

Лабораторная работа 4: Работа со встроенными модулями Nodejs

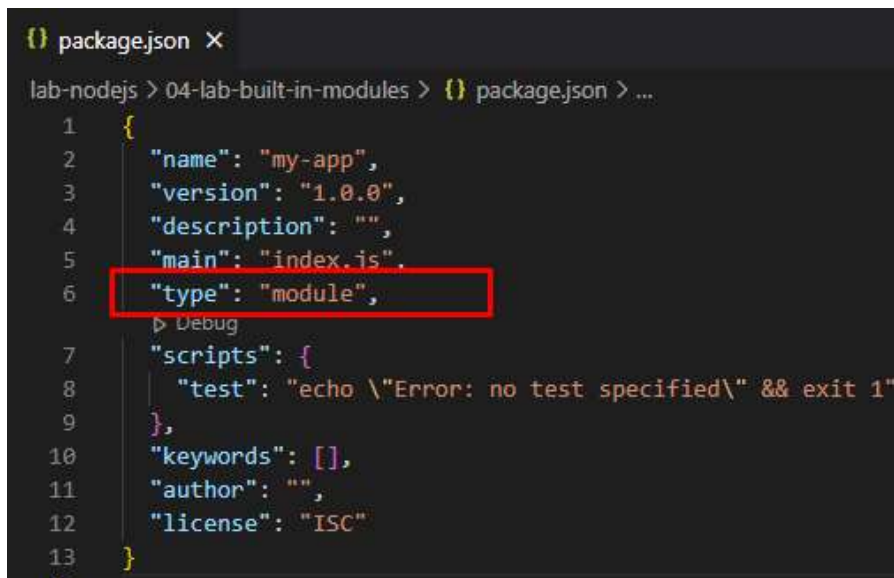
В этой работе мы инициализируем папку проекта, подключим существующие встроенные модули и запустим скрипты с этими модулями

1. Откройте в консоли папку **04-lab-built-in-modules** (сейчас она играет роль папки нашего проекта) и инициализируйте её командой

```
npm init -y
```

2. В файле **package.json** укажите ключ-значение

```
"type": "module" :
```



```
{  
  "name": "my-app",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "type": "module",  
  "scripts": {  
    "test": "echo \\\"Error: no test specified\\\" && exit 1"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC"  
}
```

3. В файле **01-http.js** импортируйте модуль **http**:

```
import http from 'http'
```

4. В файле **01-http.js** пропишите выводы в консоль следующих свойств объекта **http**:

```
http.METHODS  
http.STATUS_CODES[200]  
http.IncomingMessage  
http.OutgoingMessage  
http.ServerResponse  
http.createServer
```

5. Запустите **node 01-http.js**, результат должен выглядеть так:

```
C:\Windows\system32\cmd.exe
C:\Users\Administrator\Desktop\nodejs\lab-nodejs\04-lab-built-in-modules>node 01-http.js
Методы HTTP: [
  'ACL',      'BIND',      'CHECKOUT',
  'CONNECT',  'COPY',      'DELETE',
  'GET',      'HEAD',      'LINK',
  'LOCK',     'M-SEARCH',  'MERGE',
  'MKACTIVITY', 'MKCALENDAR', 'MKCOL',
  'MOVE',     'NOTIFY',    'OPTIONS',
  'PATCH',   'POST',     'PROPFIND',
  'PROPPATCH', 'PURGE',    'PUT',
  'REBIND',   'REPORT',   'SEARCH',
  'SOURCE',   'SUBSCRIBE', 'TRACE',
  'UNBIND',   'UNLINK',   'UNLOCK',
  'UNSUBSCRIBE'
]
Статус 200: OK
Входящее сообщение: [Function: IncomingMessage]
Ответ: [Function: OutgoingMessage]
ответ сервера: [Function: ServerResponse]
создание сервера: [Function: createServer]

C:\Users\Administrator\Desktop\nodejs\lab-nodejs\04-lab-built-in-modules>
```

6. Возьмите код сервера со страницы <https://nodejs.org/en/docs/guides/getting-started-guide/> и вставьте его в файл **02-простой-вебсервер.js**. Примечание: перепишите импорт модуля `http` с *CJS* на *ES6* :

```
import http from 'http'

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

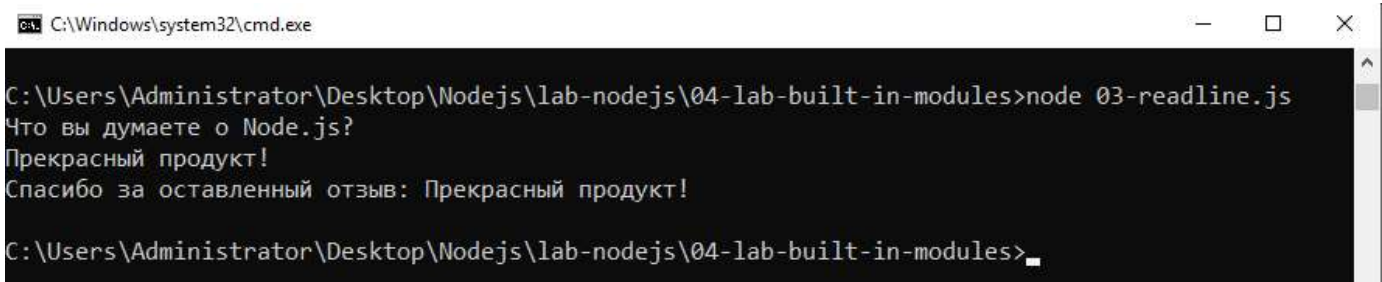
7. Запустите сервер командой `node 02-простой-вебсервер.js` :

```
C:\Windows\system32\cmd.exe - node 02-простой-вебсервер.js
C:\Users\Administrator\Desktop\nodejs\lab-nodejs\04-lab-built-in-modules>node 02-простой-вебсервер.js
Server running at http://127.0.0.1:3000/
```

8. Если ошибок нет, откройте в браузере адрес <http://127.0.0.1:3000/> и убедитесь что страница работает. Ура!:



9. Перейдите в консоль и остановите сервер сочетанием клавиш `Ctrl + C`
10. Перепишите код файла **02-простой-вебсервер.js** так, чтобы сервер возвращал HTML разметку в виде заголовка первого уровня `<h1>Привет, мир</h1>`, а цвет заголовка был синим.
Протестируйте скрипт: запустите сервер и убедитесь что нужный результат показывается в браузере по адресу <http://127.0.0.1:3000/>
11. Изучите содержимое файла **03-readline.js** и запустите его в Nodejs `node 03-readline.js` :

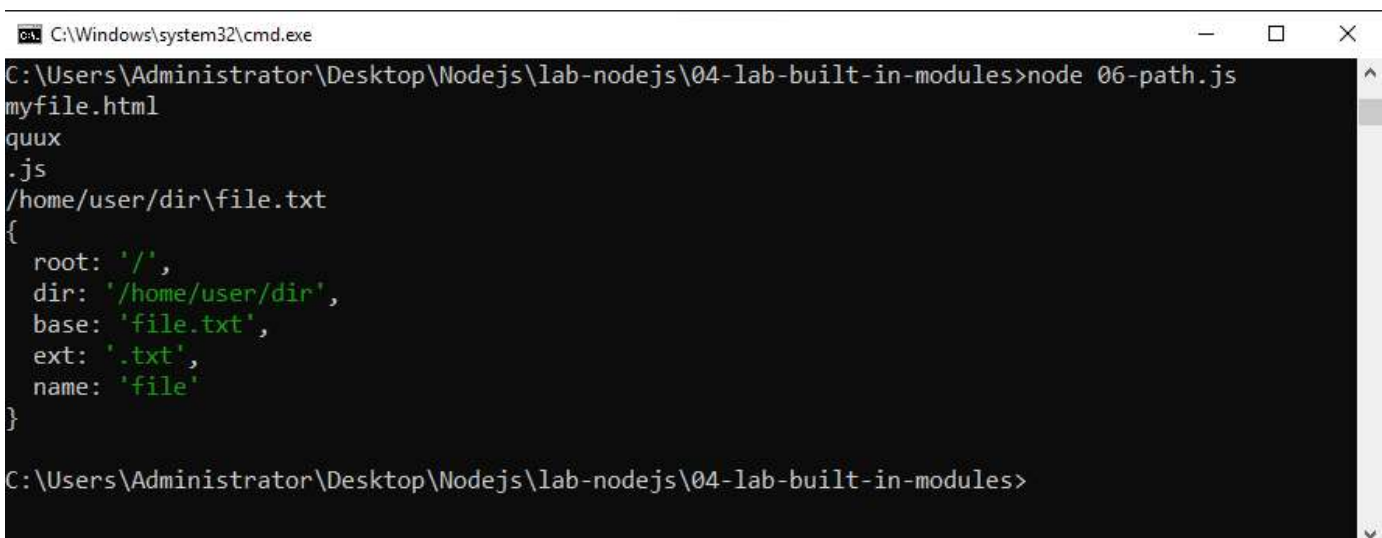


```
C:\Windows\system32\cmd.exe
C:\Users\Administrator\Desktop\nodejs\lab-nodejs\04-lab-built-in-modules>node 03-readline.js
Что вы думаете о Node.js?
Прекрасный продукт!
Спасибо за оставленный отзыв: Прекрасный продукт!
C:\Users\Administrator\Desktop\nodejs\lab-nodejs\04-lab-built-in-modules>
```

12. Изучите содержимое файла **04-URL.js** и запустите его в Nodejs `node 04-URL.js` (в нем парсится URL: <https://user:pass@sub.example.com:8080/p/a/t/h?query=string#hash>):

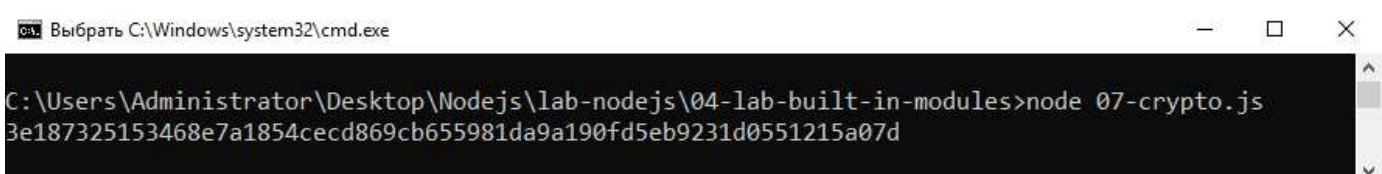
```
Url {
  protocol: 'https:',
  slashes: true,
  auth: 'user:pass',
  host: 'sub.example.com:8080',
  port: '8080',
  hostname: 'sub.example.com',
  hash: '#hash',
  search: '?query=string',
  query: 'query=string',
  pathname: '/p/a/t/h',
  path: '/p/a/t/h?query=string',
  href: 'https://user:pass@sub.example.com:8080/p/a/t/h?query=string#hash'
}
```

13. Изучите содержимое файла **06-path.js** и запустите его в Nodejs `node 06-path.js` :



```
C:\Windows\system32\cmd.exe
C:\Users\Administrator\Desktop\nodejs\lab-nodejs\04-lab-built-in-modules>node 06-path.js
myfile.html
quux
.js
/home/user/dir/file.txt
{
  root: '/',
  dir: '/home/user/dir',
  base: 'file.txt',
  ext: '.txt',
  name: 'file'
}
C:\Users\Administrator\Desktop\nodejs\lab-nodejs\04-lab-built-in-modules>
```

14. Изучите содержимое файла **07-crypto.js** и запустите его в Nodejs `node 07-crypto.js` :



```
C:\Windows\system32\cmd.exe
C:\Users\Administrator\Desktop\nodejs\lab-nodejs\04-lab-built-in-modules>node 07-crypto.js
3e187325153468e7a1854cecd869cb655981da9a190fd5eb9231d0551215a07d
```

15. Запустите сервер на основе файла **08-server.js** и посмотрите в браузере как он работает.

Примечание: обратите внимание на порт, в этом примере используется 8000

16. Создайте файл **09-events.js** с кодом и запустите его:

```
import { EventEmitter } from 'node:events';

class MyEmitter extends EventEmitter {}

const myEmitter = new MyEmitter();
myEmitter.on('event', () => {
  console.log('событие произошло!!');
});
myEmitter.emit('event');
```