

Лабораторная работа №2

Задача 1.

Отдача контента через Node.js

Создадим следующую структуру каталога сайта:

site

- img -
 - logo.gif
 - photo.gif

- Index.html
- about.html
- contact.html
- style.css
- script.js

Файл index.html содержит следующую разметку

```
<html>
<head>
  <title>Node-page</title>
  <link rel=stylesheet href="styles.css" type=text/css>
  <script src ="script.js" type= "text/javascript"></script>
</head>
<body>
  <hl><img src = "img/logo.gif" />Сайт на Node.js</hl>
  <div class = "main menu">
    <p><a href="about.html">О нас</a></p>
    <p><a href="about.html">Контакты</a></p>
  </div>
  <img src= "img/photo.gif" />
</body>
</html>
```

Сначала нам потребуется подключить модули:

```
var http = require("http");
var fs = require('fs');
var url = require("url");
```

Попробуем отдать контент уже опробованным нами способом:

```
http.createServer(function onRequest(request, response) {  
  var pathname = url.parse(request.url).pathname;  
  console.log("Получен запрос " + pathname);  
  
  fs.readFile('index.html', 'utf8', function(error, data) {  
    if (error) {  
      console.log( 'Could not find or open file for  
reading\n');  
    } else {  
      response.writeHead (200, ('Content-Type':'text/html '));  
      response.end(data);  
    }  
  })  
}).listen(8080);
```

Запустите браузер с адресом <http://localhost:8080> и наблюдайте что будет выводиться в консоли? Почему так происходит ?

Задача 2.

В предыдущем Задаче 1 представлено все разнообразие контента, которое нам надо в нужном формате отдать браузеру. Если бы этот контент состоял только из html-страниц, можно было бы ограничиться следующим кодом:

```
http.createServer(function onRequest(request, response) {
    var postData = "";
    var pathname = url.parse(request.url).path;
    if(pathname == '/')
        pathname = '/index.html';
    // чтобы убрать начальный слэш
    pathname = pathname.substring(1, pathname.length);
    fs.readFile (pathname, 'utf8', function (err, data) {
        if (err) {
            console.log( 'Could not find or open file' +pathname
+' for reading\n ');
        }
        else {
            response.writeHead(200, {'Content-Type': 'text/html'});
            response.end(data);
        }
    })
}).listen(8080);
```

Здесь у нас присутствуют не только javascript- и CSS-файлы, которые современный браузер, поморщившись, примет и как 'text/html', но и изображения для которых правильный Content-type обязателен. Поэтому дополним наш код - сначала добавим еще один, небесполезный модуль:

var path = require('path'); - он отвечает за операции с путями в файловой системе

Далее создадим объект с mime-типами:

```
var mimeTypes = {
    '.js' : 'text/javascript',
    '.html': 'text/html',
    '.css' : 'text/css',
    '.jpg' : 'image/jpeg',
    '.gif' : 'image/gif'
};
```

Ну а теперь немного изменим код отдачи контента:

```
fs.readFile(pathname, 'utf8', function (err, data)
    if (err) {
        console.log('Could not find or open file ' +
            pathname + ' for reading\n');
    } else (
        response.writeHead(200, {'Content-Type':
            mimeTypes[path.extname(pathname)]});
        response.end(data);
    })
```

Запустите данный код и посмотрите весь ли контент получен браузером?

Задача 3.

Уже лучше, но картинок мы все равно не увидим. Причина проста - это бинарные данные, и читаются они другим способом. Нам немного нужно модифицировать код:

```
http.createServer(function onRequest(request, response) {
  var pathname = url.parse(request.url).path;
  if(pathname == '/') { pathname = '/index.html';}
  var extname = path.extname(pathname);
  console.log(extname);
  var mimeType = mimeTypes(path.extname(pathname));
  pathname = pathname.substring(1, pathname.length);
  if( (extname == ".gif") || (extname==".jpg") ) {
    var img = fs.readFileSync(' ./'+pathname);
    response.writeHead(200, { 'Content-Type': mimeType });
    response.end(img, 'binary');
  } else {
    fs.readFile(pathname, 'utf8', function(err, data){
      if (err) {
        console.log('Could not find or open file ' +
pathname + ' for reading\n' );
      } else {
        console.log(pathname+" "+mimeType);
        response.writeHead(200, { 'Content-Type':
mimeType});
        response.end(data);
      }
    })
  }
}).listen(8080);
```

Запустите код, проверти полную отдачу контента.