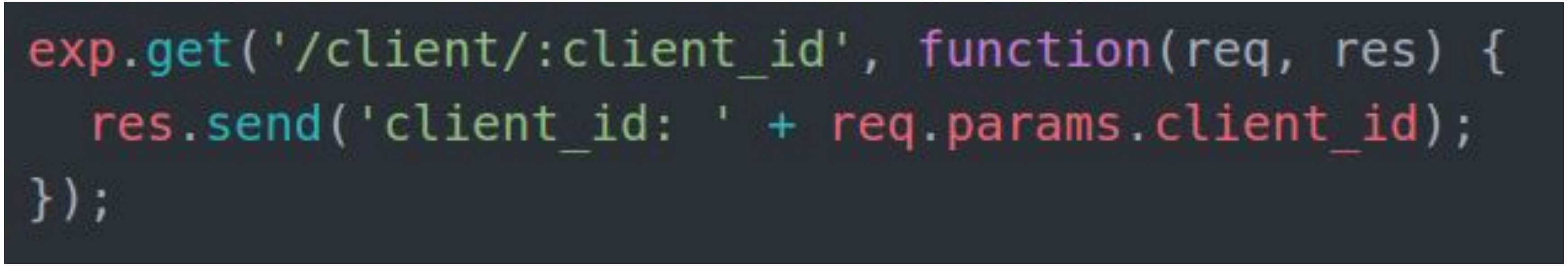
```
$ npm install express --save
```

KEYNOTE: Express, State of the Union ...
node
JS Interactive
KEYNOTE: Express, State of the Union
Doug Wilson, Express

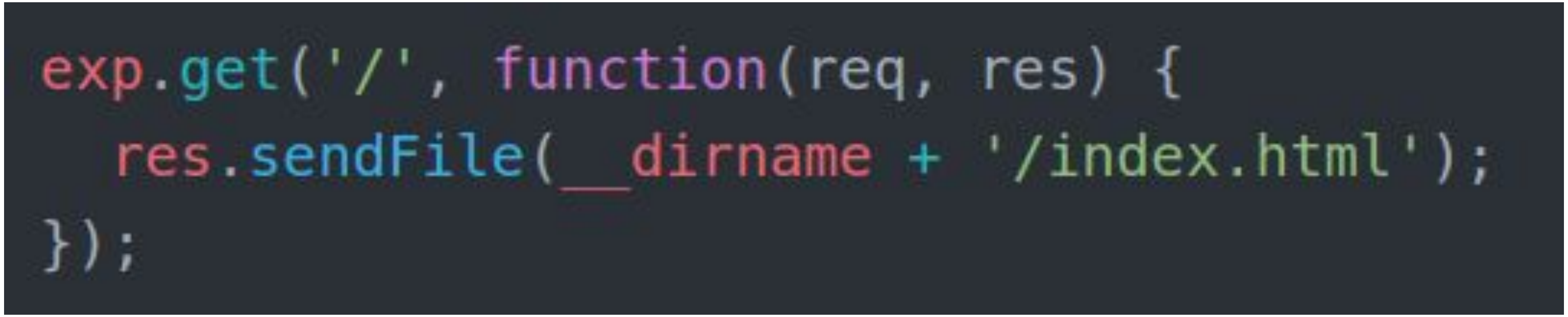


```
var express = require('express');
var exp = express();

exp.get('/', function(req, res) {
  res.send('Test test 123 123');
});

exp.listen(3000);
```

Dynamic URL



```
exp.get('/client/:client_id', function(req, res) {
  res.send('client_id: ' + req.params.client_id);
});
```



```
exp.get('/', function(req, res) {
  res.sendFile(__dirname + '/index.html');
});
```

type

sendStatus

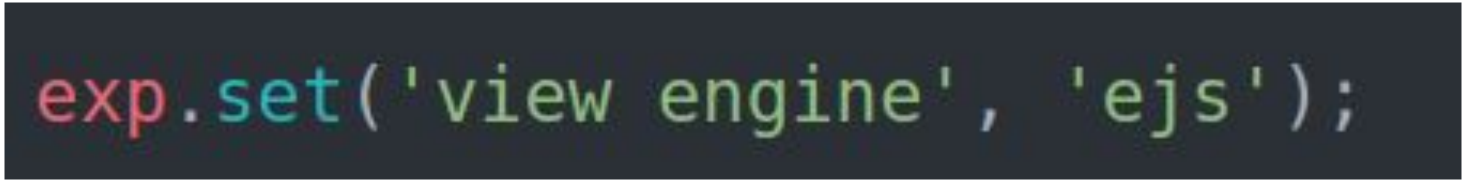
```
res.status(500).send('Something broke!');
```

http://expressjs.com/ru/api.html

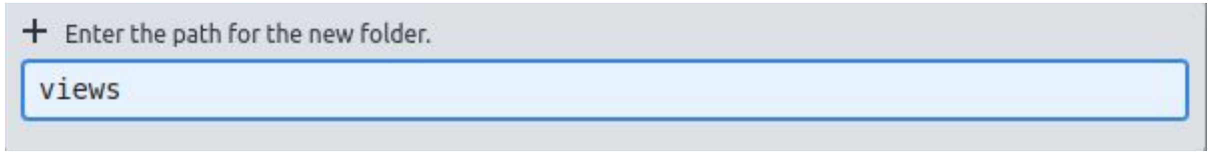
Express JS

Шаблонизатор https://ejs.co/

```
npm
install
ejs
```



```
exp.set('view engine', 'ejs');
```

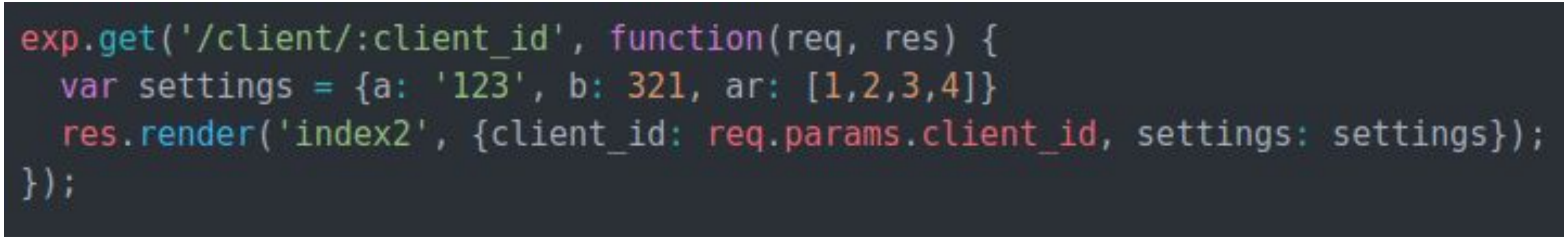


Enter the path for the new folder.

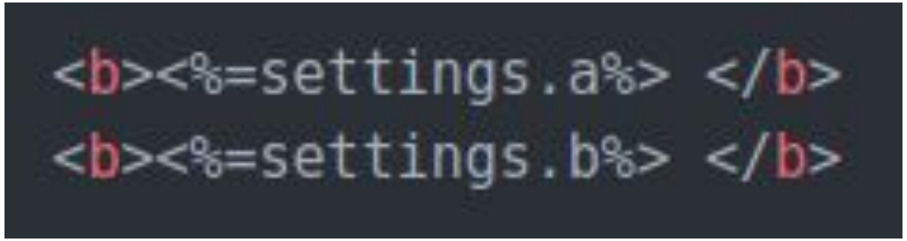
```
views
```

```
exp.get('/client/:client_id', function(req, res) {
  res.render('index2', {client_id: req.params.client_id});
});
```

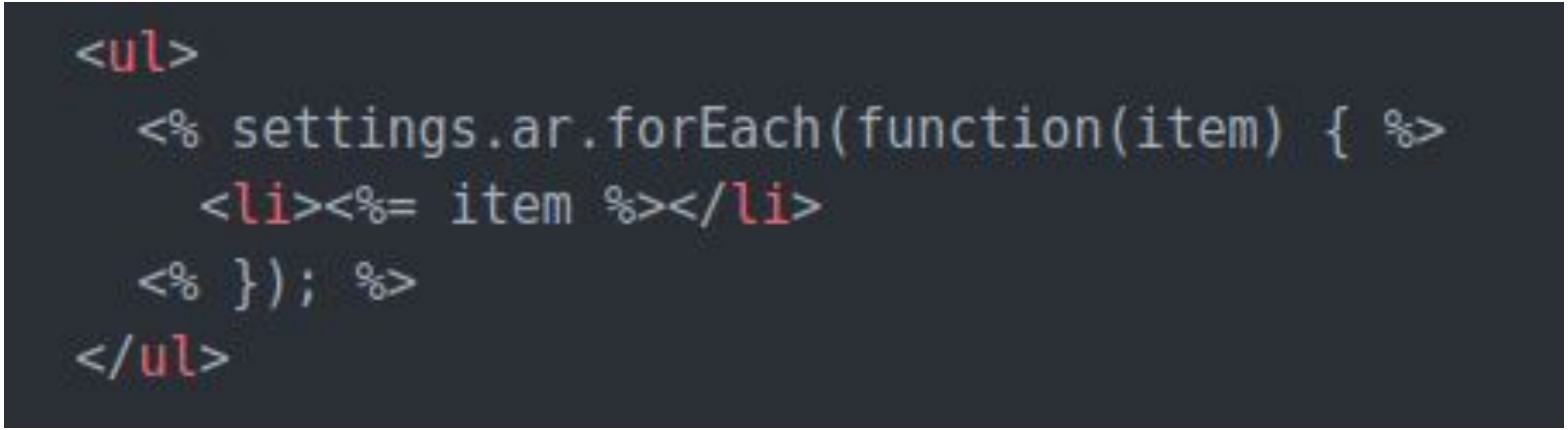
```
<body>
  <h1>CLIENT_ID: <%=client_id%> </h1>
</body>
```



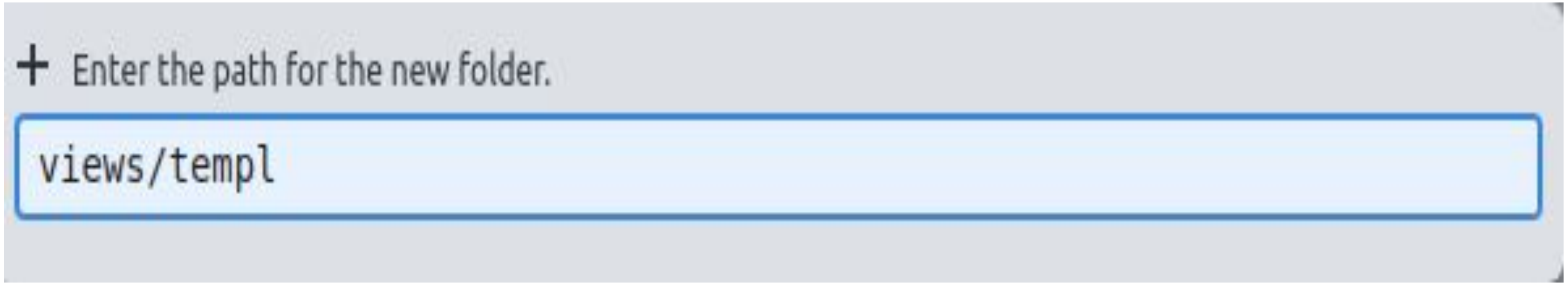
```
exp.get('/client/:client_id', function(req, res) {
  var settings = {a: '123', b: 321, ar: [1,2,3,4]}
  res.render('index2', {client_id: req.params.client_id, settings: settings});
});
```



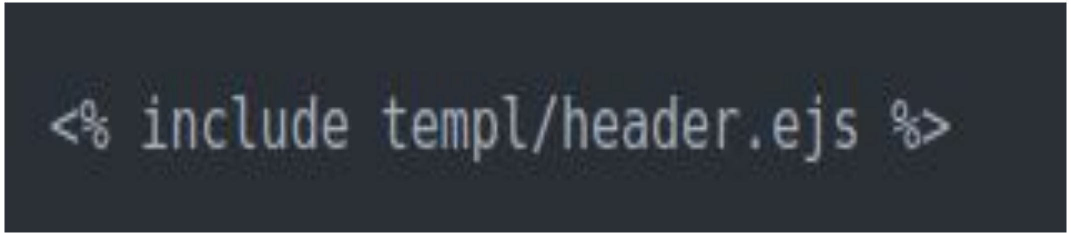
```
<b><%=settings.a%> </b>
<b><%=settings.b%> </b>
```



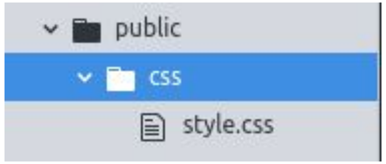
```
<ul>
  <% settings.ar.forEach(function(item) { %>
    <li><%= item %></li>
  <% }); %>
</ul>
```


+ Enter the path for the new folder.

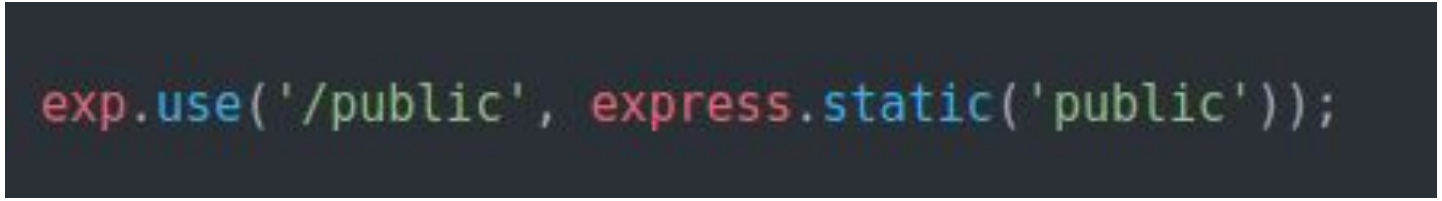
```
views/templ
```



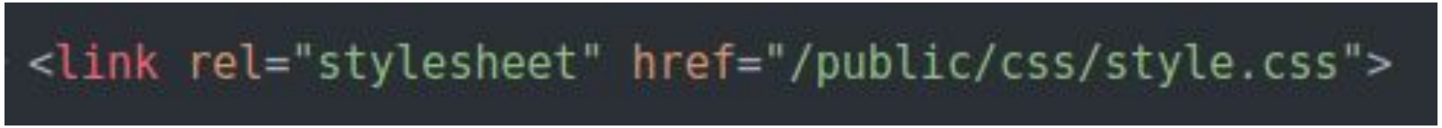
```
<% include templ/header.ejs %>
```



```
public
  css
    style.css
```



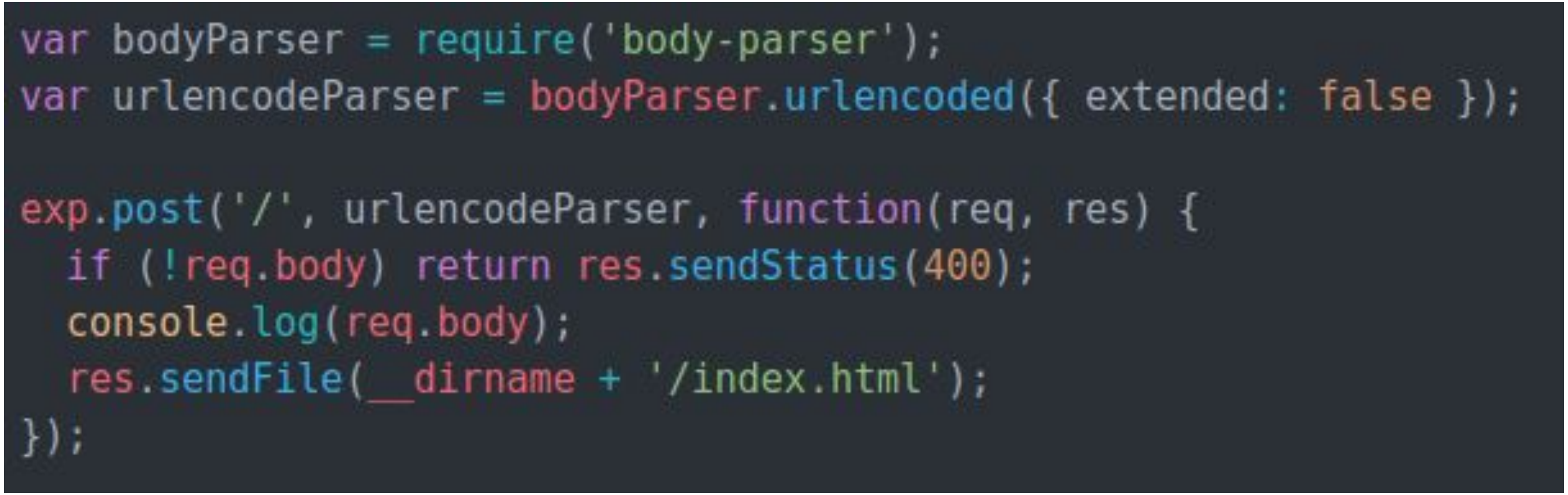
```
exp.use('/public', express.static('public'));
```



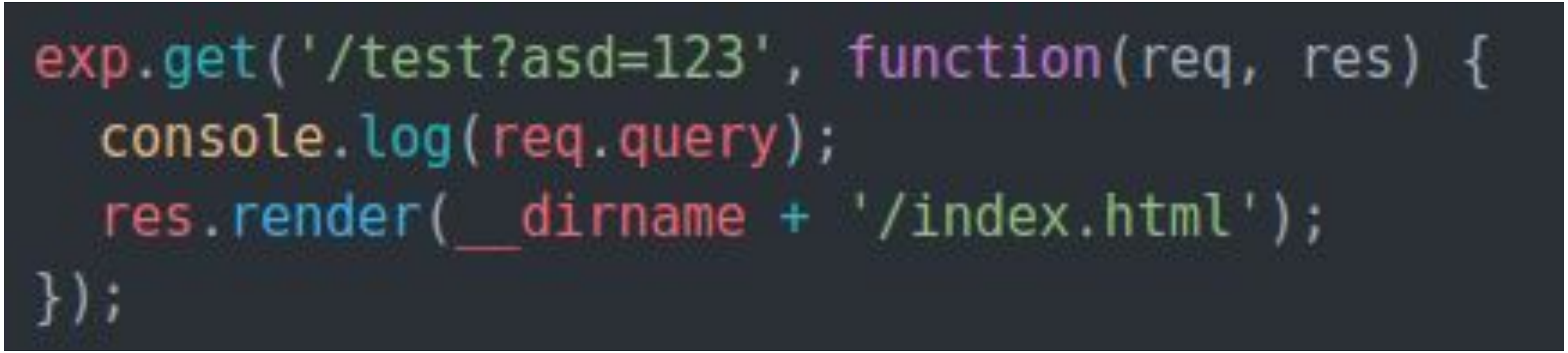
```
<link rel="stylesheet" href="/public/css/style.css">
```

Получить данные из формы

npm install body-parser



```
var bodyParser = require('body-parser');
var urlencodeParser = bodyParser.urlencoded({ extended: false });

exp.post('/', urlencodeParser, function(req, res) {
  if (!req.body) return res.sendStatus(400);
  console.log(req.body);
  res.sendFile(__dirname + '/index.html');
});
```



```
exp.get('/test?asd=123', function(req, res) {
  console.log(req.query);
  res.render(__dirname + '/index.html');
});
```

https://mongoosejs.com/

elegant mongodb object modeling for node.js

```
npm
install
mongoose
```

https://mlab.com

mLab

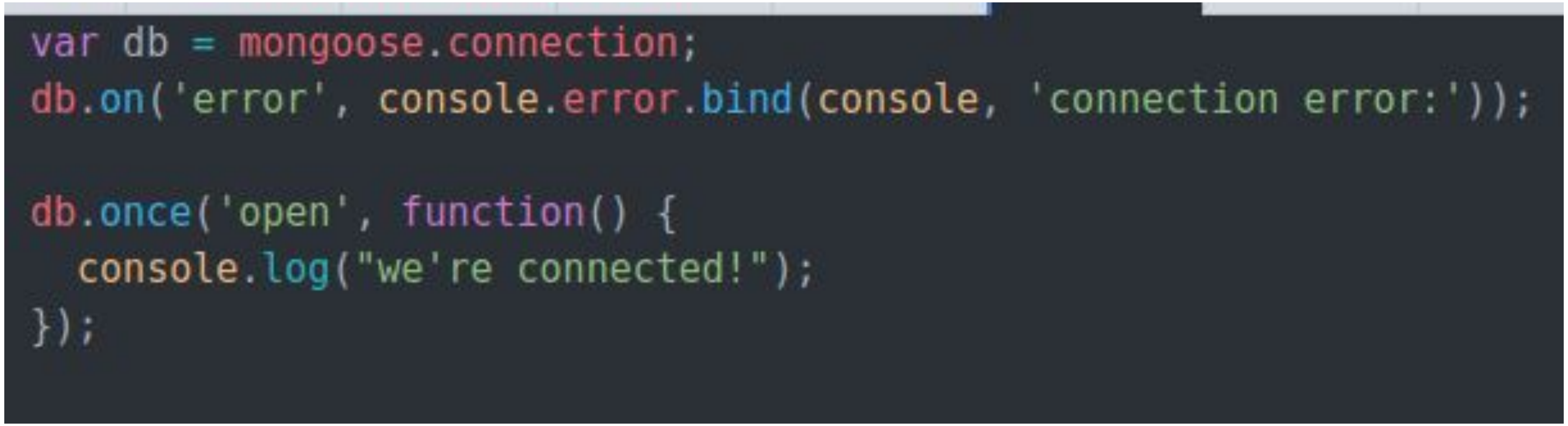
A MongoDB Company

DATABASE
COLLECTION
USER

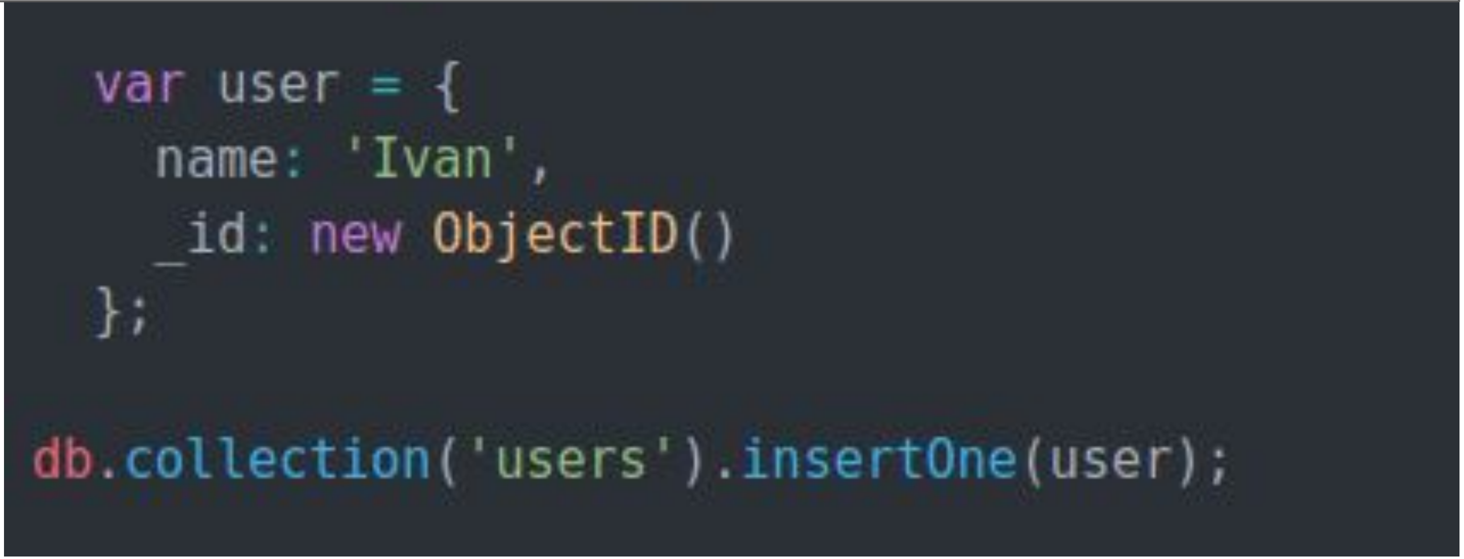
mongoDB

CONNECTION

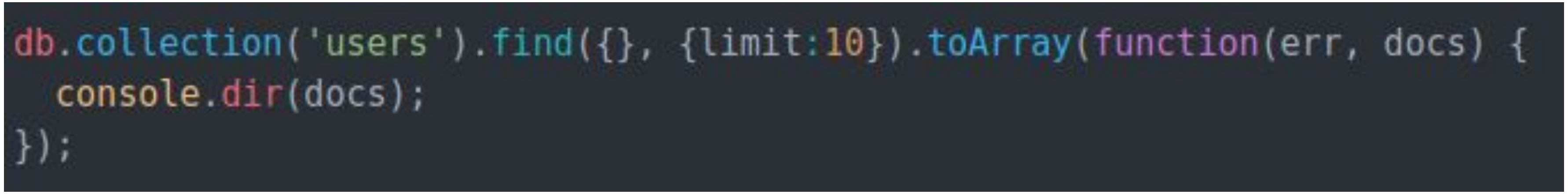
```
const mongoose = require('mongoose');

mongoose.connect('mongodb://evgeniy:qwerty1
23@ds125422.mlab.com:25422/testdb',
{useNewUrlParser: true });
```



```javascript
var db = mongoose.connection;
db.on('error', console.error.bind(console, 'connection error:'));

db.once('open', function() {
  console.log("we're connected!");
});
```

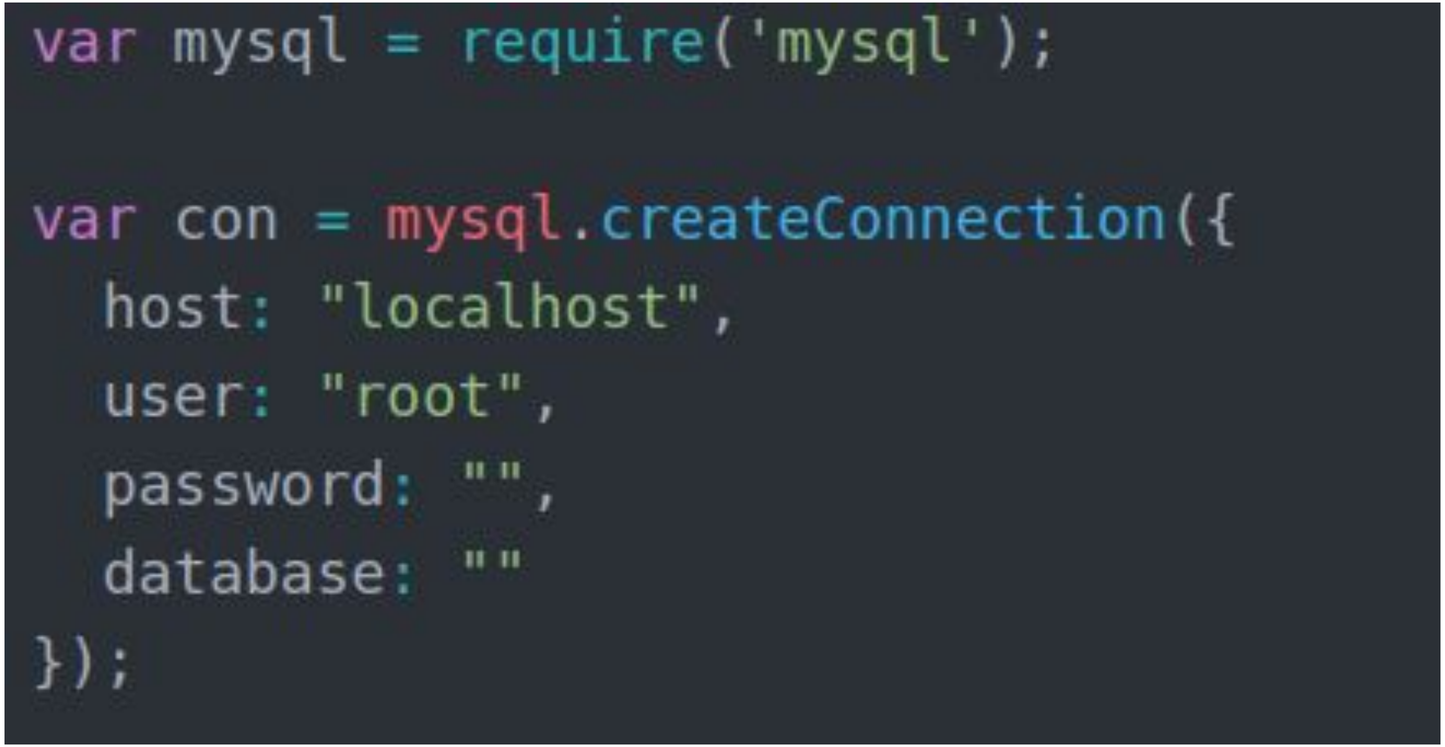


```
  var user = {
    name: 'Ivan',
    _id: new ObjectID()
  };

db.collection('users').insertOne(user);
```



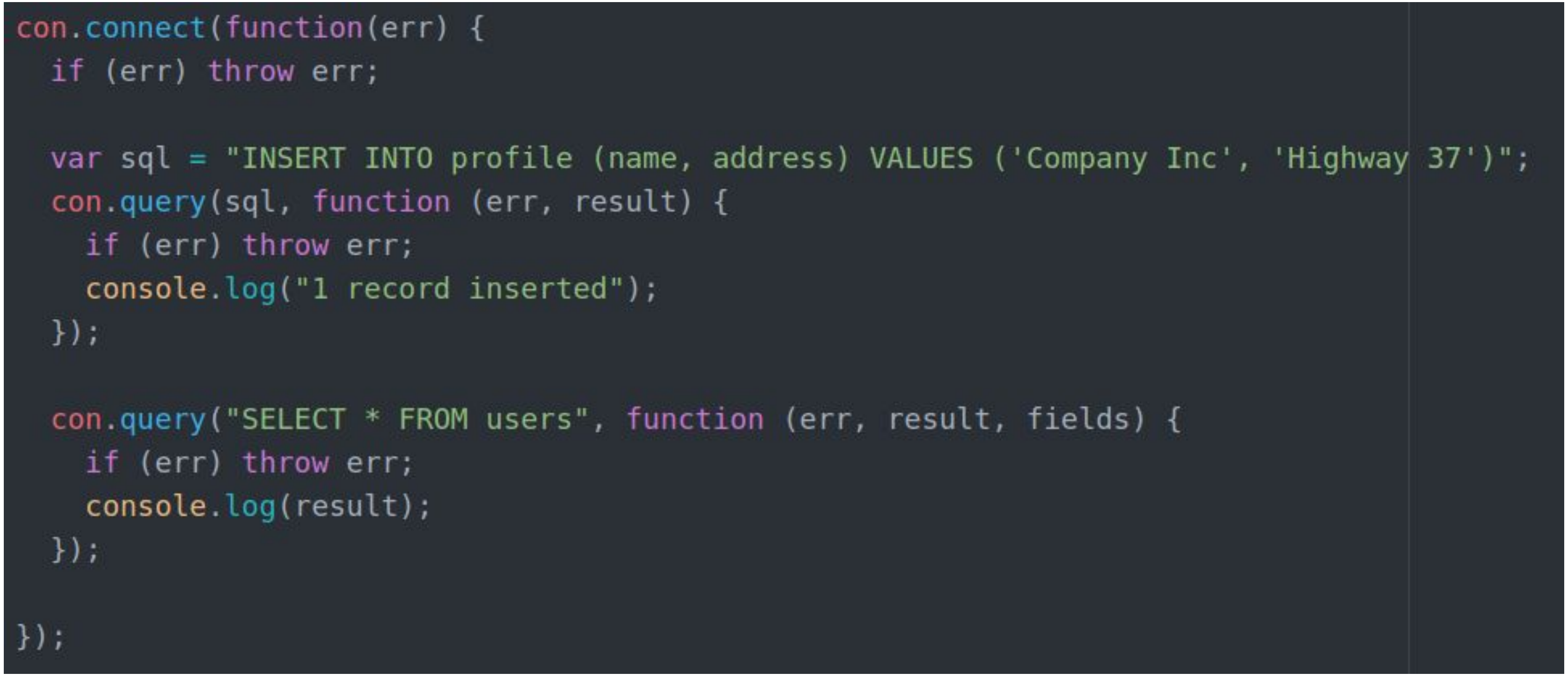
```
db.collection('users').find({}, {limit:10}).toArray(function(err, docs) {
  console.dir(docs);
});
```

NPM INSTALL MYSQL



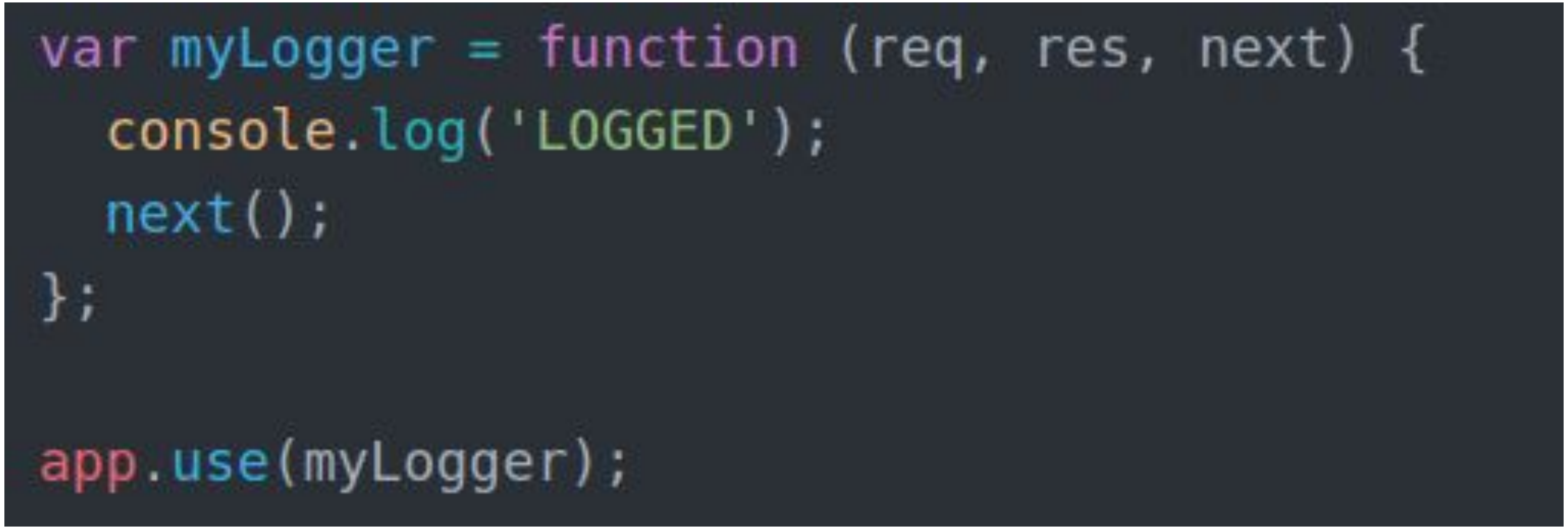
```
var mysql = require('mysql');

var con = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "",
  database: ""
});
```

```
host: "82.200.170.122",
user: "testuser",
password: "qwerty123123",
database: "test"
```

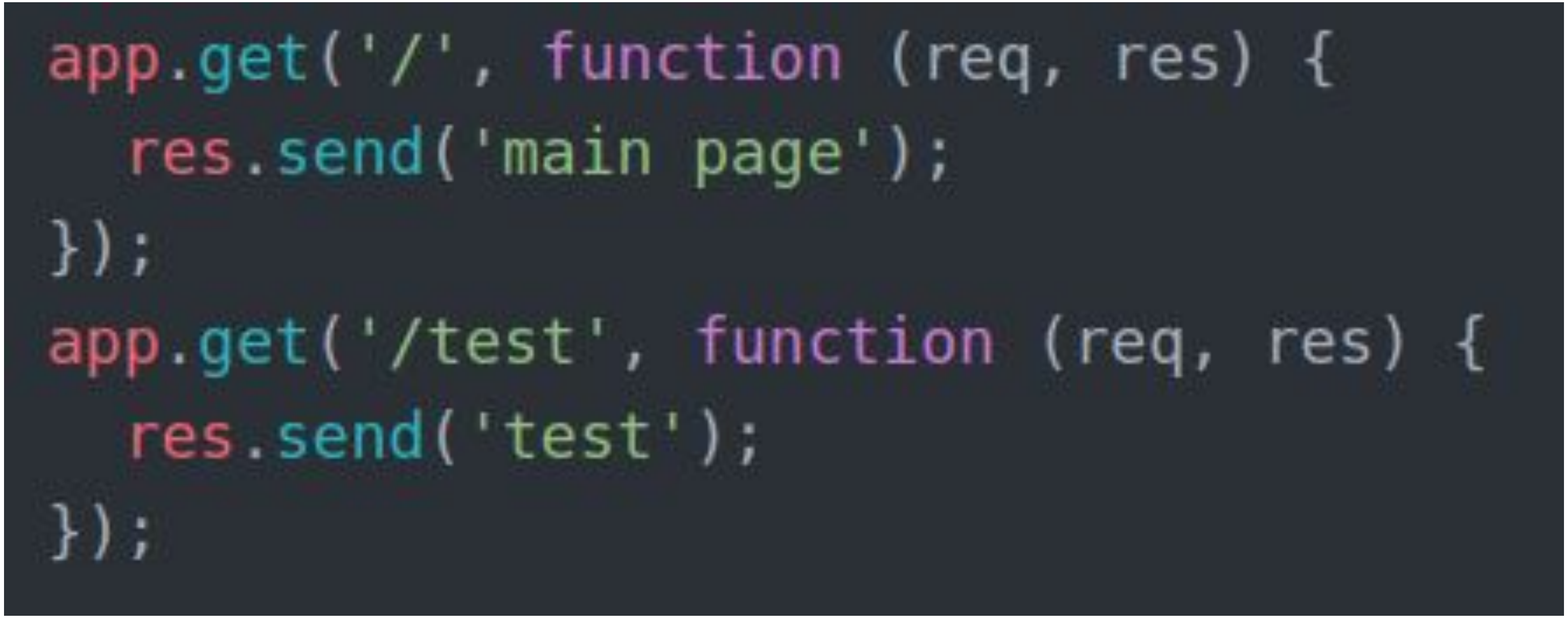


```
con.connect(function(err) {
  if (err) throw err;

  var sql = "INSERT INTO profile (name, address) VALUES ('Company Inc', 'Highway 37')";
  con.query(sql, function (err, result) {
    if (err) throw err;
    console.log("1 record inserted");
  });

  con.query("SELECT * FROM users", function (err, result, fields) {
    if (err) throw err;
    console.log(result);
  });

});
```

middleware
промежуточная обработка



```
var myLogger = function (req, res, next) {
  console.log('LOGGED');
  next();
};

app.use(myLogger);
```



```
app.get('/', function (req, res) {
  res.send('main page');
});
app.get('/test', function (req, res) {
  res.send('test');
});
```

```
var cookieParser = require('cookie-parser')

app.use(cookieParser())
```

```
app.get('/', function(req, res) {
 console.log('Cook: ', req.cookies)
})
```

Axios—это JavaScript-библиотека для выполнения HTTP-запросов в Node.js

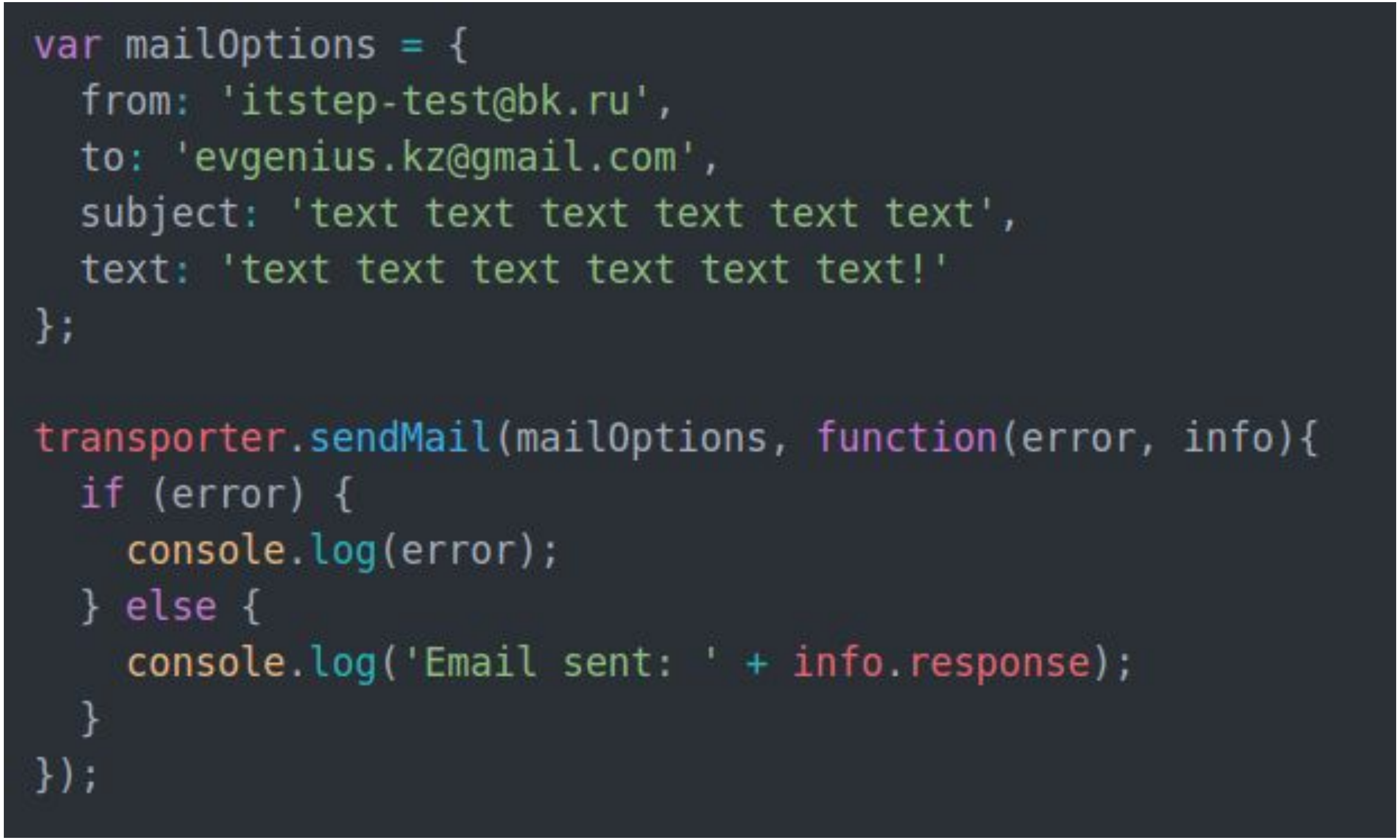
```
var axios = require('axios');
```

```
axios.get('/login')
  .then(function (response) {
    console.log(response.data);
  })
  .catch(function (error) {
    console.log(error);
  });
```

```
axios.get('http://mysite.kz')
  .then(function (response) {
    console.log(response.data);
  })
  .catch(function (error) {
    console.log(error);
  });
```

nodemailer

```
var transporter = nodemailer.createTransport({
  host: 'smtp.mail.ru',
  port: 465,
  secure: true,
  auth: {
    user: 'itstep-test@bk.ru',
    pass: '48o^xjamYKCT'
  }
});
```



```
var mailOptions = {
  from: 'itstep-test@bk.ru',
  to: 'evgenius.kz@gmail.com',
  subject: 'text text text text text text',
  text: 'text text text text text text!'
};

transporter.sendMail(mailOptions, function(error, info){
  if (error) {
    console.log(error);
  } else {
    console.log('Email sent: ' + info.response);
  }
});
```