



Web-технологии

Применение Node.js.
Основы Deno

- Работа с модулями
- Использование npm и yarn, готовые модули
- Событийная модель в Node.js
- Создание простейшего сервера на Node.js
 - Эхо-сервер, параметры запроса, заголовки
- Журналирование (логгирование)
- Внешние сервера для журналирования (логгирования)
 - Rollbar – попытка отловить ошибки раньше пользователей
 - Sentry – прослеживание JavaScript
- Работа с файлами и потоками
- «Асинхронный try-catch»
- npm
- Deno
 - Установка, настройка среды
 - Разрешения, примеры

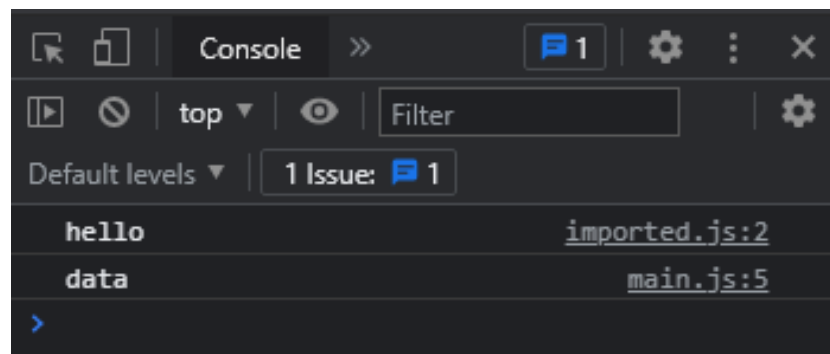
<http://learn.javascript.ru/screencast/nodejs>

<https://nodejs.org/en/docs/guides/>

Модули в браузере

3

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
</head>
<body>
  <script src="main.js" type="module"></script>
</body>
</html>
```



main.js

```
import hello from "./imported.js"
import {data} from "./imported.js"
```

```
hello()
console.log(data)
```

.js, .mjs... ,... as... *... then...

imported.js

```
function hello() {
  console.log("hello")
}
let data = "data"
export {data}
export default hello
```

Деление на модули

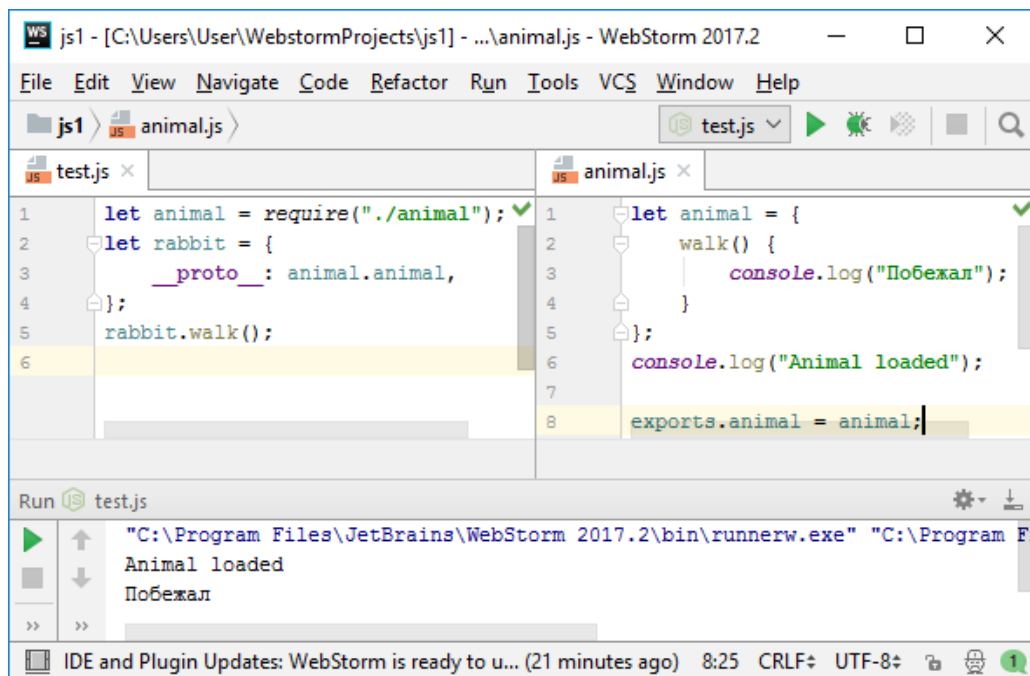
```

let animal = require("./animal");
let rabbit = {
  __proto__: animal.animal,
};
rabbit.walk();

let animal = {
  walk() {
    console.log("Побежал");
  }
};
console.log("Animal loaded");

exports.animal = animal;

```

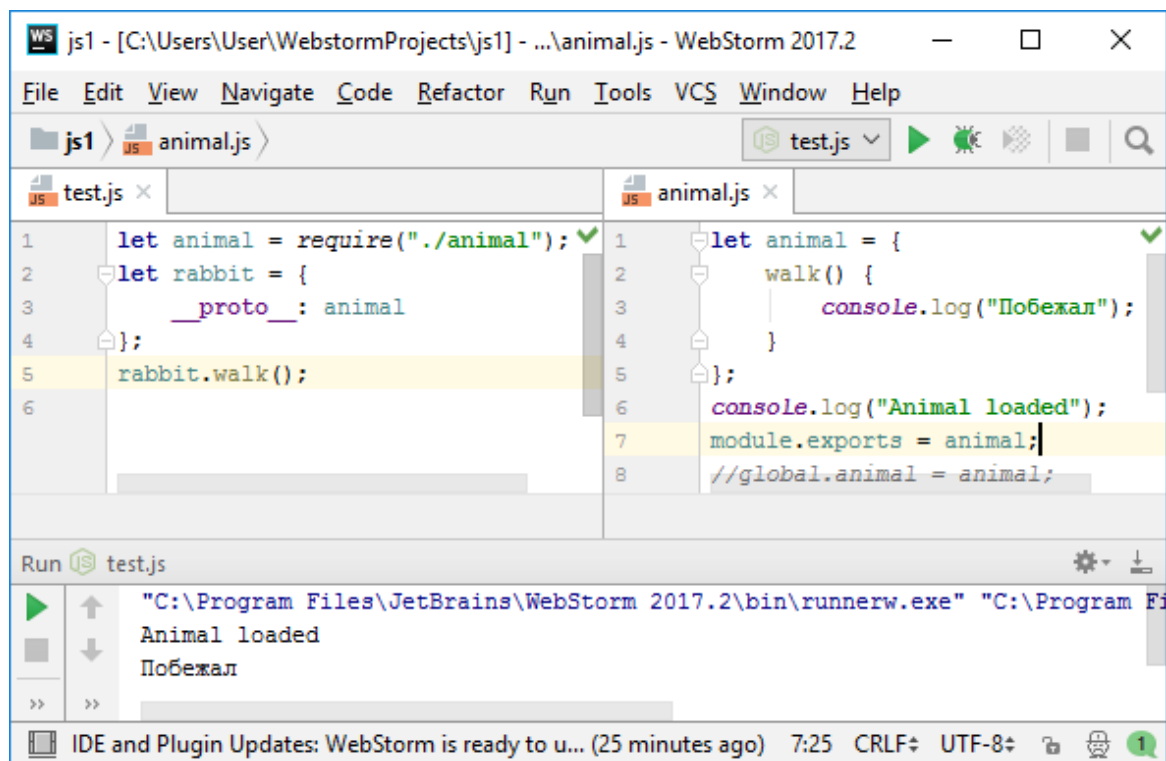


Глобальные переменные

5

```
let animal = require("./animal");
let rabbit = {
  __proto__: animal
};
rabbit.walk();

let animal = {
  walk() {
    console.log("Побежал");
  }
};
console.log("Animal loaded");
module.exports = animal;
//global.animal = animal;
```

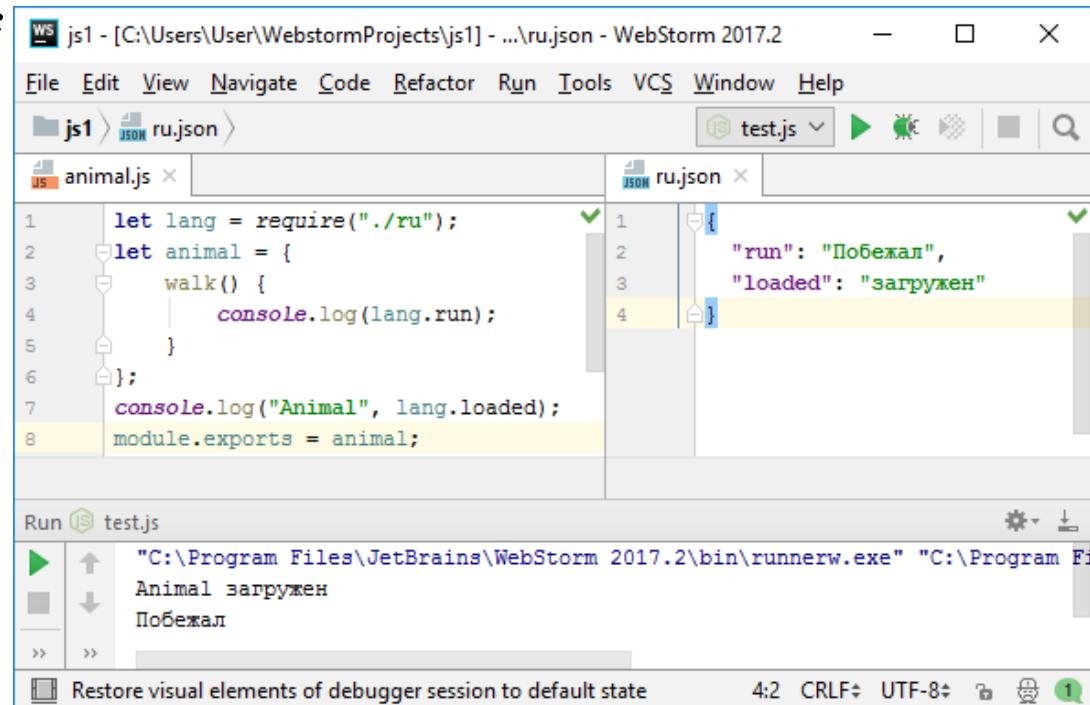


Использование JSON

6

```
let lang = require("./ru");
let animal = {
  walk() {
    console.log(lang.run);
  }
};
console.log("Animal", lang.loaded);
module.exports = animal;
```

```
{
  "run": "Побежал",
  "loaded": "загружен"
}
```



Использование папок для модулей

7

```
js1 - [C:\Users\User\WebstormProjects\js1] - ... \test.js - WebStorm 2017.2
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help

js1 > test.js >

test.js x index.js x

js1 C:\Users\User\W

- animal
 - index.js
 - ru.json
 - test.js
- External Libraries

```
1 let animal = require("../animal");
2 let rabbit = {
3   __proto__: animal
4 };
5 rabbit.walk();
6
```

```
1 let lang = require("../ru");
2 let animal = {
3   walk() {
4     console.log(lang.run);
5   }
6 };
7 console.log("Animal", lang.loaded);
8 module.exports = animal;
```

Run test.js

"C:\Program Files\JetBrains\WebStorm 2017.2\bin\runnerw.exe" "C:\Program Files\nodejs\node.exe"

Animal загружен

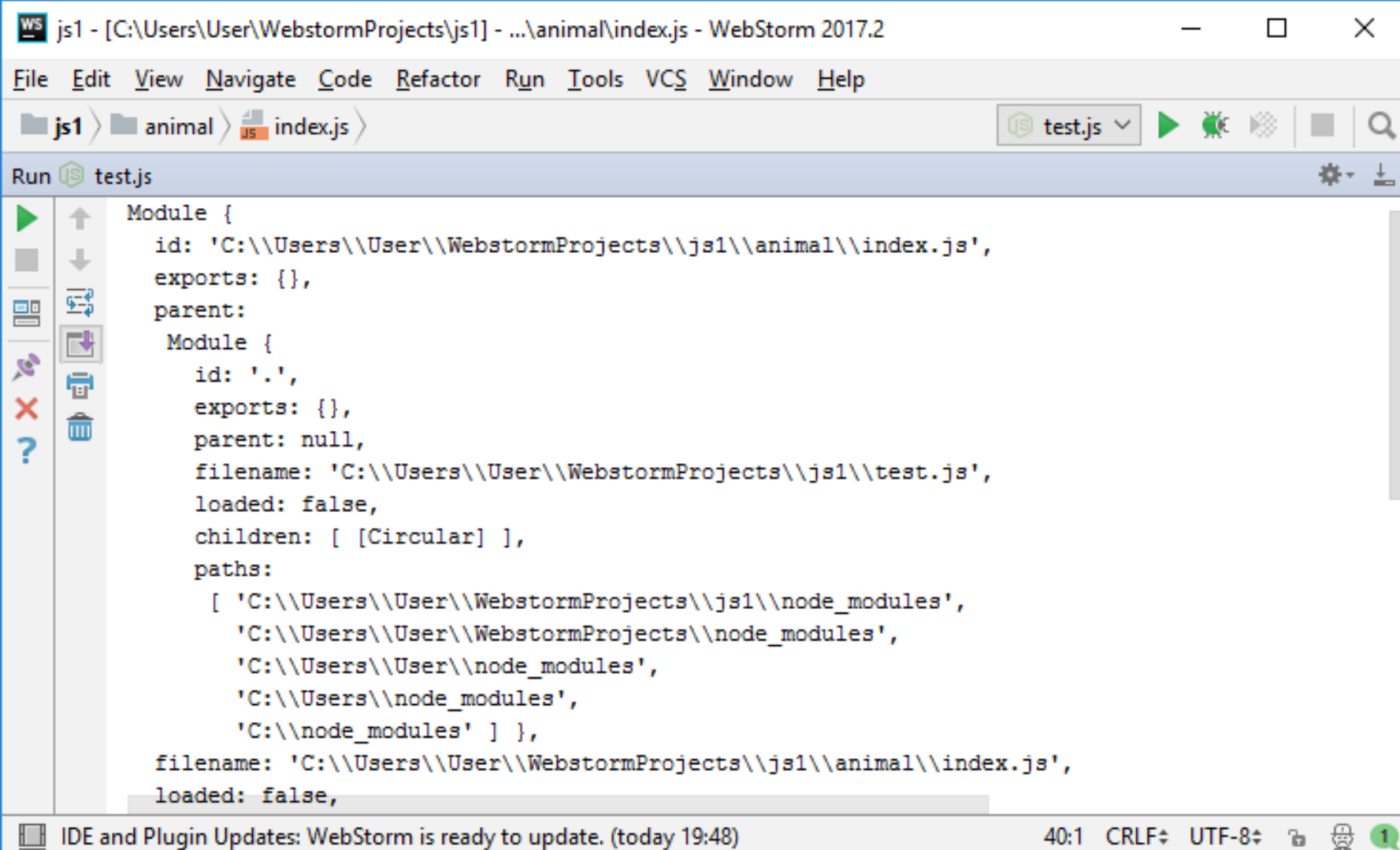
Побежал

Process finished with exit code 0

1:33 CRLF UTF-8

Скрипт должен называться **index.js**

Объект module



The screenshot shows the WebStorm 2017.2 IDE interface. The title bar indicates the project is 'js1' located at 'C:\Users\User\WebstormProjects\js1', with the active file being '...animal\index.js'. The menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The breadcrumb navigation shows the path 'js1 > animal > index.js'. The Run toolbar shows 'test.js' selected, along with play, debug, and other execution icons. The console output displays the 'module' object structure for the current file.

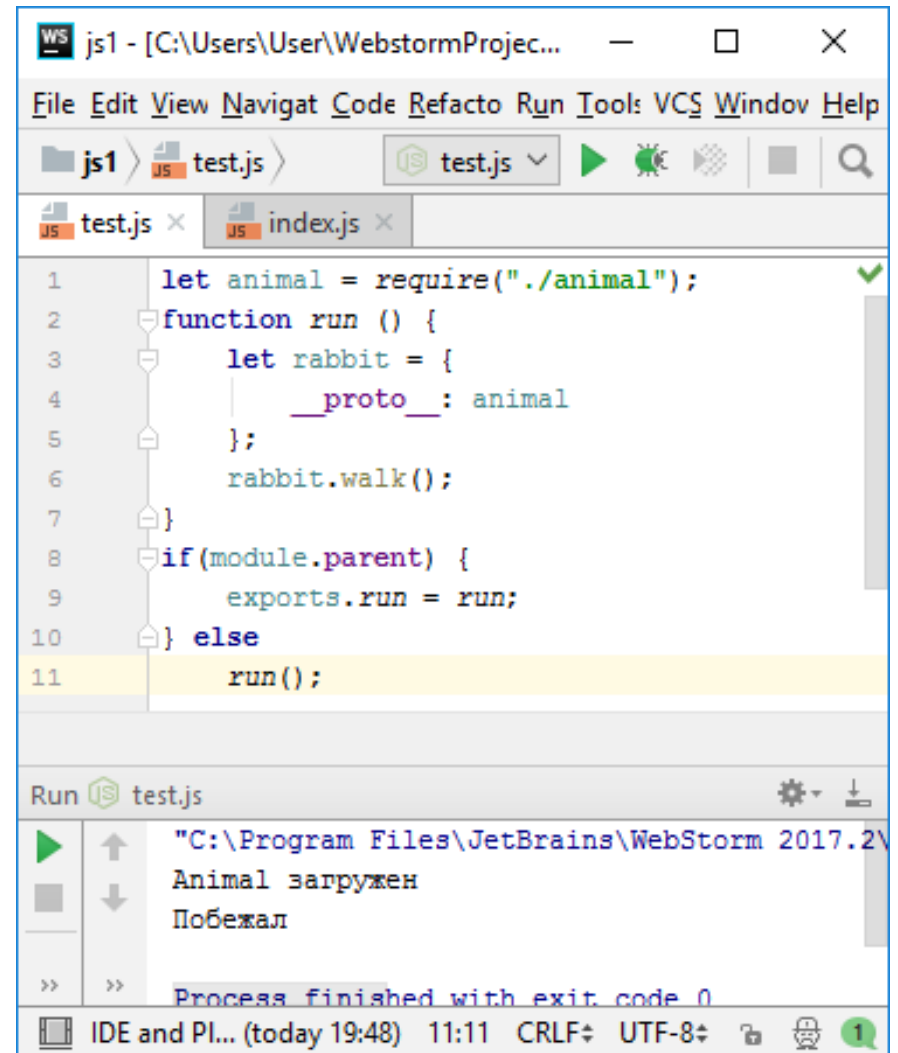
```
Module {
  id: 'C:\\Users\\User\\WebstormProjects\\js1\\animal\\index.js',
  exports: {},
  parent:
    Module {
      id: '.',
      exports: {},
      parent: null,
      filename: 'C:\\Users\\User\\WebstormProjects\\js1\\test.js',
      loaded: false,
      children: [ [Circular] ],
      paths:
        [ 'C:\\Users\\User\\WebstormProjects\\js1\\node_modules',
          'C:\\Users\\User\\WebstormProjects\\node_modules',
          'C:\\Users\\User\\node_modules',
          'C:\\Users\\node_modules',
          'C:\\node_modules' ] },
  filename: 'C:\\Users\\User\\WebstormProjects\\js1\\animal\\index.js',
  loaded: false,
```

The status bar at the bottom shows 'IDE and Plugin Updates: WebStorm is ready to update. (today 19:48)', the current time '40:1', and encoding settings 'CRLF' and 'UTF-8'.

module.parent

9

```
let animal = require("./animal");  
function run () {  
    let rabbit = {  
        __proto__: animal  
    };  
    rabbit.walk();  
}  
if(module.parent) {  
    exports.run = run;  
} else  
    run();
```



The screenshot shows an IDE window titled 'js1 - [C:\Users\User\WebstormProjec...'. The editor displays a file named 'test.js' with the following code:

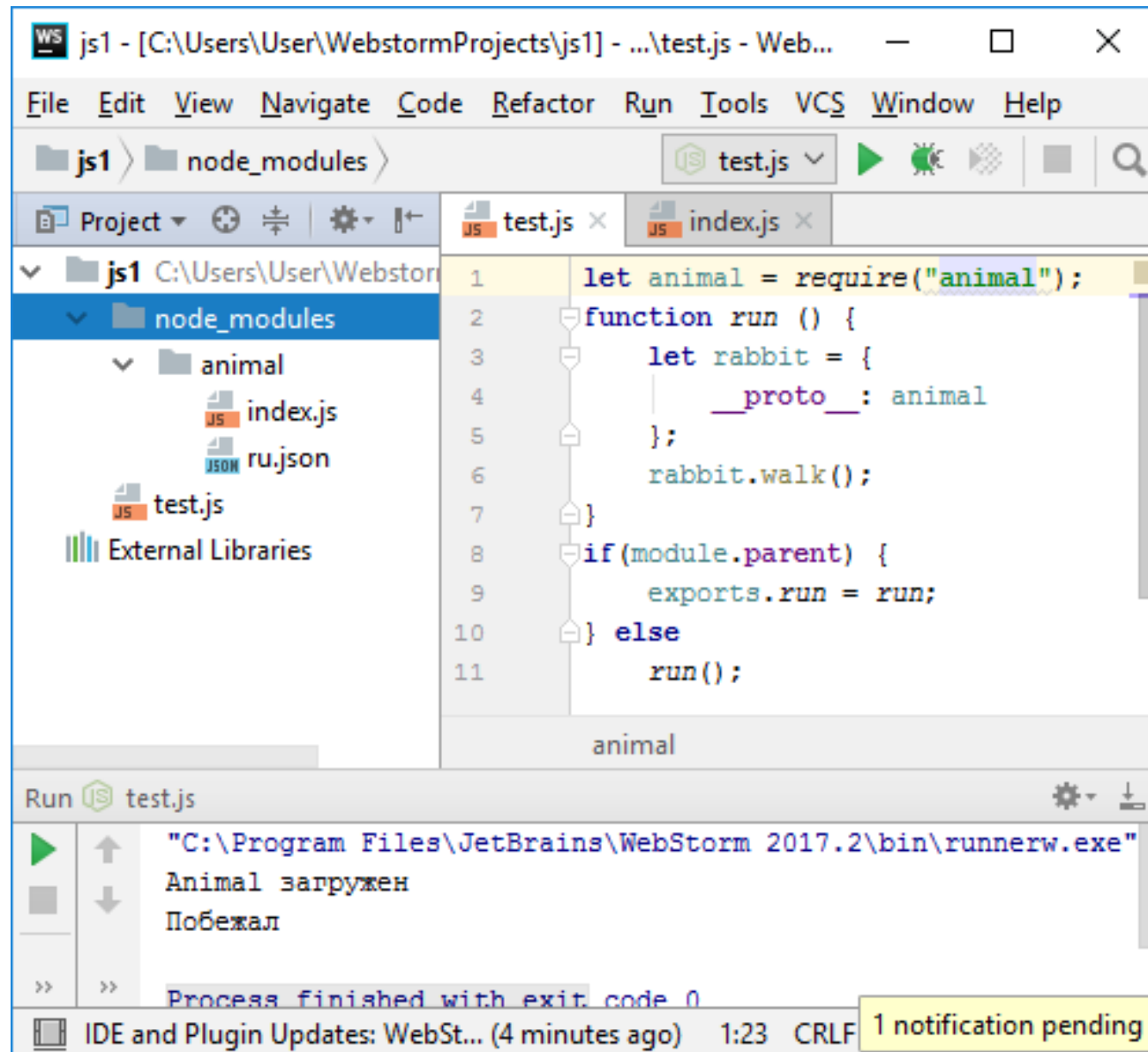
```
1 let animal = require("./animal");  
2 function run () {  
3     let rabbit = {  
4         __proto__: animal  
5     };  
6     rabbit.walk();  
7 }  
8 if(module.parent) {  
9     exports.run = run;  
10 } else  
11     run();
```

The code is executed, and the output is shown in the Run console at the bottom. The console displays the following messages:

```
Run test.js  
"C:\Program Files\JetBrains\WebStorm 2017.2\...  
Animal загружен  
Побежал  
Process finished with exit code 0
```

node_modules

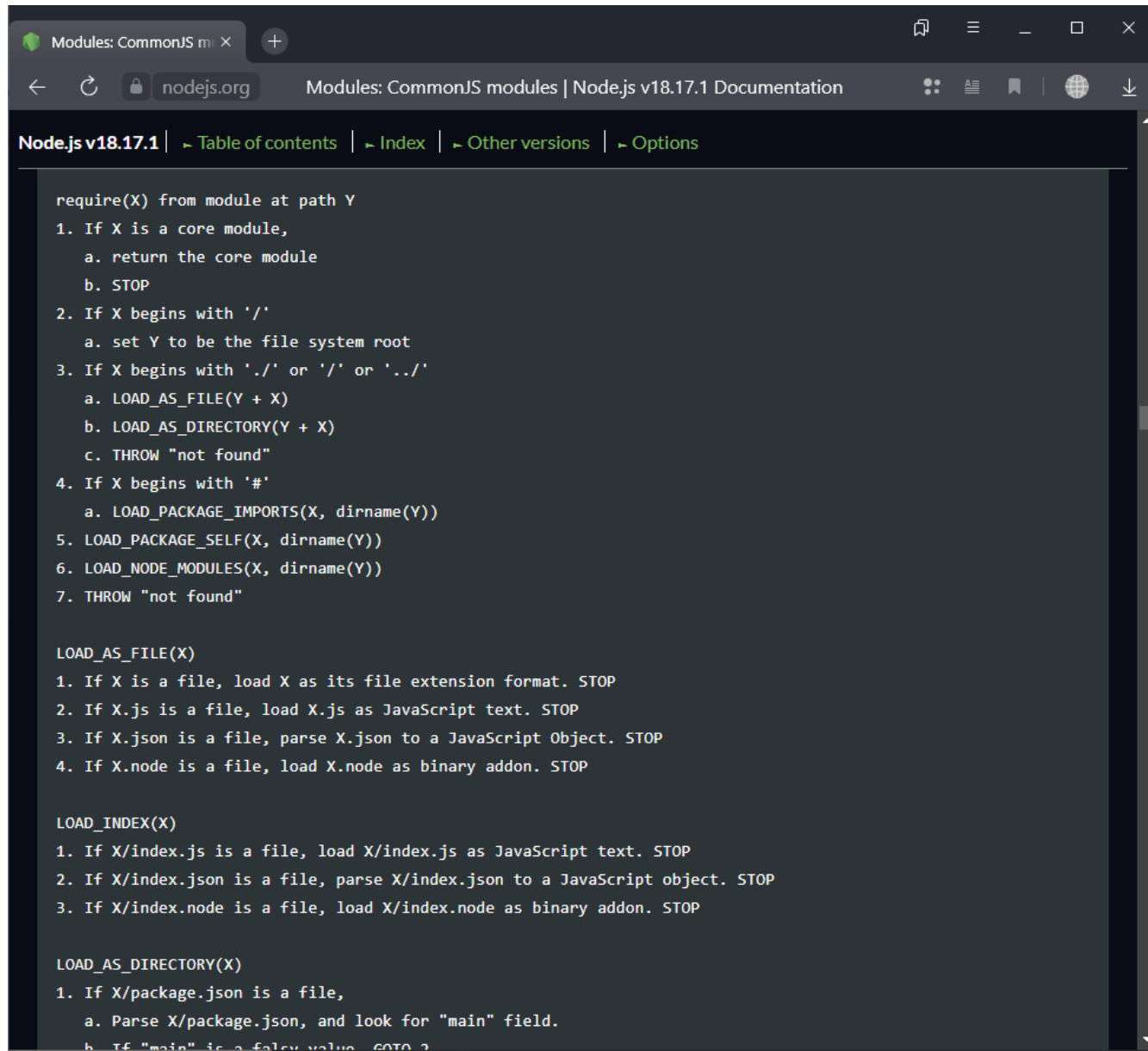
10



Последовательность загрузки модулей

11

<https://nodejs.org/dist/latest-v18.x/docs/api/modules.html#all-together>



The screenshot shows a web browser displaying the Node.js v18.17.1 documentation page for modules. The page title is "Modules: CommonJS modules | Node.js v18.17.1 Documentation". The breadcrumb navigation shows "Node.js v18.17.1" followed by links to "Table of contents", "Index", "Other versions", and "Options". The main content area is dark-themed and contains the following text:

```
require(X) from module at path Y
```

1. If X is a core module,
 - a. return the core module
 - b. STOP
2. If X begins with '/'
 - a. set Y to be the file system root
3. If X begins with './' or '/' or '../'
 - a. LOAD_AS_FILE(Y + X)
 - b. LOAD_AS_DIRECTORY(Y + X)
 - c. THROW "not found"
4. If X begins with '#'
 - a. LOAD_PACKAGE_IMPORTS(X, dirname(Y))
5. LOAD_PACKAGE_SELF(X, dirname(Y))
6. LOAD_NODE_MODULES(X, dirname(Y))
7. THROW "not found"

LOAD_AS_FILE(X)

1. If X is a file, load X as its file extension format. STOP
2. If X.js is a file, load X.js as JavaScript text. STOP
3. If X.json is a file, parse X.json to a JavaScript Object. STOP
4. If X.node is a file, load X.node as binary addon. STOP

LOAD_INDEX(X)

1. If X/index.js is a file, load X/index.js as JavaScript text. STOP
2. If X/index.json is a file, parse X/index.json to a JavaScript object. STOP
3. If X/index.node is a file, load X/index.node as binary addon. STOP

LOAD_AS_DIRECTORY(X)

1. If X/package.json is a file,
 - a. Parse X/package.json, and look for "main" field.
 - b. If "main" is a false value, GOTO 2.

CommonJS vs. ES6 modules

12

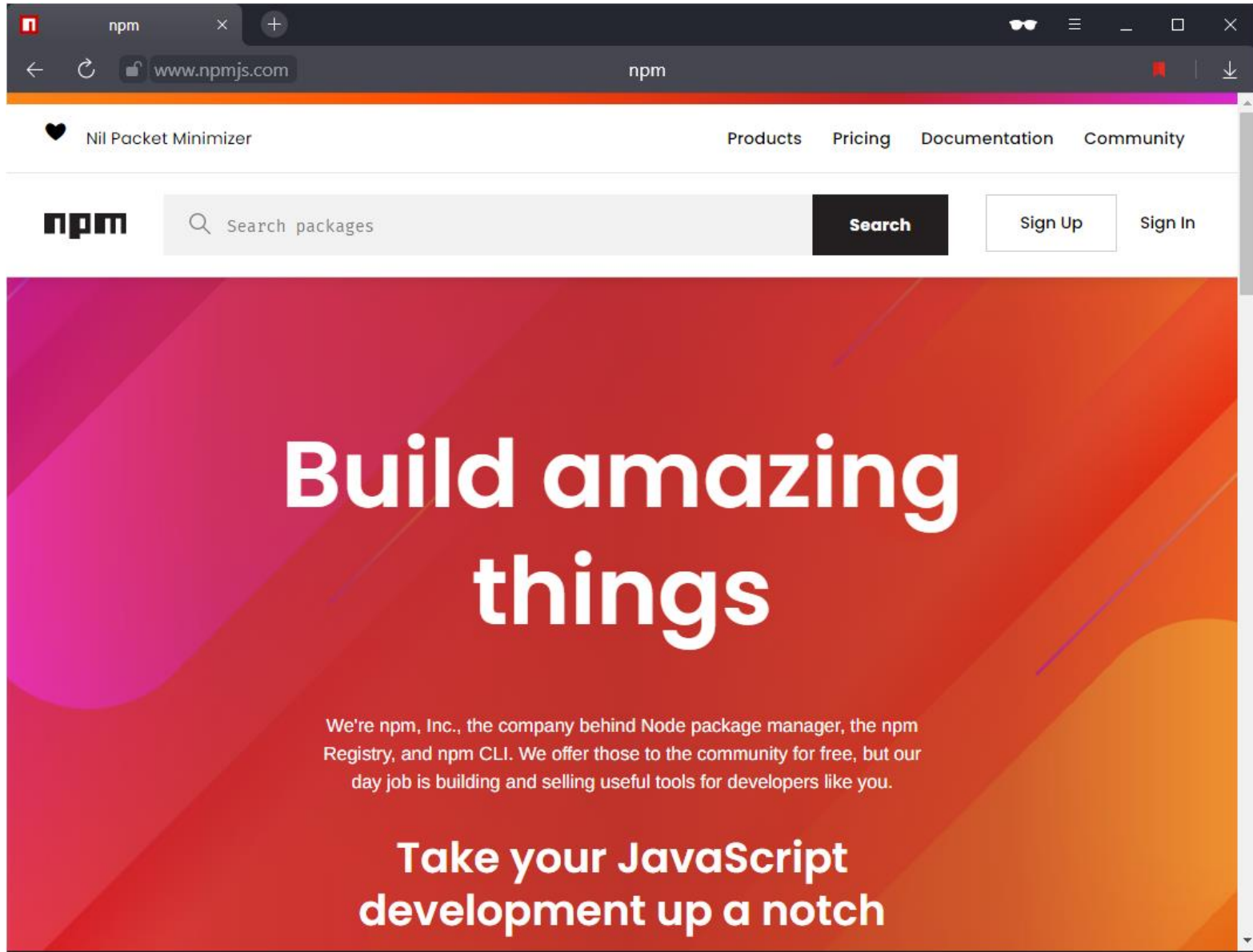
CommonJS

- Экспорт (один)
 - `module.exports`, `exports`
- Импорт
 - `require`
- Загрузка
 - runtime
 - не «поднимает» импорт
- Экспортирует «значение»
- Переменные верхнего уровня
 - `arguments`, `require`, `module`, `exports`, `__filename`, `__dirname`
- Не поддерживает синтаксис ES6 модулей

ES6

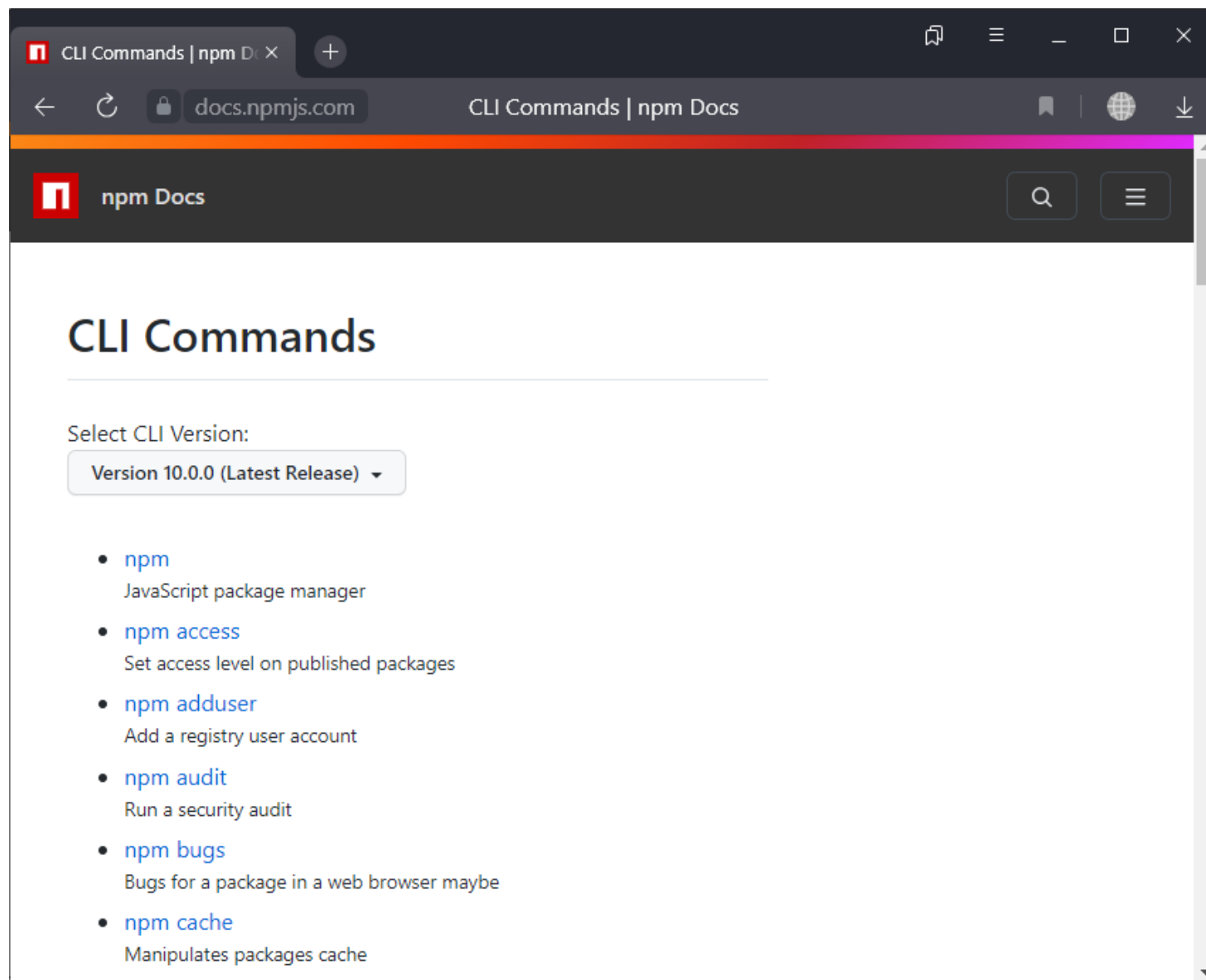
- Экспорт (множественный)
 - `export`
- Импорт
 - `import`
- Загрузка
 - статическая компиляция
 - «поднимает» импорт
- Экспортирует «ссылку»
- Нет переменных верхнего уровня
- Поддерживает синтаксис CommonJS

<https://www.npmjs.com/>



Команды CLI npm

14



<https://docs.npmjs.com/cli/v10/commands>

npm init – инициализация

15

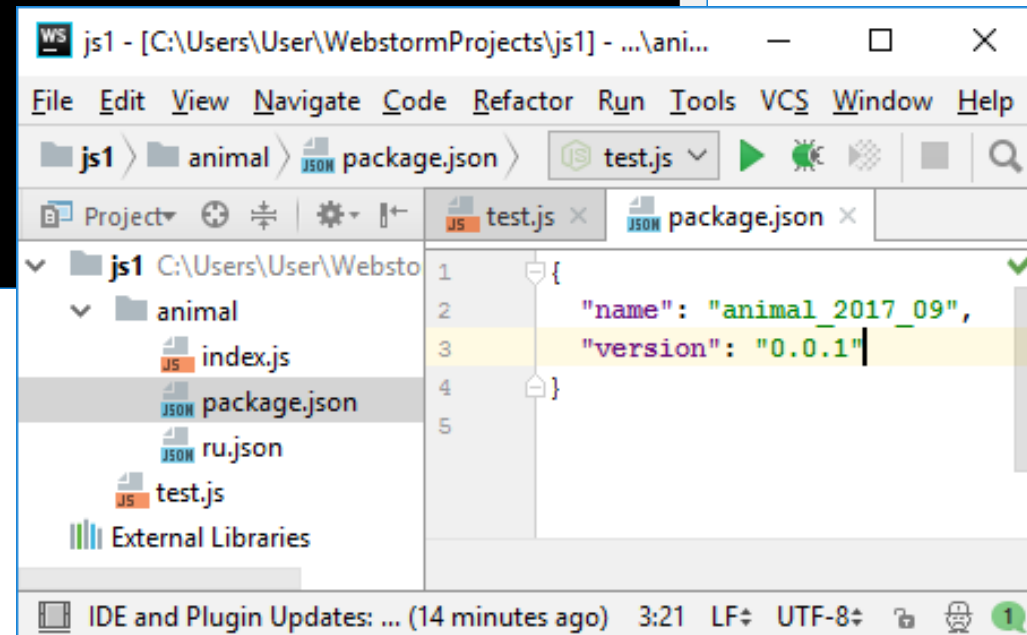
```
npm
C:\Users\User\WebstormProjects\js1\animal>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg> --save` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
name: (animal) animal_2017_09
version: (1.0.0) 0.0.1
description: Без описания
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\User\WebstormProjects\js1\animal\package.json:

{
  "name": "animal_2017_09",
  "version": "0.0.1",
  "description": "Без описания",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
```



package.json

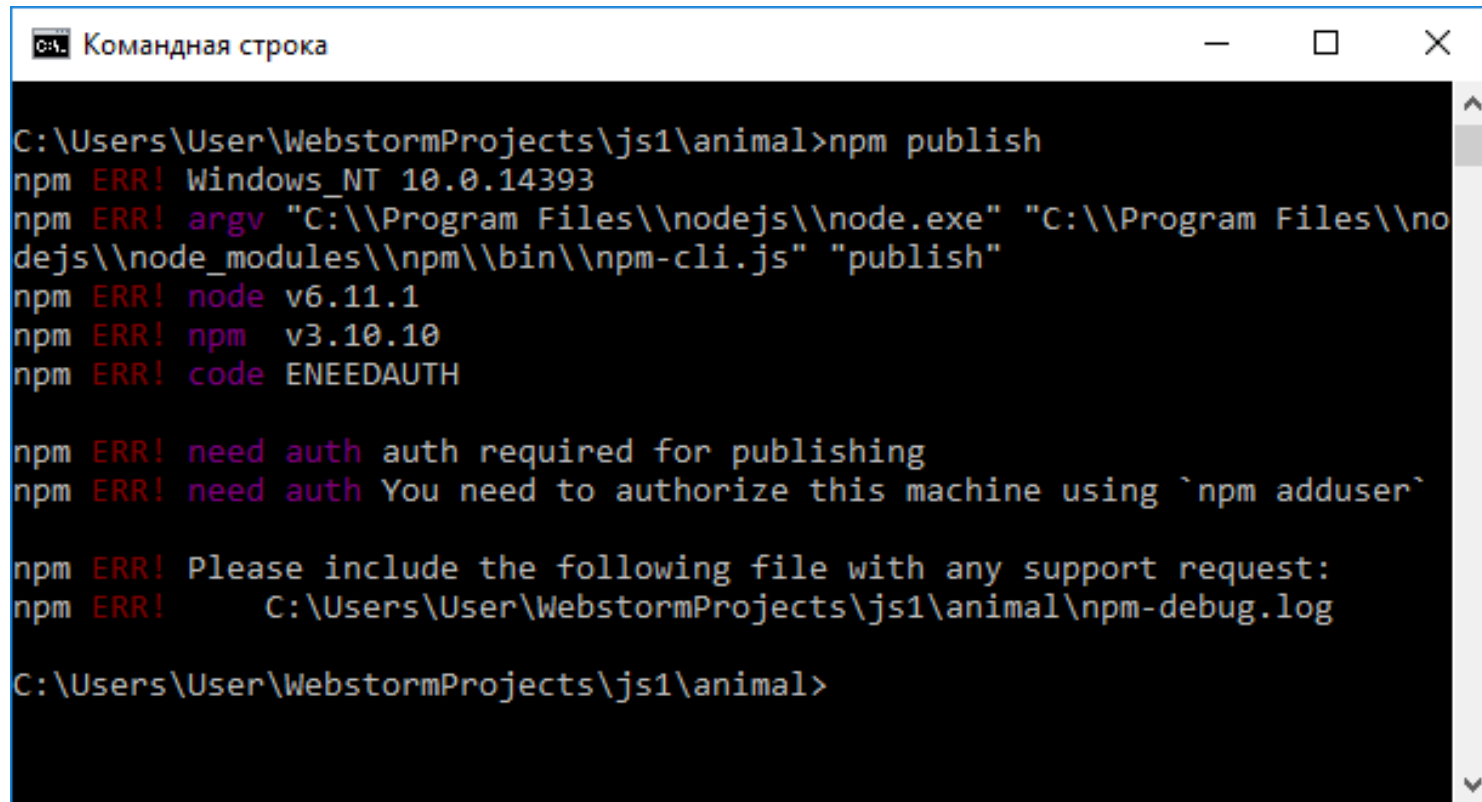
Пример

- Создание: **npm init**
- **name** – имя пакета
- **version** – версия пакета
- **description** – описание
- **author** – автор
- **script** – содержит дополнительные команды для **npm**
 - **start** – **node index.js**
- **main** – главная точка входа в проекте
- **devDependencies** – зависимости на время разработки
- **dependencies** – зависимости пакета
- **license** – лицензия
- **keywords** – ключевые слова, описывающие пакет

```
{  
  "name": "my-project",  
  "version": "0.0.1",  
  "description": "This is a project.",  
  "author": "Someone <someone@example.com>",  
  "contributors": [{  
    "name": "Someone Else",  
    "email": "else@example.com"  
  }],  
  "keywords": ["improves", "searching"]  
}
```


npm publish – публикация

17



```
Командная строка

C:\Users\User\WebstormProjects\js1\animal>npm publish
npm ERR! Windows_NT 10.0.14393
npm ERR! argv "C:\\Program Files\\nodejs\\node.exe" "C:\\Program Files\\nodejs\\node_modules\\npm\\bin\\npm-cli.js" "publish"
npm ERR! node v6.11.1
npm ERR! npm v3.10.10
npm ERR! code ENEEDAUTH

npm ERR! need auth auth required for publishing
npm ERR! need auth You need to authorize this machine using `npm adduser`

npm ERR! Please include the following file with any support request:
npm ERR!      C:\Users\User\WebstormProjects\js1\animal\npm-debug.log

C:\Users\User\WebstormProjects\js1\animal>
```

npm help

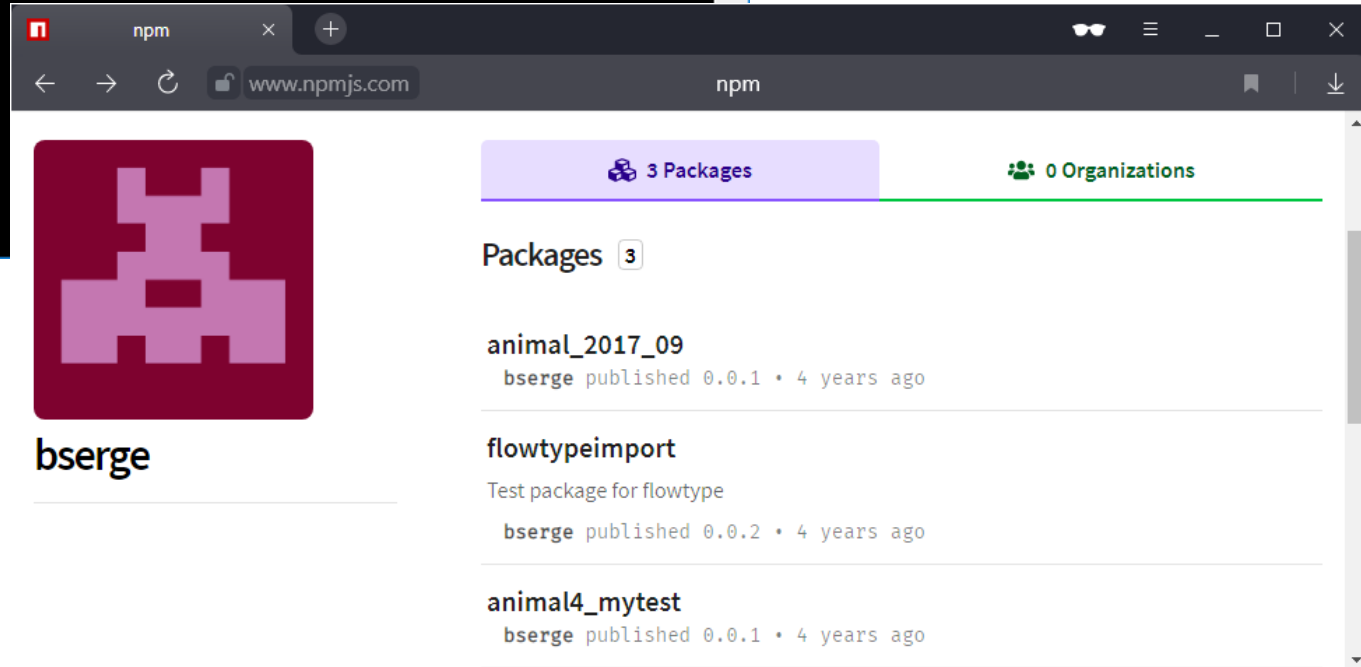
npm adduser – добавление пользователя 18
npm publish – публикация

```
Командная строка

C:\Users\User\WebstormProjects\js1\animal>npm adduser
Username: bserge
Password:
Email: (this IS public) constr17@bk.ru
Logged in as bserge on https://registry.npmjs.org/.

C:\Users\User\WebstormProjects\js1\animal>npm publish
+ animal_2017_09@0.0.1

C:\Users\User\WebstormProjects\js1\animal>
```



<https://www.npmjs.com/~bserge>

npm s animal_2017_09 – поиск

npm i animal_2017_09 – установка

```
Командная строка

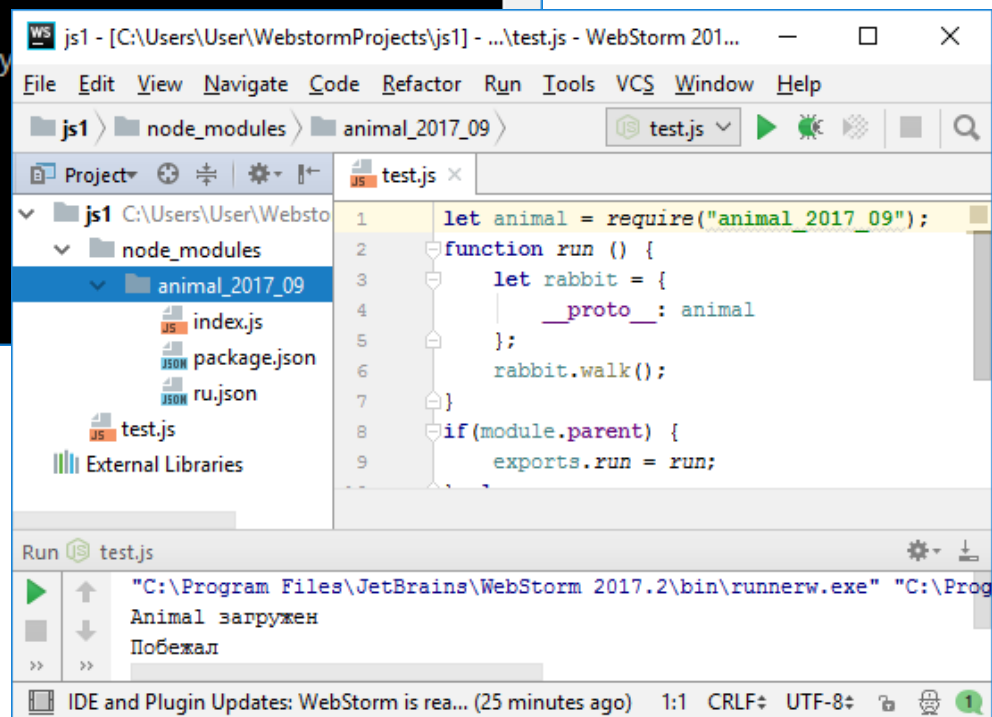
C:\Users\User\WebstormProjects\js1>npm s animal_2017_09
npm WARN notice update to the newest npm client for improved search results:
  npmjs.com/get-npm
NAME      DESCRIPTION  AUTHOR  DATE      VERSION  KEYWORDS
animal_2017_09  =bserge 2017-09-16 0.0.1

C:\Users\User\WebstormProjects\js1>npm i animal_2017_09
C:\Users\User\WebstormProjects\js1
`-- animal_2017_09@0.0.1

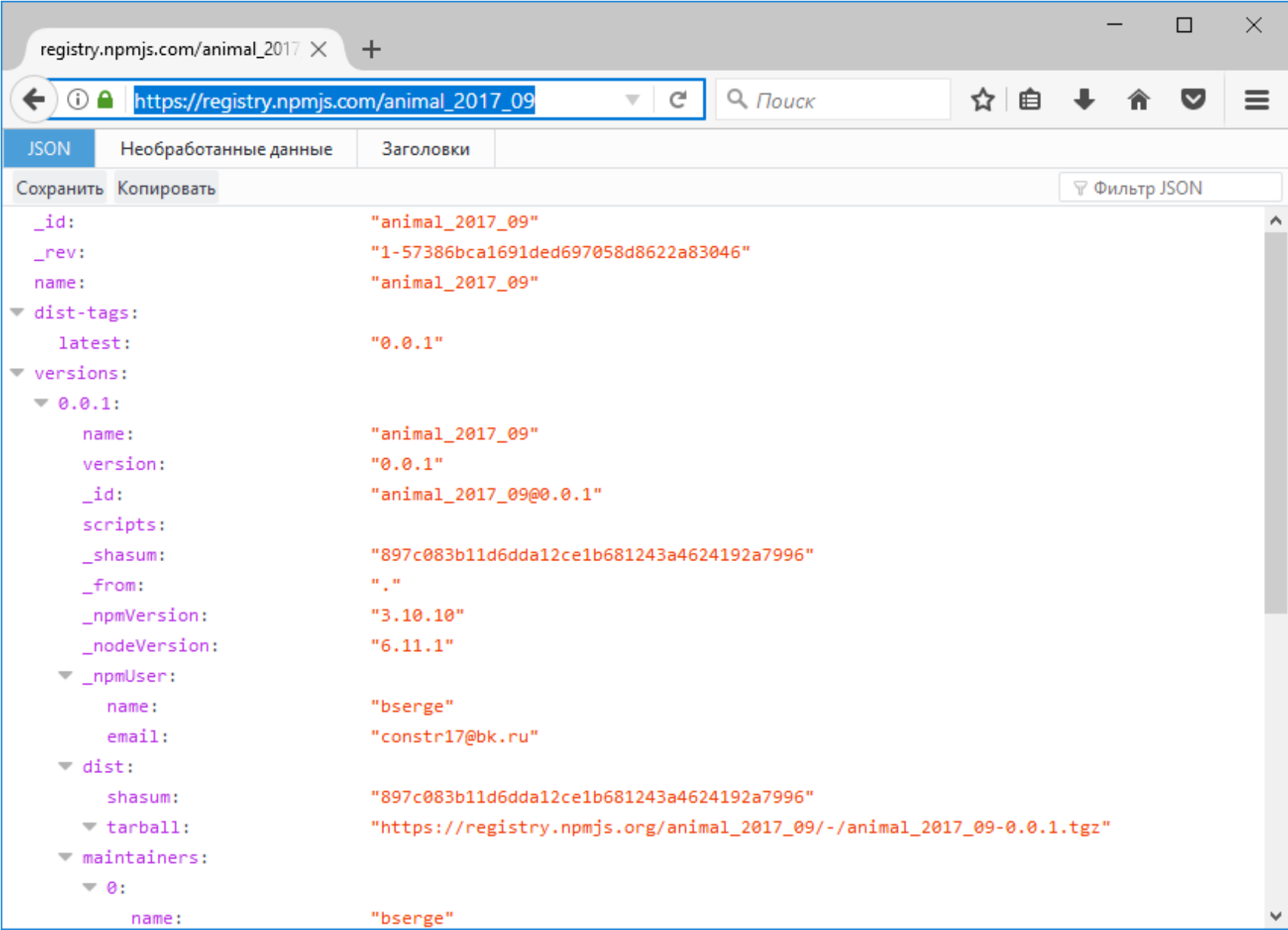
npm WARN enoent ENOENT: no such file or directory
ormProjects\js1\package.json'
npm WARN js1 No description
npm WARN js1 No repository field.
npm WARN js1 No README data
npm WARN js1 No license field.

C:\Users\User\WebstormProjects\js1>_
```

Глобальная установка
npm i -g animal_2017_09



https://registry.npmjs.com/animal_2017_09 20



The screenshot shows a web browser window displaying the npm registry page for the package 'animal_2017_09'. The address bar shows the URL 'https://registry.npmjs.com/animal_2017_09'. The page has a tabbed interface with 'JSON' selected. Below the tabs are buttons for 'Сохранить' (Save) and 'Копировать' (Copy), and a 'Фильтр JSON' (JSON Filter) input field. The main content area displays the JSON data for the package, which is partially expanded to show the 'versions' section.

```
{
  "_id": "animal_2017_09",
  "_rev": "1-57386bca1691ded697058d8622a83046",
  "name": "animal_2017_09",
  "dist-tags": {
    "latest": "0.0.1"
  },
  "versions": {
    "0.0.1": {
      "name": "animal_2017_09",
      "version": "0.0.1",
      "_id": "animal_2017_09@0.0.1",
      "scripts": {},
      "_shasum": "897c083b11d6dda12ce1b681243a4624192a7996",
      "_from": ".",
      "_npmVersion": "3.10.10",
      "_nodeVersion": "6.11.1",
      "_npmUser": {
        "name": "bserge",
        "email": "constr17@bk.ru"
      },
      "dist": {
        "shasum": "897c083b11d6dda12ce1b681243a4624192a7996",
        "tarball": "https://registry.npmjs.org/animal_2017_09/-/animal_2017_09-0.0.1.tgz"
      },
      "maintainers": [
        {
          "name": "bserge"
        }
      ]
    }
  }
}
```

npm – использование

- Инициация проекта
 - **npm init**
- Добавление модуля
 - **npm install [package]**
 - Глобальный модуль
 - **npm install -g [package]**
- Добавление зависимости в **package.json**
 - **npm install --save [package]**
- Удаление
 - **npm uninstall [package]**
- Загрузить всё, что указано в **package.json**
 - **npm install**
- Публикация пакета
 - **npm adduser**
 - **npm login**
 - **npm publish**

npx module_name

npx -p package_name module_name

npx --help

```
Command Prompt
C:\Users\User>npx --help
Execute binaries from npm packages.

npx [options] <command>[@version] [command-arg]...

npx [options] [-p|--package <package>]... <command> [command-arg]...

npx [options] -c '<command-string>'

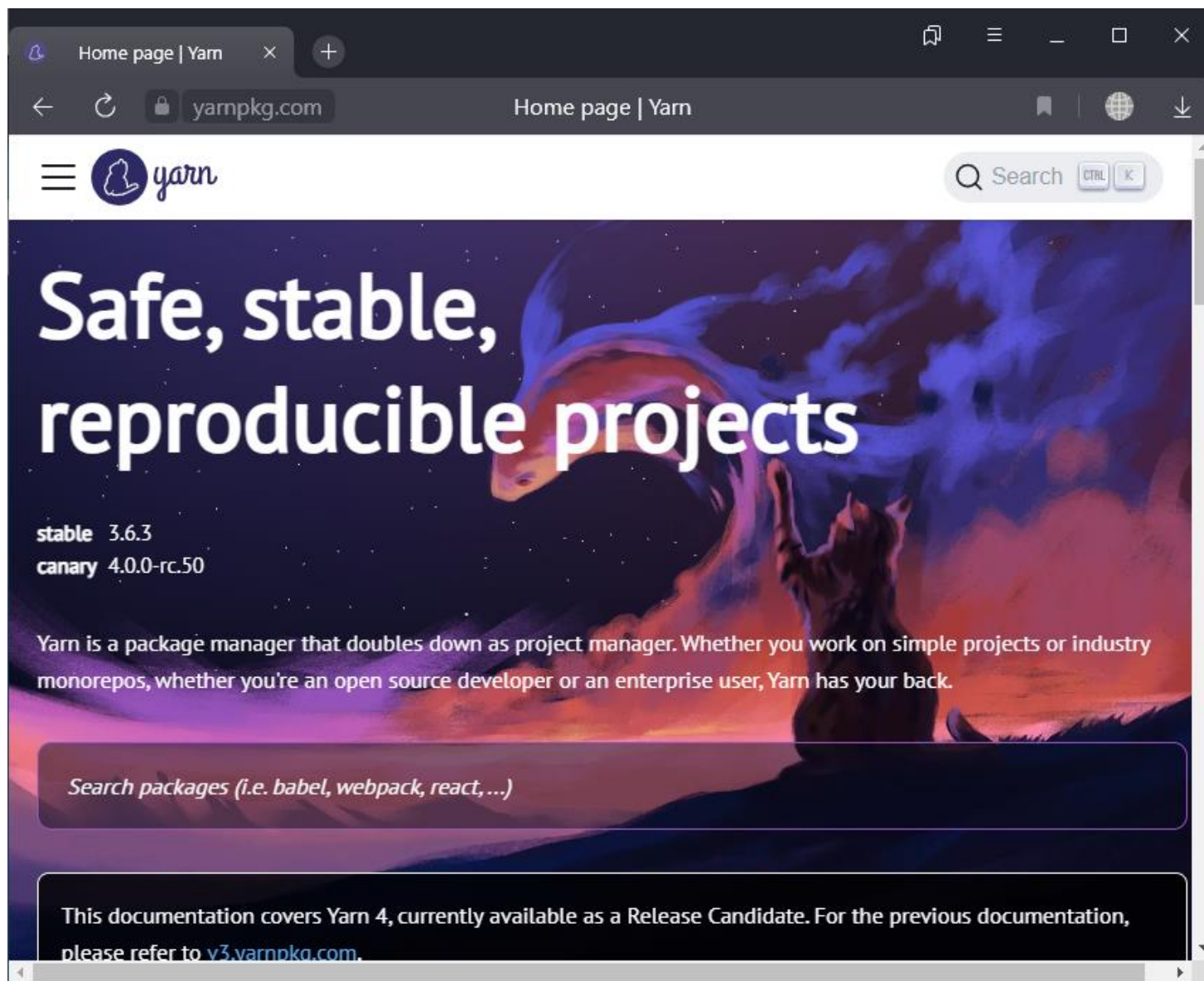
npx --shell-auto-fallback [shell]

Options:
  --package, -p          Package to be installed.                  [string]
  --cache                Location of the npm cache.                 [string]
  --always-spawn         Always spawn a child process to execute the command. [boolean]
  --no-install           Skip installation if a package is missing.  [boolean]
  --userconfig           Path to user npmrc.                        [string]
  --call, -c            Execute string as if inside `npm run-script`. [string]
  --shell, -s           Shell to execute the command with, if any.  [string] [default: false]
  --shell-auto-fallback Generate shell code to use npx as the "command not found" fallback. [string] [choices: "", "bash", "fish", "zsh"]
  --ignore-existing      Ignores existing binaries in $PATH, or in the local project. This forces npx to do a temporary install and use the latest version. [boolean]
  --quiet, -q           Suppress output from npx itself. Subcommands will not be affected. [boolean]
  --npm                 npm binary to use for internal operations. [string] [default: "C:\Users\User\AppData\Roaming\npm\node_modules\npm\bin\npm-cli.js"]
  --node-arg, -n        Extra node argument when calling a node binary. [string]
  --version, -v         Show version number [boolean]
  --help, -h           Show help [boolean]
```

- Запуск локально установленных пакетов
- Выполнение пакета, который не установлен
- Тестирование разных версий пакетов

yarn – аналог npm

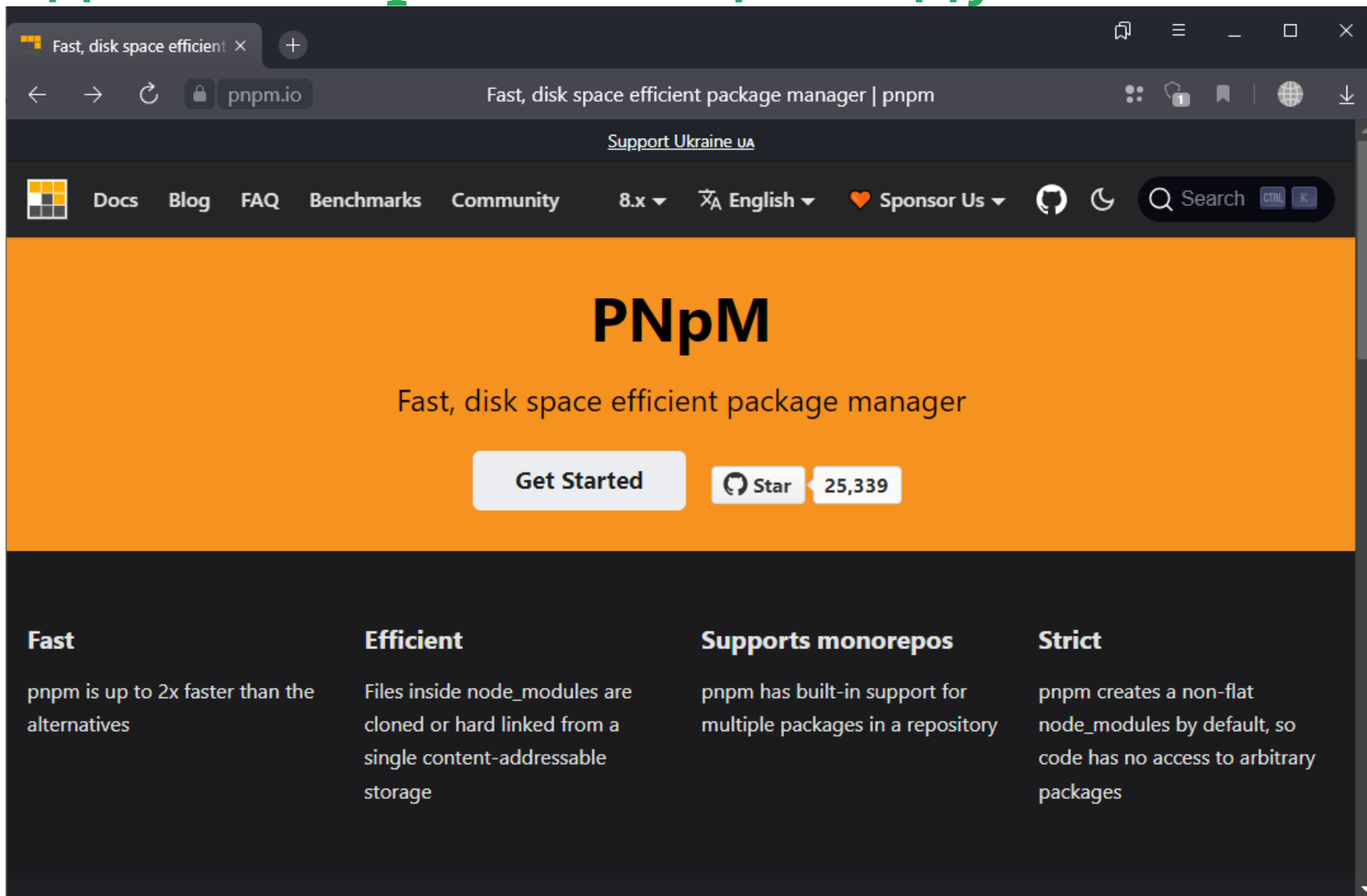
<https://yarnpkg.com/>



yarn – использование

- Инициация проекта
 - **yarn init**
- Добавление модуля
 - **yarn add [package]**
 - **yarn add [package]@[version]**
 - **yarn add [package]@[tag]**
- Добавление зависимости в **devDependencies**
 - **yarn add [package] --dev**
- Обновление зависимости
 - **yarn upgrade [package]**
 - **yarn upgrade [package]@[version]**
 - **yarn upgrade [package]@[tag]**
- Удаление
 - **yarn remove [package]**
- Загрузить всё, что указано в **package.json**
 - **yarn**
 - **yarn install**

pnpm – аналог npm с ведением единого хранилища модулей



The screenshot shows the pnpm.io website in a web browser. The browser's address bar displays 'pnpm.io' and the page title is 'Fast, disk space efficient package manager | pnpm'. The website's navigation bar includes links for Docs, Blog, FAQ, Benchmarks, and Community, along with a version selector (8.x), language selector (English), and a 'Sponsor Us' button. A search bar is also present. The main content area features the pnpm logo, the tagline 'Fast, disk space efficient package manager', a 'Get Started' button, and a GitHub Star badge showing 25,339 stars. Below this, four key features are highlighted: Fast (2x faster than alternatives), Efficient (cloned or hard linked files), Supports monorepos (built-in support for multiple packages), and Strict (non-flat node_modules by default).

Fast, disk space efficient package manager | pnpm

Support Ukraine ua

Docs Blog FAQ Benchmarks Community 8.x English Sponsor Us Search

PNpM

Fast, disk space efficient package manager

Get Started Star 25,339

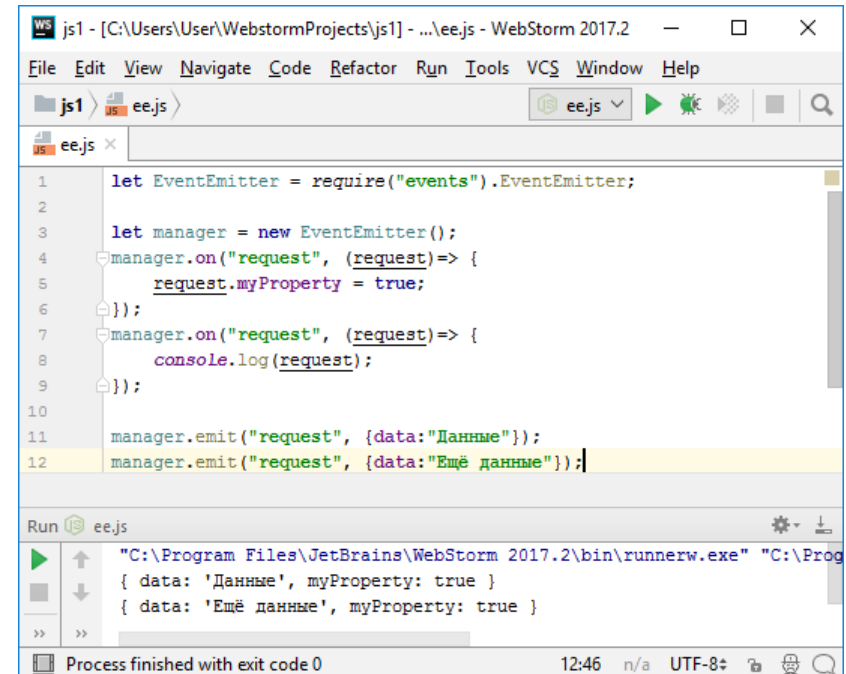
Fast	Efficient	Supports monorepos	Strict
pnpm is up to 2x faster than the alternatives	Files inside node_modules are cloned or hard linked from a single content-addressable storage	pnpm has built-in support for multiple packages in a repository	pnpm creates a non-flat node_modules by default, so code has no access to arbitrary packages

<https://pnpm.io/> - доступен только через VPN

Обработка событий (EventEmitter)

26

```
let EventEmitter = require("events").EventEmitter;
let manager = new EventEmitter();
manager.on("request", (request)=> {
    request.myProperty = true;
});
manager.on("request", (request)=> {
    console.log(request);
});
manager.emit("request", {data:"Данные"});
manager.emit("request", {data:"Ещё данные"});
```

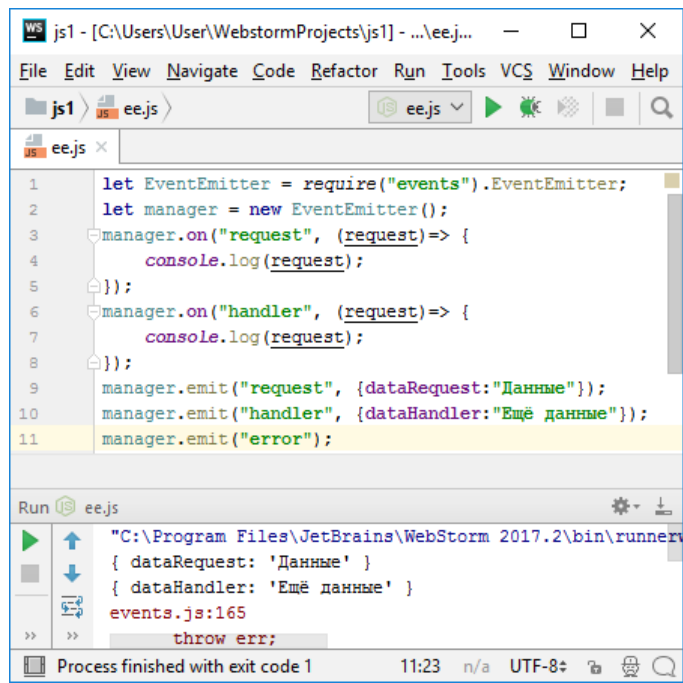


```
js1 - [C:\Users\User\WebstormProjects\js1] - ...ee.js - WebStorm 2017.2
File Edit View Navigate Code Refactor Run Tools VCS Window Help
js1 ee.js
ee.js x
1 let EventEmitter = require("events").EventEmitter;
2
3 let manager = new EventEmitter();
4 manager.on("request", (request)=> {
5     request.myProperty = true;
6 });
7 manager.on("request", (request)=> {
8     console.log(request);
9 });
10
11 manager.emit("request", {data:"Данные"});
12 manager.emit("request", {data:"Ещё данные"});

Run ee.js
"C:\Program Files\JetBrains\WebStorm 2017.2\bin\runnerw.exe" "C:\Prog
{ data: 'Данные', myProperty: true }
{ data: 'Ещё данные', myProperty: true }

Process finished with exit code 0 12:46 n/a UTF-8
```

```
let EventEmitter = require("events").EventEmitter;
let manager = new EventEmitter();
manager.on("request", (request)=> {
    console.log(request);
});
manager.on("handler", (request)=> {
    console.log(request);
});
manager.emit("request", {dataRequest:"Данные"});
manager.emit("handler", {dataHandler:"Ещё данные"});
manager.emit("error");
```



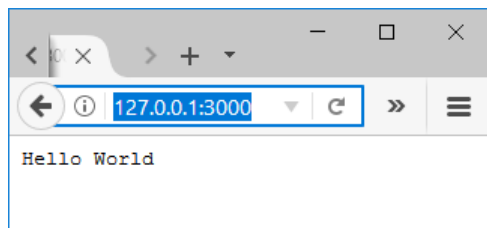
```
js1 - [C:\Users\User\WebstormProjects\js1] - ...eej...
File Edit View Navigate Code Refactor Run Tools VCS Window Help
js1 ee.js
ee.js
1 let EventEmitter = require("events").EventEmitter;
2 let manager = new EventEmitter();
3 manager.on("request", (request)=> {
4     console.log(request);
5 });
6 manager.on("handler", (request)=> {
7     console.log(request);
8 });
9 manager.emit("request", {dataRequest:"Данные"});
10 manager.emit("handler", {dataHandler:"Ещё данные"});
11 manager.emit("error");

Run ee.js
"C:\Program Files\JetBrains\WebStorm 2017.2\bin\runner...
{ dataRequest: 'Данные' }
{ dataHandler: 'Ещё данные' }
events.js:165
    throw err;

Process finished with exit code 1 11:23 n/a UTF-8
```

Простейший сервер (1)

28



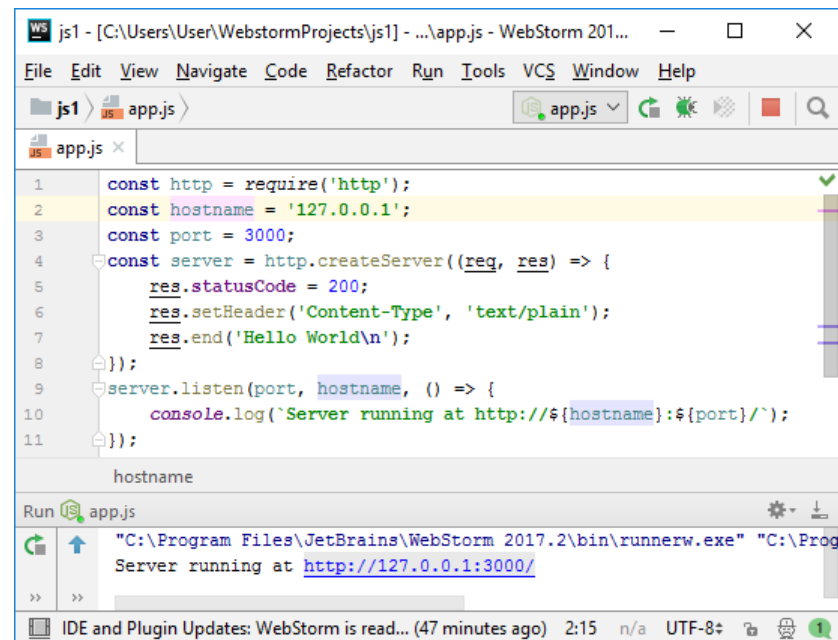
```
const http = require('http');
```

```
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\n');  
});
```

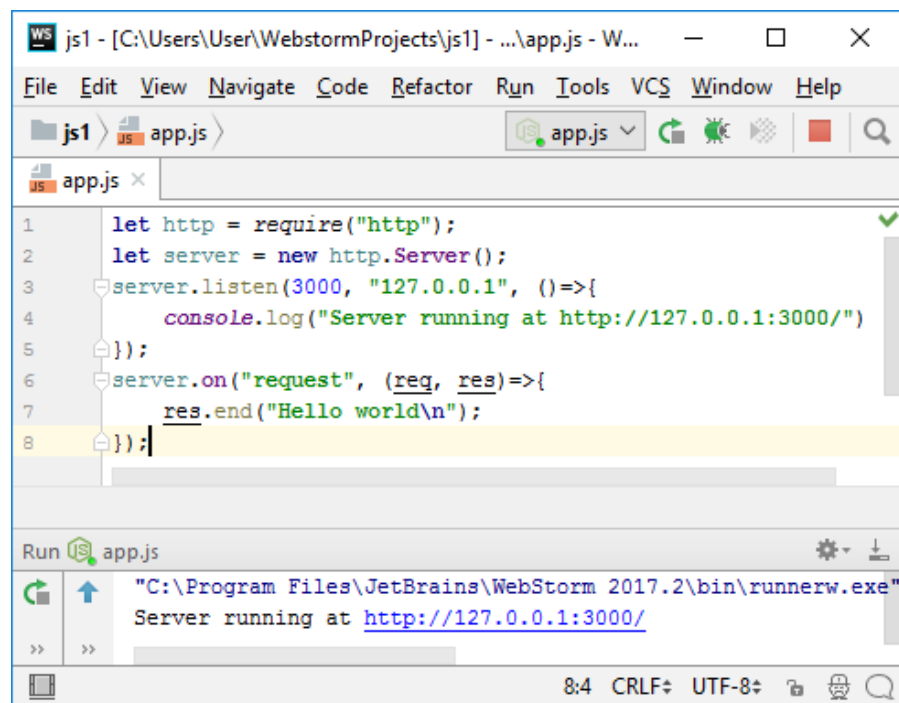
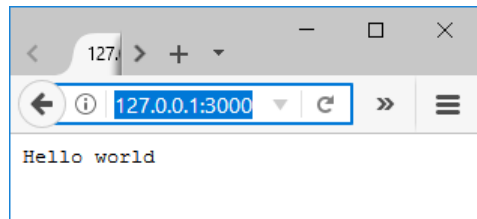
```
server.listen(port, hostname, () => {  
  console.log(`Server running at http://${hostname}:${port}/`);  
});
```



Простейший сервер (2)

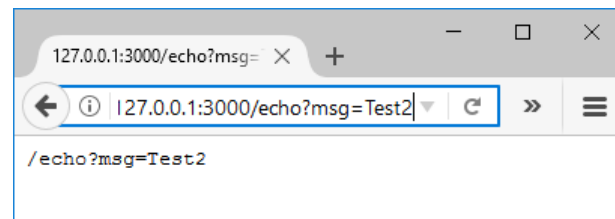
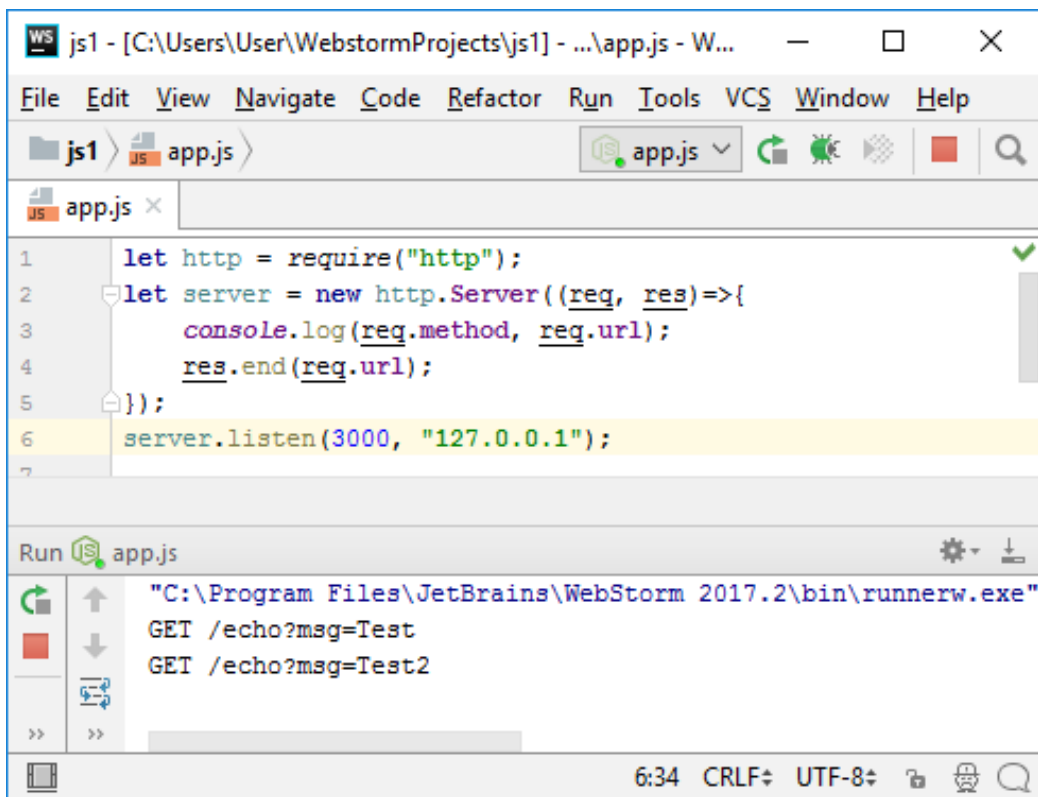
29

```
let http = require("http");
let server = new http.Server();
server.listen(3000, "127.0.0.1", ()=>{
    console.log("Server running at http://127.0.0.1:3000/")
});
server.on("request", (req, res)=>{
    res.end("Hello world\n");
});
```



Echo-сервер

```
let http = require("http");  
let server = new http.Server((req, res)=>{  
    console.log(req.method, req.url);  
});  
server.listen(3000, "127.0.0.1");
```



HTTPS-сервер

31

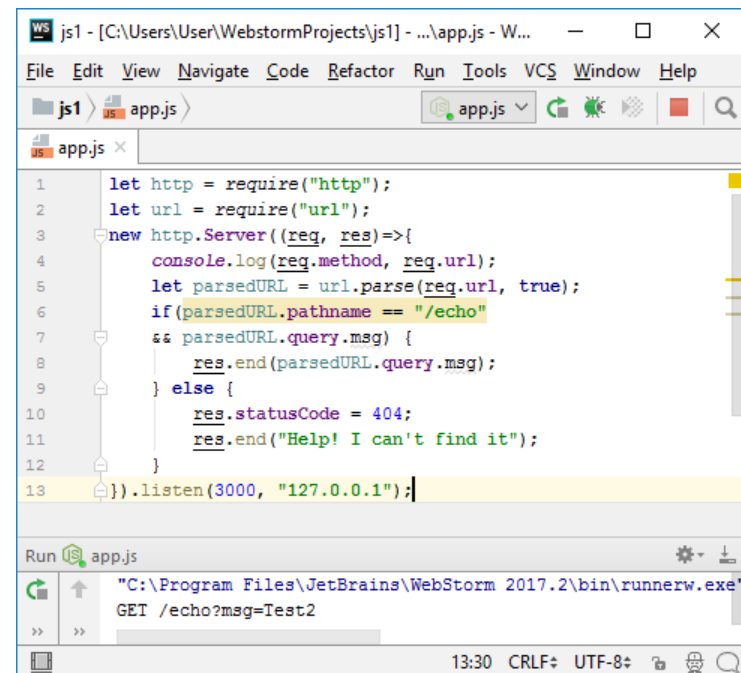
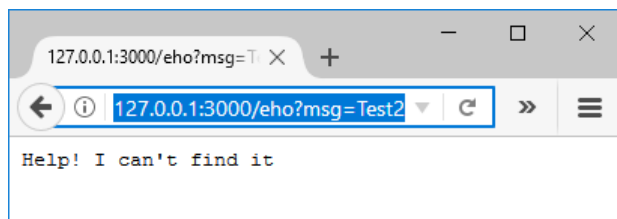
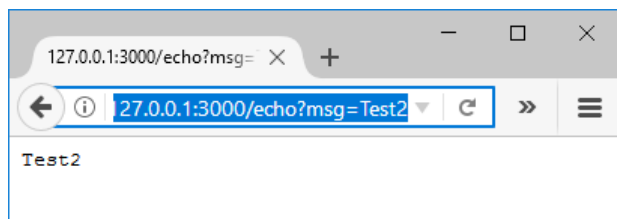
```
// curl -k https://localhost:443/  
const https = require('https');  
const fs = require('fs');  
  
const options = {  
  key: fs.readFileSync('cert/example.com_nginx.key'),  
  cert: fs.readFileSync('cert/example.com_nginx.crt')  
}; // Важно! Последовательное чтение файлов  
  
https.createServer(options, (req, res) => {  
  res.writeHead(200);  
  res.end('hello world\n');  
}).listen(443);
```

```
openssl req -new -sha256 -key example.com_nginx.key -out example.com.csr
```

Разбор параметров запроса

32

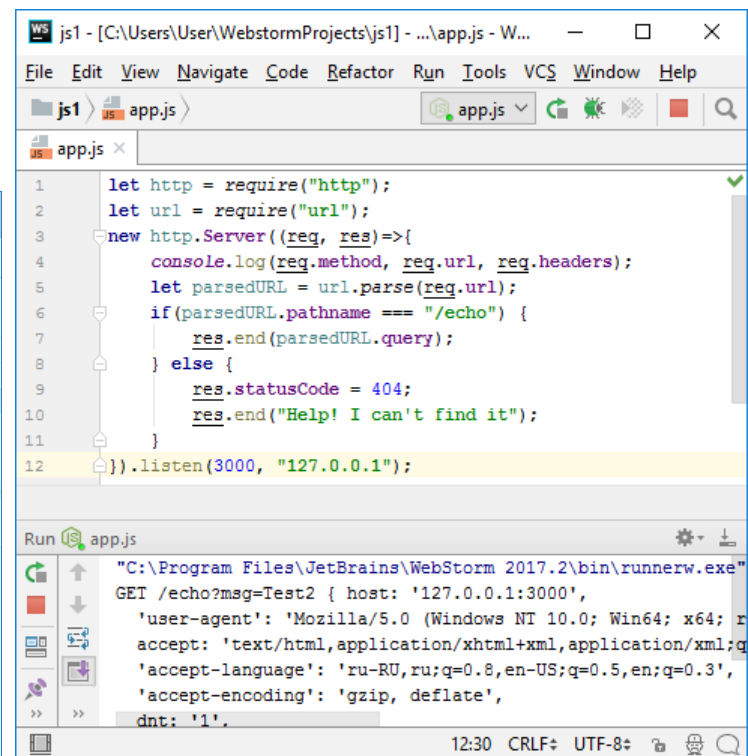
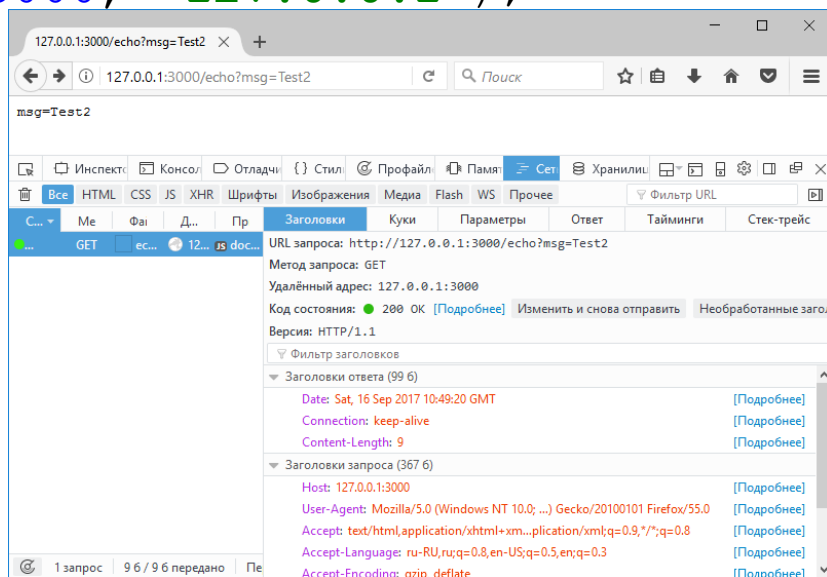
```
let http = require("http");
let url = require("url");
new http.Server((req, res)=>{
  console.log(req.method, req.url);
  let parsedURL = url.parse(req.url, true);
  if(parsedURL.pathname == "/echo"
  && parsedURL.query.msg) {
    res.end(parsedURL.query.msg);
  } else {
    res.statusCode = 404;
    res.end("Help! I can't find it")
  }
}).listen(3000, "127.0.0.1");
```



Заголовки

```
let http = require("http");
let url = require("url");
new http.Server((req, res)=>{
  console.log(req.method, req.url, req.headers);
  let parsedURL = url.parse(req.url);
  if(parsedURL.pathname === "/echo") {
    res.end(parsedURL.query);
  } else {
    res.statusCode = 404;
    res.end("Help! I can't find it")
  }
}).listen(3000, "127.0.0.1");
```

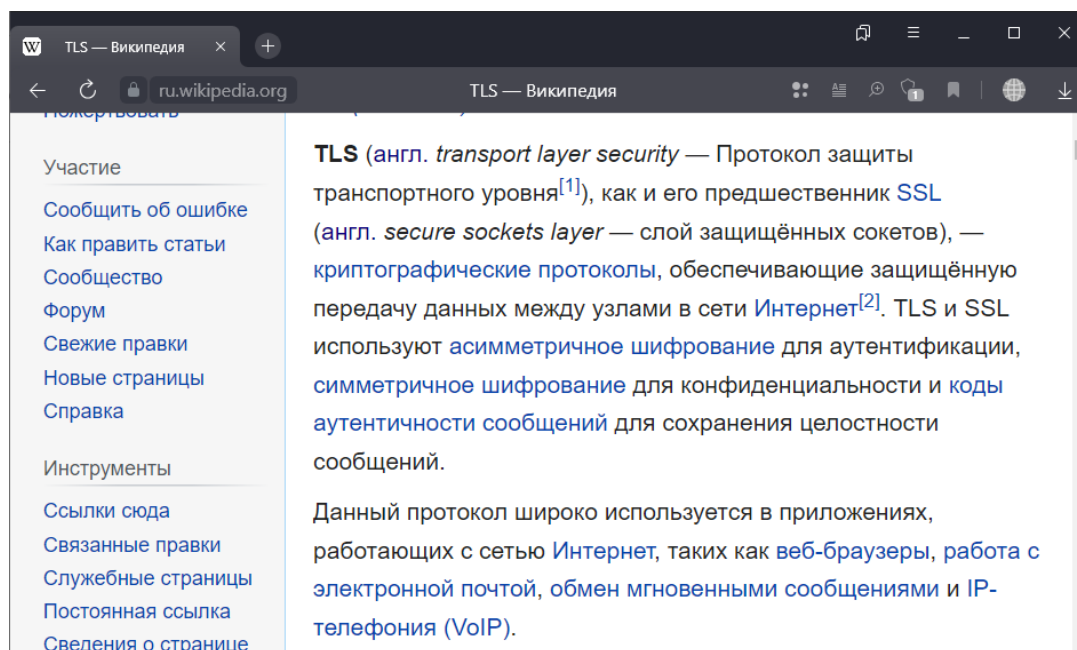
`res.setHeader(name, value)`



TLS сокет: сервер и клиент

Подготовка ключей

- `openssl genrsa -out private-key.pem 1024`
- `openssl req -new -key private-key.pem -out csr.pem`
- `openssl x509 -req -in csr.pem -signkey private-key.pem -out public-cert.pem`



<https://ru.wikipedia.org/wiki/TLS>

TLS сокет-сервер

```
var fs = require('fs');
var options = {
  key: fs.readFileSync('private-key.pem'),
  cert: fs.readFileSync('public-cert.pem')
};
var server = require('tls').createServer(options, function(socket) {
  socket.write("Message from server"); // Приветствие
  socket.on('data', function(data) { // Печать данных
    console.log('Recieved: '${data.toString()}'');
  });
  socket.on('end', function() { // Передача завершена
    console.log('EOT');
  });
});
server.listen(1337, '127.0.0.1', function() { // Старт
  console.log("I'm listening at %s, on port %s", '127.0.0.1', 1337);
});
server.on('error', function(error) {
  console.error(error); // Обработка ошибки
  server.destroy(); // Удаление
});
```

TLS сокет-клиент

36

```
var fs = require('fs');
var options = { // Сертификаты
  key: fs.readFileSync('private-key.pem'),
  cert: fs.readFileSync('public-cert.pem'),
  rejectUnauthorized: false
};
var client = require('tls').connect(1337, '127.0.0.1', options, function() {
  if (client.authorized) // Проверка авторизации
    console.log("Connection authorized by a Certificate Authority.");
  else
    console.log("Connection not authorized: " + client.authorizationError);
  client.write("Message from client"); // Приветствие
});
client.on("data", function(data) {
  console.log(`Recieved: '${data.toString()}'`);
  client.end(); // Закрывает соединение
});
client.on('close', function() {
  console.log("Connection closed");
});
client.on('error', function(error) {
  console.error(error); // Отображение ошибки
  client.destroy(); // Закрытие после ошибки
});
```

Журнал сервера:

I'm listening at 127.0.0.1, on port 1337
Recieved: 'Message from client'
EOT

Журнал клиента:

Connection not authorized:
DEPTH_ZERO_SELF_SIGNED_CERT
Recieved: 'Message from server'
Connection closed

npm i debug

37

The screenshot shows the npm website for the 'debug' package. The package is version 4.3.4, published a year ago, and is public. It has 1 dependency, 48,021 dependents, and 71 versions. The package is available on GitHub at github.com/debug-js/debug. The homepage is also github.com/debug-js/debug#.... The weekly downloads are 228,721,018. The license is MIT. The unpacked size is 42.4 kB and the total files are 7.

debug

4.3.4 • Public • Published a year ago

[Readme](#) [Code](#) [Beta](#) [1 Dependency](#) [48 021 Dependents](#) [71 Versions](#)

debug

build unknown coverage unknown [Slack](#) backers 13 sponsors 7

Install

```
> npm i debug
```

Repository

github.com/debug-js/debug

Homepage

github.com/debug-js/debug#...

Weekly Downloads

228 721 018

Version	License
4.3.4	MIT

Unpacked Size	Total Files
42.4 kB	7

The screenshot shows a Windows command prompt window titled 'Командная строка'. The user has run the command `npm i debug` in the directory `C:\Users\User\WebstormProjects\js1`. The output shows the installation of `debug@3.0.1` and `ms@2.0.0`. There are several warnings from npm: `ENOENT: no such file or directory, open 'C:\Users\User\WebstormProjects\js1\package.json'`, `js1 No description`, `js1 No repository field.`, `js1 No README data`, and `js1 No license field.`

```
C:\Users\User\WebstormProjects\js1>npm i debug
C:\Users\User\WebstormProjects\js1
-- debug@3.0.1
-- ms@2.0.0

npm WARN enoent ENOENT: no such file or directory, open 'C:\Users\User\WebstormProjects\js1\package.json'
npm WARN js1 No description
npm WARN js1 No repository field.
npm WARN js1 No README data
npm WARN js1 No license field.

C:\Users\User\WebstormProjects\js1>
```

Демонстрация debug

app.js

```
var debug = require('debug')('http')
    , http = require('http')
    , name = 'My App';

debug('booting %o', name);
http.createServer(function(req, res){
    debug(req.method + ' ' + req.url);
    res.end('hello\n');
}).listen(3000, function(){
    debug('listening');
});
```

```
require('./worker');
```

worker.js

```
var a = require('debug')('worker:a')
    , b = require('debug')('worker:b');

function work() {
    a('doing lots of uninteresting work');
    setTimeout(work, Math.random() * 1000);
}

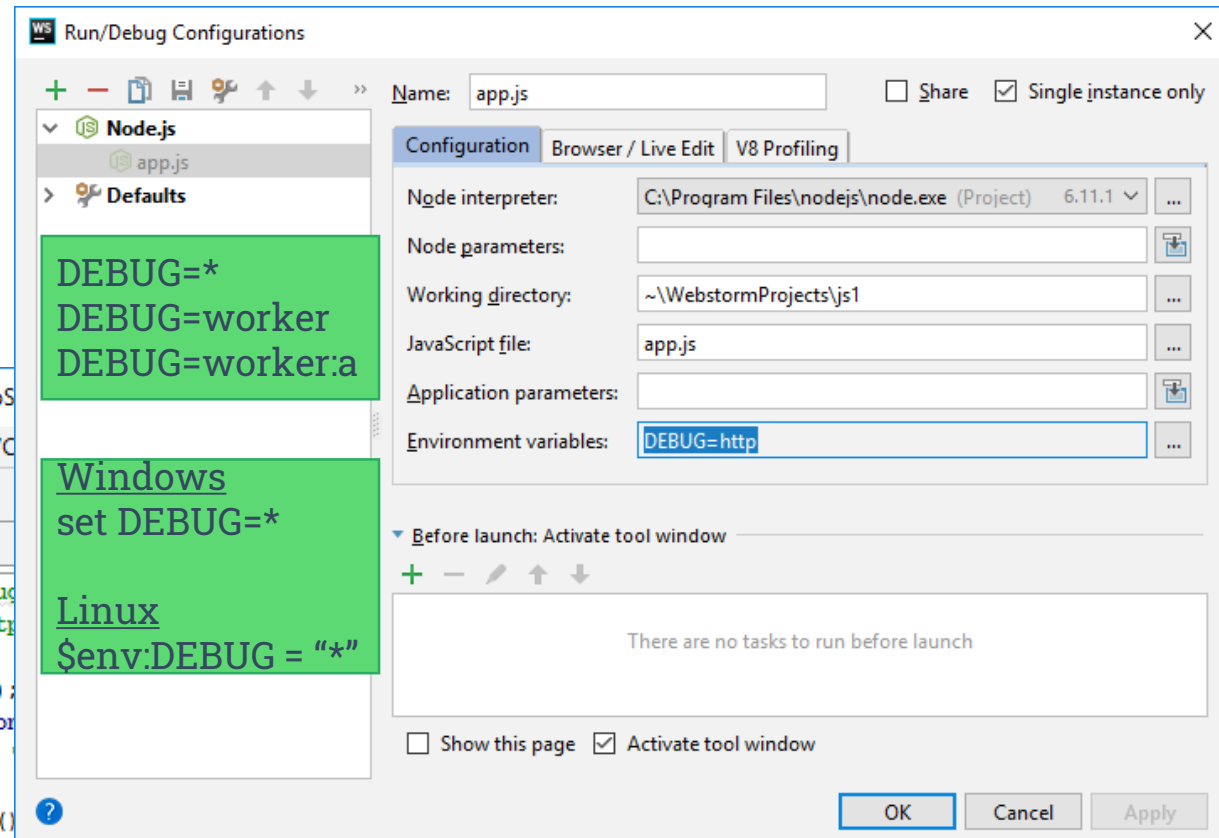
work();

function workb() {
    b('doing some work');
    setTimeout(workb, Math.random() * 2000);
}

workb();
```

DEBUG=http

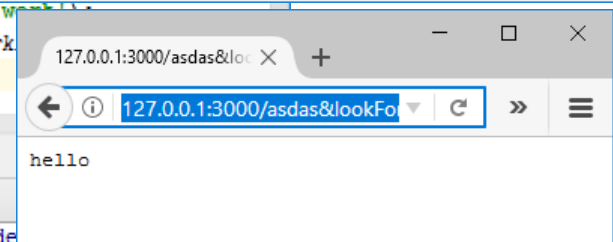
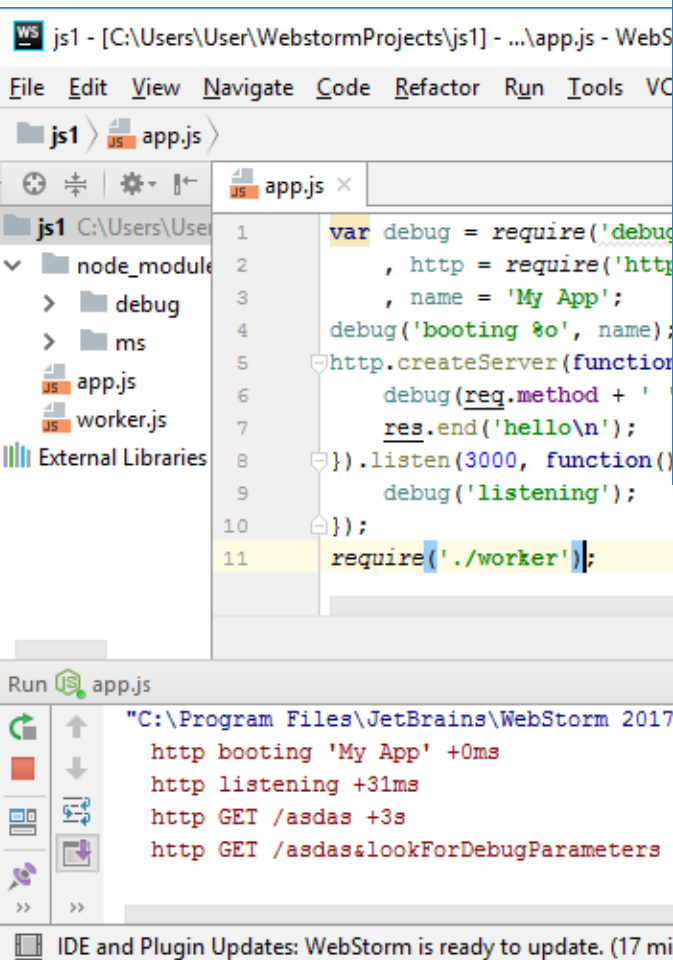
39



DEBUG=*
DEBUG=worker
DEBUG=worker:a

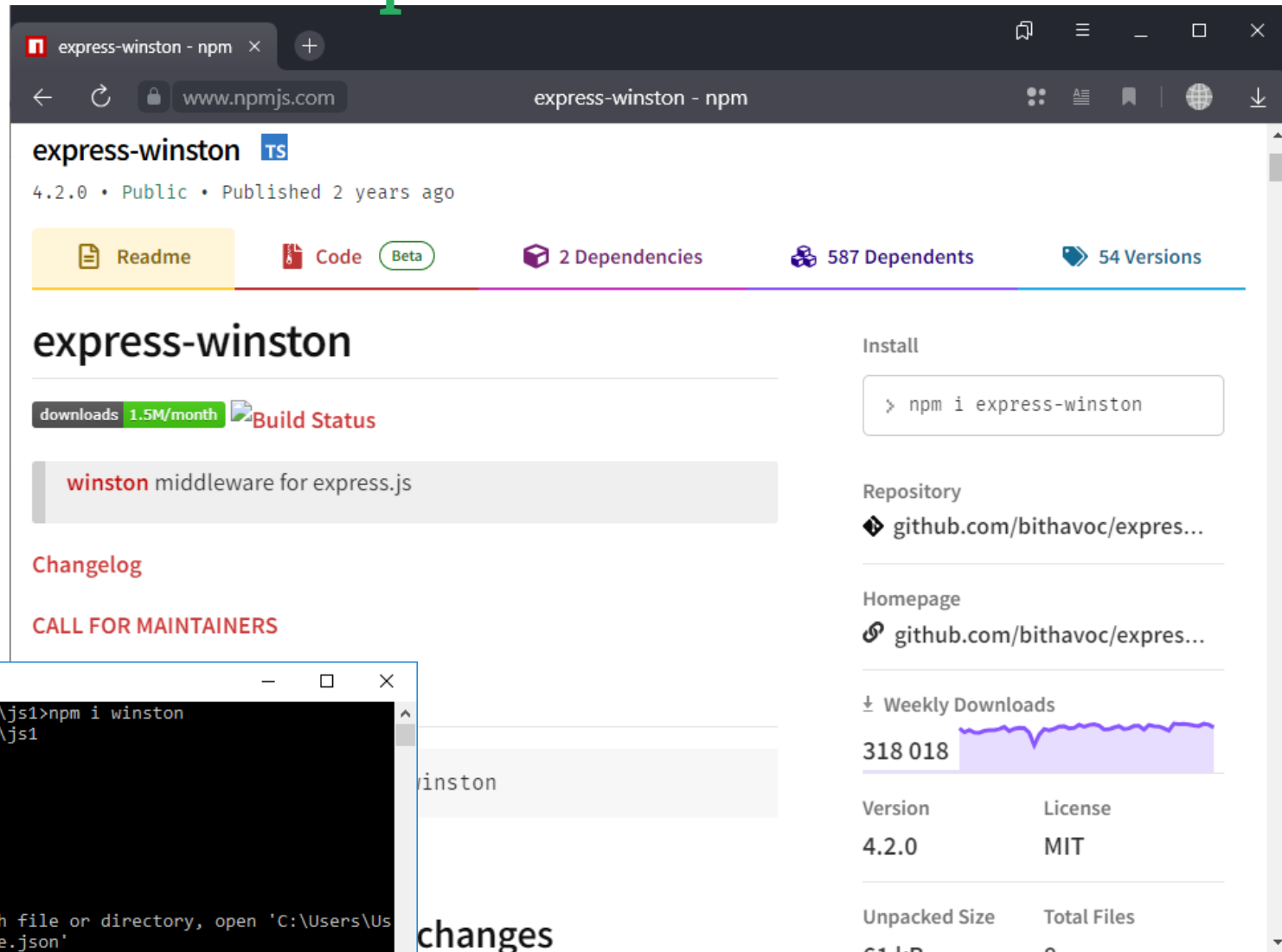
Windows
set DEBUG=*

Linux
\$env:DEBUG = "*"



npm i winston express-winston

40



The screenshot shows the npm package page for **express-winston**. The page header includes the package name, version (4.2.0), and publication status (Public, 2 years ago). Below this are links for Readme, Code (Beta), Dependencies (2), Dependents (587), and Versions (54). The main section features the package name, a download rate of 1.5M/month, and a Build Status link. A description states it is 'winston middleware for express.js'. On the right, there is an 'Install' section with a code block for `npm i express-winston`, a 'Repository' link to `github.com/bithavoc/express-winston`, and a 'Homepage' link to the same repository. A 'Weekly Downloads' chart shows 318,018 downloads. At the bottom, a table lists the version (4.2.0) and license (MIT).

express-winston TS

4.2.0 • Public • Published 2 years ago

[Readme](#) [Code](#) Beta [2 Dependencies](#) [587 Dependents](#) [54 Versions](#)

express-winston

downloads **1.5M/month** [Build Status](#)

winston middleware for express.js

[Changelog](#)

CALL FOR MAINTAINERS

Install

```
> npm i express-winston
```

Repository

[github.com/bithavoc/express...](https://github.com/bithavoc/express-winston)

Homepage

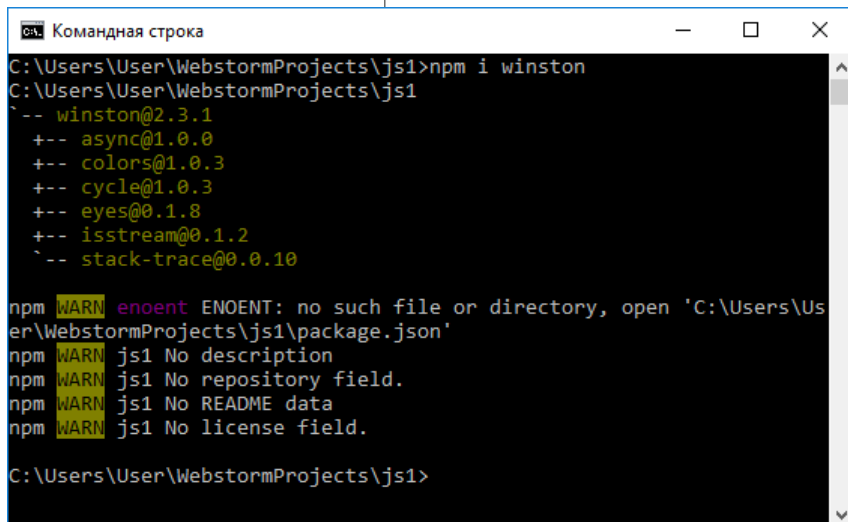
[github.com/bithavoc/express...](https://github.com/bithavoc/express-winston)

Weekly Downloads

318 018

Version	License
4.2.0	MIT

Unpacked Size	Total Files
61 kB	8



The screenshot shows a Windows command prompt window titled 'Командная строка'. The user has entered the command `npm i winston` in a directory `C:\Users\User\WebstormProjects\js1`. The output shows the installation of `winston@2.3.1` and its dependencies: `async@1.0.0`, `colors@1.0.3`, `cycle@1.0.3`, `eyes@0.1.8`, `isstream@0.1.2`, and `stack-trace@0.0.10`. Below this, there are several warning messages from npm indicating that the `package.json` file is missing or incomplete for the project.

```
Командная строка
C:\Users\User\WebstormProjects\js1>npm i winston
C:\Users\User\WebstormProjects\js1>
-- winston@2.3.1
+-- async@1.0.0
+-- colors@1.0.3
+-- cycle@1.0.3
+-- eyes@0.1.8
+-- isstream@0.1.2
+-- stack-trace@0.0.10

npm WARN enoent ENOENT: no such file or directory, open 'C:\Users\User\WebstormProjects\js1\package.json'
npm WARN js1 No description
npm WARN js1 No repository field.
npm WARN js1 No README data
npm WARN js1 No license field.

C:\Users\User\WebstormProjects\js1>
```


Демонстрация winston

app.js

```
var winston = require('winston')
    , http = require('http')
    , name = 'My App';

winston.error('booting %o', name);
http.createServer(function(req, res) {
    winston.info(req.method + ' ' + req.url);
    res.end('hello\n');
}).listen(3000, function() {
    winston.info('listening');
});

require('./worker');
```

worker.js

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

function work() {
    winston.log("debug", 'doing lots of uninteresting work');
    setTimeout(work, Math.random() * 1000);
}

work();

function workb() {
    winston.debug('doing some work');
    setTimeout(workb, Math.random() * 2000);
}

workb();
```

Логгирование по умолчанию

42

The screenshot displays the WebStorm 2017.2 IDE interface. The top toolbar shows the 'Run' button (a green play icon). The main editor area is split into two panes. The left pane shows the file explorer with a project named 'js1' containing 'app.js' and 'worker.js'. The right pane shows the code for 'worker.js'.

```
var winston = require('winston')
, http = require('http')
, name = 'My App';
winston.error('booting %o', name);
http.createServer(function(req, res){
  winston.info(req.method + ' ' + req.url);
  res.end('hello\n');
}).listen(3000, function(){
  winston.info('listening');
});
require('./worker');
```

The bottom pane shows the 'Run' console for 'app.js'. It displays the following output:

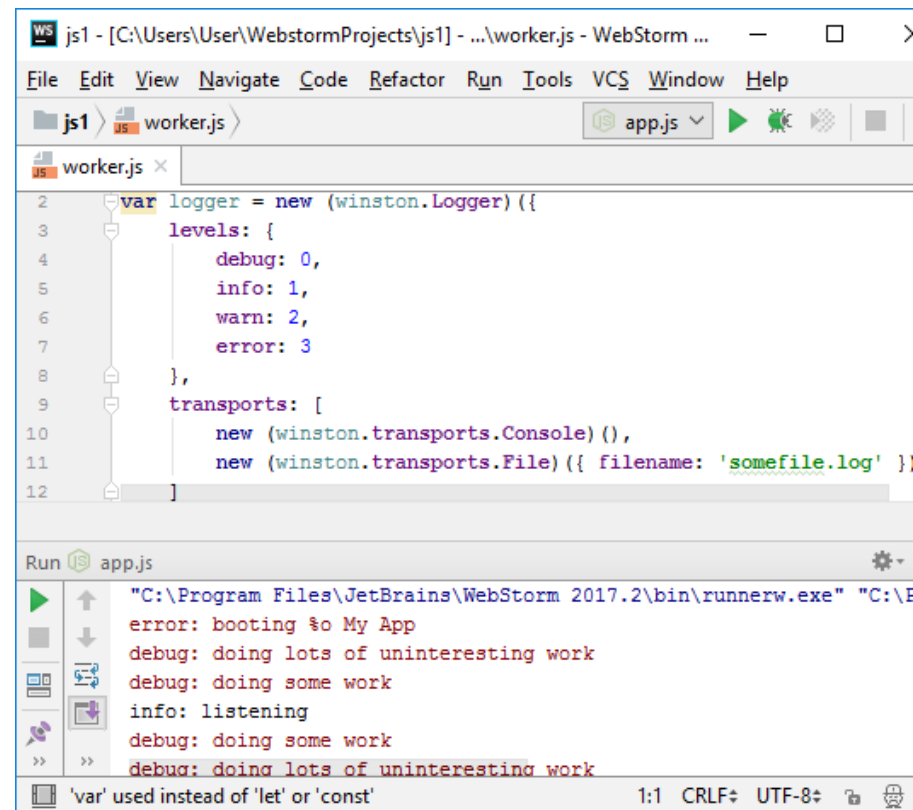
```
"C:\Program Files\JetBrains\WebStorm 2017.2\bin\runnerw.exe" "C:\Program Files\nodejs\node.exe" C:\User
error: booting %o My App
info: listening
info: GET /
```

In the bottom left corner, a browser window is open at the address '127.0.0.1:3000/'. The browser's address bar shows '127.0.0.1:3000' and the page content displays 'hello'.

Настраиваемое логгирование

43

```
var logger = new (winston.Logger) ({
  levels: {
    debug: 0,
    info: 1,
    warn: 2,
    error: 3
  },
  transports: [
    new (winston.transports.Console) (),
    new (winston.transports.File) (
      { filename: 'somefile.log' } )
  ]
});
```



Rollbar | Error Logging & Tracking Service for Software Teams

Rollbar Log in Try free

Don't just track errors, continuously improve your Customer Experience

Proactively discover, predict, and resolve errors in real-time with Rollbar's error monitoring platform.

Sign Up for Free

or

Sign Up with GitHub

Monitors 100-200 concurrent A/B experiments

duolingo

Severin Hacker
Co-founder & CTO

twilio salesforce circleci duolingo wealthsimple +Babbel DOLLAR SHAVE CLUB

- Java
- Node
- PHP
- JavaScript
- Python
- Android
- iOS
- .Net

Rollbar – установка

45

- Node.js
- `npm install --save rollbar`
- `yarn add -D rollbar`
- `let Rollbar = require("rollbar");`
- `let rollbar = new Rollbar("access token");`
- `rollbar.log("TestError: Hello World!");`
- JavaScript
- `<script>`
- `var _rollbarConfig = {`
- `accessToken: "access token",`
- `captureUncaught: true,`
- `captureUnhandledRejections:`
- `true,`
- `payload: {`
- `environment: "production"`
- `}`
- `};`
- `// Rollbar Snippet`
- `// ...`
- `// End Rollbar Snippet`
- `</script>`

Tiestovyi-proiekt - Dashboard - X

https://rollbar.com/bserge3/

Welcome to Rollbar! What's next? 1 of 4 steps complete

- ✓ Integrate a Notifier
- Invite Team Members
- Connect Source Control
- Set Up Deploy Tracking

Top 10 items in last 24 hours

view all Any Environment

No occurrences of active items. [View all items.](#)

Hourly Error/Critical Occurrences

No data

Top 5 active items new/reactivated last day

No active items were new or reactivated in the last day.

Daily Error/Critical Occurrences

No data

7day Trend

Count	Title
2	#1 Hello world!

Tiestovyi-proiekt - Items - Rollbar - X

https://rollbar.com/bserge3/

Welcome to Rollbar! What's next? 1 of 4 steps complete

- ✓ Integrate a Notifier
- Invite Team Members
- Connect Source Control
- Set Up Deploy Tracking

Count: 1 Seen: [all dates](#) Activated: [all dates](#) Live Update

Learn more about advanced search

Search

Critical Error Warning Info Debug

SOURCE Show All ENVIRONMENT Show All OWNER Show All STATUS Active

Total	Last 1	Item	Envi...	Level	Owner
2	2 days	#1 Hello world!	deve...	Debug	

Hello world!

Previous page Next page

Download CSV

Сообщения

- Slack, HipChat, Flowdock, Campfire

Контроль исходного кода

- GitHub, Bitbucket, GitLab

Отслеживания ошибок

- JIRA, Trello, GitHub, Bitbucket, GitLab, Pivotal, Asana, Sprintly

Управление инцидентами

- PagerDuty, VictorOps, OpsGenie, Datadog

https://sentry.io/for/javascript/

48



The screenshot shows the Sentry website for JavaScript Error and Performance Monitoring. The browser tab is titled "JavaScript Error and Performance Monitoring | Sentry". The URL bar shows "sentry.io". The navigation bar includes the Sentry logo, links for "PRODUCT >", "PRICING", "DOCS", "SIGN IN", and a "GET STARTED" button. The main content area features a large yellow "JS" logo, the heading "JavaScript Error and Performance Monitoring", and a paragraph: "Resolve JavaScript errors with max efficiency, not max effort. Get actionable insights to resolve JavaScript performance issues with the ability to track, debug, and resolve JavaScript errors across platforms." Below this are two buttons: "TRY SENTRY FOR FREE" and "SEE THE DOCS".

JavaScript Error and Performance Monitoring | Sentry

SENTRY PRODUCT > PRICING DOCS SIGN IN GET STARTED

JS

JavaScript Error and Performance Monitoring

Resolve JavaScript errors with max efficiency, not max effort. Get actionable insights to resolve JavaScript performance issues with the ability to track, debug, and resolve JavaScript errors across platforms.

TRY SENTRY FOR FREE SEE THE DOCS

- JavaScript
- .NET
- Python
- Android
- PHP
- Django
- Java
- Flask
- Ruby
- Laravel
- iOS
- Rails
- Node
- React
- Go

Sentry – установка

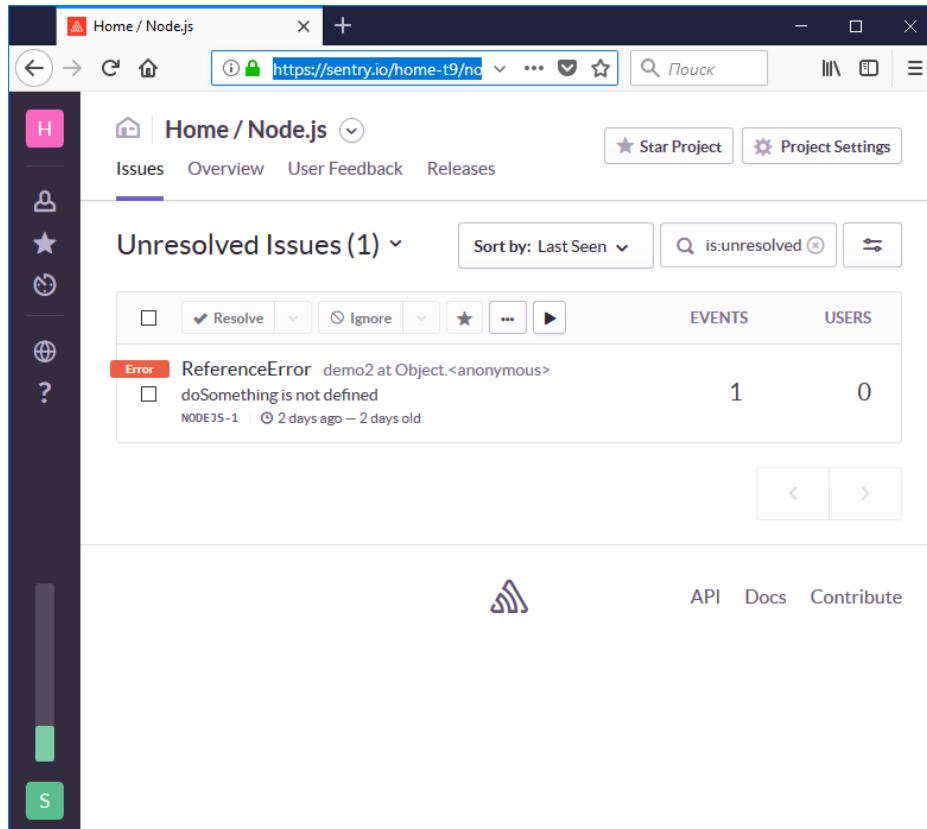
49

- Node.js
- `npm install raven --save`
- `yarn add -D raven`
- `let Raven = require('raven');`
- `Raven.config('your dsn').install();`
- `try {`
 - `doSomething(a[0]);`
- `} catch (e) {`
 - `Raven.captureException(e);`
- `}`

- JavaScript

```
<script  
src="https://cdn.ravenjs.co  
m/<VERSION>/raven.min.js"  
>  
</script>
```

```
<script>  
    Raven.config('your  
    dsn').install()  
    try {  
        doSomething(a[0]);  
    } catch (e) {  
        Raven.captureException(e);  
    }  
</script>
```



Home / Node.js

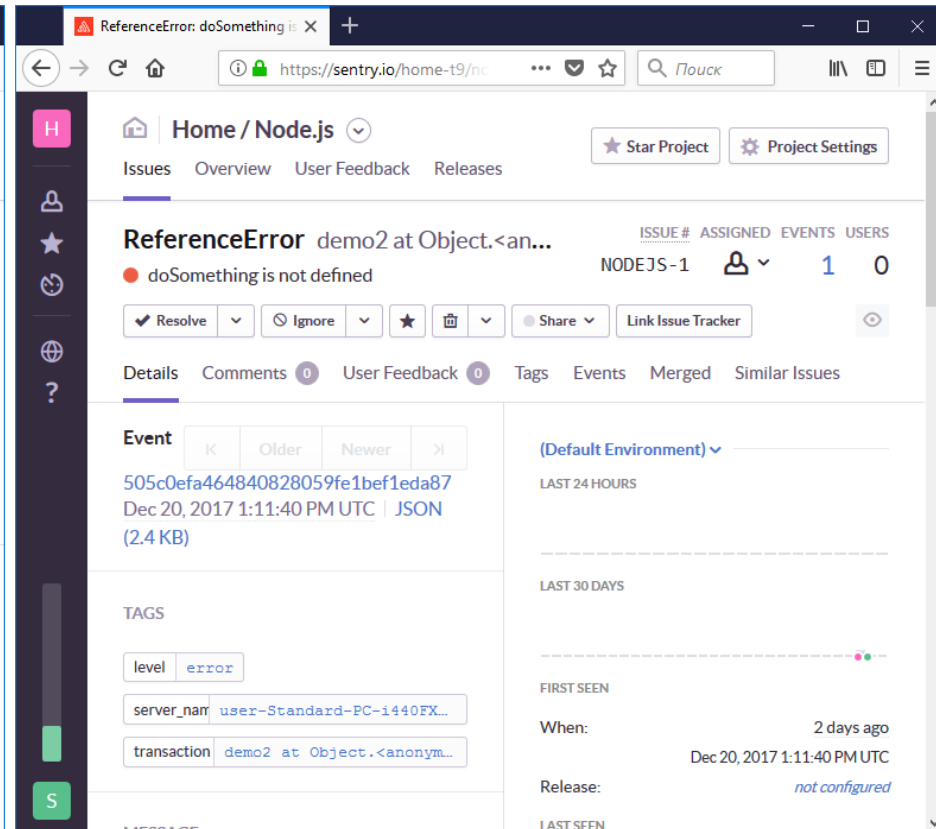
Issues Overview User Feedback Releases

Star Project Project Settings

Unresolved Issues (1) Sort by: Last Seen is:unresolved

	EVENTS	USERS
Error ReferenceError demo2 at Object.<anonymous> doSomething is not defined NODEJS-1 2 days ago — 2 days old	1	0

API Docs Contribute



ReferenceError: doSomething is not defined

Home / Node.js

Issues Overview User Feedback Releases

Star Project Project Settings

ReferenceError demo2 at Object.<anonymous> ISSUE # ASSIGNED EVENTS USERS
doSomething is not defined NODEJS-1 1 0

Resolve Ignore Share Link Issue Tracker

Details Comments (0) User Feedback (0) Tags Events Merged Similar Issues

Event 505c0efa464840828059fe1bef1eda87
Dec 20, 2017 1:11:40 PM UTC | JSON (2.4 KB)

TAGS

level error
server_name user-Standard-PC-i440FX...
transaction demo2 at Object.<anonym...

(Default Environment) LAST 24 HOURS
LAST 30 DAYS
FIRST SEEN
When: 2 days ago
Dec 20, 2017 1:11:40 PM UTC
Release: not configured
LAST SEEN

GitHub

Heroku

Slack

SessionStack

Datadog

Trello

Bitbucket

Jira

HipChat

PagerDuty

Asana

Twilio

GitLab

Yubico

Segment

OneLogin

Auth0

Okta

Hosted
Graphite

GCP

Pivotal
Tracker

Grove

OpsGenie

Campfire

Flowdock

Pushover

Lighthouse

Phabricator

Redmine

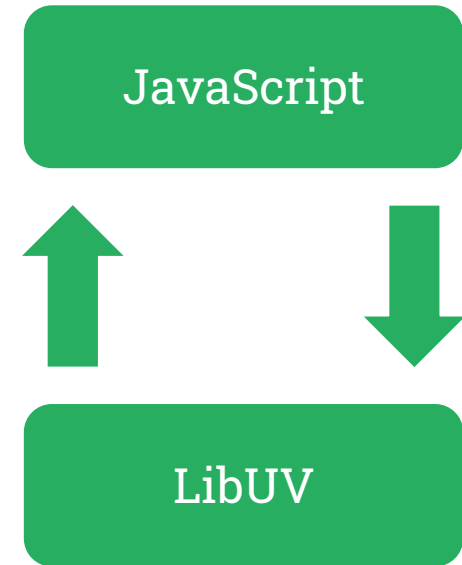
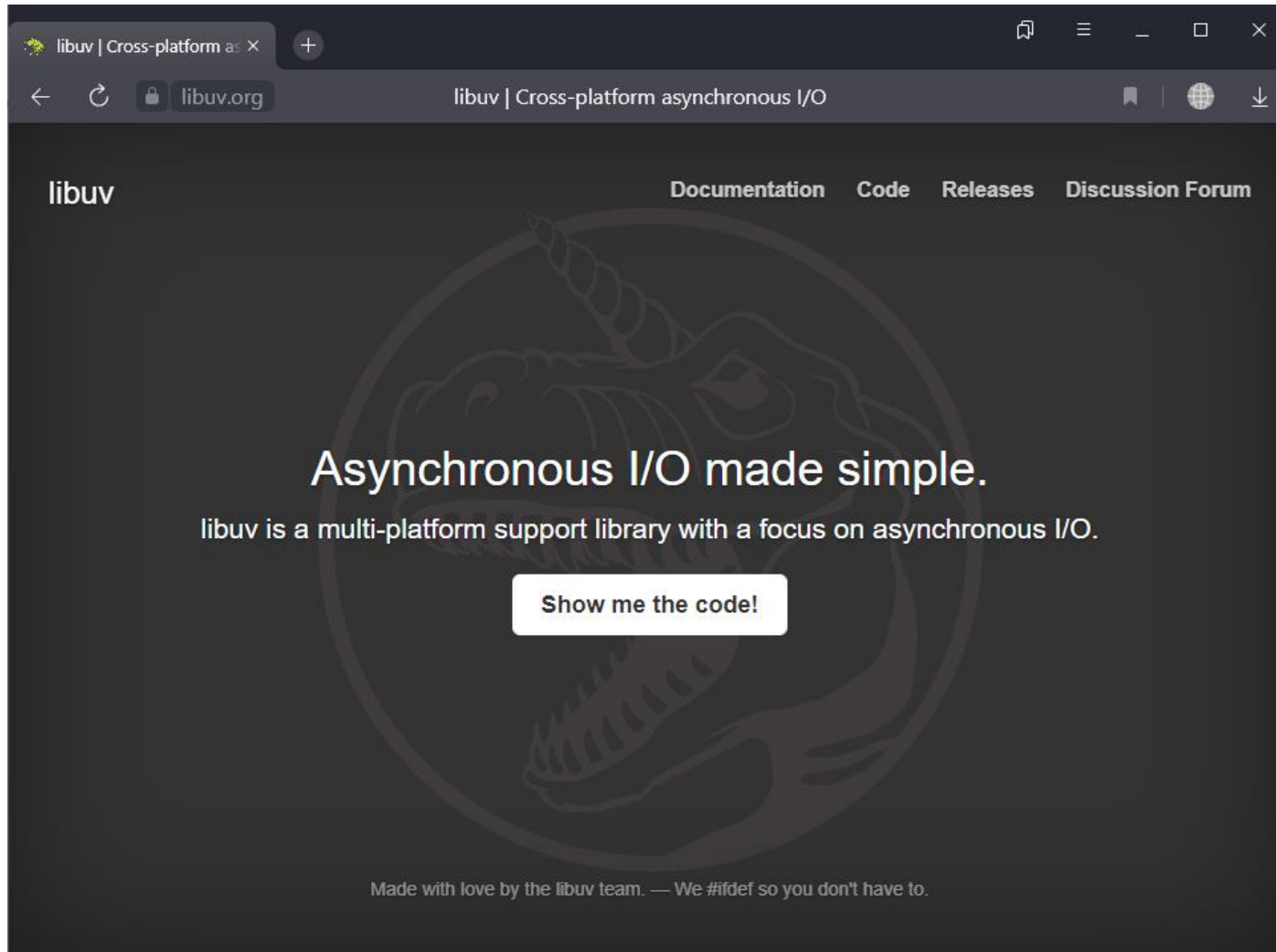
Teamwork

Taiga

Sprint.ly

YouTrack

VictorOps

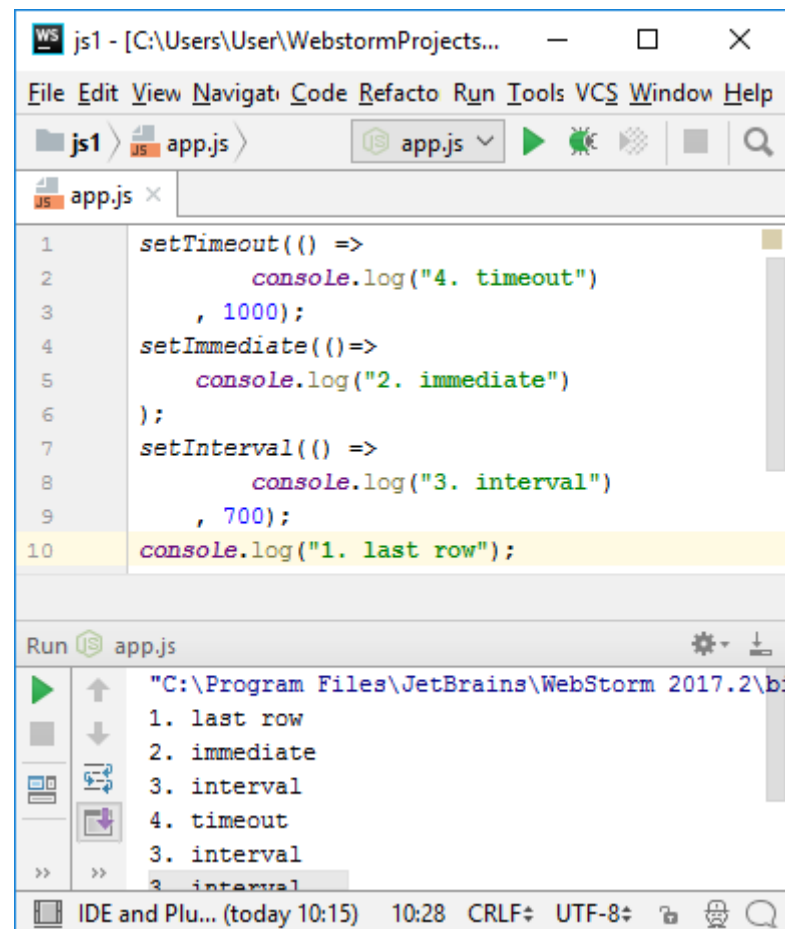


<https://libuv.org/>

Вызов с задержкой

53

```
setTimeout(() =>
    console.log("4. timeout")
    , 1000);
setImmediate(()=>
    console.log("2. immediate")
);
setInterval(() =>
    console.log("3. interval")
    , 700);
console.log("1. last row");
```



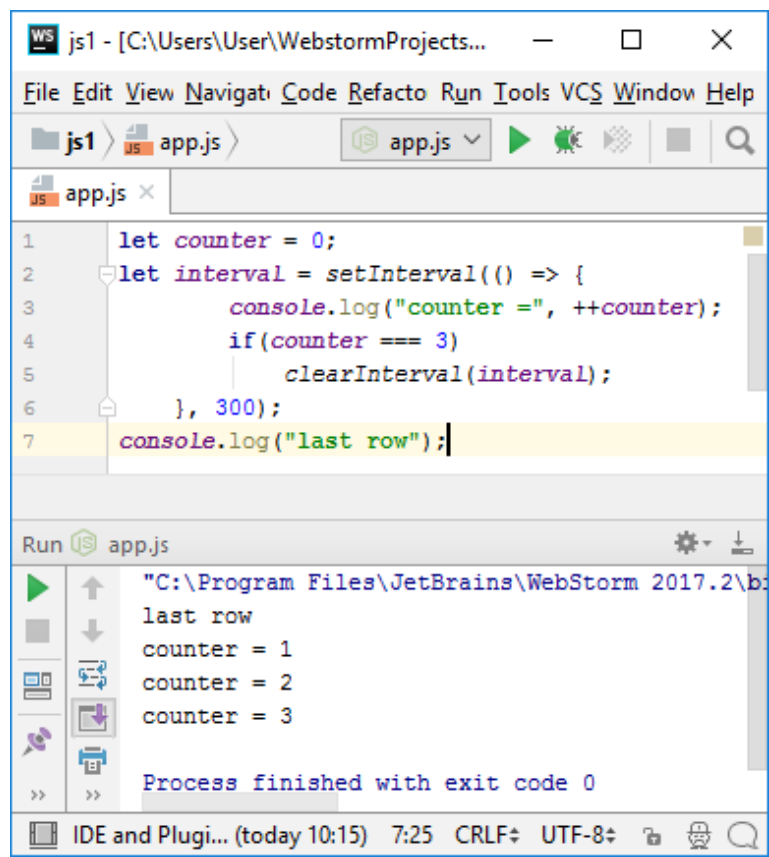
Периодические задачи

54

```
let counter = 0;
let interval = setInterval(() => {
    console.log("counter =", ++counter);
    if(counter === 3)
        clearInterval(interval);
}, 300);
console.log("last row");
```

Альтернатива:

```
let counter = 0;
setTimeout(work, 300);
function work() {
    console.log("counter =", ++counter);
    if(counter !== 3)
        setTimeout(work, 300);
}
console.log("last row");
```

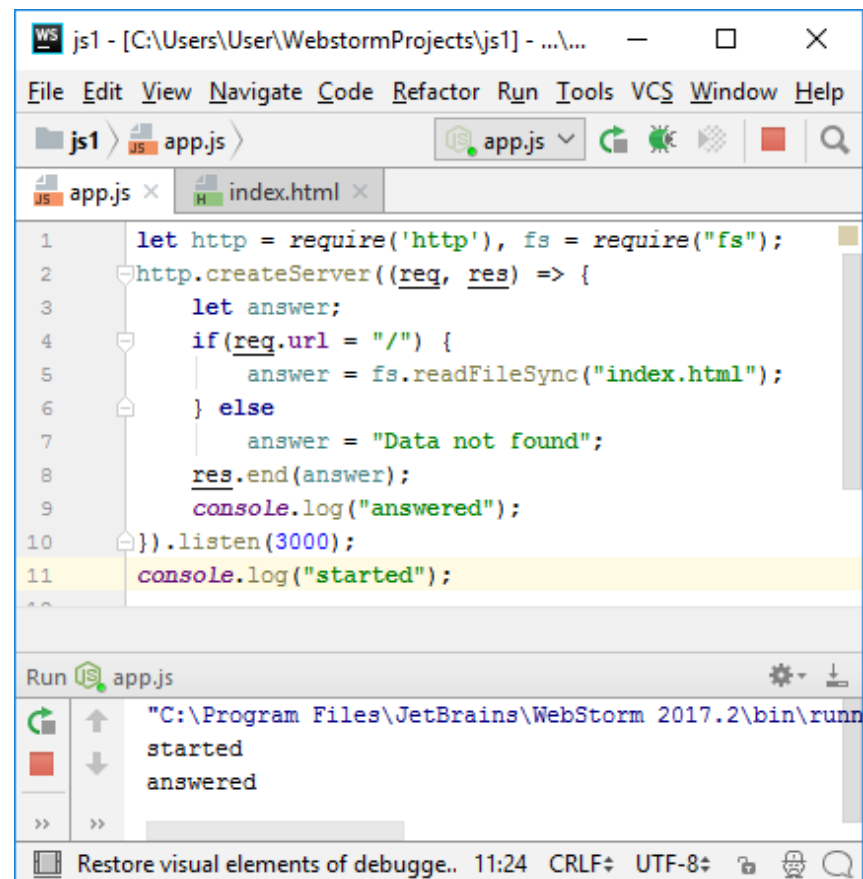


Синхронное чтение файла

55

```
let http = require('http'), fs = require("fs");
http.createServer((req, res) => {
  let answer;
  if(req.url = "/") {
    answer = fs.readFileSync("index.html");
  } else
    answer = "Data not found";
  res.end(answer);
  console.log("answered");
}).listen(3000);
console.log("started");
```

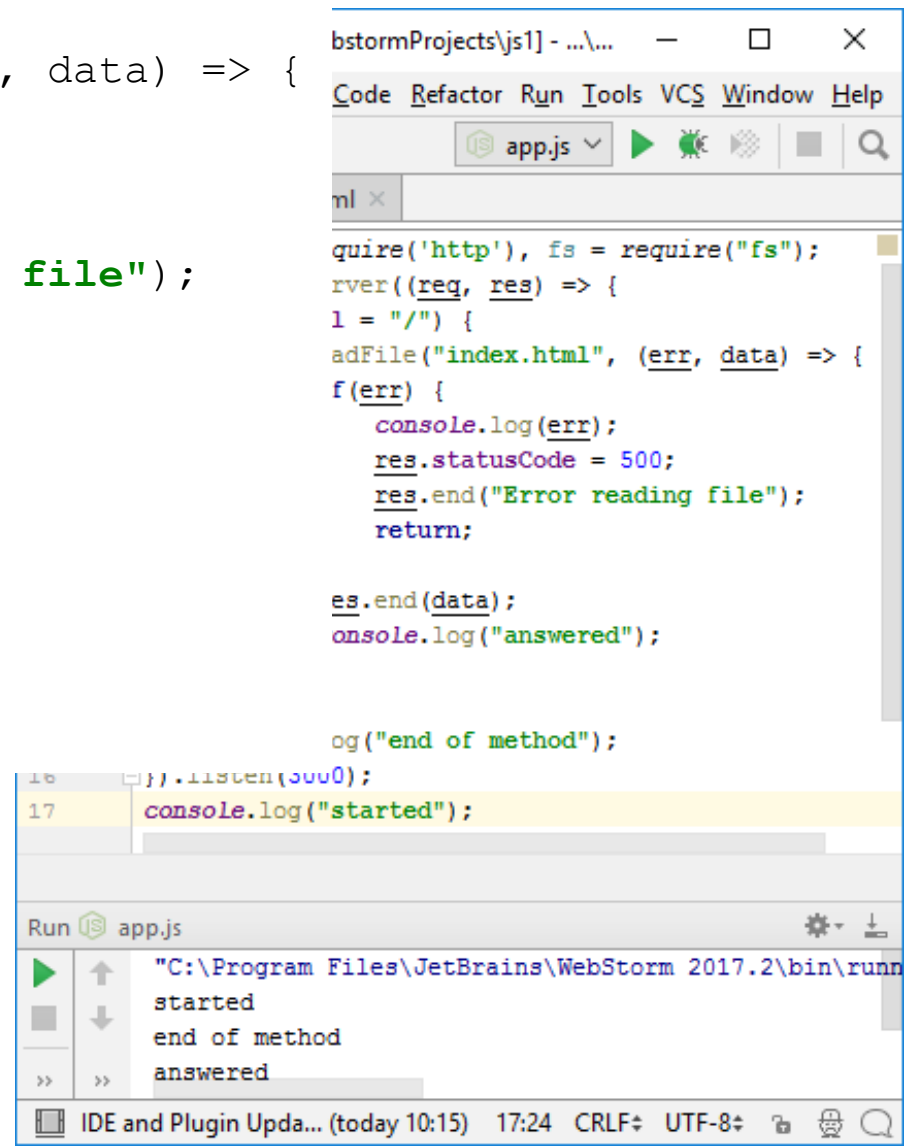
Добавить try-catch



Асинхронное чтение файла

56

```
let http = require('http'), fs = require("fs");
http.createServer((req, res) => {
  if(req.url === "/") {
    fs.readFile("index.html", (err, data) => {
      if(err) {
        console.log(err);
        res.statusCode = 500;
        res.end("Error reading file");
        return;
      }
      res.end(data);
      console.log("answered");
    });
  }
  console.log("end of method");
}).listen(3000);
console.log("started");
```



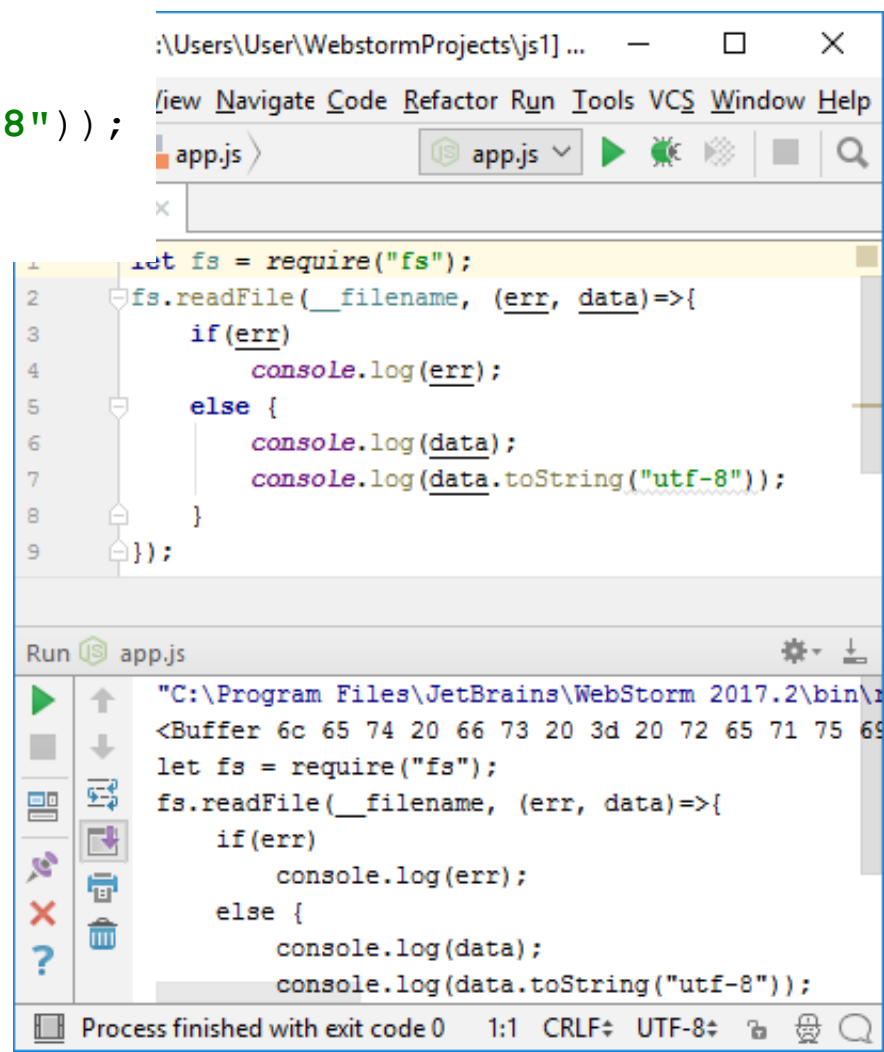
Чтение бинарных данных

57

```
let fs = require("fs");
fs.readFile(__filename, (err, data)=>{
  if(err)
    console.log(err);
  else {
    console.log(data);
    console.log(data.toString("utf-8"));
  }
});
```

Альтернатива:

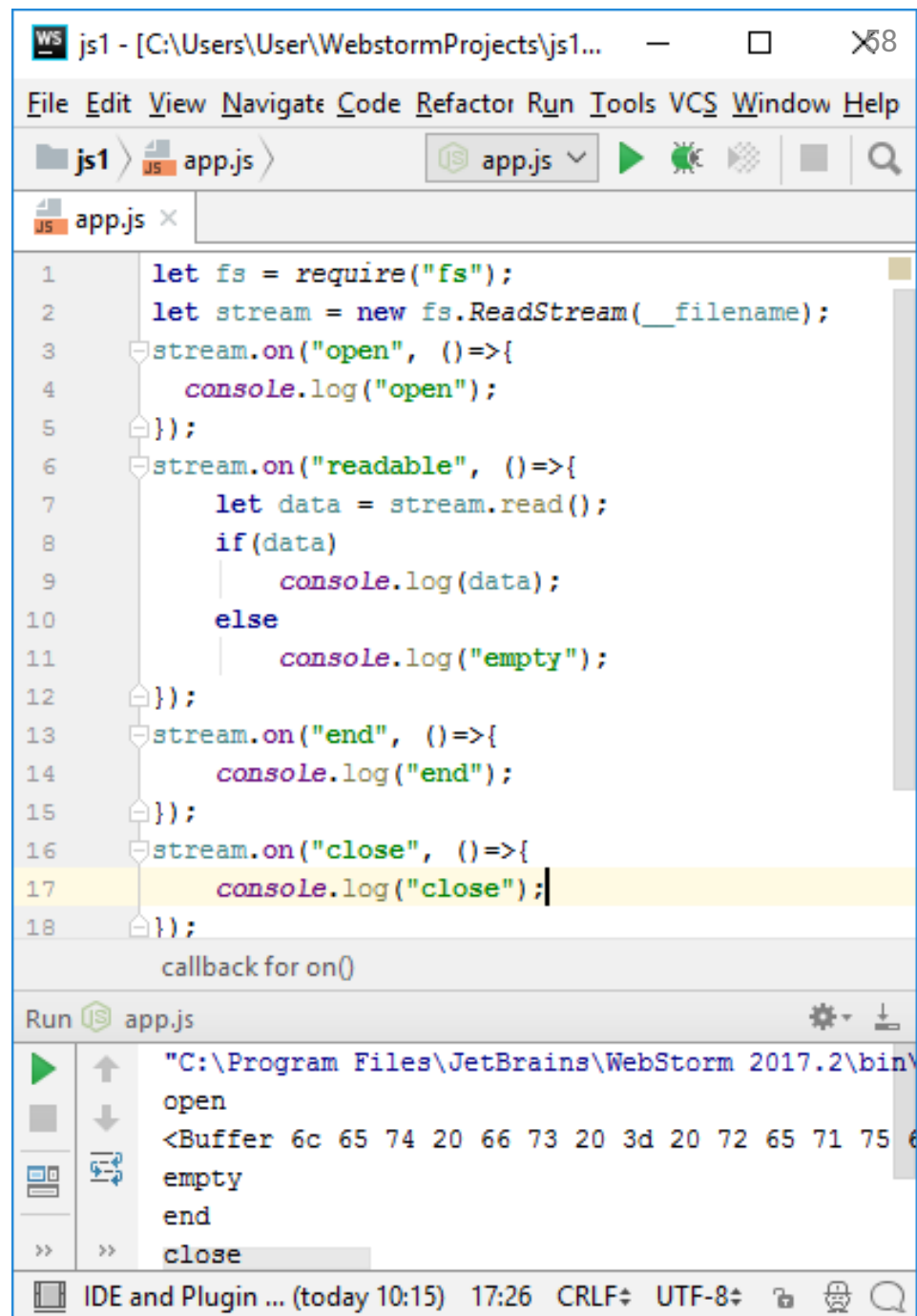
```
let fs = require("fs");
fs.readFile(__filename,
{encoding: "utf-8"},
(err, data)=>{
  if(err)
    console.log(err);
  else
    console.log(data);
});
```



Работа с потоком

```
let fs = require("fs");
let stream =
  new fs.ReadStream(__filename);
stream.on("open", ()=>{
  console.log("open");
});
stream.on("readable", ()=>{
  let data = stream.read();
  if(data)
    console.log(data);
  else
    console.log("empty");
});
stream.on("end", ()=>{
  console.log("end");
});
stream.on("close", ()=>{
  console.log("close");
});
```

Буфер 64k

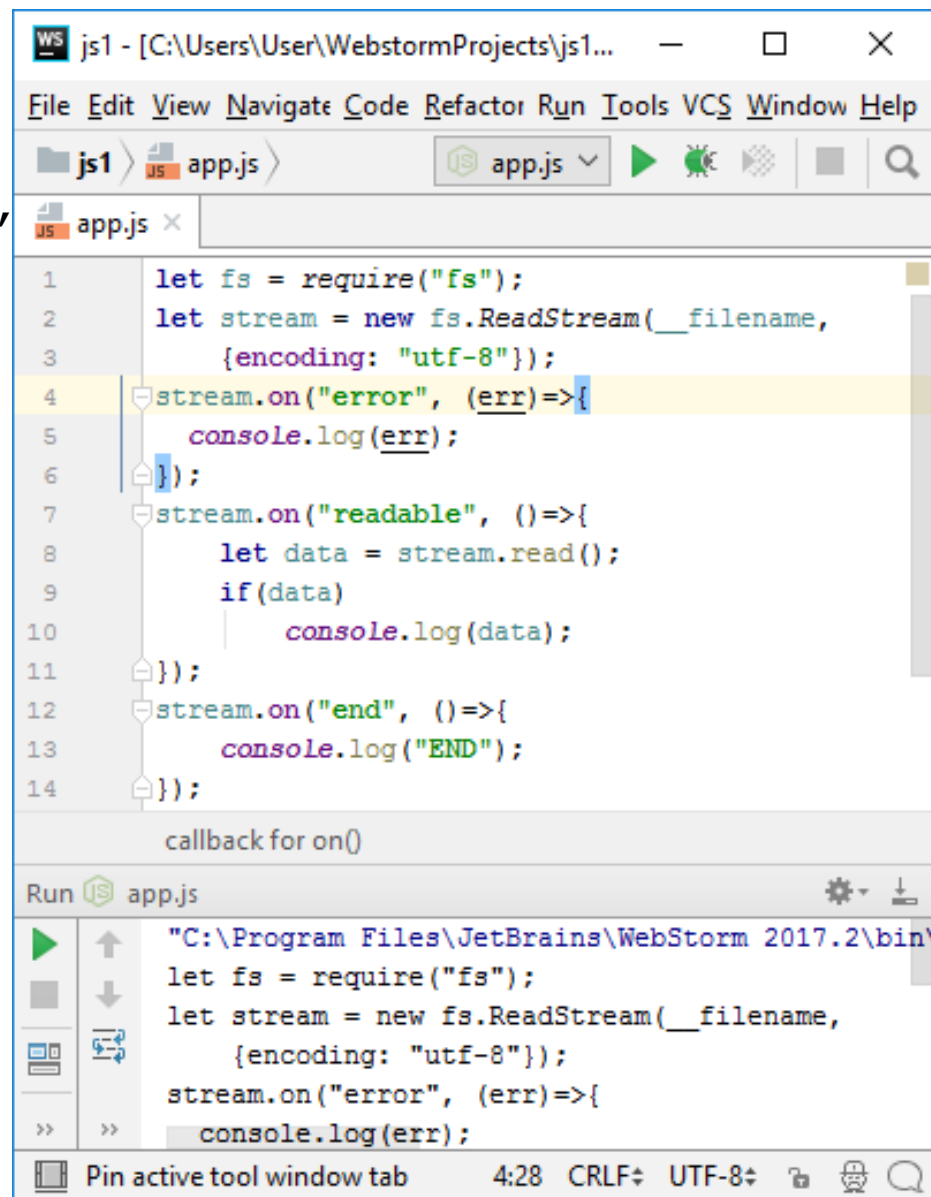


```
js1 - [C:\Users\User\WebstormProjects\js1...
File Edit View Navigate Code Refactor Run Tools VCS Window Help
js1 app.js app.js
app.js x
1 let fs = require("fs");
2 let stream = new fs.ReadStream(__filename);
3 stream.on("open", ()=>{
4   console.log("open");
5 });
6 stream.on("readable", ()=>{
7   let data = stream.read();
8   if(data)
9     console.log(data);
10  else
11    console.log("empty");
12 });
13 stream.on("end", ()=>{
14   console.log("end");
15 });
16 stream.on("close", ()=>{
17   console.log("close");
18 });
callback for on()
Run app.js
" C:\Program Files\JetBrains\WebStorm 2017.2\bin\
open
<Buffer 6c 65 74 20 66 73 20 3d 20 72 65 71 75 6
empty
end
close
IDE and Plugin ... (today 10:15) 17:26 CRLF UTF-8
```

Обработка ошибок при работе с потоками

```
let fs = require("fs");
let stream =
    new fs.ReadStream(__filename,
        {encoding: "utf-8"});
stream.on("error", (err)=>{
    console.log(err);
});
stream.on("readable", ()=>{
    let data = stream.read();
    if(data)
        console.log(data);
});
stream.on("end", ()=>{
    console.log("END");
});
```

Универсализм для
всех потоков



Чтение и отправка в поток по частям

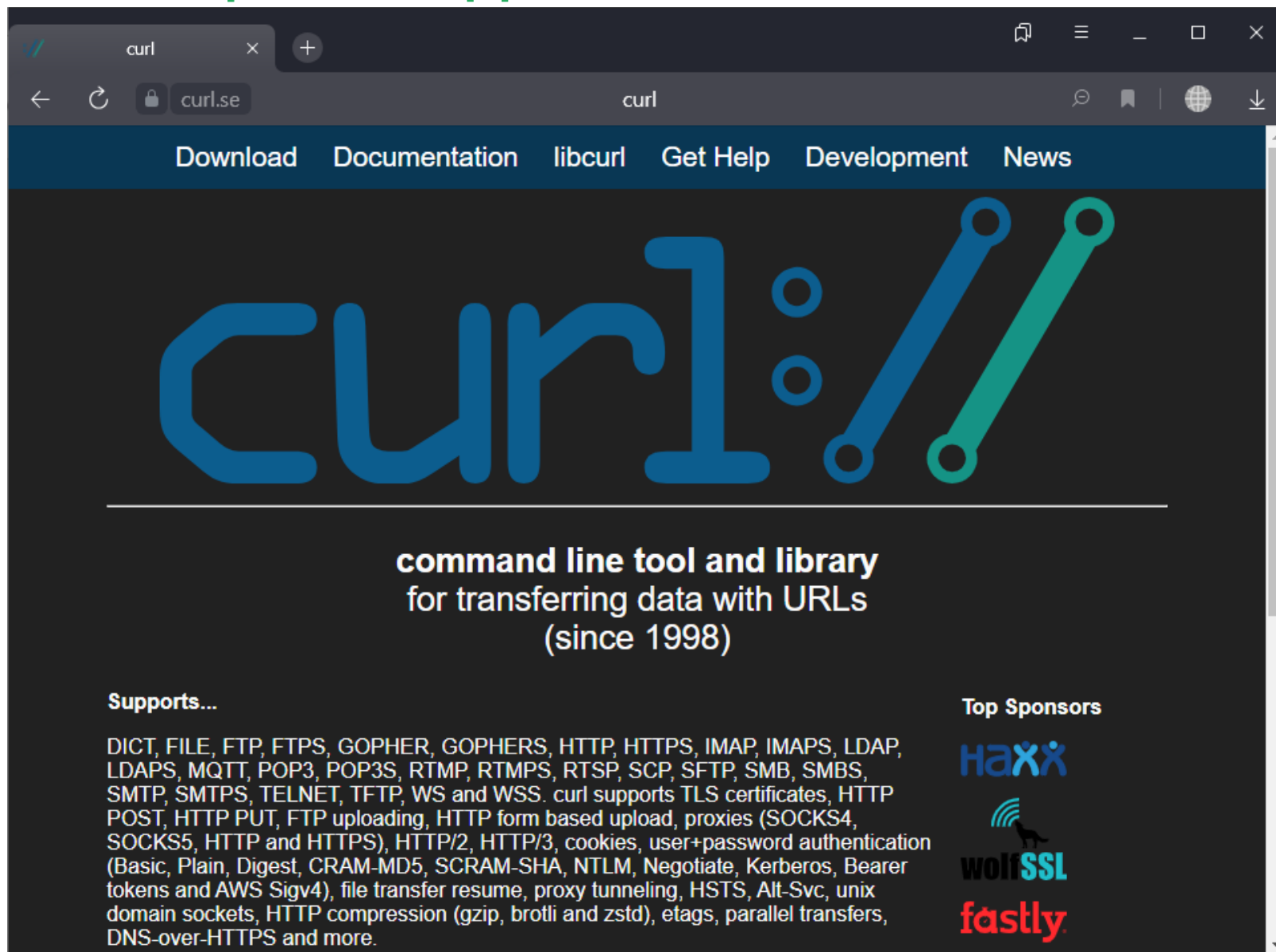
60

```
let http = require("http");
let fs = require("fs");
new http.Server((req, res)=>{
  if(req.url=="/")
    sendFile(new fs.ReadStream("index.html"), res);
}).listen(3000);
function sendFile(file, res){
  file.on("readable", write);
  function write() {
    let data = file.read();
    if(data && !res.write(data)) {
      file.removeListener("readable", write);
      res.once("drain", ()=>{
        file.on("readable", write);
        write();
      });
    }
  }
  file.on("end", () => res.end());
}
```

file.pipe(res);

Имитация медленного клиента

61



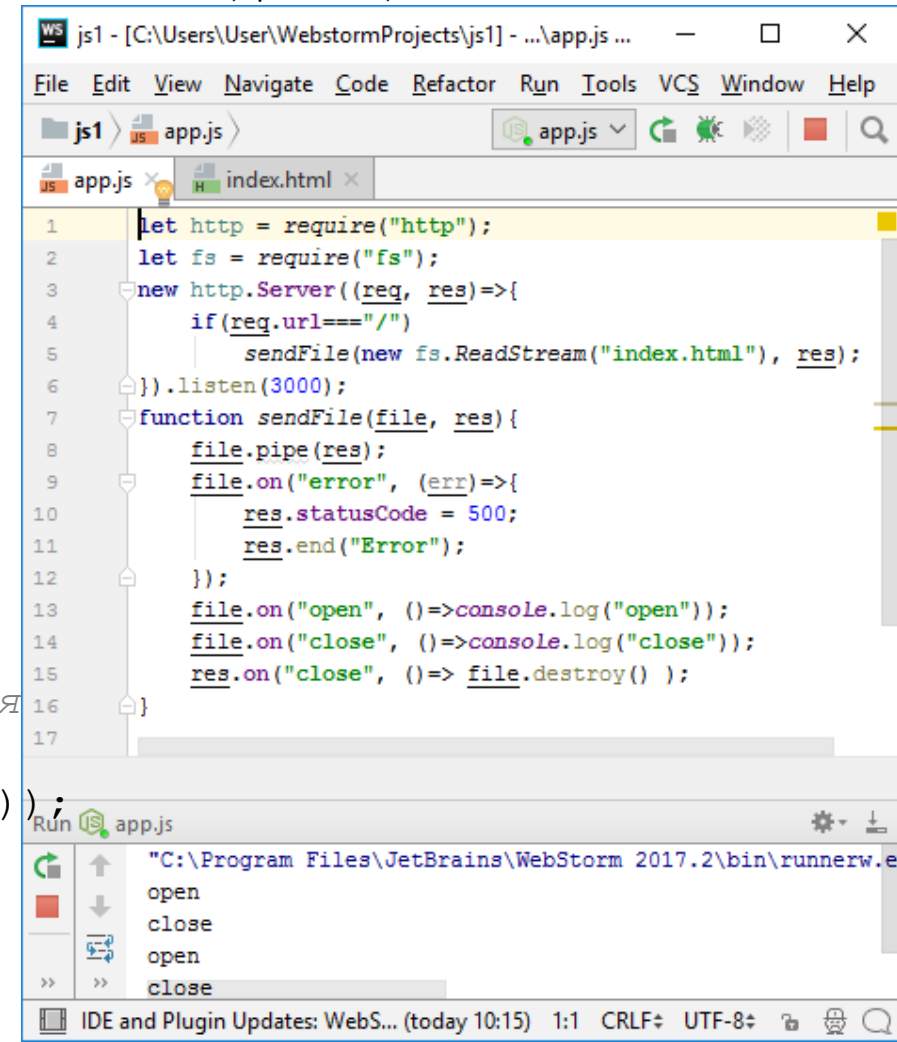
<https://curl.se/>

```
curl --limit 1k http://localhost:3000
```

Передача данных медленному клиенту

62

```
let http = require("http");
let fs = require("fs");
new http.Server((req, res)=>{
  if(req.url==="/")
    sendFile(new fs.ReadStream("index.html"), res);
}).listen(3000);
function sendFile(file, res){
  file.pipe(res);
  file.on("error", (err)=>{
    res.statusCode = 500;
    res.end("Error");
  });
  file.on("open", ()=>
    console.log("open")
  );
  file.on("close", ()=>
    console.log("close")
  );
  // Заккрытие соединения до окончания
  // передачи файла
  res.on("close", ()=> file.destroy());
}
```

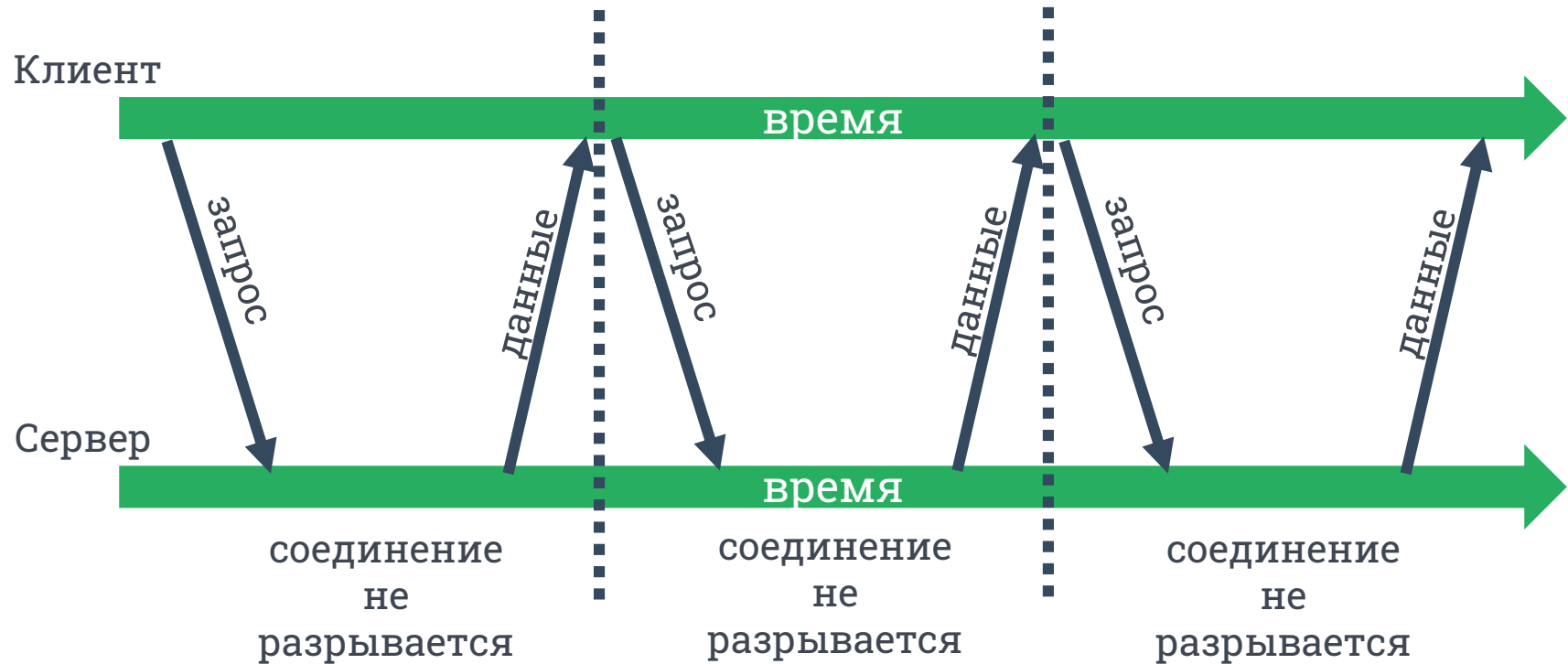


The screenshot shows an IDE window titled 'js1 - [C:\Users\User\WebstormProjects\js1] - ...\app.js ...'. The code editor displays the same JavaScript code as the previous block. Below the editor, the 'Run' console is visible, showing the output of the console.log statements: 'open' and 'close'.

```
js1 - [C:\Users\User\WebstormProjects\js1] - ...\app.js ...
File Edit View Navigate Code Refactor Run Tools VCS Window Help
js1 > app.js >
app.js x index.html x
1 let http = require("http");
2 let fs = require("fs");
3 new http.Server((req, res)=>{
4   if(req.url==="/")
5     sendFile(new fs.ReadStream("index.html"), res);
6   }).listen(3000);
7   function sendFile(file, res){
8     file.pipe(res);
9     file.on("error", (err)=>{
10       res.statusCode = 500;
11       res.end("Error");
12     });
13     file.on("open", ()=>console.log("open"));
14     file.on("close", ()=>console.log("close"));
15     res.on("close", ()=> file.destroy() );
16   }
17
Run app.js
"C:\Program Files\JetBrains\WebStorm 2017.2\bin\runnerw.e
open
close
open
close
```

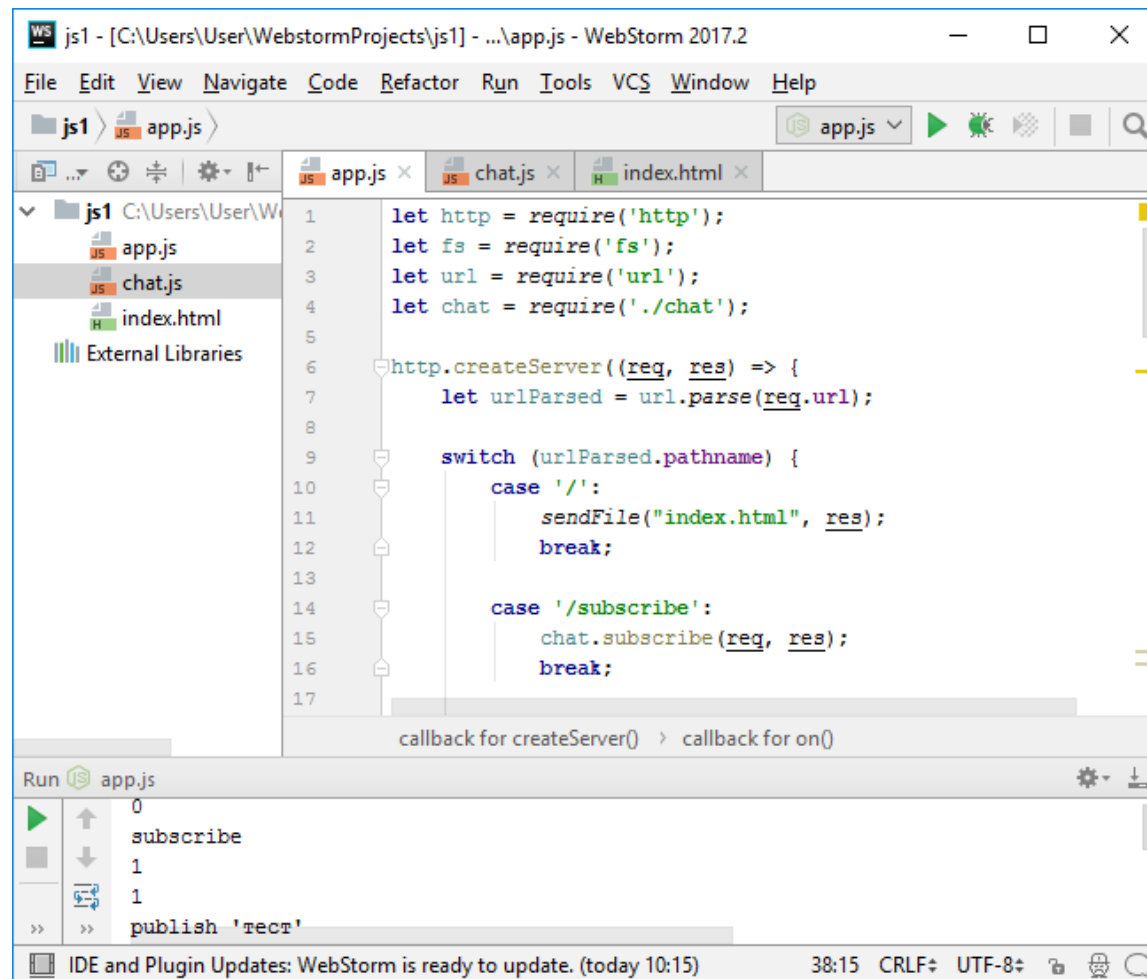
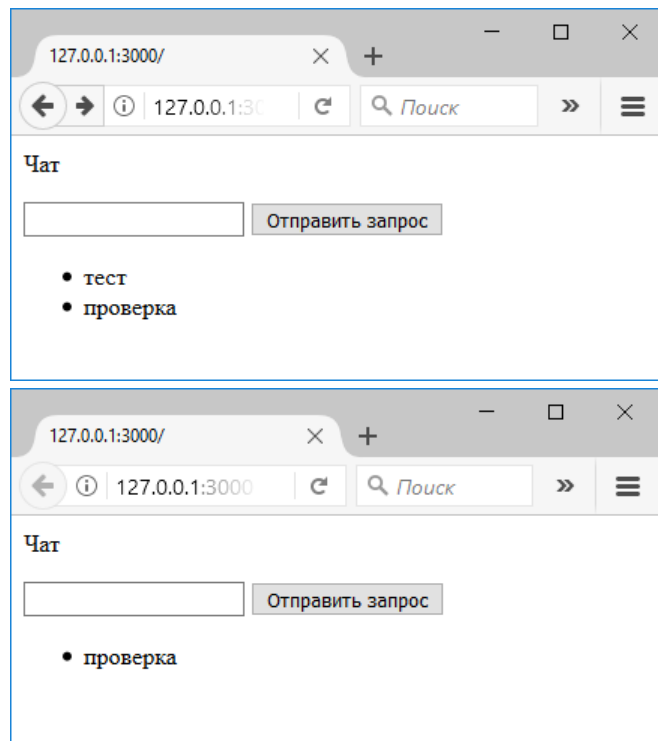
Long-polling («длинные запросы»)

63



Чат через «длинные запросы»

64

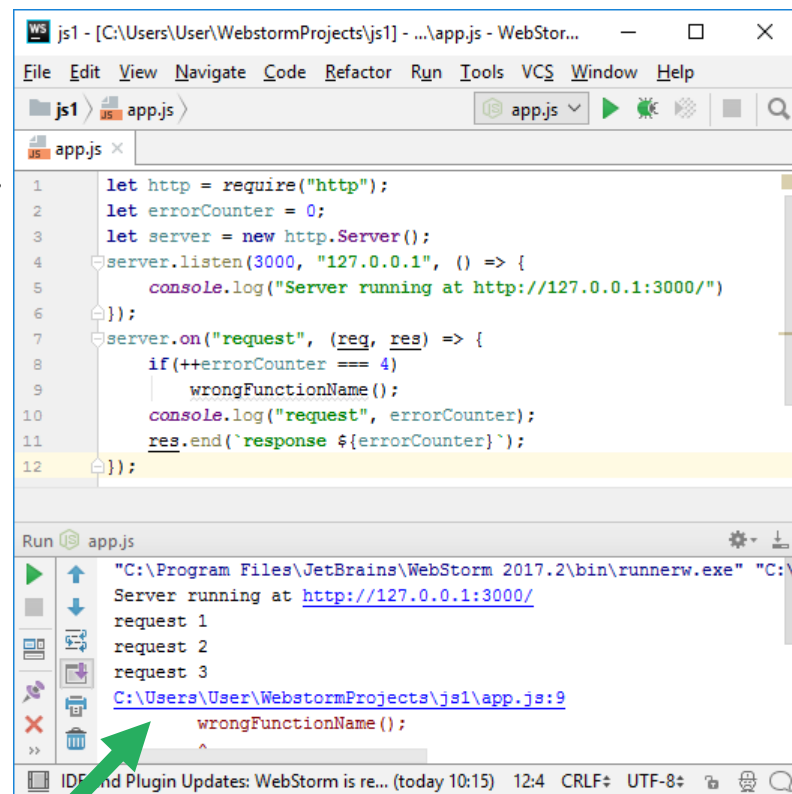
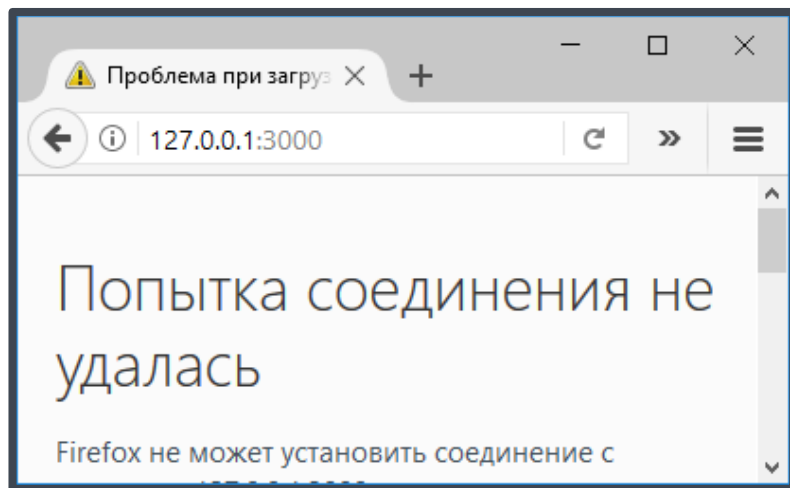


№27 «Чат через long-polling, чтение POST»

<http://learn.javascript.ru/screencast/nodejs>

Домены (1)

```
let http = require("http");
let errorCounter = 0;
let server = new http.Server();
server.listen(3000, "127.0.0.1", () => {
    console.log("Server running at http://127.0.0.1:3000/")
});
server.on("request", (req, res) => {
    if(++errorCounter === 4)
        wrongFunctionName();
    console.log("request", errorCounter);
    res.end(`response ${errorCounter}`);
});
```



<https://nodejs.org/en/docs/guides/domain-postmortem/>
<https://nodejs.org/dist/latest-v18.x/docs/api/domain.html>

Домены (2)

66

```
let domain = require("domain");  
let server = domain.create();
```

```
server.on("error", (err) => {  
    console.log("Домен перехватил ошибку %s", err);  
});
```

```
server.run(()=>{  
    wrongFunctionName();  
});
```

benchmark

22,000 requests/second



17,000 requests/second

Domain | Node.js v18.17.x

nodejs.org Domain | Node.js v18.17.1 Document

Node.js v18.17.1 document

Table of contents | Index | Other versions | Options

Table of contents

- Domain **deprecated**
 - Warning: Don't ignore errors!
 - Additions to Error objects

js1 - [C:\Users\User\WebstormProjects\js1] - ...app.js...

File Edit View Navigate Code Refactor Run Tools VCS Window Help

js1 app.js

app.js

```
1 let domain = require("domain");  
2 let server = domain.create();  
3  
4 server.on("error", (err) => {  
5     console.log("Домен перехватил ошибку %s", err);  
6 });  
7  
8 server.run(()=>{  
9     wrongFunctionName();  
10 });
```

Run app.js

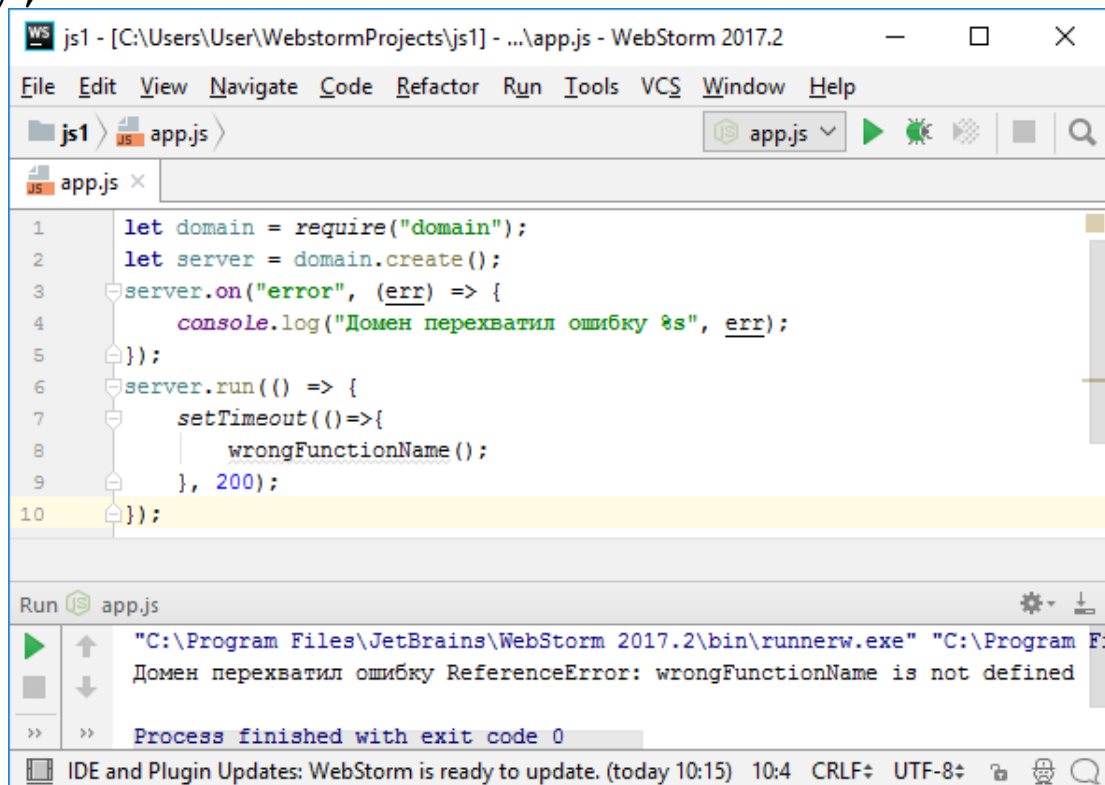
"C:\Program Files\JetBrains\WebStorm 2017.2\bin\runnerw..."
Домен перехватил ошибку ReferenceError: wrongFunctionName
Process finished with exit code 0

IDE and Plugin Updates: Web... (today 10:15) 1:1 CRLF UTF-8

Домены (3)

67

```
let domain = require("domain");
let server = domain.create();
server.on("error", (err) => {
    console.log("Домен перехватил ошибку %s", err);
});
server.run(() => {
    setTimeout(()=>{
        wrongFunctionName();
    }, 200);
});
```



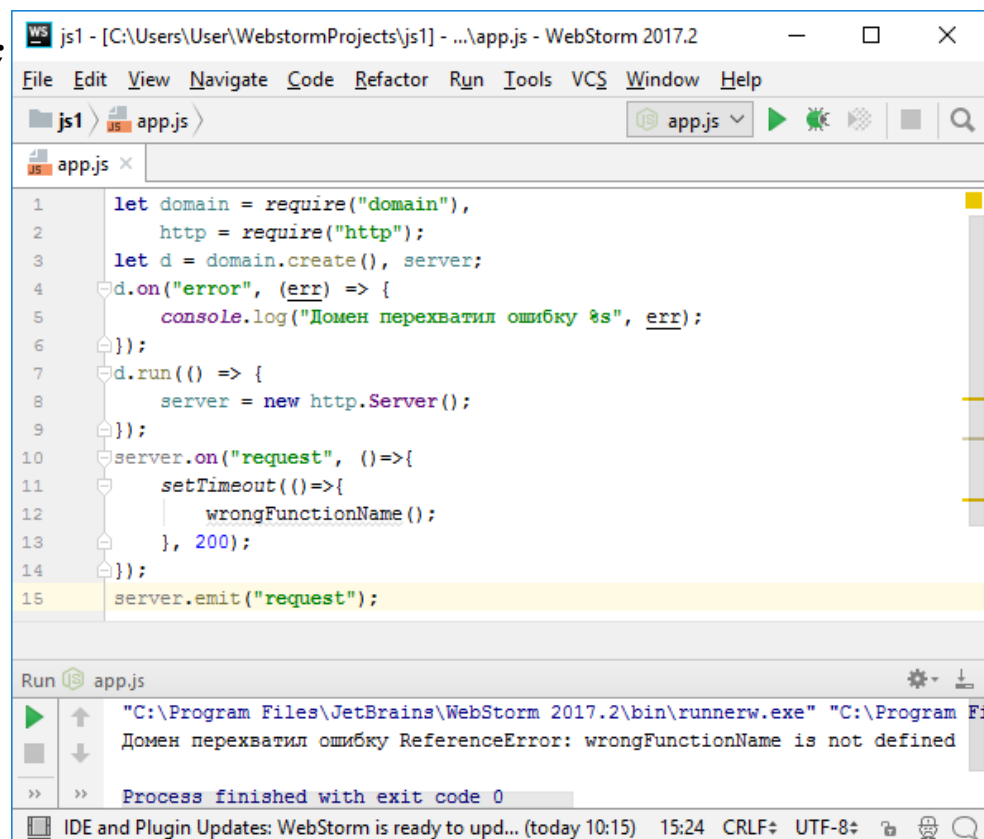
Домены (4)

68

```
let domain = require("domain"),
    http = require("http");

let d = domain.create(), server;
d.on("error", (err) => {
    console.log("Домен перехватил ошибку %s", err);
});
d.run(() => {
    server = new http.Server();
});
server.on("request", ()=>{
    setTimeout(()=>{
        wrongFunctionName();
    }, 200);
});
server.emit("request");
```

Попробуйте создать
экземпляр Server вне
run()

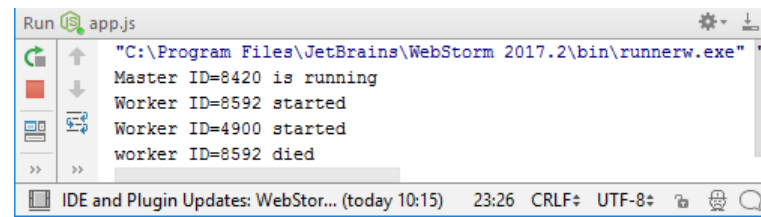
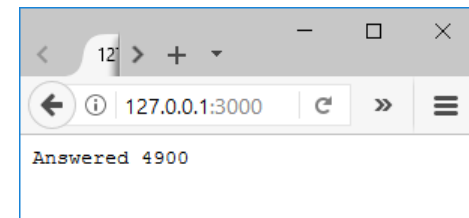


ИСПОЛЬЗОВАНИЕ НЕСКОЛЬКИХ ПРОЦЕССОРОВ

```
const cluster = require('cluster');
const http = require('http');
const numCPUs = require('os').cpus().length;
const list = [];

if (cluster.isMaster) {
  console.log(`Master ID=${process.pid} is running`);
  // Fork workers
  for (let i = 0; i < numCPUs; i++)
    list.push(cluster.fork());
  cluster.on('exit', (worker, code, signal) => {
    console.log(`worker ID=${worker.process.pid} died`);
  });
} else {
  // Worker могут разделять любые TCP-соединения
  http.createServer((req, res) => {
    res.writeHead(200);
    res.end(`Answered ${process.pid}`);
  }).listen(3000);
  console.log(`Worker ID=${process.pid} started`);
}

if (list[0])
  list[0].disconnect();
```

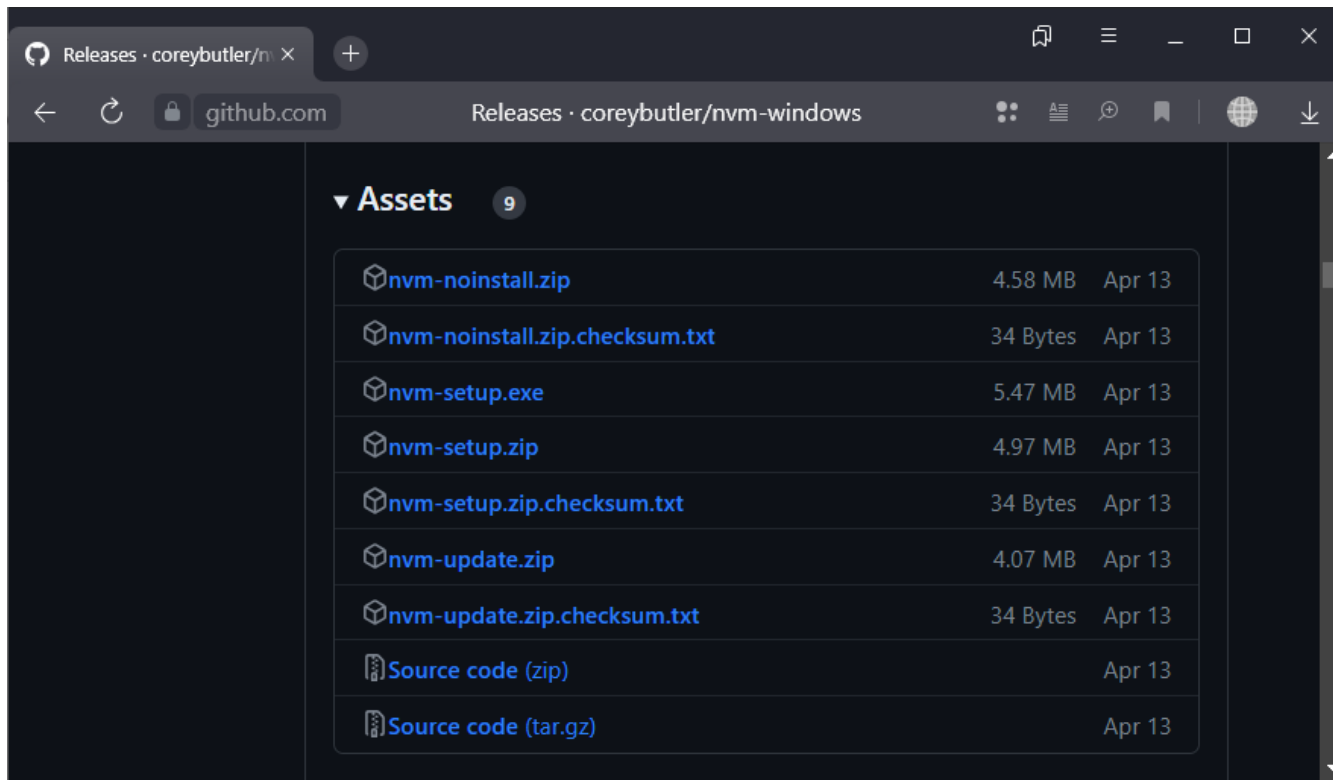


Node Version Manager

Поддержка разных версий Node.js

Windows

<https://github.com/coreybutler/nvm-windows/releases>



Linux

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.37.2/install.sh | bash
```

src

<https://github.com/nvm-sh/nvm>

nvm — ИСПОЛЬЗОВАНИЕ

71

```
Командная строка
C:\Users\sbelyaev\AppData\Roaming\nvm>nvm

Running version 1.1.7.

Usage:

nvm arch                : Show if node is running in 32 or 64 bit mode.
nvm install <version> [arch] : The version can be a node.js version or "latest" for the latest stable version.
                                Optionally specify whether to install the 32 or 64 bit version (defaults to system arch).
                                Set [arch] to "all" to install 32 AND 64 bit versions.
                                Add --insecure to the end of this command to bypass SSL validation of the remote download server.
nvm list [available]      : List the node.js installations. Type "available" at the end to see what can be installed. Aliased as ls.
nvm on                   : Enable node.js version management.
nvm off                  : Disable node.js version management.
nvm proxy [url]          : Set a proxy to use for downloads. Leave [url] blank to see the current proxy.
                                Set [url] to "none" to remove the proxy.
nvm node_mirror [url]    : Set the node mirror. Defaults to https://nodejs.org/dist/. Leave [url] blank to use default url.
nvm npm_mirror [url]     : Set the npm mirror. Defaults to https://github.com/npm/cli/archive/. Leave [url] blank to default url.
nvm uninstall <version>  : The version must be a specific version.
nvm use [version] [arch] : Switch to use the specified version. Optionally specify 32/64bit architecture.
                                nvm use <arch> will continue using the selected version, but switch to 32/64 bit mode.
nvm root [path]          : Set the directory where nvm should store different versions of node.js.
                                If <path> is not set, the current root will be displayed.
nvm version              : Displays the current running version of nvm for Windows. Aliased as v.
```

nvm install 14.17.6
nvm list
nvm use 14.17.6

```
Командная строка
C:\Users\sbelyaev\AppData\Roaming\nvm>node --version
v14.17.6

C:\Users\sbelyaev\AppData\Roaming\nvm>nvm list

* 14.17.6 (Currently using 64-bit executable)
  9.11.2

C:\Users\sbelyaev\AppData\Roaming\nvm>nvm use 9.11.2
Now using node v9.11.2 (64-bit)

C:\Users\sbelyaev\AppData\Roaming\nvm>node --version
v9.11.2
```

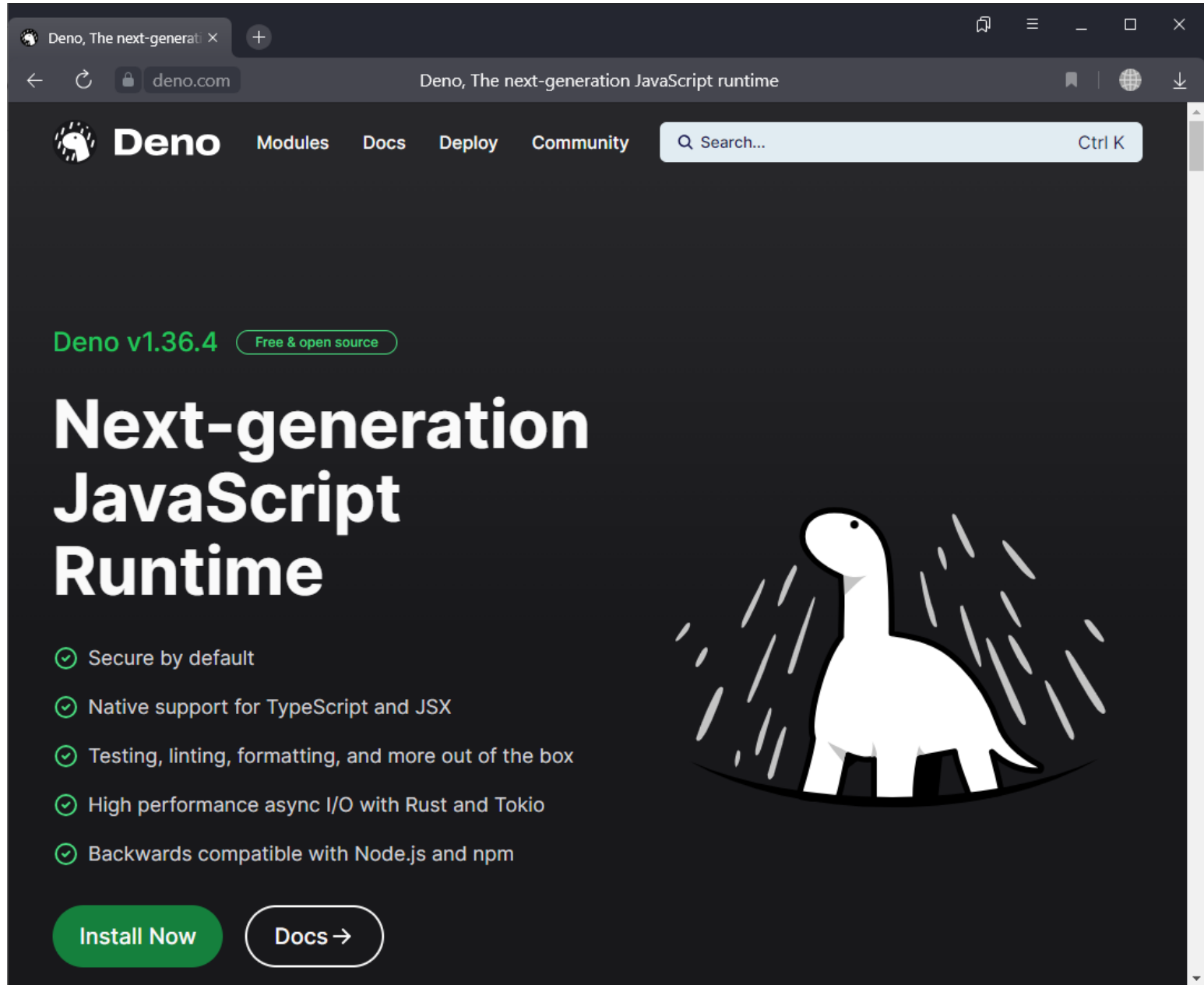
10 вещей, о которых я сожалею в Node.js

Ryan Dahl

- Отказались от Promise
 - Были удалены в феврале 2010
- Безопасность
 - Доступ к диску без ограничений
- Система сборки
 - Сложности с разработкой на C
- package.json
 - прт в стандартной поставке, централизованный, лишняя абстракция, лишняя информация
- node_modules
 - Сложный алгоритм, проблемы с доступом из браузера
- require без расширения
- Единый index.js

<https://deno.land/>

73



The image is a screenshot of a web browser displaying the Deno website. The browser's address bar shows 'deno.com' and the page title is 'Deno, The next-generation JavaScript runtime'. The website has a dark theme. At the top, there is a navigation bar with the Deno logo, links for 'Modules', 'Docs', 'Deploy', and 'Community', and a search bar with the placeholder text 'Search...' and a 'Ctrl K' shortcut. Below the navigation bar, the main content area features the text 'Deno v1.36.4' in green, followed by a green button that says 'Free & open source'. The main heading is 'Next-generation JavaScript Runtime' in large white text. To the right of the heading is a white illustration of a dinosaur standing in the rain. Below the heading, there is a list of features, each preceded by a green checkmark: 'Secure by default', 'Native support for TypeScript and JSX', 'Testing, linting, formatting, and more out of the box', 'High performance async I/O with Rust and Tokio', and 'Backwards compatible with Node.js and npm'. At the bottom, there are two buttons: a green 'Install Now' button and a white 'Docs →' button.

Deno, The next-generation JavaScript runtime

Deno

Modules Docs Deploy Community

Search... Ctrl K

Deno v1.36.4 Free & open source

Next-generation JavaScript Runtime

- ✓ Secure by default
- ✓ Native support for TypeScript and JSX
- ✓ Testing, linting, formatting, and more out of the box
- ✓ High performance async I/O with Rust and Tokio
- ✓ Backwards compatible with Node.js and npm

Install Now Docs →

Deno

<https://github.com/denoland/deno>

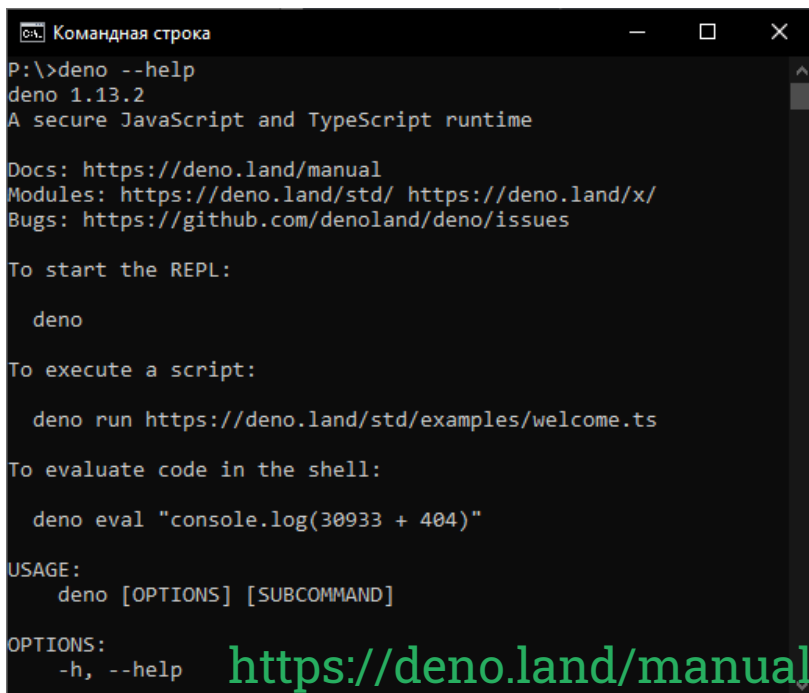
- Безопасность
 - По умолчанию без права работы с файловой системой
- Компилятор TypeScript
 - Поддержка TypeScript «из коробки»
- Упрощение системы модулей
 - Не совместимо с модулями Node
 - Импорты относительные или абсолютные URL
 - Всегда с расширением (js, ts)
- Другое
 - Совместим с браузерами (`window`, а не `global`)
 - Поддерживает await верхнего уровня

Shell (Mac, Linux):

- `curl -fsSL https://deno.land/x/install/install.sh | sh`

PowerShell (Windows):

- `irm https://deno.land/install.ps1 | iex`



```
Командная строка
P:\>deno --help
deno 1.13.2
A secure JavaScript and TypeScript runtime

Docs: https://deno.land/manual
Modules: https://deno.land/std/ https://deno.land/x/
Bugs: https://github.com/denoland/deno/issues

To start the REPL:

  deno

To execute a script:

  deno run https://deno.land/std/examples/welcome.ts

To evaluate code in the shell:

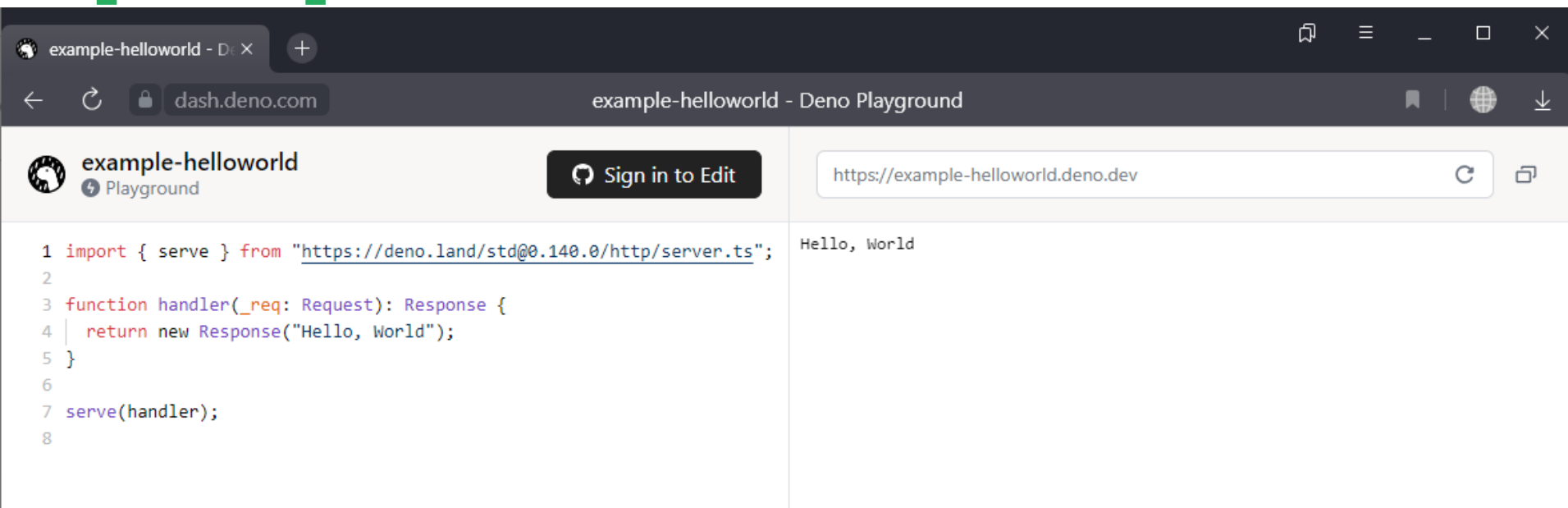
  deno eval "console.log(30933 + 404)"

USAGE:
  deno [OPTIONS] [SUBCOMMAND]

OPTIONS:
  -h, --help
```

Пример использования Deno

76



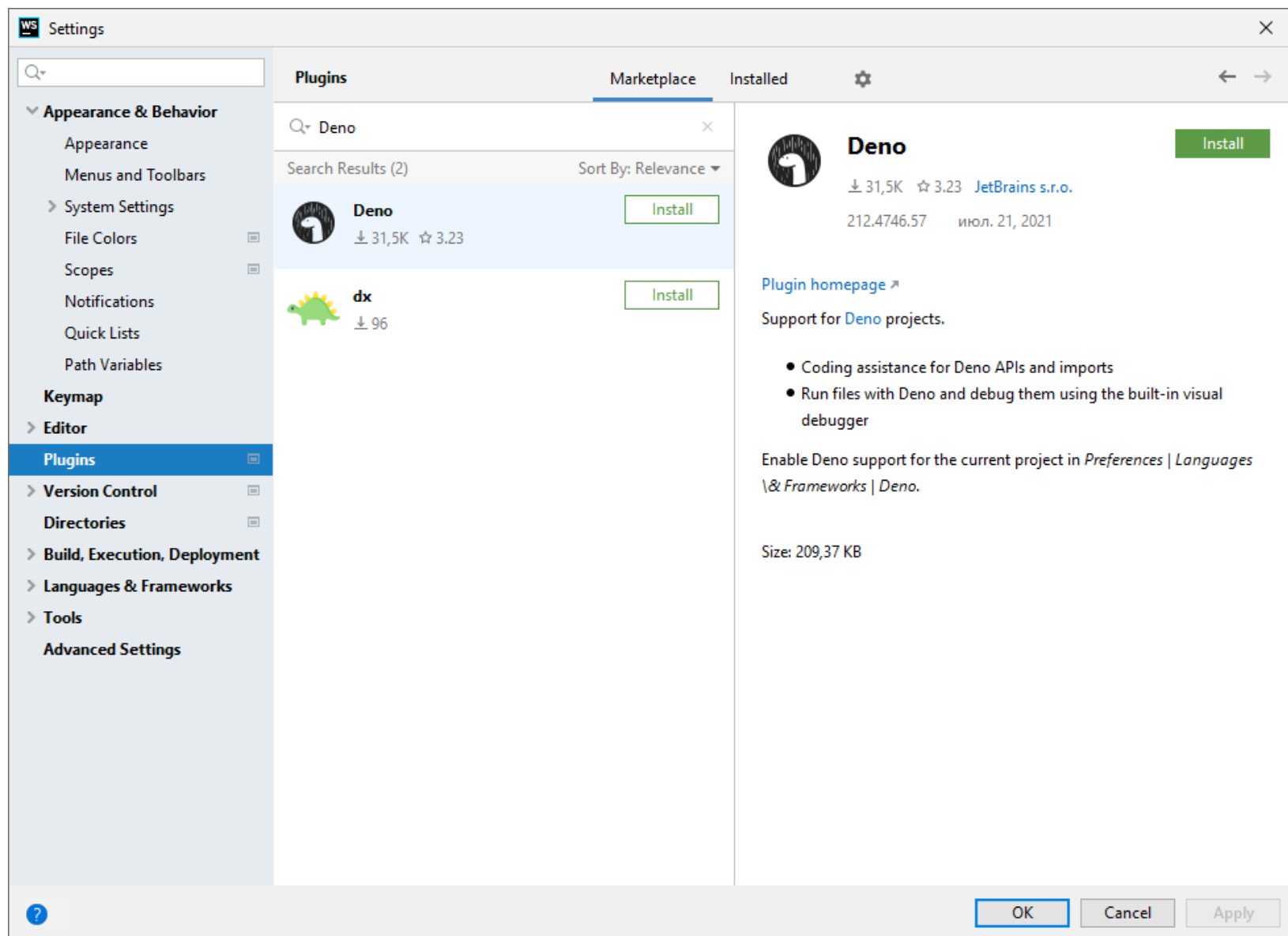
<https://dash.deno.com/playground/example-helloworld>



<https://deno.com/deploy/docs/examples>

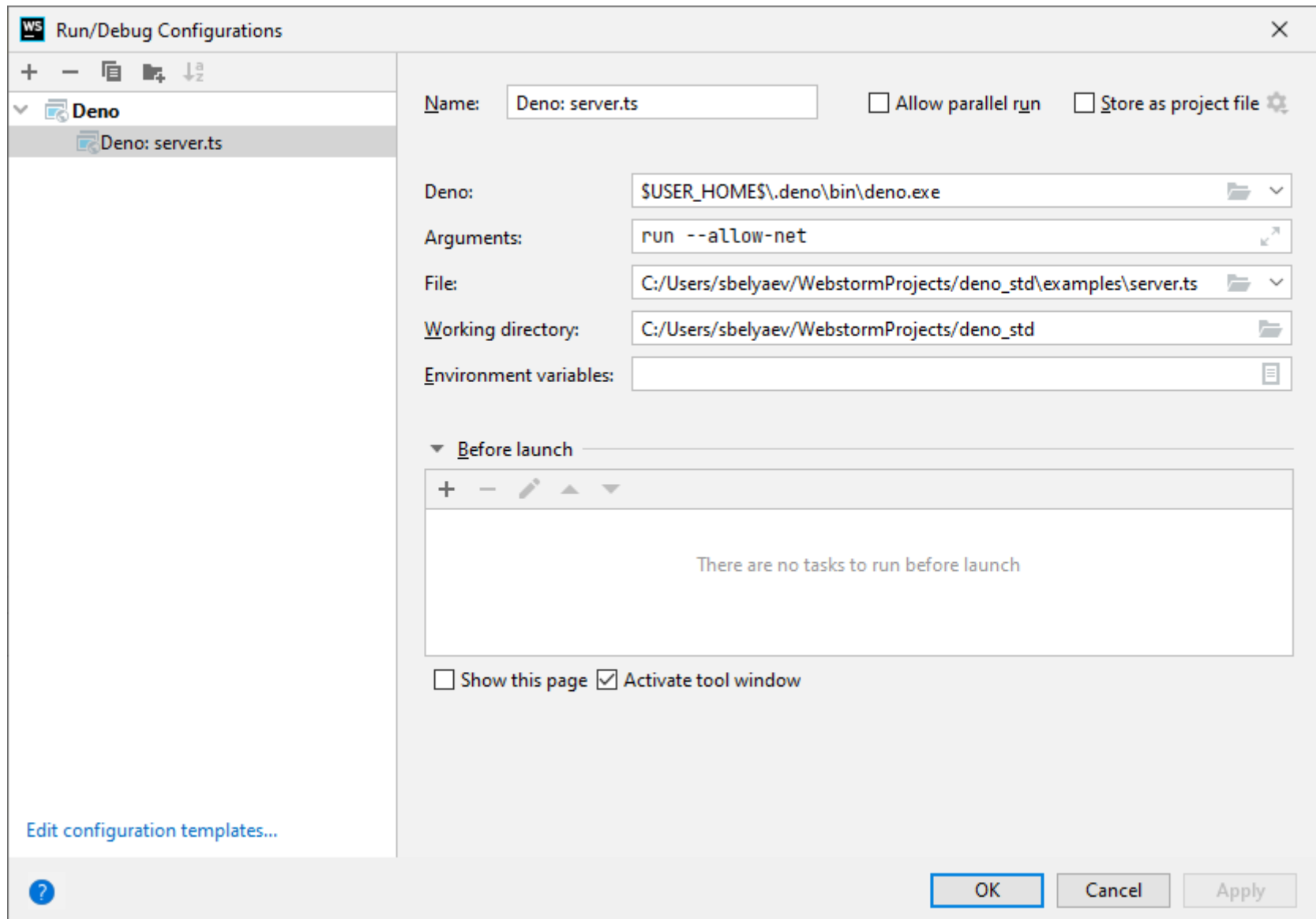
Установка Deno в WebStorm

77



Настройка конфигурации

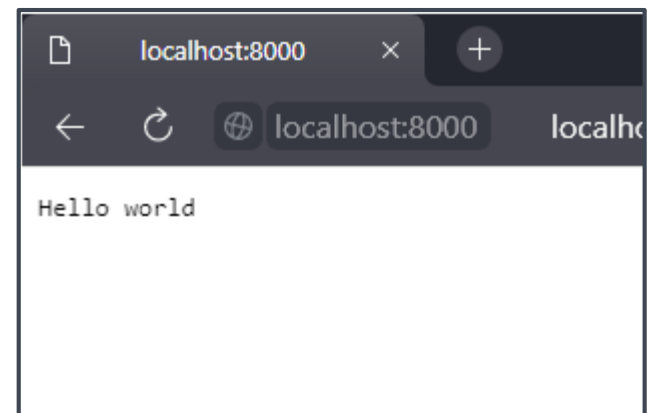
78



Пример программы (1)

79

```
const listener = Deno.listen({ port: 8000 });
console.log("http://localhost:8000/");
for await (const conn of listener) {
  (async () => {
    const requests = Deno.serveHttp(conn);
    for await (const { respondWith } of requests) {
      console.log(requests);
      respondWith(new Response("Hello world"));
    }
  })();
}
```



```
Командная строка - deno run --allow-net server.ts
C:\Users\sbelyaev\WebstormProjects\deno_std\examples>deno run --allow-net server.ts
Check file:///C:/Users/sbelyaev/WebstormProjects/deno_std/examples/server.ts
http://localhost:8000/
HttpConn { managedResources: Set { 6 } }
HttpConn { managedResources: Set { 7 } }
```

Пример программы (2)

```
// Не забудьте параметр --allow-net
const url = Deno.args[0] // Чтение параметров командной строки
const res = await fetch(url) // Чтение URL
const body = new Uint8Array(await res.arrayBuffer()) // Получение
await Deno.stdout.write(body) // Вывод в консоль
```

<https://example.com>

```
<!doctype html>
```

```
<html>
```

```
<head>
```

```
  <title>Example Domain</title>
```

```
  <meta charset="utf-8" />
```

```
  <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1" />
```

```
  <style type="text/css">
```

```
    body {
```

```
  ...
```

Варианты запуска

```
deno run main.ts
```

```
deno run https://mydomain.com/main.ts
```

```
cat main.ts | deno run -
```


The screenshot shows the Deno Runtime APIs documentation page. The browser address bar displays 'deno.land'. The page header includes the Deno logo, navigation links for 'Modules', 'Docs', 'Deploy', and 'Community', and a search bar. The 'Runtime APIs' tab is selected, and the version 'v1.36.4' is chosen from a dropdown menu. A sidebar on the left lists various API categories, including 'Broadcast Channel', 'Cache API', 'Compression Streams API', 'Console and Debugging', 'DOM APIs', 'DOM Events', 'Encoding API', 'Errors', 'ES Modules', 'Fetch API', 'File System', 'HTTP Server', and 'I/O'. The main content area provides an overview of the APIs, explaining that they are built into the Deno CLI and are a combination of web platform APIs and Deno-specific APIs. It also mentions that non-standard, Deno-specific APIs are in the 'Deno' namespace. The page lists several functional categories: 'Broadcast Channel' (with 'BroadcastChannel' and 'BroadcastChannelEventMap'), 'Cache API' (with 'Cache', 'CacheQueryOptions', 'CacheStorage', and 'caches'), and 'Compression Streams API' (with 'CompressionStream' and 'DecompressionStream').

Runtime APIs | Deno

Manual Runtime APIs

v1.36.4 Latest

☐ Show Unstable API

- ▶ Broadcast Channel
- ▶ Cache API
- ▶ Compression Streams API
- ▶ Console and Debugging
- ▶ DOM APIs
- ▶ DOM Events
- ▶ Encoding API
- ▶ Errors
- ▶ ES Modules
- ▶ Fetch API
- ▶ File System
- ▶ HTTP Server
- ▶ I/O

There are APIs that are built into the Deno CLI that are beyond those that are built-ins for JavaScript. They are a combination of web platform APIs Deno has implemented and Deno specific APIs.

We try to keep non-standard, Deno specific, APIs in the `Deno` namespace. We have grouped the APIs into the following functional categories.

Broadcast Channel

- `BroadcastChannel`
- `BroadcastChannelEventMap`

Cache API

- `Cache`
- `CacheQueryOptions`
- `CacheStorage`
- `caches`

Compression Streams API

- `CompressionStream` An API for compressing a stream of data.
- `DecompressionStream` An API for decompressing a stream of data.

Разрешения Deno

82

- **--allow-env=<allow-env>**
 - Разрешение устанавливать значения переменным среды
- **--allow-hrtime**
 - Разрешение на точное вычисление времени
- **--allow-net=<allow-net>**
 - Разрешение на доступ к сети
- **--allow-ffi**
 - Разрешение на использование библиотек
- **--allow-read=<allow-read>**
 - Разрешение на чтение файлов ОС
- **--allow-run=<allow-run>**
 - Разрешение на запуск подпроцессов
- **--allow-write=<allow-write>**
 - Разрешение на запись файлов
- **-A, --allow-all**
 - Все разрешения

https://deno.land/std@0.201.0 – библиотека стандартных модулей

The screenshot shows the Deno Standard Modules page. The browser address bar displays 'std@0.201.0 | Deno' and 'deno.land'. The page header includes the Deno logo, navigation links (Modules, Docs, Deploy, Community), a search bar, and a 'Ctrl K' shortcut. The main content area is titled 'std 0.201.0' and describes it as the 'Deno standard library'. It provides links to 'View Documentation' and 'View Source'. Under 'Attributes', it lists 'Official Deno project' and 'Includes Deno configuration'. The 'Repository' section points to 'denoland/deno_std'. The 'Current version released' is 'a week ago'. A 'Versions' list on the left shows versions from 0.195.0 to 0.201.0, with 0.201.0 marked as 'Latest'. The right sidebar contains sections for 'Deno Standard Modules' (with a Codecov 88% badge), 'Releases' (explaining independent tagging), and 'How to use' (recommending tagged releases).

std 0.201.0
Deno standard library

[View Documentation](#)
[View Source](#)

Attributes

- Official Deno project
- Includes Deno configuration

Repository

[denoland/deno_std](#)

Current version released
a week ago

Versions

- 0.201.0 (Latest)
- 0.200.0
- 0.199.0
- 0.198.0
- 0.197.0
- 0.196.0
- 0.195.0

Deno Standard Modules

codecov 88%

These modules do not have external dependencies and they are reviewed by the Deno core team. The intention is to have a standard set of high quality code that all Deno projects can use fearlessly.

Contributions are welcome!

Releases

Standard library is currently tagged independently of Deno version. This will change once the library is stabilized.

To check compatibility of different version of standard library with Deno CLI see [this list](#).

How to use

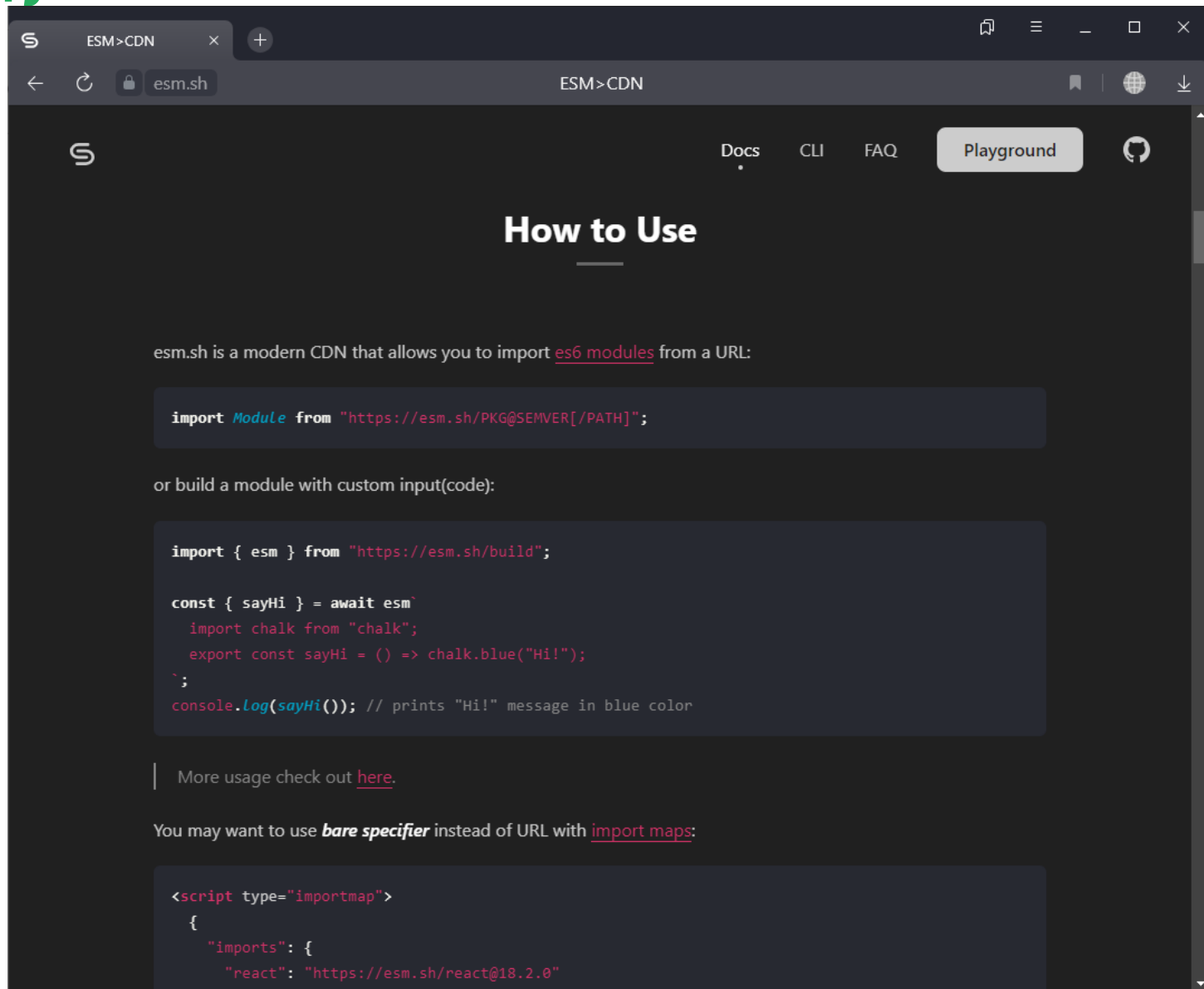
These modules will eventually be tagged in accordance with Deno releases but as of today we do not yet consider them stable and so we version the standard modules differently from the Deno runtime to reflect this.

It is strongly recommended that you link to tagged releases to avoid unintended updates and breaking changes.

Don't link to / import any module whose path

https://esm.sh/ - использование модулей NPM

84



Вопросы для самопроверки

- Где поддерживаются модули (браузер, сервер)?
- В чем отличие CommonJS и ES6 модулей?
- Для чего нужны и чем отличаются npm, yarn, pnpm?
- В чем отличия в применении модулей debug и winston?
- Для чего нужны rollbar и sentry?
- Чем отличаются setTimeout, setImmediate, setInterval?
- Для чего используются domain и почему они deprecated?
- Как можно использовать несколько процессов в JS?
- Для чего нужен nvm?
- В чем ключевые отличия Deno от NodeJS?