

# Лабораторная работа 6: Модуль fs и работа с файловой системой

В этой работе при помощи модуля `fs` мы создадим, прочитаем, допишем, получим информацию и удалим файлы.

1. Откройте в консоли папку **06-lab-fs-module** и инициализируйте её командой

```
npm init -y
```

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

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

3. В файле **06-lab-fs-module/01-write.js** создайте файл при помощи API обещаний (Promises):

```
import { writeFile } from 'node:fs/promises';

const data = 'Ехал Грека через реку\nВидит Грека в речке рак';
const promise = writeFile('message.txt', data);
await promise;
```

4. В файле **06-lab-fs-module/02-read.js** создайте файл при помощи API обещаний (Promises):

```
import { readFile } from 'node:fs/promises';

try {
  const filePath = new URL('message.txt', import.meta.url);
  console.log(`filePath:${filePath}`)
  const contents = await readFile(filePath, { encoding: 'utf8' });
  console.log(contents);
} catch (err) {
  console.error(err.message);
}
```

5. В файле **06-lab-fs-module/03-append.js.js** добавьте новые строки в файл **message.txt**:

```
import { appendFile } from 'node:fs/promises';

try {
  const filePath = new URL('message.txt', import.meta.url)
  const data = '\nСунул грека руку в реку,\nпак за руку греку – цап!'
  const contents = await appendFile(filePath, data)
} catch (err) {
  console.error(err.message);
}
```

6. Выполните скрипт `node 03-append.js.js`, в файле **message.txt** появятся новые строки:

Ехал Грека через реку  
Видит Грека в речке рак  
Сунул грека руку в реку,  
рак за руку греку – цап!

7. Импортируйте функцию `stat` и получите информацию о переименованном файле:

```
import { stat } from 'node:fs/promises';

try {
  const path = new URL('newMessage.txt', import.meta.url)
  const result = await stat(path)

  console.log(JSON.stringify(result, null, ' '))
} catch (err) {
  console.error(err.message);
}
```

8. Результат вызова `node 05-stat.js` будет выглядеть как объект с данными о файле:

```
{
  "dev": 917013410,
  "mode": 33206,
  "nlink": 1,
  "uid": 0,
  "gid": 0,
  "rdev": 0,
  "blksize": 4096,
  "ino": 844424930804980,
  "size": 82,
  "blocks": 0,
  "atimeMs": 1672748257734.003,
  "mtimeMs": 1672748257734.003,
  "ctimeMs": 1672748587188.702,
  "birthtimeMs": 1672738869963.7307,
  "atime": "2023-01-03T12:17:37.734Z",
  "mtime": "2023-01-03T12:17:37.734Z",
  "ctime": "2023-01-03T12:23:07.189Z",
  "birthtime": "2023-01-03T09:41:09.964Z"
}
```

9. Познакомьтесь с примером сервера на Nodejs с использованием работы с файлами на сайте MDN

[https://developer.mozilla.org/en-US/docs/Learn/Server-side/Node\\_server\\_without\\_framework](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Node_server_without_framework)

```

import * as fs from 'node:fs';
import * as http from 'node:http';
import * as path from 'node:path';

const PORT = 8000;

const MIME_TYPES = {
  default: 'application/octet-stream',
  html: 'text/html; charset=UTF-8',
  js: 'application/javascript; charset=UTF-8',
  css: 'text/css',
  png: 'image/png',
  jpg: 'image/jpeg',
  gif: 'image/gif',
  ico: 'image/x-icon',
  svg: 'image/svg+xml',
};

const STATIC_PATH = path.join(process.cwd(), './static');

const toBool = (() => true, () => false);

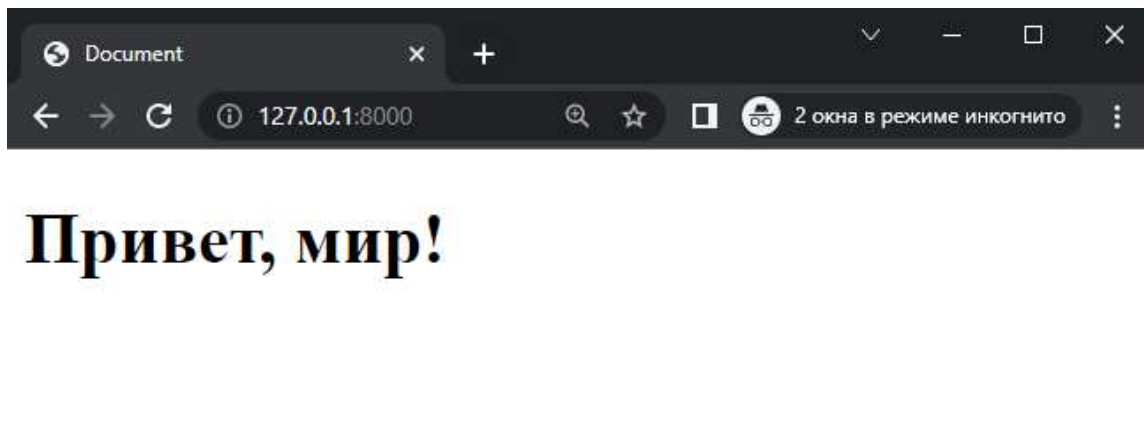
const prepareFile = async (url) => {
  const paths = [STATIC_PATH, url];
  if (url.endsWith('/')) paths.push('index.html');
  const filePath = path.join(...paths);
  const pathTraversal = !filePath.startsWith(STATIC_PATH);
  const exists = await fs.promises.access(filePath).then(...toBool);
  const found = !pathTraversal && exists;
  const streamPath = found ? filePath : STATIC_PATH + '/404.html';
  const ext = path.extname(streamPath).substring(1).toLowerCase();
  const stream = fs.createReadStream(streamPath);
  return { found, ext, stream };
};

http.createServer(async (req, res) => {
  const file = await prepareFile(req.url);
  const statusCode = file.found ? 200 : 404;
  const mimeType = MIME_TYPES[file.ext] || MIME_TYPES.default;
  res.writeHead(statusCode, { 'Content-Type': mimeType });
  file.stream.pipe(res);
  console.log(`${req.method} ${req.url} ${statusCode}`);
}).listen(PORT);

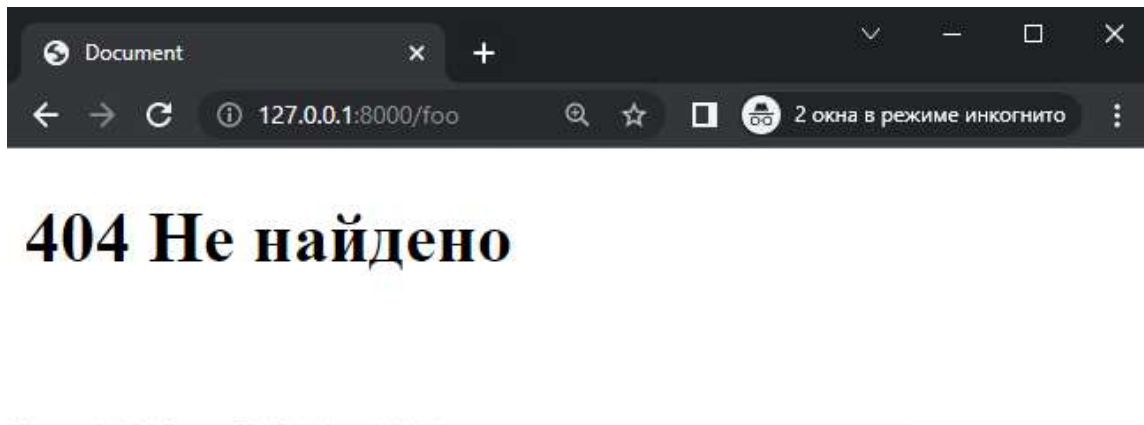
console.log(`Server running at http://127.0.0.1:${PORT}/`);

```

10. Запустите сервер `node 06-server-with-files.js` . *Примечание: в консоли сервер останавливается сочетанием клавиш `Ctrl + C`*
11. Откройте страницу `http://127.0.0.1:8000` в браузере, убедитесь что показывается файл **06-lab-fs-module/static/index.html** :



12. Откройте несуществующую страницу `http://127.0.0.1:8000/foo` в браузере, убедитесь что показывается файл **06-lab-fs-module/static/404.html**



13. Если вы забыли/не знали синтаксис `import * as fs from 'node:fs'` изучите страницу об [импорте всего содержимого модуля](#)
14. Подключите CSS и JS файлы из **06-lab-fs-module/static** и убедитесь, что после загрузки страницы изменения видны в браузере/консоли