



Web-технологии

Сборка приложений, новые возможности
JavaScript

- Автоматизация задач с помощью GULP
 - src, dest, обработка стилей, скриптов
- Сборка модулей (bundle) с помощью Webpack
 - простая конфигурация (entry, output)
 - использование загрузчиков и плагинов
 - контроль изменений кода при разработке
 - отладка web-приложений
- Новые возможности JavaScript
 - Web-сообщения, web-worker
 - Web-сокеты (ws, socket.io)
 - Возможности history, sessionStorage, localStorage, cookie
 - Авторизация с Passportjs
 - Sequelize ORM
 - WebAssembly

<https://gulpjs.com/>

<https://babeljs.io/>

<https://webpack.js.org/>

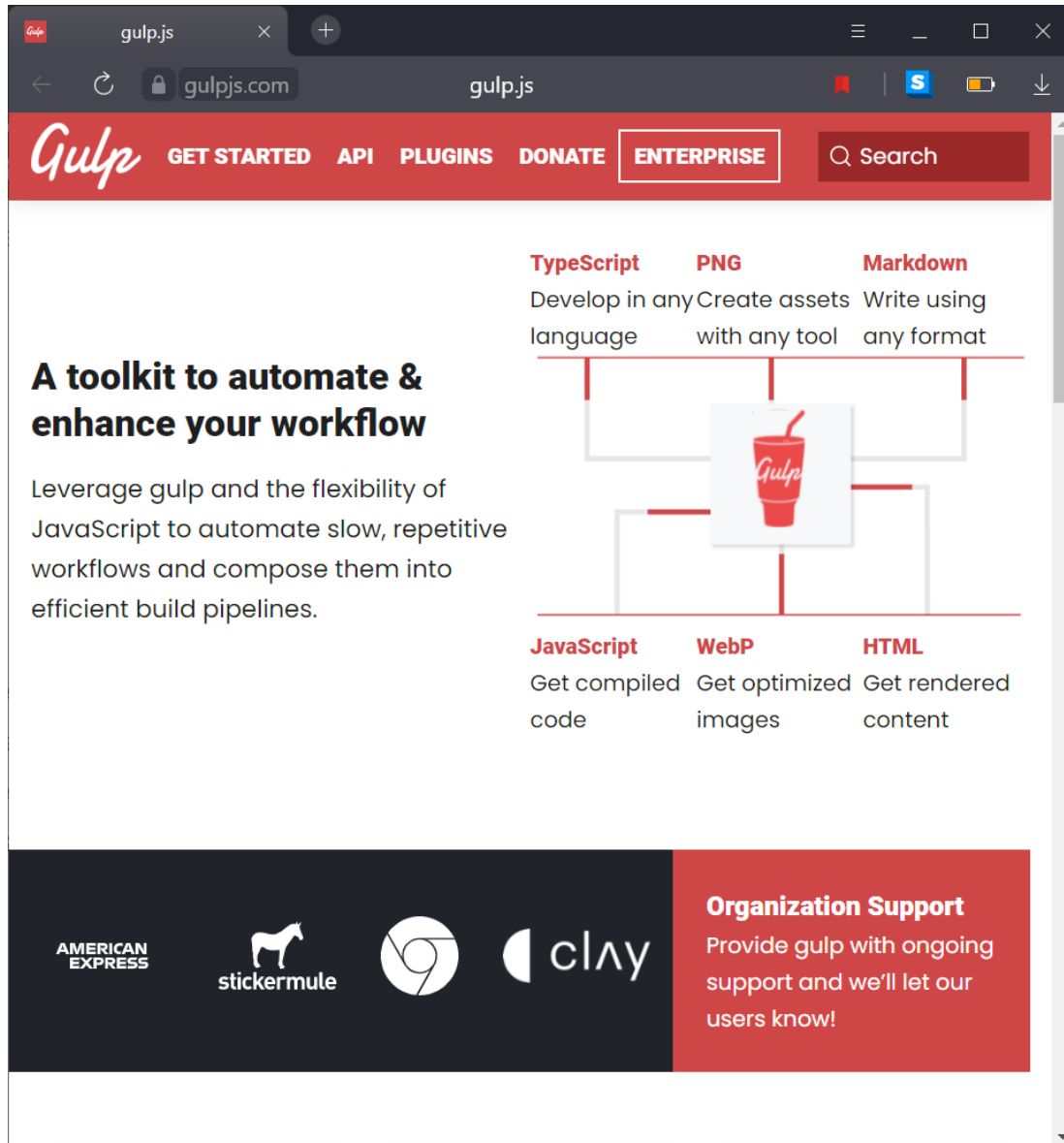
<https://socket.io/>

<http://www.passportjs.org/>

<https://sequelize.org/>

<https://webassembly.org/>

GULP – автоматизация рабочего процесса³



Автоматическое выполнение часто используемых задач

- минификация
 - тестирование
 - объединение файлов
- Написан на JavaScript
Команды записываются на JavaScript

```
npm i --global gulp-cli
npm install --save-dev gulp
gulp --version
> CLI version: 2.3.0
> Local version: 4.0.2
```

<https://gulpjs.com/>

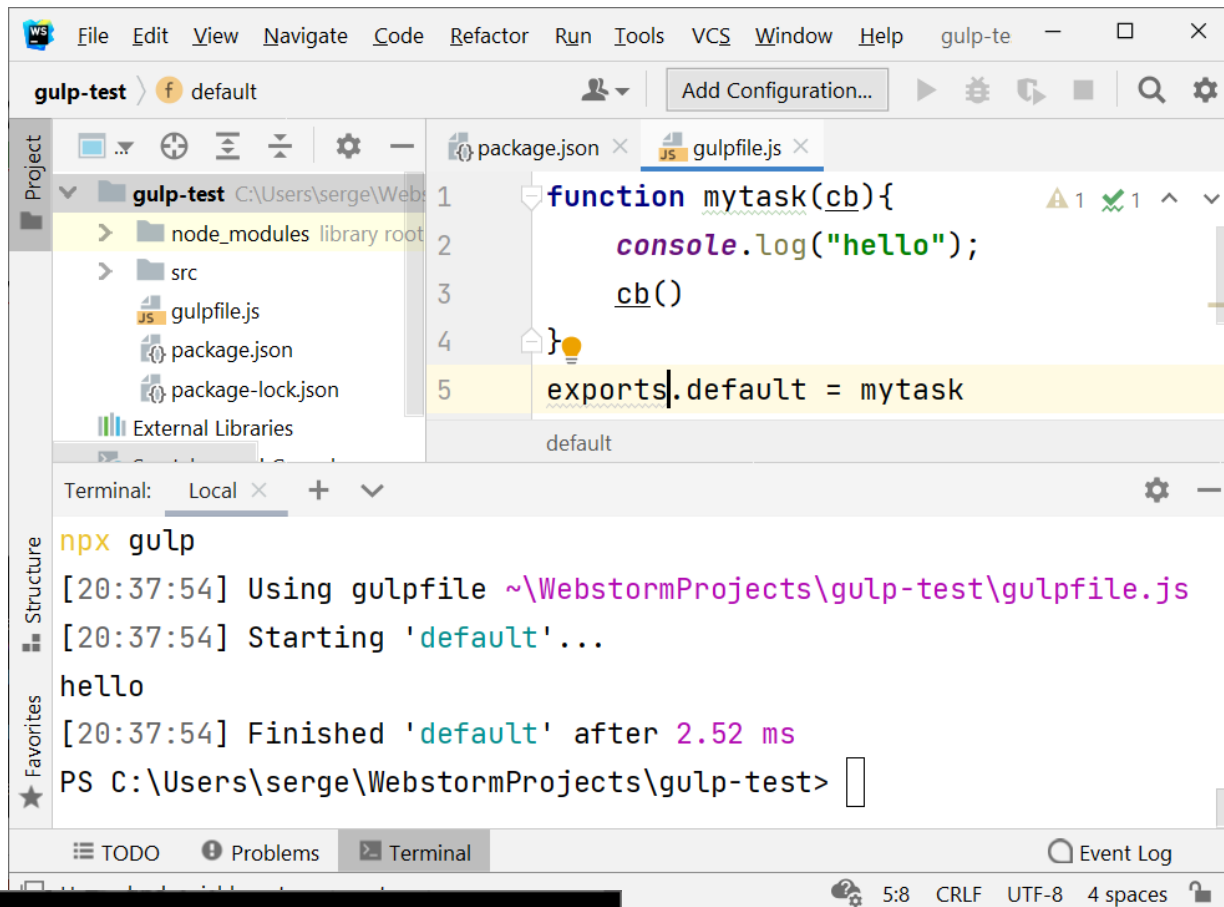
GULP. Простейшая задача / default

4

gulpfile.js

```
function mytask(cb){  
  console.log("hello");  
  cb()  
}  
exports.default = mytask
```

В данном случае
запуск через **npx**, т.к.
я не использовал
глобальную
установку

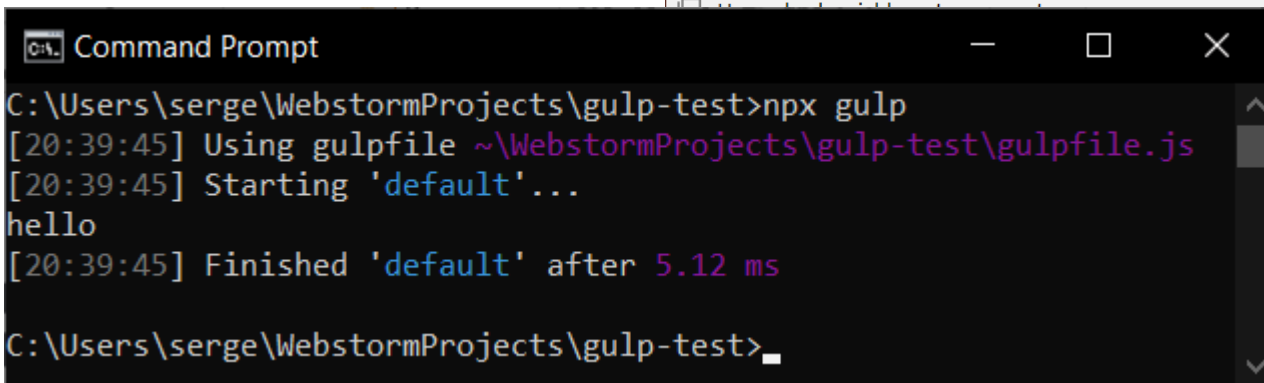


The screenshot shows the Visual Studio Code editor interface. The file explorer on the left shows a project named 'gulp-test' with files 'package.json', 'package-lock.json', 'gulpfile.js', and a 'src' directory. The editor window displays 'gulpfile.js' with the following code:

```
1 function mytask(cb){  
2   console.log("hello");  
3   cb()  
4 }  
5 exports.default = mytask
```

The terminal at the bottom shows the command 'npx gulp' and its output:

```
npx gulp  
[20:37:54] Using gulpfile ~\WebstormProjects\gulp-test\gulpfile.js  
[20:37:54] Starting 'default'...  
hello  
[20:37:54] Finished 'default' after 2.52 ms  
PS C:\Users\serge\WebstormProjects\gulp-test>
```



The screenshot shows a Windows Command Prompt window with the following text:

```
C:\Users\serge\WebstormProjects\gulp-test>npx gulp  
[20:39:45] Using gulpfile ~\WebstormProjects\gulp-test\gulpfile.js  
[20:39:45] Starting 'default'...  
hello  
[20:39:45] Finished 'default' after 5.12 ms  
C:\Users\serge\WebstormProjects\gulp-test>
```

Для TypeScript –
переименовать в
gulpfile.ts

- **src** – выбор файлов в соответствии с шаблоном
- **dest** – записывает файлы в директории
- **symlink** – создание символьных ссылок вместо копирования файлов
- **task** – объявление задач
- **lastRun** – получение timestamp последней успешной задачи
- **parallel** – запуск задач в параллель
- **series** – последовательный запуск задач
- **watch** – контроль изменений файлов
- **tree** – получение дерева задач

GULP. Последовательный запуск / series

6

gulpfile.js

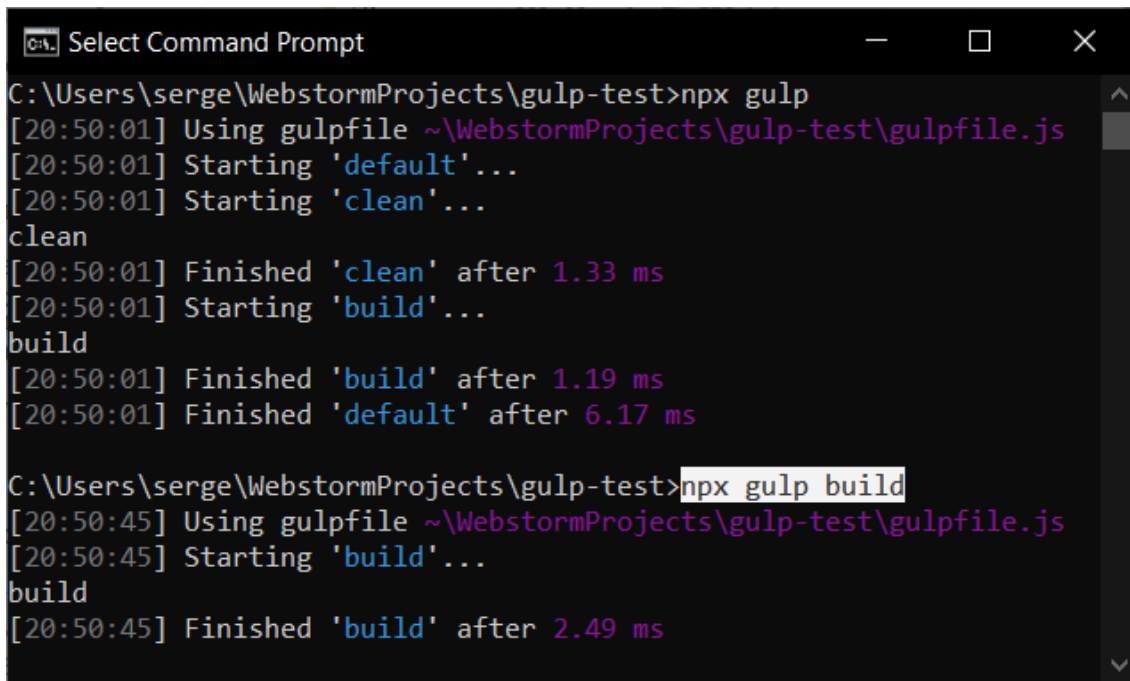
```
const {series} = require("gulp")
function build(cb){
  console.log("build");
  cb()
}
function clean(cb){
  console.log("clean");
  cb()
}
exports.build = build
exports.default = series(clean, build)
```

> **gulp**

- default
 - clean
 - build

> **gulp build**

- build



```
C:\Users\serge\WebstormProjects\gulp-test>npx gulp
[20:50:01] Using gulpfile ~\WebstormProjects\gulp-test\gulpfile.js
[20:50:01] Starting 'default'...
[20:50:01] Starting 'clean'...
clean
[20:50:01] Finished 'clean' after 1.33 ms
[20:50:01] Starting 'build'...
build
[20:50:01] Finished 'build' after 1.19 ms
[20:50:01] Finished 'default' after 6.17 ms

C:\Users\serge\WebstormProjects\gulp-test>npx gulp build
[20:50:45] Using gulpfile ~\WebstormProjects\gulp-test\gulpfile.js
[20:50:45] Starting 'build'...
build
[20:50:45] Finished 'build' after 2.49 ms
```

- Запуск **default**
- Запуск задачи (в примере – **build**)

GULP. Параллельный запуск / parallel

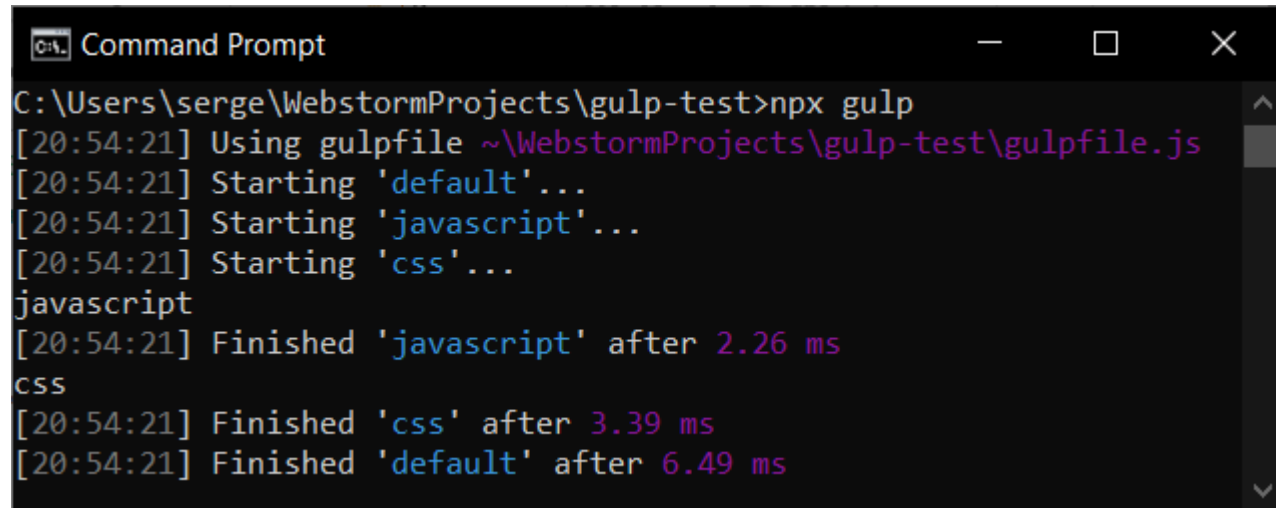
7

gulpfile.js

```
const {parallel} = require("gulp")
exports.default = parallel(javascript, css)
function javascript(cb){
  console.log("javascript");
  cb()
}
function css(cb){
  console.log("css");
  cb()
}
```

> gulp

- default
 - javascript, css



```
Command Prompt
C:\Users\serge\WebstormProjects\gulp-test>npx gulp
[20:54:21] Using gulpfile ~\WebstormProjects\gulp-test\gulpfile.js
[20:54:21] Starting 'default'...
[20:54:21] Starting 'javascript'...
[20:54:21] Starting 'css'...
javascript
[20:54:21] Finished 'javascript' after 2.26 ms
css
[20:54:21] Finished 'css' after 3.39 ms
[20:54:21] Finished 'default' after 6.49 ms
```

// Возможно комбинирование

```
exports.build = series(clean, parallel(css, javascript));
```

GULP. Асинхронная обработка

8

gulpfile.js

```
const { src, dest } = require('gulp');
const { EventEmitter } = require('events');
const fs = require('fs')
function stream() { // Задача с потоком
  return src('*.js').pipe(dest('build/'));
}
function eventEmitter() { // Задача с EventEmitter
  const emitter = new EventEmitter();
  setTimeout(() => emitter.emit('finish'), 250);
  return emitter;
}
function callback(cb) {
  cb(); // cb() вызывается после окончания async-работы
}
function callbackError(cb) {
  cb(new Error('kaboom')); // Аналогично с ошибкой
}
function passingCallback(cb) { // Передача cb асинхронной задаче
  fs.readFile('package.json', cb);
}
async function asyncAwaitTask() { // Использование async
  const { version } = JSON.parse(fs.readFileSync('package.json', 'utf8'));
  await Promise.resolve(version); // Можно явно вызвать await
}
exports.default = asyncAwaitTask;
```


GULP. Работа с файлами / .pipe

9

gulpfile.js

```
const { src, dest } = require('gulp');
const babel = require('gulp-babel');
// Не забудьте установить @babel/core @babel/preset-env
// Не забудьте настроить .babelrc
const uglify = require('gulp-uglify');
```

```
exports.copy = function() { // Копирование файлов
  return src('src/*.js').pipe(dest('build/'));
}
exports.babel = function() { // Применение babel
  return src('src/*.js')
    .pipe(babel()) // presets будут взяты из .babelrc
    .pipe(dest('build/'));
}
exports.uglify = function() {
  return src('src/*.js')
    .pipe(babel())
    .pipe(src('vendor/*.js')) // Доп.источник
    .pipe(uglify()) // Минификация
    .pipe(dest('build/'));
}
```

.babelrc

```
{
  "presets":
    ["@babel/preset-env"]
}
```

<https://only-to-top.ru/blog/tools/2019-10-20-gulp-babel.html>

<https://babeljs.io/>

Результат babel + uglify

10

Исходный файл в src/

```
/**
 * Summation
 * @param a
 * @param b
 * @returns {*}
 */
export function sum(a, b) {
  let s = a + b
  return s
}
```

Результат uglify в build/

```
function sum(u,n){return u+n}export{sum};
```

Результат babel в build/

```
"use strict";
Object.defineProperty(exports, "__esModule", {
  value: true
});
exports.sum = sum;
/**
 * Summation
 * @param a
 * @param b
 * @returns {*}
 */
function sum(a, b) {
  var s = a + b;
  return s;
}
```

Результат babel+uglify в build/

```
"use strict";function sum(e,s){return
e+s}Object.defineProperty(exports,"__esM
odule",{value:!0}),exports.sum=sum;
```

GULP. Фазы вывода.

Переименование. Спецсимволы

gulpfile.js

```
const { src, dest } = require('gulp');
const babel = require('gulp-babel');
const uglify = require('gulp-uglify');
const rename = require('gulp-rename');

exports.default = function() {
  return src('src/**/*.js') // Любая вложенность
    .pipe(babel())
    .pipe(dest('build/')) // Исходный
    .pipe(uglify())
    .pipe(rename({ extname: '.min.js' }))
    .pipe(dest('build/')); // Переименованный
}
```

В результате в **build/** находятся исходные файлы, подготовленные **babel**, а также минифицированные с постфиксом **.min.js**

Спецсимволы:

- * – любая последовательность
- ** – любая вложенность
- ! – отрицание

Примеры для src

- '*.js'
- 'scripts/**/*.js'
- ['scripts/**/*.js', '!scripts/vendor/**']
- ['scripts/**/*.js', '!scripts/vendor/**', 'scripts/vendor/react.js']
- ['**/*.js', '!node_modules/**']

GULP. Создание sourcemaps

12

gulpfile.js

```
const { src, dest } = require('gulp');  
const babel = require('gulp-babel');  
const uglify = require('gulp-uglify');  
const rename = require('gulp-rename');
```

По умолчанию sourcemaps
отключены.

Для встроенных в dest нужно
передать { sourcemaps: true }

```
exports.default = function() {  
  return src('src/**/*.js', { sourcemaps: true }) // Любая вложенность  
    .pipe(babel())  
    .pipe(dest('build/')) // Исходный  
    .pipe(uglify())  
    .pipe(rename({ extname: '.min.js' }))  
    .pipe(dest('build/', { sourcemaps: "." })); // Переименованный  
}
```

build/sum.min.js.map

```
{"version":3,"sources":["sum.js"],"names":["sum","a","b"],"mappings":"aAMO,SAASA,IAAIC,EAAGC,GALvB,OADAD,EAAAC,E","file":"sum.min.js","sourcesContent":["/**\r\n *  
Summation\r\n * @param a\r\n * @param b\r\n * @returns {*} \r\n */\r\nexport function  
sum(a, b) {\r\n  let s = a + b\r\n  return s\r\n}"]}
```

GULP. Наблюдение за изменениями

13

gulpfile.js

/ watch

```
const { watch, series } = require('gulp');
function clean(cb) {
  // тело функции пропущено
  cb();
}
function javascript(cb) {
  // тело функции пропущено
  cb();
}
function css(cb) {
  // тело функции пропущено
  cb();
}
exports.default = function() {
  // Можно вызывать одну задачу
  watch('src/*.css', css);
  // Можно вызвать несколько задач
  watch('src/*.js', series(clean, javascript));
};
exports.watchDelay = function() {
  // Можно вызывать задачу с задержкой
  watch('src/*', { delay: 500 }, function(cb) {
    // тело функции пропущено
    cb()
  })
}
```

Как и в **src** в **watch** можно передавать массивы путей

GULP. Настройка папок в dest

14

gulpfile.js

```
const { src, dest } = require('gulp');
```

```
exports.default = function() {  
  return src('src/**/*')  
    .pipe(dest((file) => { // Деление на папки  
      switch (file.extname) {  
        case ".js":  
        case ".json":  
          return "script";  
        case ".css":  
          return "css";  
        default:  
          return "dest";  
      }  
    })  
  )  
}
```

Файлы будут разнесены по папкам:

- js, json → script
- css → css
- остальные → dest

GULP. Работа со стилями / less, css

15

gulpfile.js

```
const { src, dest } = require('gulp');
const less = require('gulp-less');
// Не забудьте установить LESS
const rename = require('gulp-rename');
const cleanCSS = require('gulp-clean-css');

paths = { // Настройка путей
  styles: {
    src: 'src/**/*.less',
    dest: 'build/styles/'
  }
}

exports.default = function() {
  return src(paths.styles.src)
    .pipe(less()) // Обработать LESS
    .pipe(cleanCSS()) // Минификация CSS
    .pipe(rename({ // Переименовать
      basename: 'main',
      suffix: '.min'
    }))
    .pipe(dest(paths.styles.dest));
}
```

src/test.less

```
.average(@x, @y) {
  @Average: ((@x + @y) / 2);
}
div {
  .average(12px, 10px); // вызов миксина
  padding: @Average; // результат
}
```



build/styles/main.min.css

```
div{padding:11px}
```

GULP. Объединение файлов / concat

16

gulpfile.js

```
const { src, dest } = require('gulp');  
const concat = require('gulp-concat');  
const uglify = require('gulp-uglify');
```

```
exports.default = function() {  
  return src("src/*.js")  
    .pipe(uglify())  
    // Объединить в один файл  
    .pipe(concat("main.min.js"))  
    .pipe(dest("build"));  
}
```

src/sum.js

```
/**  
 * Summation  
 * @param a  
 * @param b  
 * @returns {*}  
 */  
export function sum(a, b) {  
  let s = a + b  
  return s  
}
```

src/sum.js

```
export function mult(a, b) {  
  return a * b  
}
```



build/main.min.js

```
function mult(t,u){return t*u}export{mult};  
function sum(u,n){return u+n}export{sum};
```


GULP. Использование с PUG

17

gulpfile.js

```
const { src, dest } = require('gulp');
const pug = require('gulp-pug');
// Не забудьте установить pug
```

```
exports.pug = function() {
  return src("src/*.pug")
    .pipe(pug({
      locals: { value: 1 },
      verbose: true
    })
  )
  .pipe(dest("build"));
}
```

value == 1

msg.pug

```
html
  head
    meta(charset="utf-8")
    title Сессия
  body
    h1 Счётчик
    -if (value === 1)
      p Добро пожаловать в первый раз
    - else
      p Не первое посещение:
        =value
```

msg.html

```
<html><head><meta charset="utf-8"/><title>Сессия</title></head><body><h1>Счётчик</h1><p>Добро
пожаловать в первый
раз</p></body></html>
```



value == 2

```
<html><head><meta charset="utf-8"/><title>Сессия</title></head><body><h1>Счётчик</h1><p>Не первое
посещение: 2</p></body></html>
```

GULP. Тестирование с Mocha /

on("error"), reporter, globals, require

gulpfile.js

```
const { src, dest, series } = require('gulp')
const mocha = require('gulp-mocha')
// Не забудьте установить mocha should
const utils = require('gulp-util')

function myassert() {
  return src(['tests/test-*.js'])
    .pipe(mocha({ reporter: 'spec' }))
    .on("error", utils.log) // Обработка ошибки
}

function myshould() {
  return src(['tests/should-*.js'])
    .pipe(mocha({ reporter: 'list',
      globals: {
        should: require('should')
      }
    }))
}

exports.default = series(myassert, myshould)
```

Пример src/sum.js

```
module.exports = function sum(a, b) {
  return a + b
}
```

Пример tests/test-sum.js

```
const assert = require("assert")
const sum = require("../src/sum.js")
it('my assert ', function () {
  assert.ok(sum(2,2) == 4, "2+2=4")
})
```

Пример результата:

```
[12:55:51] Starting 'myassert'...
  ✓ assert
  1 passing (4ms)
[12:55:51] Finished 'myassert' after 370 ms
```

Если ошибку `.on("error")` не обрабатывать, а она произойдёт, то всё выполнение скрипта прекратится

GULP. Mocha, es6-модули / import, gulpfile.babel

gulpfile.babel.js

```
import { src } from 'gulp'
import mocha from 'gulp-mocha'
import utils from 'gulp-util'

function myassert() {
  return src(['tests/test-*.js'])
    .pipe(mocha({ reporter: 'spec',
      require: '@babel/register'
    }))
    .on("error", utils.log)
}
exports.default = myassert
```

.babelrc

```
{
  "presets": ["@babel/preset-env"]
}
```

src/sum.js

```
function sum(a, b) {
  let s = a + b
  return s
}
export {sum}
```

tests/test-sum.js

```
import assert from "assert"
import {sum} from "../src/sum.js"

it('assert ', function () {
  assert.ok(sum(2,2) == 4, "2+2=4")
})
```

Пример результата:

```
[15:44:32] Starting 'default'...
√ assert
1 passing (6ms)
[15:44:33] Finished 'default' after 1.1 s
```

Необходимые библиотеки babel

```
npm i --save-dev @babel/core @babel/polyfill @babel/preset-env @babel/register
```

GULP. Использование с TypeScript

20

gulpfile.js

```
const gulp = require("gulp");  
const gulpTS = require("gulp-typescript");  
const ts = gulpTS.createProject("tsconfig.json");
```

```
exports.default = function () {  
  return ts.src()  
    .pipe(ts())  
    .js.pipe(gulp.dest("build"));  
}
```

tsconfig.json

```
{  
  "compilerOptions": {  
    "target": "es5",  
    "module": "commonjs",  
    "strict": true  
  },  
  "include": [  
    "tsc/**/*.ts"  
  ],  
  "exclude": [  
    "node_modules",  
    "**/*.spec.ts"  
  ]  
}
```

tsc/hello.ts

```
function hello(compiler: string) {  
  console.log(Hello from `${compiler}`);  
}  
hello("TypeScript");
```



build/hello.js

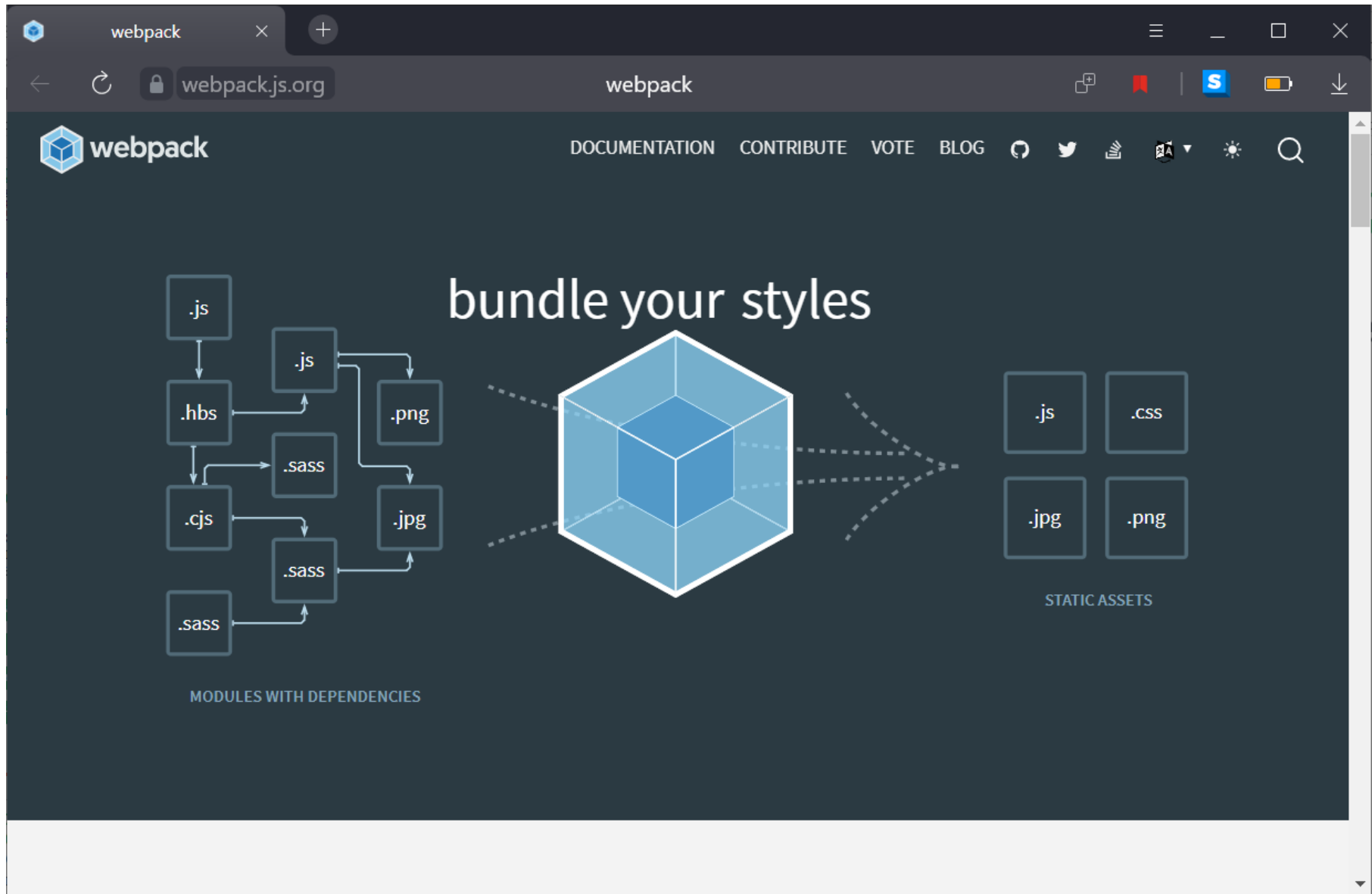
```
"use strict";  
function hello(compiler) {  
  console.log(Hello from  + compiler);  
}  
hello("TypeScript");
```

```
npm i --save-dev typescript gulp-typescript  
tsc -init
```

<https://www.typescriptlang.org/docs/handbook/gulp.html>

Webpack – сборщик модулей (бандлов)

21

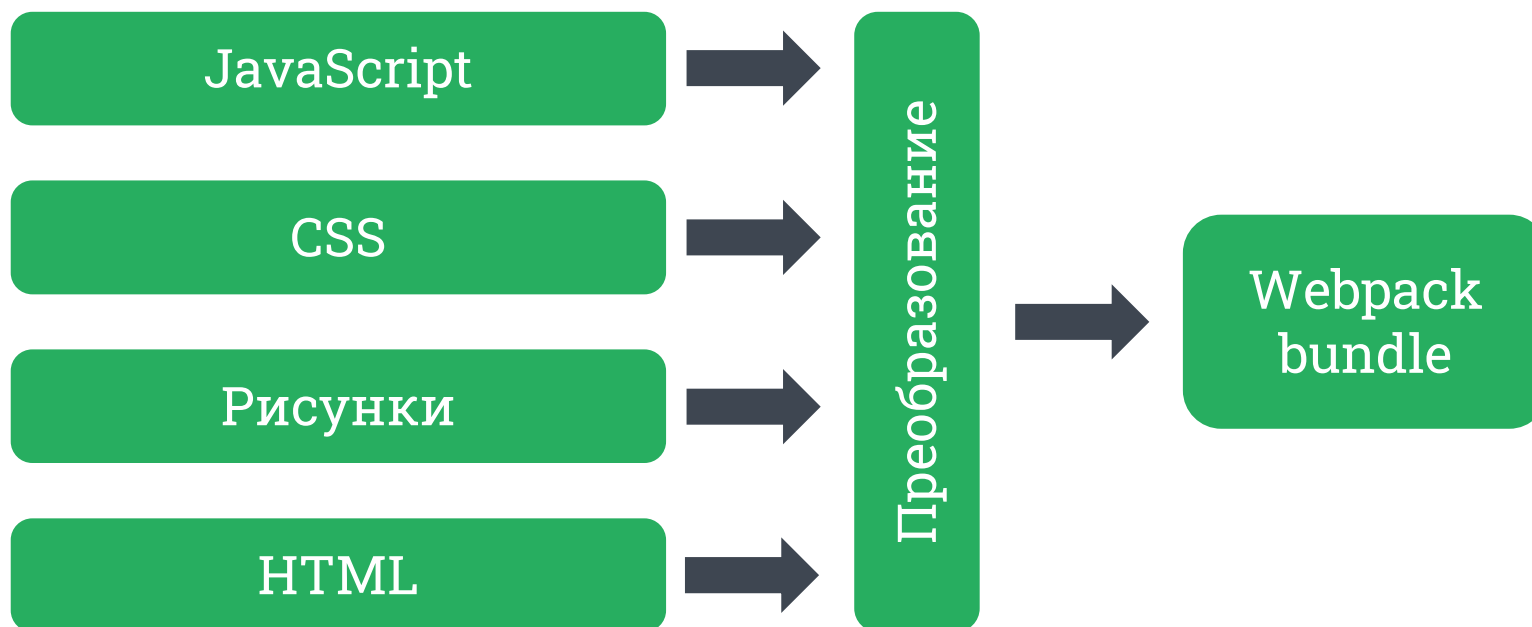


<https://webpack.js.org/>

Webpack – инструмент для сборки

22

- Webpack на основании модулей с зависимостями генерирует статические ресурсы, представляющие эти модули



`npm i -D webpack webpack-cli`

<https://github.com/webpack/webpack>
<https://github.com/webpack/webpack-cli>

Некоторые важные особенности Webpack

- Поддержка нескольких точек входа
- Управление ресурсами проекта
- «Горячая» замена модулей
- Поддержка режимов development / production
- Разделение кода `...mode: 'production'...`
- «Ленивая» загрузка
- Создание библиотек для webpack
- Поддержка модулей ES2015, CommonJS и AMD
- Поддержка TypeScript
- Поддержка Progressive Web Application (PWA)
 - визуальное и функциональное «преобразование» сайта в приложение
 - с 2020 года **офлайн-работу** с PWA могут обеспечивать iOS, Android, Windows, Linux, macOS и Chrome OS в браузерах Chrome, Safari, Firefox, Edge и Samsung Internet

Основные элементы Webpack

24

Entry – точки входа

```
module.exports = {entry: './path/to/my/entry/file.js'};
```

Output – вывод результата

```
const path = require('path');  
module.exports = {entry: './path/to/my/entry/file.js',  
  output: {path: path.resolve(__dirname, 'dist'),  
    filename: 'my-first-webpack.bundle.js' }};
```

Loaders – загрузчики

```
...module: {rules: [{test: /\.txt$/, use: 'raw-loader'}]}...
```

Plugins – подключаемые компоненты

```
...plugins: [new webpack.optimize.UglifyJsPlugin(),  
  new HtmlWebpackPlugin({template: './src/index.html'})]...
```


Webpack. Минимальная конфигурация

package.json

```
{
  "name": "webpack-prj",
  "version": "1.0.0",
  "scripts": {
    "run": "node src/index.js",
    "build": "webpack"
  },
  "devDependencies": {
    "webpack": "^5.58.2",
    "webpack-cli": "^4.9.1"
  }
}
```

src/sum.js

```
module.exports = function sum(a, b) {
  return a + b
}
```

src/mult.js

```
module.exports = function mult(a, b) {
  return a * b
}
```

webpack.config.js

```
module.exports = {
  entry: './src/index.js',
  output: {
    filename: 'bundle.js'
  }
}
```

src/index.js

```
const sum = require("./sum.js")
const mult = require("./mult.js")
console.log(sum(2, 2))
console.log(mult(3, 3))
```



dist/bundle.js

```
((()=>{var o={632:o=>{o.exports=function(o,r){return o*r}},548:o=>{o.exports=function(o,r){return o+r}}},r={};function t(n){var e=r[n];if(void 0!==e)return e.exports;var s=r[n]={exports:{}};return o[n](s,s.exports,t),s.exports}(()=>{const o=t(548),r=t(632);console.log(o(2,2)),console.log(r(3,3))})()})();
```

Результат работы webpack bundle.js

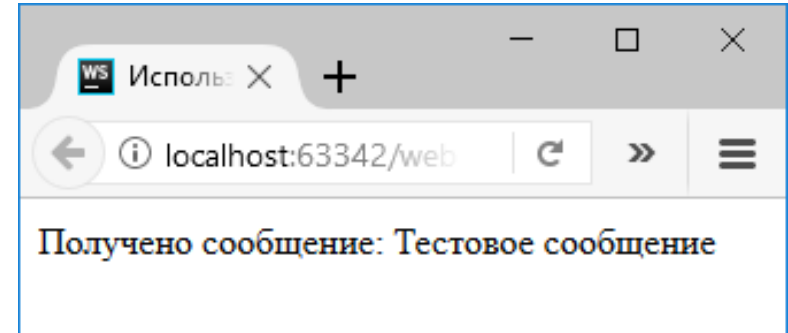
```
(() => {
  var o = {           // Объект "o" с функциями
    632: o => {         // Вернуть функцию умножения
      o.exports = function (o, r) { // "o" внутри и "o" снаружи - разные
        return o * r           // "r" внутри и "r" снаружи - разные
      }
    }, 548: o => {      // Вернуть функцию сложения
      o.exports = function (o, r) { // "o" внутри и "o" снаружи - разные
        return o + r           // "r" внутри и "r" снаружи - разные
      }
    }
  }, r = {};           // Пустой объект "r"
  function t(n) {      // Функция инициализации
    var e = r[n];
    if (void 0 !== e) return e.exports; // Повторный вызов? - вернуть сохранённый объект
    var s = r[n] = {exports: {}};      // Создать новый, сохранить в "r"
    return o[n](s, s.exports, t), s.exports // Инициализировать функцию, вернуть функцию
  }
  (() => {             // Вызов созданной функции
    const o = t(548), r = t(632);      // Инициализация функций
    console.log(o(2, 2)), console.log(r(3, 3)) // Вызов функций
  })()
})();                  // Вызов созданной функции
```

```
0 !== undefined // true
void 0 !== undefined // false
0 !== {} // true
void 0 !== {} // true
```

webpack. Запуск из консоли + HTML 27

index.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Использование Webpack</title>
</head>
<body>
<span id="mymsg"></span>
<script src="bundle.js"></script>
</body>
</html>
```



msg.js

```
module.exports = function(msg, element) {
  element.innerHTML = `Получено сообщение: ${msg}`;
}
```

main.js

```
import msg from "./msg";
msg("Тестовое сообщение", document.getElementById("mymsg"));
```

```
webpack --output-filename ./dist/bundle.js --entry ./src/main.js
```

Точки входа / entry

webpack.config.js

```
module.exports = {  
  entry : './src/index.js'  
}  Создаст dist/main.js
```

webpack.config.js

```
module.exports = {  
  entry : {  
    main: './src/index.js'  
  }  
}  Создаст dist/main.js
```

webpack.config.js

```
module.exports = {  
  entry :  
    ['./src/index1.js', './src/index2.js']  
}  Создаст dist/main.js
```

dependOn, filename, import,
library, runtime, publicPath

webpack.config.js

```
module.exports = {  
  entry : {  
    app: './src/index1.js',  
    adminApp: './src/index2.js'  
  }  
}  Создаст dist/app.js,  
dist/adminApp.js
```

webpack.config.js

```
module.exports = {  
  entry : {  
    app: './src/sum.js',  
    adminApp: {  
      dependOn: 'app',  
      import: './src/index.js'  
    }  
  }  
}  Создаст dist/app.js,  
dist/adminApp.js  
Второй будет зависеть  
от первого
```

Вывод результата / output

webpack.config.js

```
module.exports = {  
  output: {  
    filename: 'bundle.js',  
  }  
}  
} Создаст dist/bundle.js
```

webpack.config.js

```
module.exports = {  
  entry: {  
    app: './src/index.js',  
    search: './src/search.js',  
  },  
  output: {  
    filename: '[name].js',  
    path: __dirname + '/dist',  
  },  
}; Создаст dist/app.js,  
dist/search.js
```

- Варианты подстановки:
 - [hash] хэш идентификатора модуля
 - [chunkhash] хэш содержимого блока
 - [name] имя модуля
 - [id] идентификатор модуля

Особенности загрузчиков / loaders

- Могут использоваться в цепочке
 - Применяются друг за другом
 - Выполняются в обратном порядке от записи
 - Возвращают JS
- Могут быть синхронными или асинхронными
- Запускаются в **Node.js** и могут делать всё, что может делать **Node.js**
- Принимают на вход параметры
- Конфигурируются специальным объектом
- Обычный модуль может экспортировать **loader**
- **Plugin** может расширять возможности **loader**
- Могут порождать произвольные файлы в результате своей работы

Доступные loaders

<https://webpack.js.org/loaders/>

Files

- val-loader, ref-loader

JSON

- cson-loader

Transpiling

- babel-loader, buble-loader, traceur-loader, ts-loader, coffee-loader, fengari-loader, elm-webpack-loader

Templating

- html-loader, pug-loader, markdown-loader, react-markdown-loader, posthtml-loader, handlebars-loader, markup-inline-loader, twig-loader, remark-loader

Styling

- style-loader, css-loader, less-loader, sass-loader, postcss-loader, stylus-loader

Frameworks

- vue-loader, angular2-template-loader

<https://webpack.js.org/awesome-webpack/#loaders>

Использование стилей / loader

32

webpack.config.js

```
module.exports = {  
  entry: './src/index.js',  
  module: {  
    rules: [  
      { test : /\.css$/,  
        exclude: /node_modules/,  
        use: [ // Экспорт модулей как CSS  
          { loader: 'style-loader' },  
          { // Импорт CSS  
            loader: 'css-loader',  
            options: {  
              modules: true  
            }  
          }  
        ]  
      }  
    ]  
  }  
}
```

Импорт CSS

```
require("./mycss.css")
```

Импорт CSS

```
import css from "./mycss.css"
```

```
npm install --save-dev style-loader css-loader
```


Использование LESS / less-loader

33

webpack.config.js

```
module.exports = {
  entry: './src/index.js',
  module: {
    rules: [
      { test : /\.less$/i,
        exclude: /node_modules/,
        use: [
          { loader: 'style-loader' },
          {
            loader: 'css-loader',
            options: { }
          }, // Загрузка и компиляция LESS
          { loader: 'less-loader' }
        ]
      },
    ]
  }
}
```



webpack.config.js

```
module.exports = {
  entry: './src/index.js',
  module: {
    rules: [
      { test : /\.less$/i,
        exclude: /node_modules/,
        use: [
          "style-loader",
          "css-loader",
          { // Загрузка и компиляция LESS
            loader: 'less-loader',
            options: {
              // Параметры
            }
          }
        ]
      },
    ]
  }
}
```

Сначала
выполнится
LESS, затем CSS

npm install --save-dev less-loader less

Использование Babel / babel-loader

34

webpack.config.js

```
module.exports = {
  entry: './src/index.js',
  module: {
    rules: [
      {
        test: /\.m?js$/,
        exclude: /(node_modules|bower_components)/,
        use: {
          loader: 'babel-loader',
          options: {
            presets: ['@babel/preset-env']
          }
        }
      }
    ]
  }
}
```

src/index.js

```
import {sum} from './sum.js'
import {mult} from './mult.js'
```

```
console.log(sum(2, 2))
console.log(mult(3, 3))
```

src/sum.js

```
export function sum(a, b) {
  return a + b
}
```

src/mult.js

```
export function mult(a, b) {
  return a * b
}
```



dist/main.js

```
((()=>{"use strict";var e=e=>{"undefined"!=typeof
Symbol&&Symbol.toStringTag&&Object.defineProperty(e,Symbol.toStringTag,{value:"Module"}),Object.
defineProperty(e,"__esModule",{value:!0}),o={};e(o);var
l={};e(l),console.log((0,o.default)(2,2)),console.log((
0,l.default)(3,3)))})();
```

npm install -D babel-loader @babel/core @babel/preset-env

Использование TypeScript / ts-loader 35

webpack.config.js

```
module.exports = {
  mode: "development",
  devtool: "inline-source-map",
  entry: "./src/app.ts",
  module: {
    rules: [
      // Все файлы `.ts` и `.tsx` обрабатываются ts-loader
      { test: /\.tsx?$/, loader: "ts-loader" }
    ]
  }
};
```

tsconfig.json

```
{
  "compilerOptions": {
    "sourceMap": true
  }
}
```

src/app.ts

```
let s:string = "Hello world"
console.log(s)
```



dist/app.js

```
/***/ ((() => { // webpackBootstrap
var __webpack_exports__ = {};
/*!*****!\
!*** ./src/app.ts ***!
\*****/
var s = "Hello world";
console.log(s);
/***/ })()
;
//# ...
```

npm install -D ts-loader typescript

Plugin – костяк Webpack

- **Webpack** создан на основе той же системы подключаемых компонентов, которая доступна в его конфигурации
- Подключаемые компоненты предлагают функции, которые не могут реализовать загрузчики

Примеры:

- **HtmlWebpackPlugin**
 - создание HTML файлов
- **NpmInstallWebpackPlugin**
 - автоустановка недостающих зависимостей в процессе разработки
- **ProgressPlugin**
 - отображение процесса компиляции
- **CopyWebpackPlugin**
 - копирование файлов и директорий

<https://webpack.js.org/plugins/>

Перечень Plugin

BannerPlugin

CommonsChunkPlugin

CompressionWebpackPlugin

ContextReplacementPlugin

CopyWebpackPlugin

DefinePlugin

DllPlugin

EnvironmentPlugin

EslintWebpackPlugin

HotModuleReplacementPlugin

HtmlWebpackPlugin

IgnorePlugin

LimitChunkCountPlugin

MinChunkSizePlugin

MiniCssExtractPlugin

NoEmitOnErrorsPlugin

NormalModuleReplacementPlugin

NpmInstallWebpackPlugin

ProgressPlugin

ProvidePlugin

SourceMapDevToolPlugin

EvalSourceMapDevToolPlugin

TerserPlugin

Пример использования plugin

webpack.config.js

```
const HtmlWebpackPlugin = require('html-webpack-plugin');
const webpack = require('webpack');
const path = require('path');
```

```
module.exports = {
  entry: './src/index.js',
  mode: 'development',
  module: {
    rules: [
      {
        test: /\.js$/,
        use: 'babel-loader',
      },
    ],
  },
  plugins: [
    // Прогресс создания модуля
    new webpack.ProgressPlugin(),
    // Использование шаблонов
    new HtmlWebpackPlugin({ template: './src/index.html' }),
  ],
};
```

Пример запуска из скрипта

```
const configuration = require('./webpack.config.js');
let compiler = webpack(configuration);
new webpack.ProgressPlugin().apply(compiler);
```

`npm install --save-dev html-webpack-plugin`

Webpack. Вызов функции / HTML

39

webpack.config.js

```
const HtmlWebpackPlugin = require('html-webpack-plugin');
const path = require('path');
module.exports = {
  entry: './src/index.js',
  mode: 'development',
  module: {
    rules: [
      { test: /\.css$/,
        exclude: /node_modules/,
        use: ['style-loader', 'css-loader']
      },
      { test: /\.m?js$/,
        exclude: /(node_modules|bower_components)/,
        use: {
          loader: 'babel-loader',
          options: { presets: ['@babel/preset-env'] }
        }
      }
    ]
  },
  plugins: [
    new HtmlWebpackPlugin({ template: './src/index.html' }),
  ],
};
```

src/index.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Использование Webpack</title>
</head>
<body>
  <div id="mymsg">Здесь будет сообщение</div>
  <button onclick="show()">Отобразить</button>
  <script src="main.js"></script>
</body>
</html>
```

Webpack. Вызов функции / JS

40

src/index.js

```
import {msg} from "./msg";
require("./mycss.css");
function show() {
  msg("Тестовое сообщение", document.getElementById("mymsg"));
}
window.show = show
```

По умолчанию
функция не
доступна в window

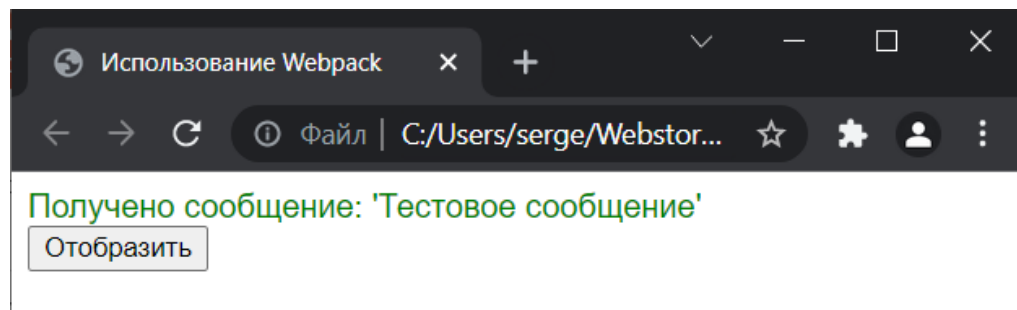
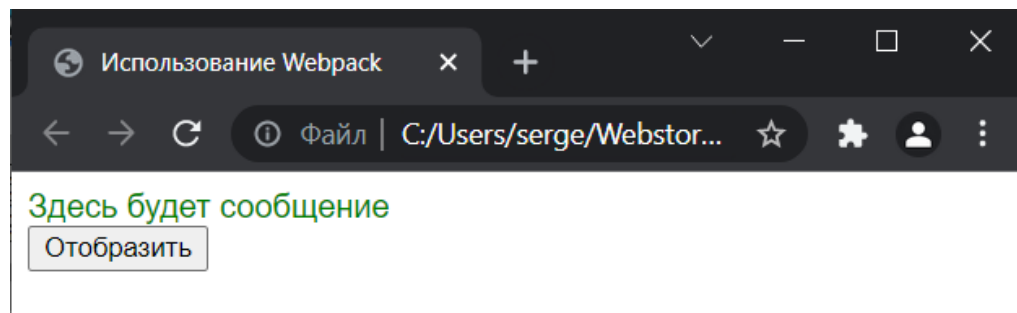
src/msg.js

```
export function msg(msg, element) {
  element.innerHTML = `Получено сообщение: '${msg}'`;
}
```

src/mycss.css

```
div {
  color: green;
  font-family: Arial;
}
```

dist/index.html



- **import** в стиле **ES2015**
- **require()** в соответствии с **CommonJS**
- **@import** внутри файлов **css/sass/less**
- **url** рисунков в стилях **url(...)** и **HTML**-тегах ****

Webpack. Контроль изменения файлов / watch, watchOption

webpack.config.js

```
const HtmlWebpackPlugin = require('html-webpack-plugin');
const path = require('path');
module.exports = {
  entry: './src/index.js',
  mode: 'development',
  module: {
    rules: [
      {
        test: /\.m?js$/,
        exclude: /(node_modules|bower_components)/,
        use: {
          loader: 'babel-loader',
          options: { presets: ['@babel/preset-env'] }
        }
      }
    ]
  },
  watch: true, // Контролировать изменения
  watchOptions: { // Задержка перед сборкой в мсек
    aggregateTimeout: 200
  },
  plugins: [
    new HtmlWebpackPlugin({ template: './src/index.html' }),
  ],
};
```

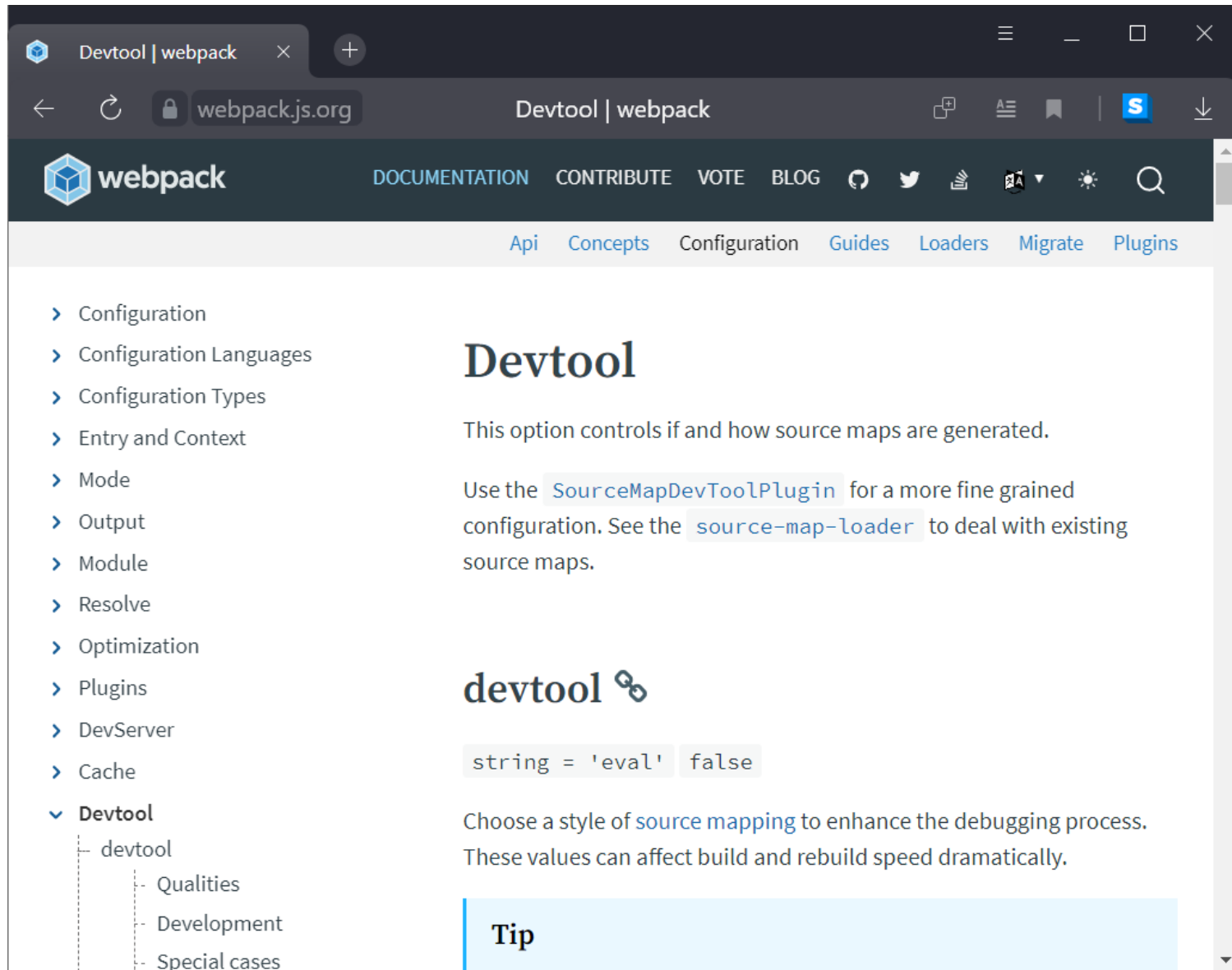
Используется
кэширование, пересборка
занимает меньше
времени

Webpack. Отдельный файл SourceMap

43

webpack.config.js

```
const HtmlWebpackPlugin = require('html-webpack-plugin');
const webpack = require('webpack');
const path = require('path');
module.exports = {
  entry: './src/index.js',
  mode: 'development',
  module: {
    rules: [
      {
        test: /\.m?js$/,
        exclude: /(node_modules|bower_components)/,
        use: {
          loader: 'babel-loader',
          options: { presets: ['@babel/preset-env'] }
        }
      }
    ]
  },
  devtool: false, // Отключение devtool
  plugins: [
    new HtmlWebpackPlugin({ template: './src/index.html' }),
    new webpack.SourceMapDevToolPlugin({
      append: '\n//# sourceMappingURL=https://example.com/sourcemap/[url]',
      filename: '[name].map',
    })
  ],
};
```



The screenshot shows the webpack.js.org website in a browser. The page title is "Devtool | webpack". The navigation bar includes links for DOCUMENTATION, CONTRIBUTE, VOTE, BLOG, and social media icons. The main content area is titled "Devtool" and explains that this option controls if and how source maps are generated. It mentions the `SourceMapDevToolPlugin` for fine-grained configuration and the `source-map-loader` for existing source maps. A code snippet shows `string = 'eval' false`. A "Tip" box at the bottom suggests choosing a style of source mapping to enhance the debugging process. The left sidebar lists various configuration options, with "Devtool" expanded to show "devtool", "Qualities", "Development", and "Special cases".

Configuration

- Configuration Languages
- Configuration Types
- Entry and Context
- Mode
- Output
- Module
- Resolve
- Optimization
- Plugins
- DevServer
- Cache
- Devtool
 - devtool
 - Qualities
 - Development
 - Special cases

Devtool

This option controls if and how source maps are generated.

Use the `SourceMapDevToolPlugin` for a more fine grained configuration. See the `source-map-loader` to deal with existing source maps.

devtool

```
string = 'eval' false
```

Choose a style of [source mapping](#) to enhance the debugging process. These values can affect build and rebuild speed dramatically.

Tip

Webpack. Отладка кода / debugger

45

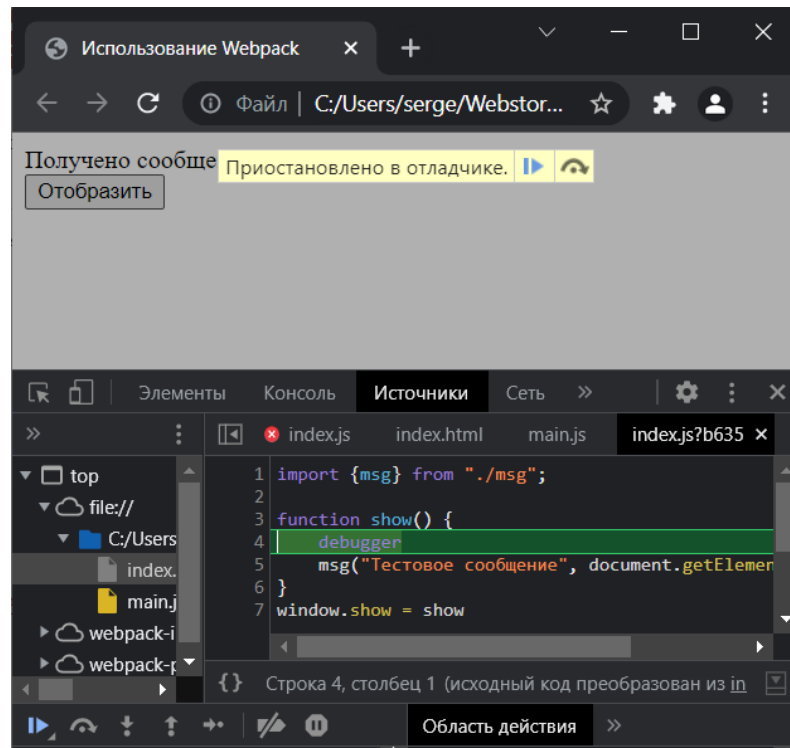
webpack.config.js

```
const HtmlWebpackPlugin = require('html-webpack-plugin');
const path = require('path');
module.exports = {
  entry: './src/index.js',
  mode: 'development',
  // build: slowest, rebuild: ok
  // production: no, quality: original
  devtool: 'eval-source-map',
  plugins: [
    new HtmlWebpackPlugin({ template: './src/index.html' }),
  ],
};
```

src/index.js

```
import {msg} from './msg';
```

```
function show() {
  debugger
  msg("Тестовое сообщение", document.getElementById("mymsg"));
}
window.show = show
```



production devtool:

- (none)
- source-map
- nosources-source-map
- hidden-nosources-source-map
- hidden-source-map

Rollup – сборщик модулей

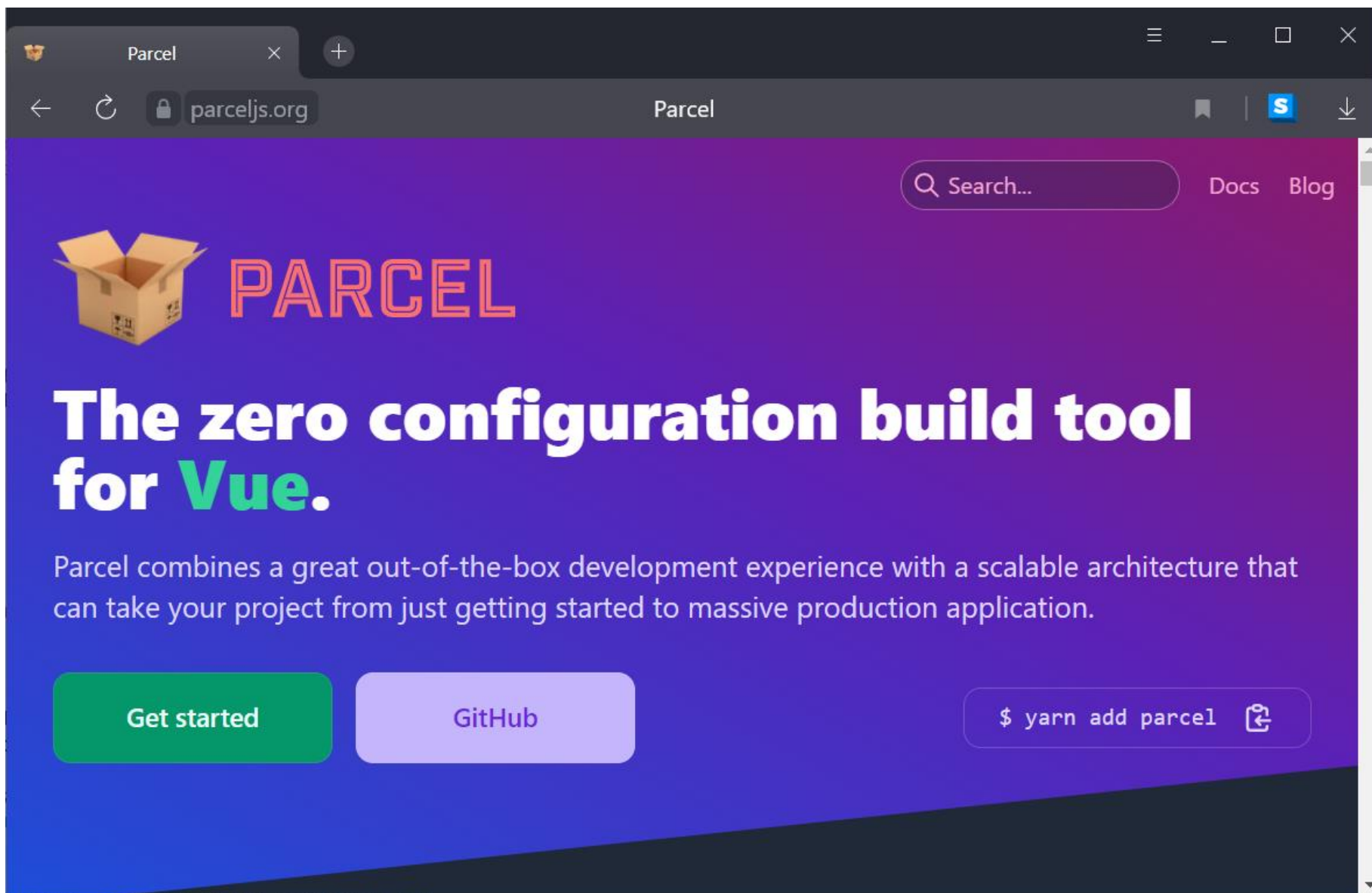
46



<https://rollupjs.org/>

Parcel – сборщик модулей

47



<https://parceljs.org/>

Esbuild – простой сборщик модулей 48

The screenshot shows the esbuild website with a dark theme. On the left is a sidebar with navigation links. The main content area features the esbuild logo, a tagline, and a horizontal bar chart comparing bundling times of different tools.

esbuild
An extremely fast JavaScript bundler

Getting Started

- Install esbuild
- Your first bundle
- Build scripts
- Bundling for the browser
- Bundling for node
- Other ways to install

API

- Transform API
- Build API
- Simple options
- Advanced options
- JS-specific details

Content Types

- JavaScript
- TypeScript
- JSX

Tool	Time
esbuild	0.33s
parcel 2	32.48s
rollup + terser	34.95s
webpack 5	41.53s

Above: the time to do a production bundle of 10 copies of the [three.js](#) library from scratch using default settings, including minification and source maps. More info [here](#).

<https://esbuild.github.io/>

Web-сообщения / postMessage

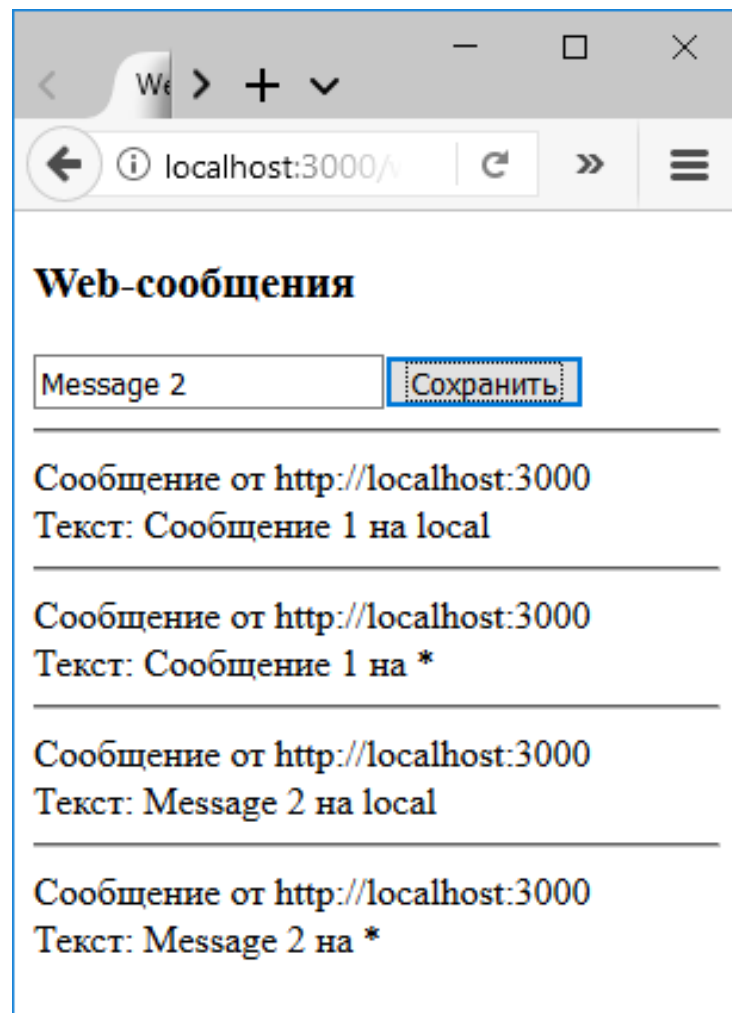
webmsg1.pug

```
doctype html
html
  head
    meta(charset="utf-8")
    script(src="/public/webmsg1.js")
    title Web-сообщения
  body(onload="init(show)")
    h3 Web-сообщения
    input#input
    button(onclick="post(input)") Сохранить
    dev#show
```

webmsg1.js

```
function init(show) {
  addEventListener("message", (e) => {
    show.innerHTML += `<hr>
      Сообщение от ${e.origin}<br>
      Текст: ${e.data}<br>`;
  });
}

function post(input) {
  let data = input.value;
  postMessage(`${data} на local`, 'http://localhost:3000');
  postMessage(`${data} на local`, 'http://localhost:8080');
  postMessage(`${data} на *`, '*');
}
```



Web-сообщения в iframe / HTML

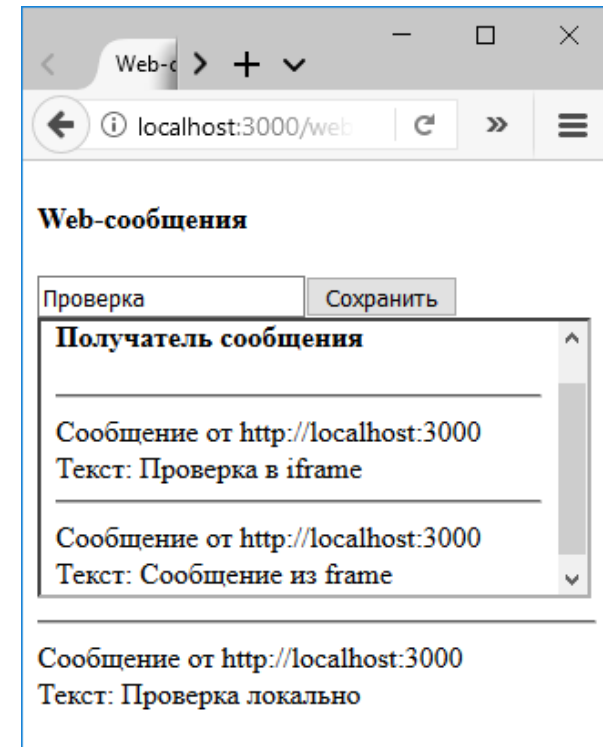
50

webmsg2.pug

```
doctype html
html
  head
    meta(charset="utf-8")
    script(src="/public/webmsg2.js")
    title Web-сообщения
  body(onload="init(show) ")
    h4 Web-сообщения
    input#input
    button(onclick="post(input, myTarget) ") Сохранить
    iframe#myTarget(src="/webmsg2frame")
    dev#show
```

webmsg2frame.pug

```
doctype html
html
  head
    meta(charset="utf-8")
    script(src="/public/webmsg2frame.js")
  body(onload="init(show) ")
    h4 Получатель сообщения
    dev#show
```



Web-сообщения в iframe / JS

51

webmsg2.js

```
function init(show) {  
  addEventListener("message", (e)=>{  
    show.innerHTML += `

---

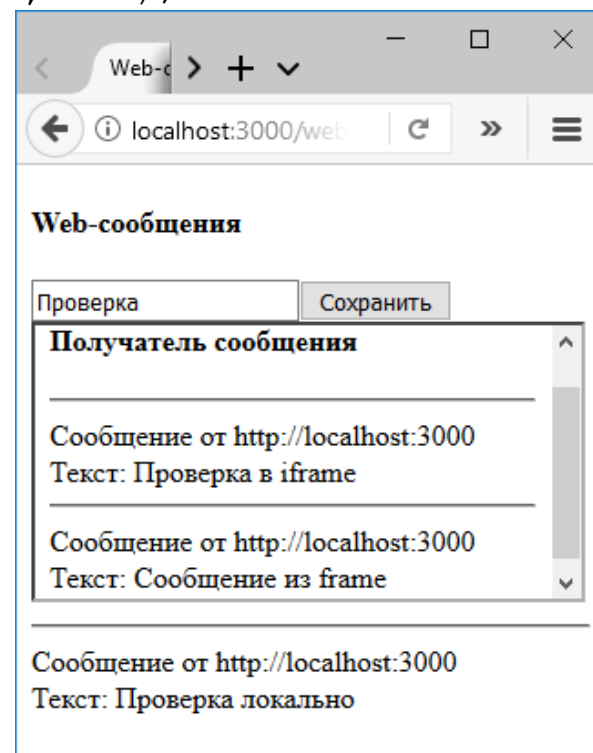
  
    Сообщение от ${e.origin}<br>  
    Текст: ${e.data}<br>`;  
  });  
}  
  
function post(input, myTarget) {  
  let data = input.value;  
  myTarget.contentWindow.postMessage(`${data} в iframe`, '*');  
  postMessage(`${data} локально`, '*');  
}
```

webmsg2frame.js

```
function init(show) {  
  addEventListener("message", (e)=>{  
    show.innerHTML += `

---

  
    Сообщение от ${e.origin}<br>  
    Текст: ${e.data}<br>`;  
    if (e.data.indexOf("iframe") > -1)  
      postMessage("Сообщение из frame", '*');  
  });  
}
```



Рабочие процессы / Worker, SharedWorker

Запуск скриптов в фоновом потоке

- Объект **Worker**
 - **postMessage(message)**
 - отправка сообщения
 - **terminate()**
 - прерывание работы **Worker**
 - событие **'message'**
 - событие **'error'**
- Экземпляр **Worker** создаётся на основании внешнего **js**-файла
- Предусмотрен импорт
 - **importScripts(file)**
 - Указывается в начале кода рабочего процесса

<https://developer.mozilla.org/ru/docs/Web/API/SharedWorker>

<https://habr.com/ru/post/261307/>

Получение сообщений от worker

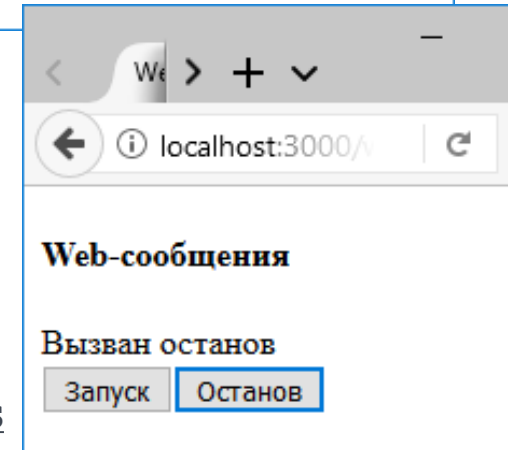
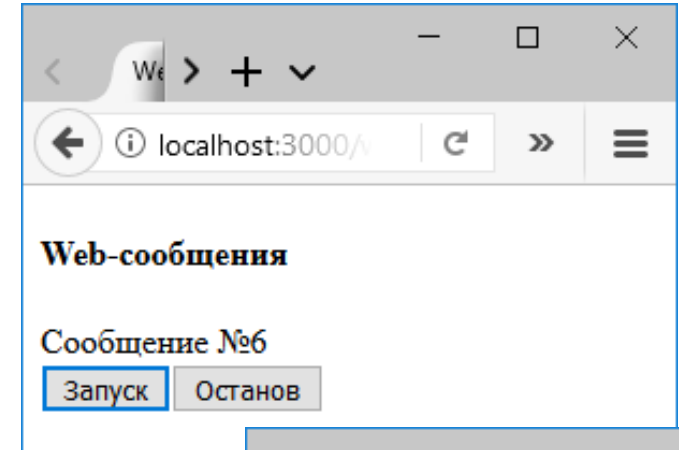
53

worker1.pug

```
html
  head
    meta(charset="utf-8")
    script(src="/public/workerShow.js")
  body
    h4 Web-сообщения
    dev#show
    br
    button(onclick="startWorker(show)") Запуск
    button(onclick="stopWorker(show)") Останов
```

workerShow.js

```
let w;
function startWorker(show) {
  if(typeof (Worker) !== undefined) {
    w = new Worker("/public/myworker.js");
    w.onmessage = (e) => {
      show.innerHTML = e.data;
    };
  } else
    show.innerHTML = "Не поддерживается";
}
function stopWorker(show) {
  if(w) {
    show.innerHTML = "Вызван останов";
    w.terminate();
  }
}
```



myworker.js

```
let idx = 0;
function timedCount() {
  postMessage(`Сообщение №${++idx}`);
  setTimeout(timedCount, 500);
}
timedCount();
```

Передача сообщений в worker

54

myworker2.js

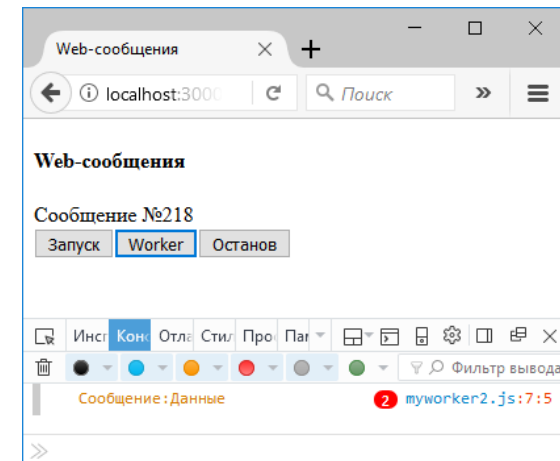
```
let idx = 0;
function timedCount() {
  postMessage(`Сообщение №${++idx}`);
  setTimeout(timedCount, 500);
}
addEventListener("message", (e)=>{
  console.log(`Сообщение:${e.data}`);
  idx += 100;
});
timedCount();
```

worker2.pug

```
doctype html
html
  head
    meta(charset="utf-8")
    script(src="/public/workerShow2.js")
    title Web-сообщения
  body
    h4 Web-сообщения
    dev#show
    br
    button(onclick="startWorker(show)") Запуск
    button(onclick="tellWorker()") Worker
    button(onclick="stopWorker(show)") Останов
```

workerShow2.js

```
let w;
function startWorker(show) {
  if(typeof (Worker) !== undefined) {
    w = new Worker("/public/myworker2.js");
    w.onmessage = (e) => {
      show.innerHTML = e.data;
    };
  } else
    show.innerHTML = "Не поддерживается";
}
function stopWorker(show) {
  if(w) {
    show.innerHTML = "Вызван останов";
    w.terminate();
  }
}
function tellWorker(show) {
  if(w)
    w.postMessage("Данные");
}
```



Web-сокеты / ws, connection, WebSocket

55

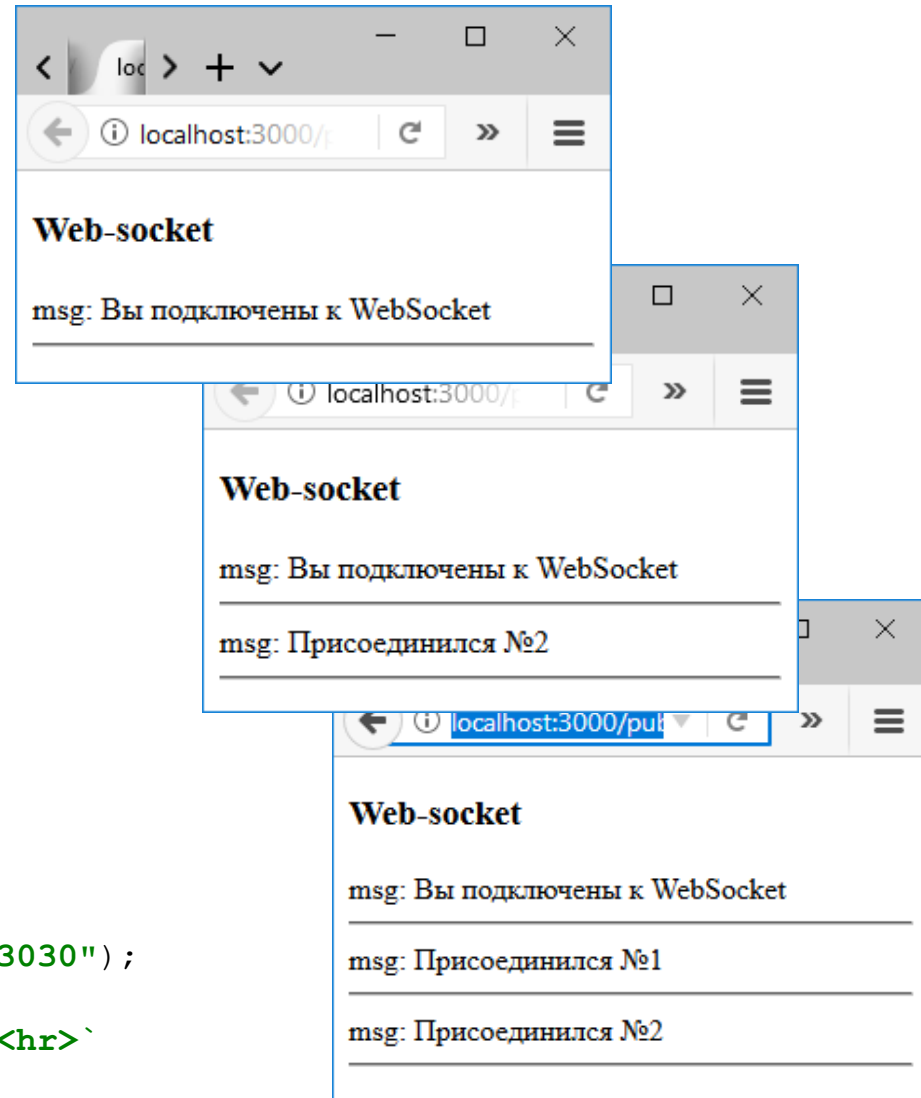
wsServer1.js

```
let WS = require("ws");
let wss = new WS.Server({port: 3030});
let clients = [];
wss.on("connection", (ws)=>{
  clients.push(ws);
  let idx = clients.indexOf(ws);
  console.log(`Новое соединение ${idx}`);
  ws.send("Вы подключены к WebSocket");
  for(w of clients)
    if(w !== ws)
      w.send(`Присоединился №${idx}`)
});
```

wsClient1.js

```
<html lang="en">
<head>
  <meta charset="UTF-8">
</head>
<body>
  <h3>Web-socket</h3>
  <div id="data"></div>
  <script>
    let ws = new WebSocket("ws://localhost:3030");
    ws.onmessage = (e)=>{
      data.innerHTML += `msg: ${e.data}<hr>`
    };
  </script>
</body>
</html>
```

npm install --save ws



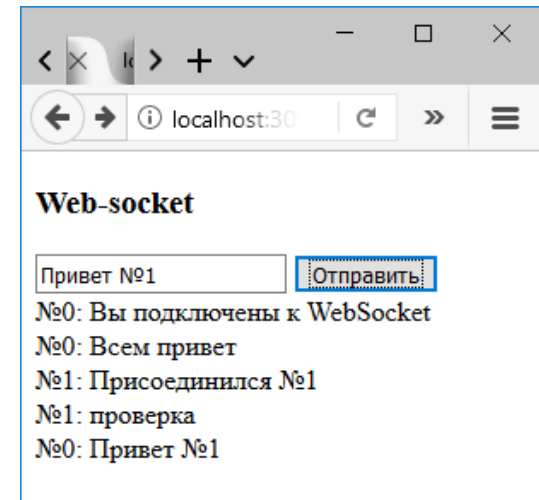
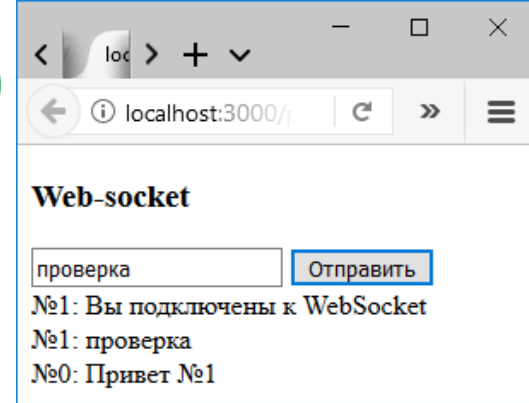
Web-сокеты / on("message")

wsServer2.js

```
let Server = require('ws').Server;
let wss = new Server({port: 3030});
let clients = [];
wss.on("connection", (ws)=>{
  clients.push(ws);
  let idx = clients.indexOf(ws);
  ws.send(JSON.stringify({type: "info", msg: "Вы подключены к WebSocket", id: idx}));
  for(w of clients)
    if(w !== ws)
      w.send(JSON.stringify({type: "info", msg: `Присоединился №${idx}`, id: idx}));
  ws.on("message", (msg)=>{
    for(w of clients)
      w.send(JSON.stringify({type: "msg", msg: msg, id: idx}));
  });
});
```

wsClient2.js

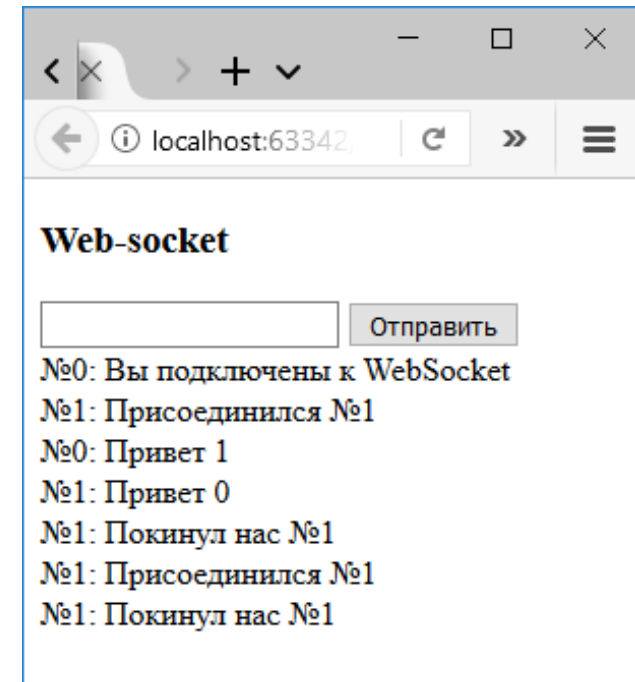
```
<input id="message">
<button onclick="send(message)">Отправить</button>
<div id="data"></div>
<script>
  var ws = new WebSocket("ws://localhost:3030");
  ws.onmessage = (e)=>{
    let message = JSON.parse(e.data);
    let elem = document.createElement("div");
    elem.appendChild(document.createTextNode(`№${message.id}: ${message.msg}`));
    data.appendChild(elem);
  };
  function send(message) {
    ws.send(message.value);
  }
</script>
```



Web-сокеты / on("close") – disconnect 57

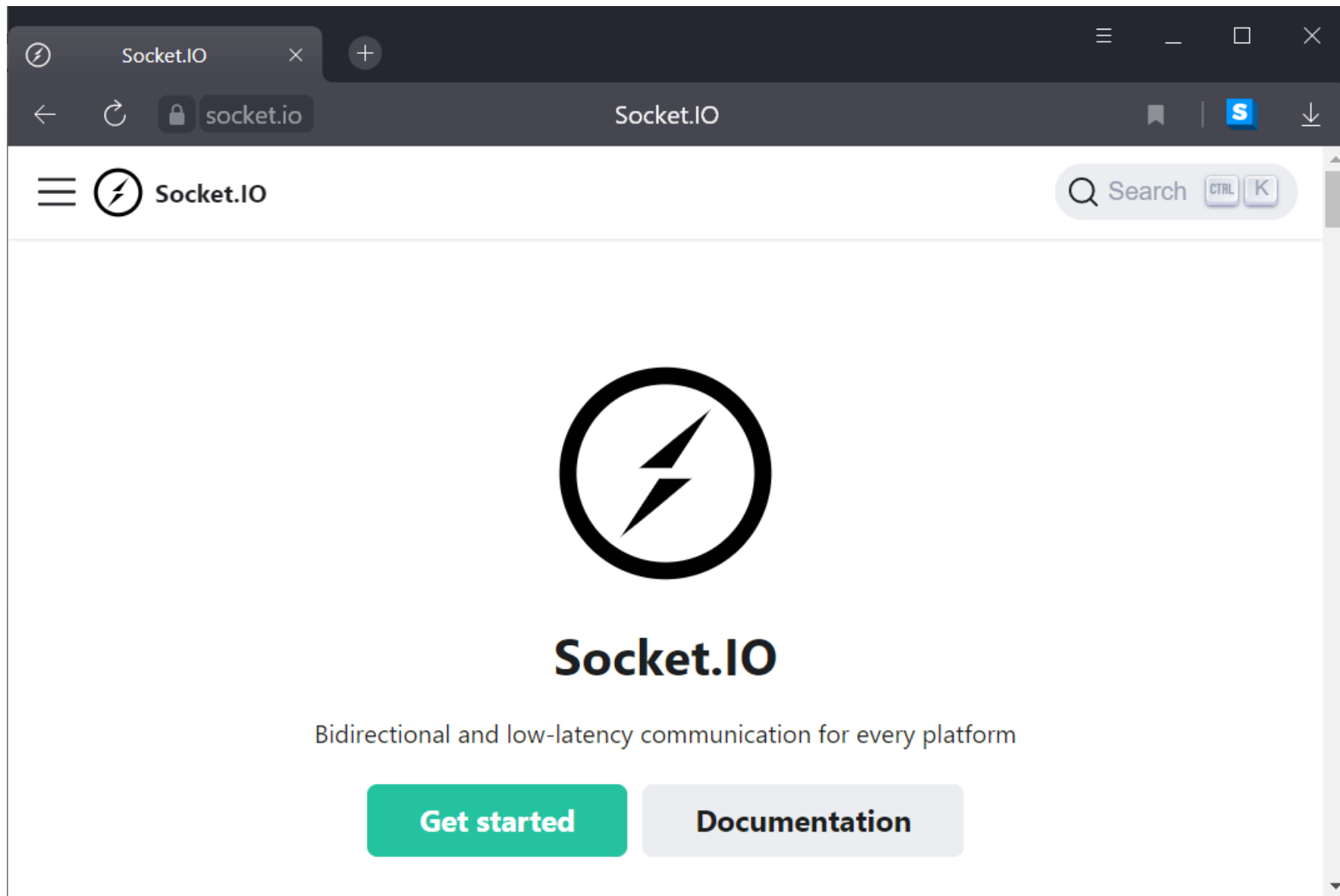
wsServer25.js

```
let Server = require('ws').Server;
let wss = new Server({port: 3030});
let clients = [];
wss.on("connection", (ws)=>{
  clients.push(ws);
  let idx = clients.indexOf(ws);
  ws.send(JSON.stringify(
    {type: "info", msg: "Вы подключены к WebSocket", id: idx}))
  );
  for(w of clients)
    if(w !== ws)
      w.send(JSON.stringify(
        {type: "info", msg: `Присоединился №${idx}`, id: idx}))
  );
  ws.on("message", (msg)=>{
    for(w of clients)
      w.send(JSON.stringify(
        {type: "msg", msg: msg, id: idx}))
  );
});
ws.on("close", ()=>{
  idx = clients.indexOf(ws);
  clients.splice(idx, 1);
  for(w of clients)
    w.send(JSON.stringify(
      {type: "msg", msg: `Покинул нас №${idx}`, id: idx}))
  );
});
});
```



Web-сокеты на основе Socket.io

58



<https://socket.io/>

Socket.IO. Поддержка Web-сокетов

59

- WebSocket
- Server-Sent Events
- XHR long polling
- XHR multipart streaming
- Forever Iframe
- JSONP Polling
- ActiveX HTMLFile
- Надёжность
 - поддержка прокси, балансировщиков нагрузки, фаерволов, антивирусов
- Поддержка восстановления соединения
- Обнаружение разъединения
- Поддержка бинарных данных
- Простое и удобное API
- Кроссбраузерность
- Поддержка нескольких пространств имён
- Создание отдельных каналов в рамках пространств имён

Socket.IO. Стандартные события

60

Сервер

- connection
- connect
- disconnecting
- disconnect
- new_namespace
- ...

Клиент

- connecting
- connect
- message
- disconnect
- reconnect
- error
- ping
- ...

Варианты подключения клиента

61

- С сервера

```
<script src="/socket.io/socket.io.js"></script>
```

- ESM-импорт с CDM

```
<script type="module">  
  import { io } from "https://cdn.socket.io/4.3.2/socket.io.esm.min.js";  
</script>
```

- Из NPM-модуля

```
import { io } from "socket.io-client";
```

Socket.IO. Соединение / connection

62

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <script src="/socket.io/socket.io.js"></script>
  <script>
    const socket = io();
    socket.on("connect", () => {
      data.innerHTML +=
        `<p>connect: ${socket.id}</p>`;
    });
    socket.on("disconnect", () => {
      data.innerHTML +=
        `<p>disconnect: ${socket.id}</p>`;
    });
    socket.on("hello", (msg)=>{
      data.innerHTML += `<p>msg: ${msg}</p>`;
    });
  </script>
</head>
<body>
<h3>Socket.IO</h3><div id="data"></div>
</body>
</html>
```

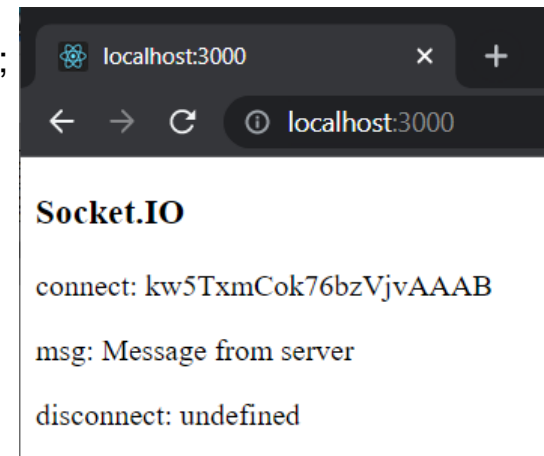
server.js

```
import { createServer } from "http";
import { Server } from "socket.io";
import fs from "fs/promises"

const httpServer = createServer(function (req, res) {
  fs.readFile("./index.html")
    .then(html => {
      res.setHeader("Content-Type", "text/html");
      res.writeHead(200);
      res.end(html);
    })
});

const io = new Server(httpServer, { /* options */ });
io.on("connection", (socket) => {
  socket.emit("hello", "Message from server")
  console.log(socket)
});
httpServer.listen(3000);
```

1. Добавьте переменную среды:
DEBUG=socket.io*
2. Посмотрите лог сервера



Socket.IO. Чат. Сервер

63

```
import { createServer } from "http";           server.js
import { Server } from "socket.io";
import fs from "fs/promises"

const httpServer = createServer(function (req, res) {
  fs.readFile("./index.html")
    .then(html => {
      res.setHeader("Content-Type", "text/html");
      res.writeHead(200);
      res.end(html);                                // Отправить "index.html" клиенту
    })
});

const io = new Server(httpServer, {});

io.on("connection", (socket) => {                // При подключении клиента
  socket.on('conn', (msg)=>{                     // Сообщение "conn"
    let time = (new Date()).toLocaleTimeString();
    console.log("connect:", socket.name = msg.name) // Сохранение имени
    socket.emit("msg", {"message": `${time} Привет ${socket.name}!`});
    socket.broadcast.emit("msg", {"message": `${time} Вошёл ${socket.name}!`});
  });

  socket.on('msg', (msg)=>{                       // Сообщение "msg"
    let time = (new Date()).toLocaleTimeString();
    console.log(msg = `${time} ${socket.name}: ${msg.value}`) // Сообщение
    socket.emit("msg", {"message": msg});          // Отправка "обратно"
    socket.broadcast.emit("msg", {"message": msg}); // Отправка всем
  });

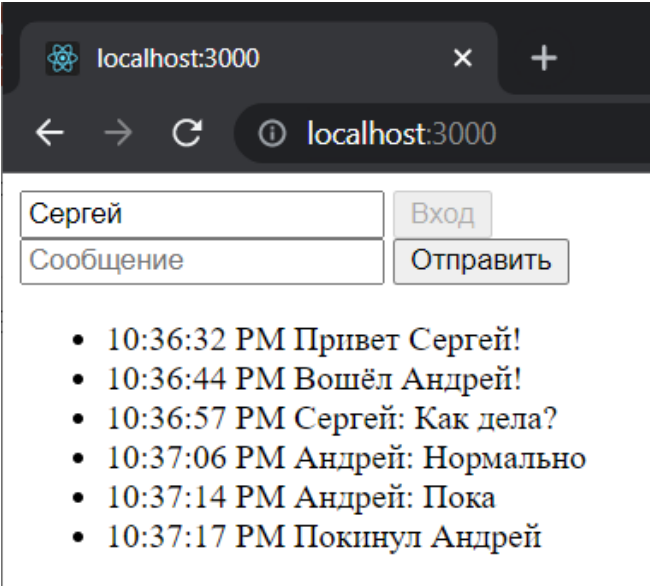
  socket.on('disconnect', (msg)=>{                // Сообщение "disconnect"
    let time = (new Date()).toLocaleTimeString();
    socket.broadcast.emit("msg", {"message": `${time} Покинул ${socket.name}`});
  });
});

httpServer.listen(3000);
```

Socket.IO. Чат. Клиент

64

```
<!DOCTYPE html>      index.html
<html lang="ru">
<head><meta charset="UTF-8">
  <script src="/socket.io/socket.io.js"></script>
  <script>
    let socket, nick
    function chat() {
      socket = io();
      nick = document.getElementById("name").value
      socket.on("connect", () => { socket.emit("conn", {"name": nick}); });
      socket.on("msg", (msg) => { addUL(msg.message); });
      document.getElementById("send").disabled = false;
      document.getElementById("login").disabled = true;
    }
    function addUL(msg) {
      const li = document.createElement("li")
      data.appendChild((li.innerHTML = msg, li))
    }
    const send = (msg) => socket && socket.emit("msg", {"name": nick, "value": msg})
  </script>
</head>
<body>
  <input id="name" placeholder="Введите имя">
  <button id="login" onclick="chat()">Вход</button><br>
  <input id="msg" placeholder="Сообщение">
  <button id="send" disabled onclick="send(msg.value)">Отправить</button><br>
  <ul id="data"></ul>
</body>
</body>
</html>
```



localhost:3000

Сергей

Сообщение

Вход

Отправить

- 10:36:32 PM Привет Сергей!
- 10:36:44 PM Вошёл Андрей!
- 10:36:57 PM Сергей: Как дела?
- 10:37:06 PM Андрей: Нормально
- 10:37:14 PM Андрей: Пока
- 10:37:17 PM Покинул Андрей

«Летучие» сообщения:
socket.volatile.emit("msg", {"message": msg})
Могут быть не доставлены клиенту

Socket.IO. Поддержка namespaces

65

На сервере

```
io.of("/orders").on("connection", (socket) => {  
  socket.on("order:list", () => {});  
  socket.on("order:create", () => {});  
});
```

```
io.of("/users").on("connection", (socket) => {  
  socket.on("user:list", () => {});  
});
```

На клиенте

```
const socket = io(); // or io("/"), the main namespace  
const orderSocket = io("/orders"); // the "orders" namespace  
const userSocket = io("/users"); // the "users" namespace
```

История (обычная)

// переместиться назад по истории

history.back()

// переместиться вперёд по истории

history.forward()

// переместиться на заданное число шагов

history.go(steps)

// количество записей в истории

history.length



https://developer.mozilla.org/ru/docs/Web/API/History_API

История (новые методы)

67

// создать новую запись в истории

`pushState(state, title, url)`

// Здесь

// state - новое состояние

// title - название для пользователя

// url - фиктивный url

// заменить запись в истории

`replaceState(state, title, url)`

// проверить объект истории

`history.state`

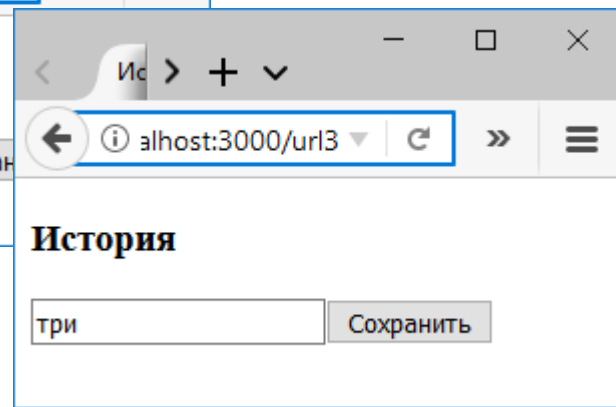
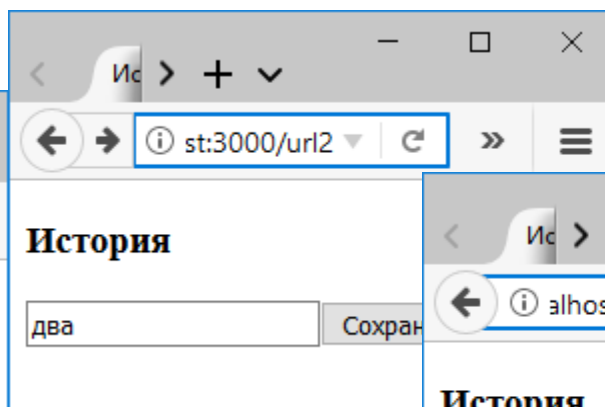
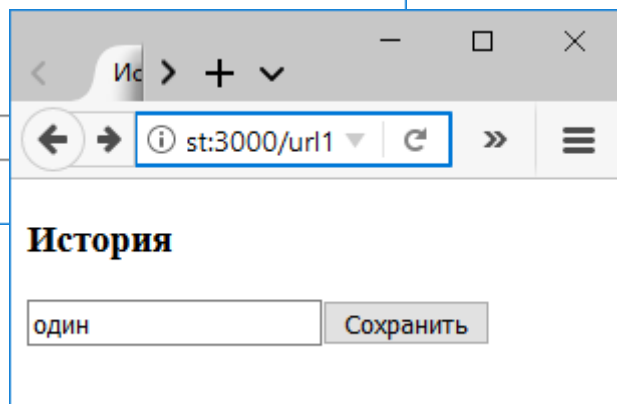
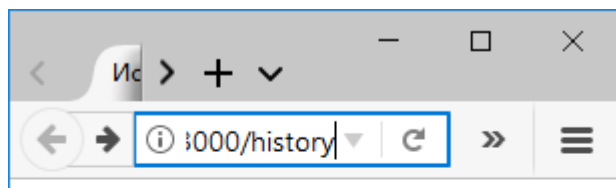
История (пример)

history.pug

```
doctype html
html
  head
    meta(charset="utf-8")
    script(src="/public/history.js")
    title История
  body(onload="init(input)")
    h3 История
    input#input
    button(onclick="store(input)") Сохранить
```

history.js

```
var idx = 0;
function init(input) {
  addEventListener("popstate", (e)=>{
    if(e.state)
      input.value = e.state.myvalue;
    else
      input.value = "";
  }, false);
}
function store(input) {
  let value = input.value;
  history.pushState({myvalue: value},
    'Заголовок', ` /url${++idx}`);
}
```



sessionStorage

local.js

69

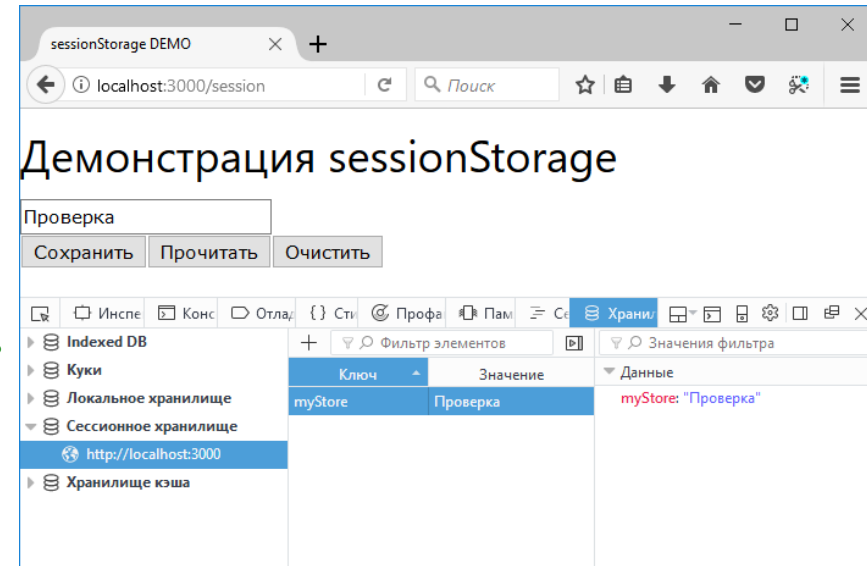
sessionStorage.pug

```
doctype html
html
  head
    meta(charset="utf-8")
    link(href='/public/w3.css', rel='stylesheet')
    script(src="/public/session.js")
    title sessionStorage DEMO
  body
    h1 Демонстрация sessionStorage
    input#source
    br
    button(onclick="store(source)") Сохранить
    button(onclick="read(source)") Прочитать
    button(onclick="source.value=''") Очистить
```

```
function store(source) {
  sessionStorage.setItem("myStore", source.value);
}
function read(source) {
  let value = sessionStorage.getItem("myStore");
  source.value = value;
}
```

Альтернативная запись session.js

```
function store(source) {
  sessionStorage["myStore"] = source.value;
}
function read(source) {
  source.value = sessionStorage["myStore"];
}
```



localStorage

local.js

70

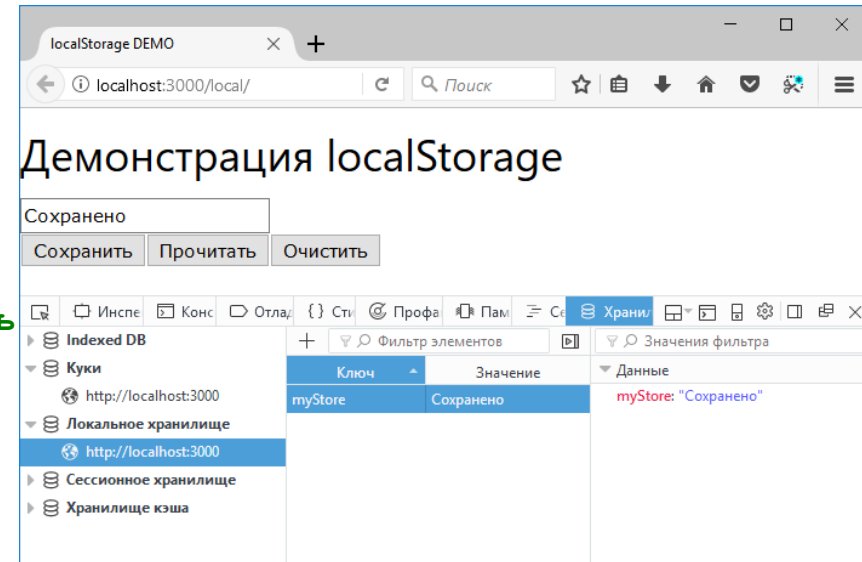
localStorage.pug

```
doctype html
html
  head
    meta(charset="utf-8")
    link(href='/public/w3.css', rel='stylesheet')
    script(src="/public/local.js")
    title localStorage DEMO
  body
    h1 Демонстрация localStorage
    input#source
    br
    button(onclick="store(source)") Сохранить
    button(onclick="read(source)") Прочитать
    button(onclick="source.value=''") Очистить
```

Альтернативная запись local.js

```
function store(source) {
  localStorage["myStore"] = source.value;
}
function read(source) {
  source.value = localStorage["myStore"];
}
```

```
function store(source) {
  localStorage.setItem("myStore", source.value);
}
function read(source) {
  let value = localStorage.getItem("myStore");
  source.value = value;
}
```



Слушатель localStorage / "storage"

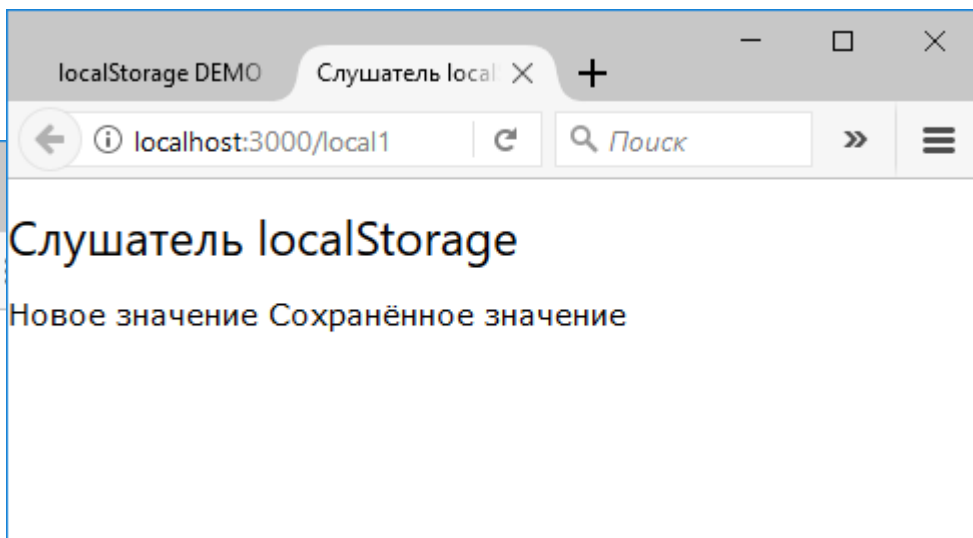
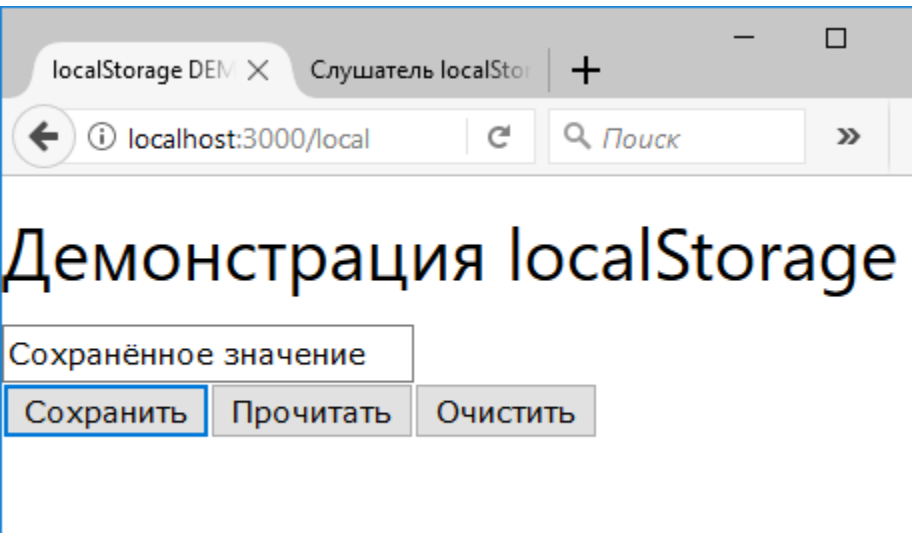
71

local1.js

```
function init() {  
  addEventListener("storage", show);  
}  
function show() {  
  let div = document.getElementById("show");  
  div.innerHTML = `Новое значение ${localStorage["myStore"]}`;  
}
```

localStorage1.pug

```
doctype html  
html  
  head  
    meta(charset="utf-8")  
    link(href="/public/w3.css", rel="stylesheet")  
    script(src="/public/local1.js")  
    title Слушатель localStorage  
  body(onload="init()")  
    h3 Слушатель localStorage  
    div#show
```



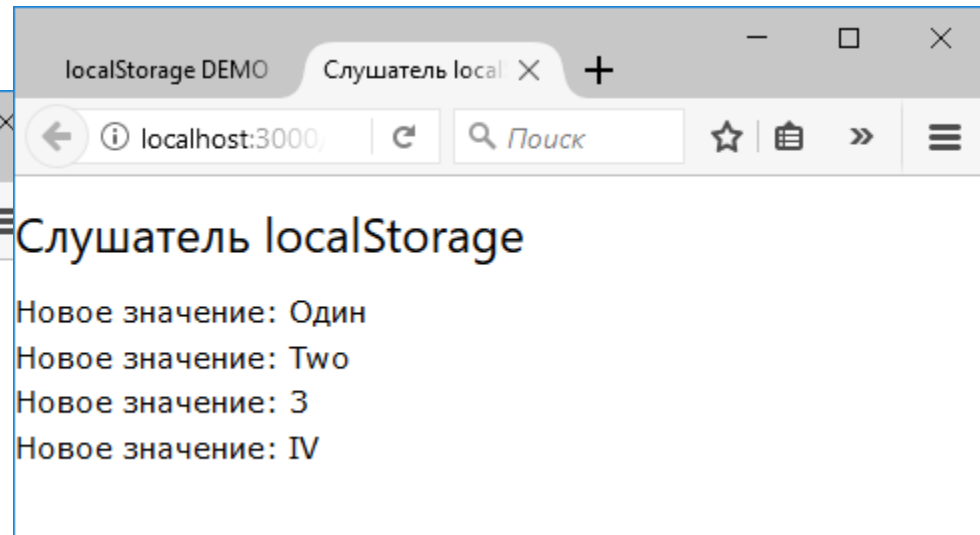
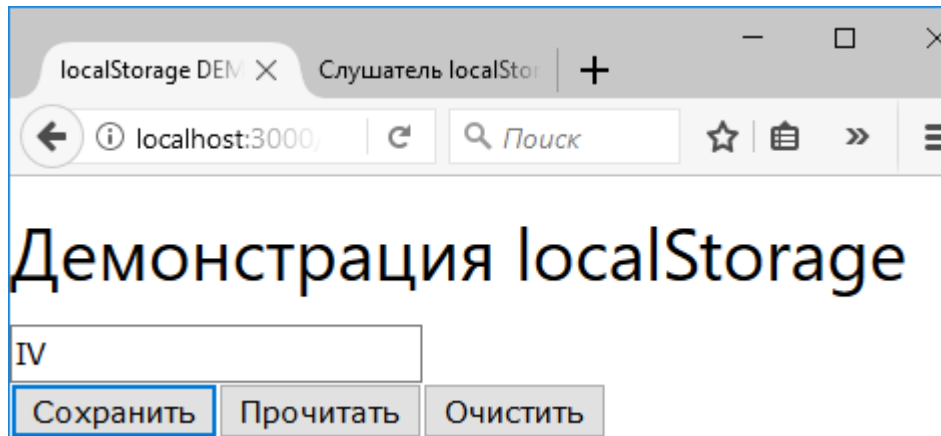
Слушатель localStorage / length, (key),⁷² [key]

local2.js

```
var showDiv;  
function init(div) {  
    showDiv = div; // Сохранение в глобальную переменную  
    addEventListener("storage", show);  
}  
function show() {  
    for(let i = 0; i < localStorage.length; i++) {  
        let key = localStorage.key(i);  
        showDiv.innerHTML+=`Новое значение: ${localStorage[key]}<br>`;  
    }  
}
```

localStorage2.pug

```
doctype html  
html  
  head  
    meta(charset="utf-8")  
    link(href='/public/w3.css', rel='stylesheet')  
    script(src="/public/local2.js")  
    title Слушатель localStorage  
  body(onload="init(msg)")  
    h3 Слушатель localStorage  
    div#msg
```

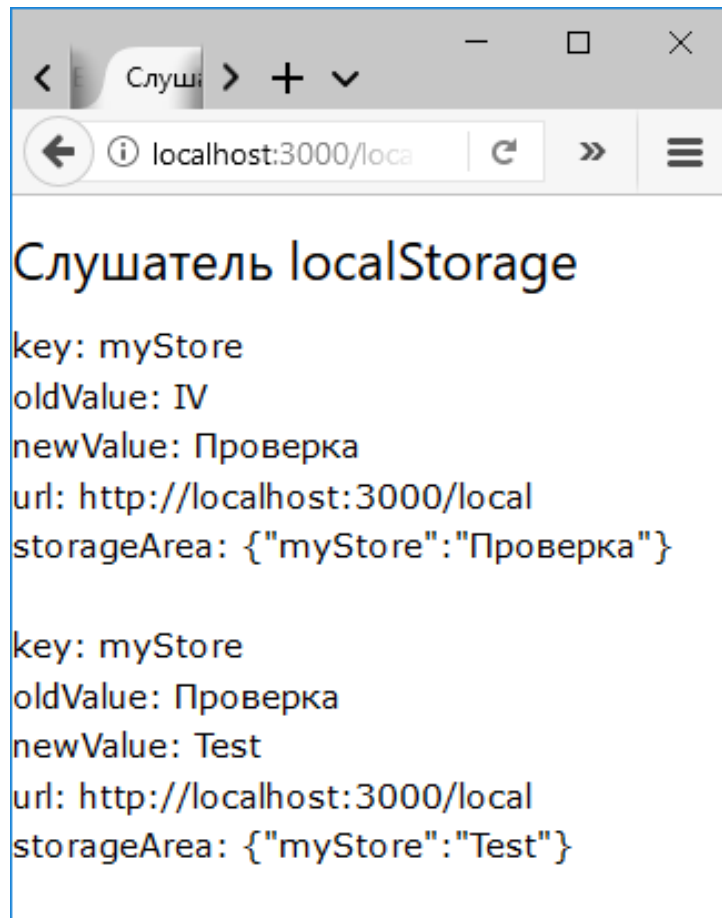


Слушатель localStorage / event

73

local3.js

```
var showDiv;  
function init(div) {  
    showDiv = div;  
    addEventListener("storage", show);  
}  
function show(event) {  
    showDiv.innerHTML += `  
key:      ${event.key}<br>  
oldValue: ${event.oldValue}<br>  
newValue: ${event.newValue}<br>  
url:      ${event.url}<br>  
storageArea: ${JSON.stringify(event.storageArea)}  
<br><br>`;  
}
```



Cookie в браузере / document.cookie

74

```
doctype html
html
  head
    meta(charset="utf-8")
    script(src="/public/cookie1.js")
    title Cookie DEMO
  body(onload="show(msg)")
    h3 Демонстрация Cookie
    div#msg
```

```
function show(msg) {
  document.cookie = "name=serge";
}
```

The screenshot shows a web browser window with the title "Cookie DEMO". The address bar displays "localhost:3000/cookie". The page content shows the heading "Демонстрация Cookie". Below the page content, the browser's developer tools are open, specifically the "Хранилища" (Storage) tab. The "Куки" (Cookies) section is expanded, showing a table of cookies for the URL "http://localhost:3000".

Имя	Домен	Путь	Данные
name	localhost	/	name: "serge" CreationTime: "Sun, 08 Oct 2017 09:31:49 GMT" Domain: "localhost" Expires: "Сессионная" HostOnly: true HttpOnly: false LastAccessed: "Sun, 08 Oct 2017 09:31:49 GMT" Path: "/" Secure: false

Cookie в браузере / expires – текст

75

```
function show(msg) {  
    document.cookie = "name=serge; expires=Thu, 07 Nov 2017 12:00:00 UTC";  
}
```

Cookie DEMO

localhost:3000/cookie

Демонстрация Cookie

Инспект Консо Отладч Стил Профайл Памя Сет **Хранили**

Indexed DB

Куки

Имя	Домен	Пу	Данные
name	localhost	/	<p>name: "serge"</p> <p>CreationTime: "Sun, 08 Oct 2017 09:31:49 GMT"</p> <p>Domain: "localhost"</p> <p>Expires: "Tue, 07 Nov 2017 12:00:00 GMT"</p> <p>HostOnly: true</p> <p>HttpOnly: false</p> <p>LastAccessed: "Sun, 08 Oct 2017 09:34:46 GMT"</p> <p>Path: "/"</p> <p>Secure: false</p>

Локальное хранилище

Сессионное хранилище

Хранилище кэша

Cookie в браузере / expires – переменная

```
function show(msg) {  
    let exp = new Date();  
    exp.setMinutes(exp.getMinutes() + 10);  
    document.cookie = `name=serge; expires=${exp.toUTCString()}`;  
}
```

Cookie DEMO

localhost:3000/cookie

Демонстрация Cookie

Инспект Консо Отладч Стил Профайл Памя Сет **Хранили**

Indexed DB

Куки

Имя	Домен	Путь
name	localhost	/

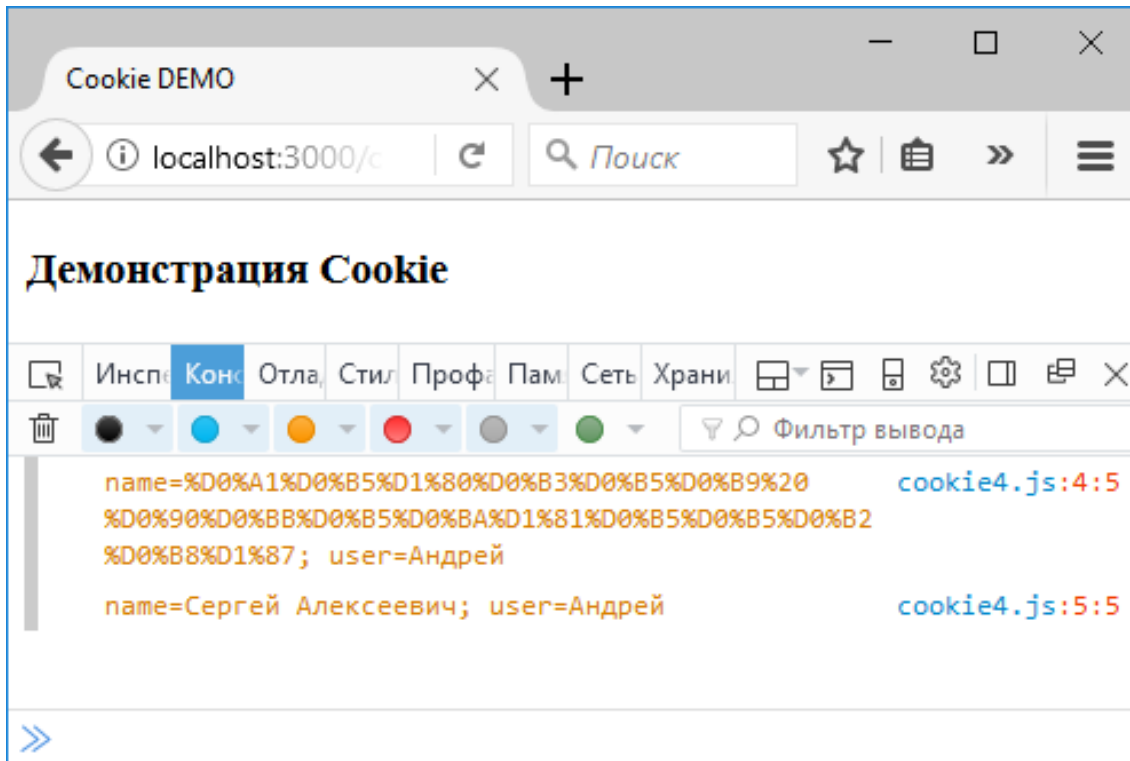
Данные

- name: "serge"
- CreationTime: "Sun, 08 Oct 2017 09:43:06 GMT"
- Domain: "localhost"
- Expires: "Sun, 08 Oct 2017 09:55:46 GMT"
- HostOnly: true
- HttpOnly: false
- LastAccessed: "Sun, 08 Oct 2017 09:45:46 GMT"
- Path: "/"
- Secure: false

Cookie в браузере / русский текст

77

```
function show(msg) {  
    let expDays = 30;  
    let exp = new Date();  
    exp.setTime(exp.getTime() + (expDays*24*60*60*1000));  
    let expires = `expires=${exp.toUTCString()}`;   
    document.cookie = `name=${encodeURIComponent("Сергей Алексеевич")};${expires}`;  
    document.cookie = `user=Андрей`;  
    console.log(document.cookie);  
    console.log(decodeURIComponent(document.cookie));  
}
```



Загрузка файла на сервер

server.js

```
import express from 'express'
import fileUpload from 'express-fileupload'
import fs from "fs/promises"
const app = express();
app.use(fileUpload());
app.get('/', function(req, res) {
  fs.readFile("./index.html")
    .then(html => {
      res.setHeader("Content-Type", "text/html");
      res.writeHead(200);
      res.end(html);
    })
})
app.post('/upload', function(req, res) {
  if (!req.files || Object.keys(req.files).length === 0)
    return res.status(400).send('No files were uploaded.');
```

const fileName = req.files.fileName;

const uploadPath = 'D:\\serge\\download\\upload\\' + fileName.name;

fileName.mv(uploadPath, **function**(err) {

if (err) **return** res.status(500).send(err);

 res.send('File uploaded!');

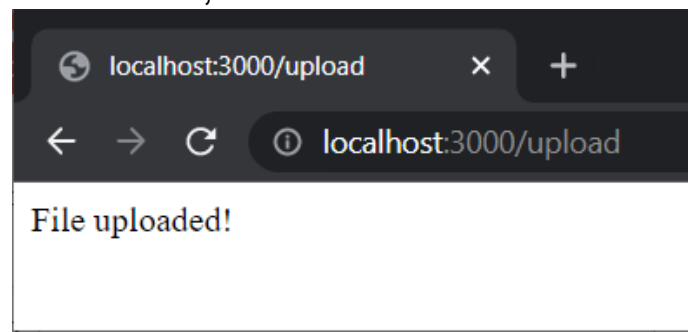
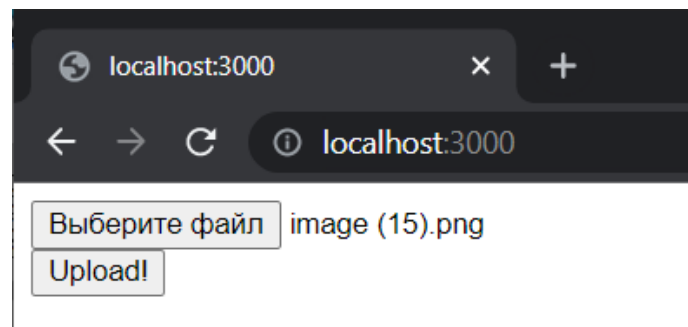
});

});

app.listen(3000)

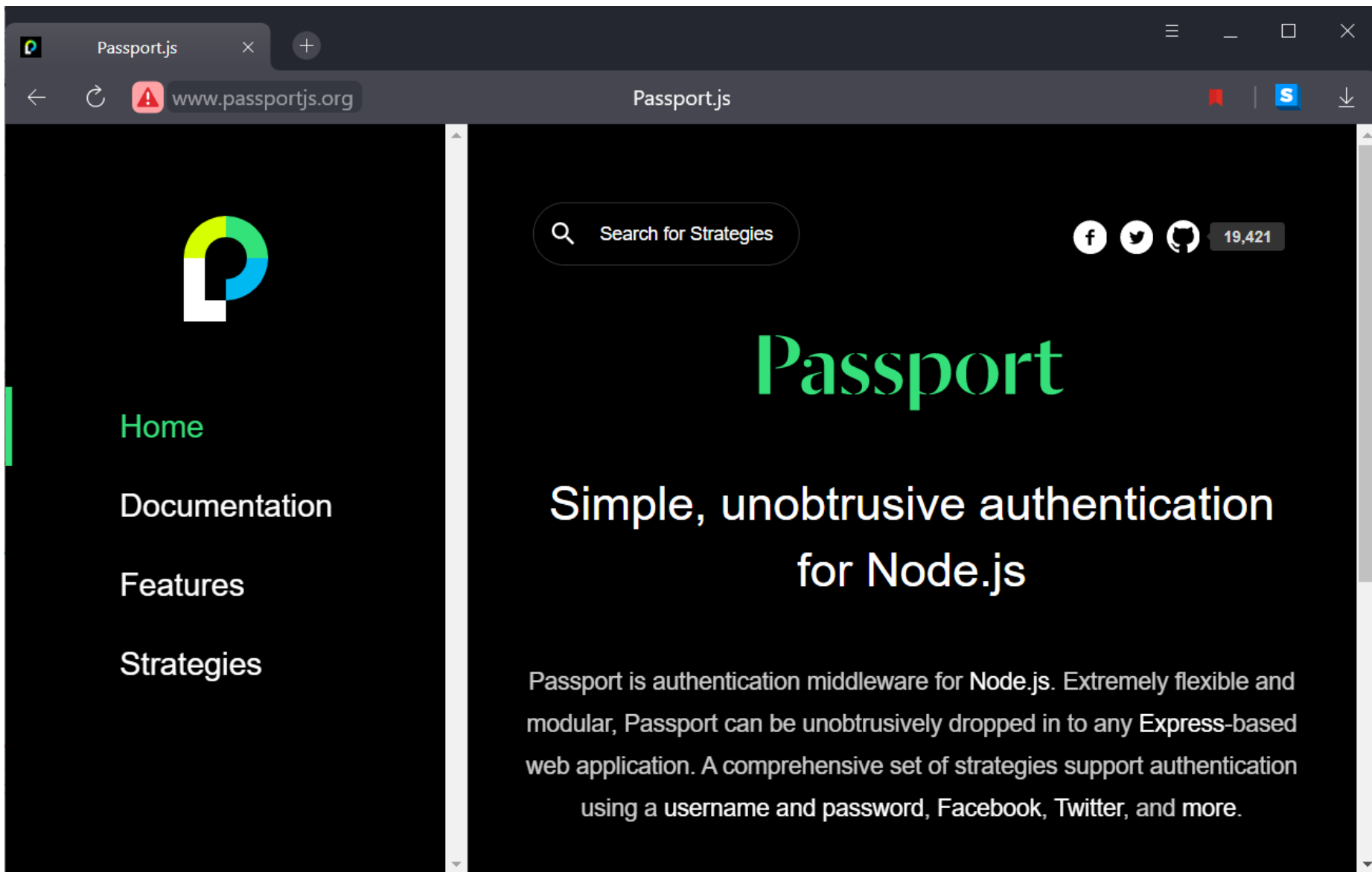
index.html

```
<html>
<body>
<form ref="fileName"
  action='http://localhost:3000/upload'
  method='post'
  enctype="multipart/form-data">
  <input type="file" name="fileName" /><br>
  <input type='submit' value='Upload!' />
</form>
</body>
</html>
```



PassportJS – авторизация

79



<http://www.passportjs.org/>

Пример приложения – HTML-страницы

login.pug

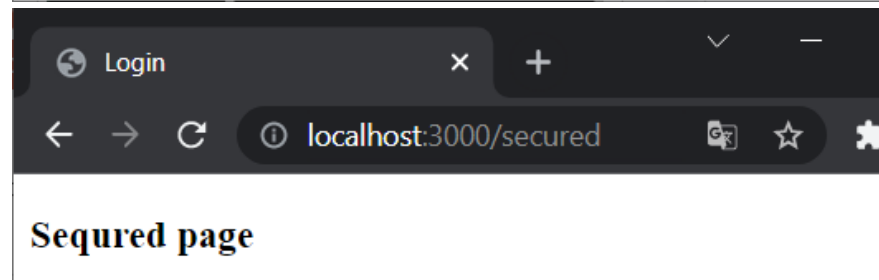
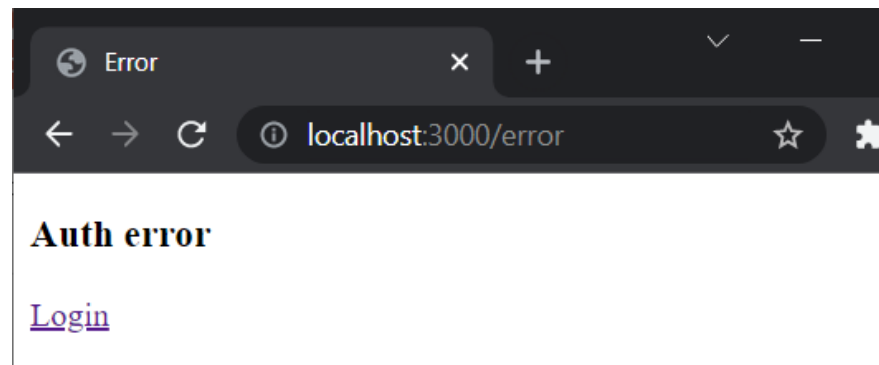
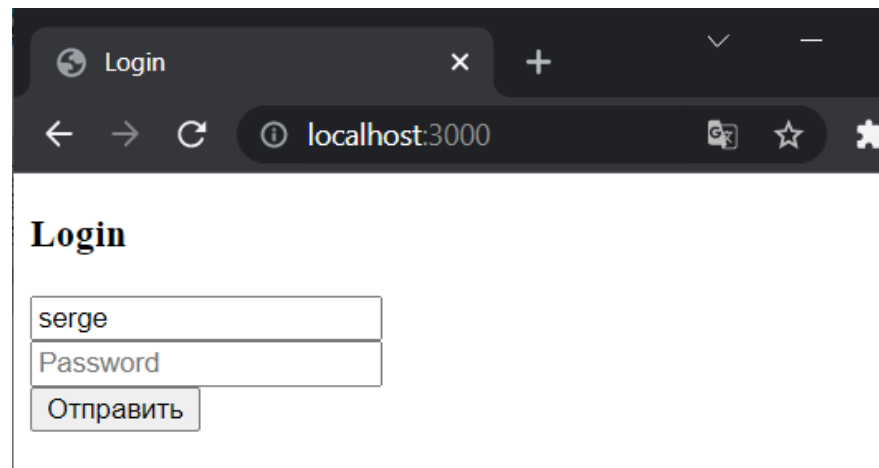
```
doctype html
html
  head
    meta(charset="utf-8")
    title Login
  body
    h3 Login
    form(method="post" action="/login")
      input(placeholder="Name" type="text" name="username" value=user)
      br
      input(placeholder="Password" type="password" name="password")
      br
      input(type="submit")
```

error.pug

```
doctype html
html
  head
    title Error
  body
    h3 Auth error
    a(href="/") Login
```

secured.pug

```
doctype html
html
  head
    title Login
  body
    h3 Secured page
```



Пример приложения – index.js (1)

81

`npm i -D passport passport-local express express-session`

// Загрузка библиотек

```
const express = require("express");  
const passport = require("passport")  
const LocalStrategy = require("passport-local").Strategy  
const session = require("express-session");
```

// Настройка сервера

```
const server = express();  
server.use(express.json());  
server.use(express.urlencoded({ extended: false }));  
server.set("view engine", "pug");  
server.set("views", "./views");  
server.use(session({ secret: "cat", resave: false, saveUninitialized: false }));
```

// Настройка passportjs

```
passport.use(new LocalStrategy(  
  function(username, password, done) {  
    findUser({ username: username }, function (err, dbUser) {  
      if (err) return done(err)  
      if (!dbUser) return done(null, false)  
      if (!dbUser.verifyPassword(password)) return done(null, false)  
      return done(null, dbUser)  
    })  
  })  
)  
passport.serializeUser(function(user, done) { done(null, user) })  
passport.deserializeUser(function(user, done) { done(null, user) })  
server.use(passport.initialize())  
server.use(passport.session())
```

Пример приложения – index.js (2)

82

// Проверка пользователя

```
const dbUser = {username: "serge", verifyPassword: (password) => password === "123"}
function findUser(loginUser, callback) {
  if (loginUser.username === dbUser.username)
    return callback(null, dbUser)
  return callback(null)
}
```

// Стартовая страница

```
server.get("/", (req, res) => {
  res.render("login", { user: "serge" })
})
```

// Проверка пароля

```
server.post("/login", passport.authenticate("local",
  { successRedirect: "/secured", failureRedirect: "/error", failureMessage: true }));
```

// Функция проверки аутентификации на основе passportjs

```
function authenticationMiddleware() {
  return function (req, res, next) {
    if (req.isAuthenticated()) return next()
    res.redirect('/error')
  }
}
```

// Защищённая страница

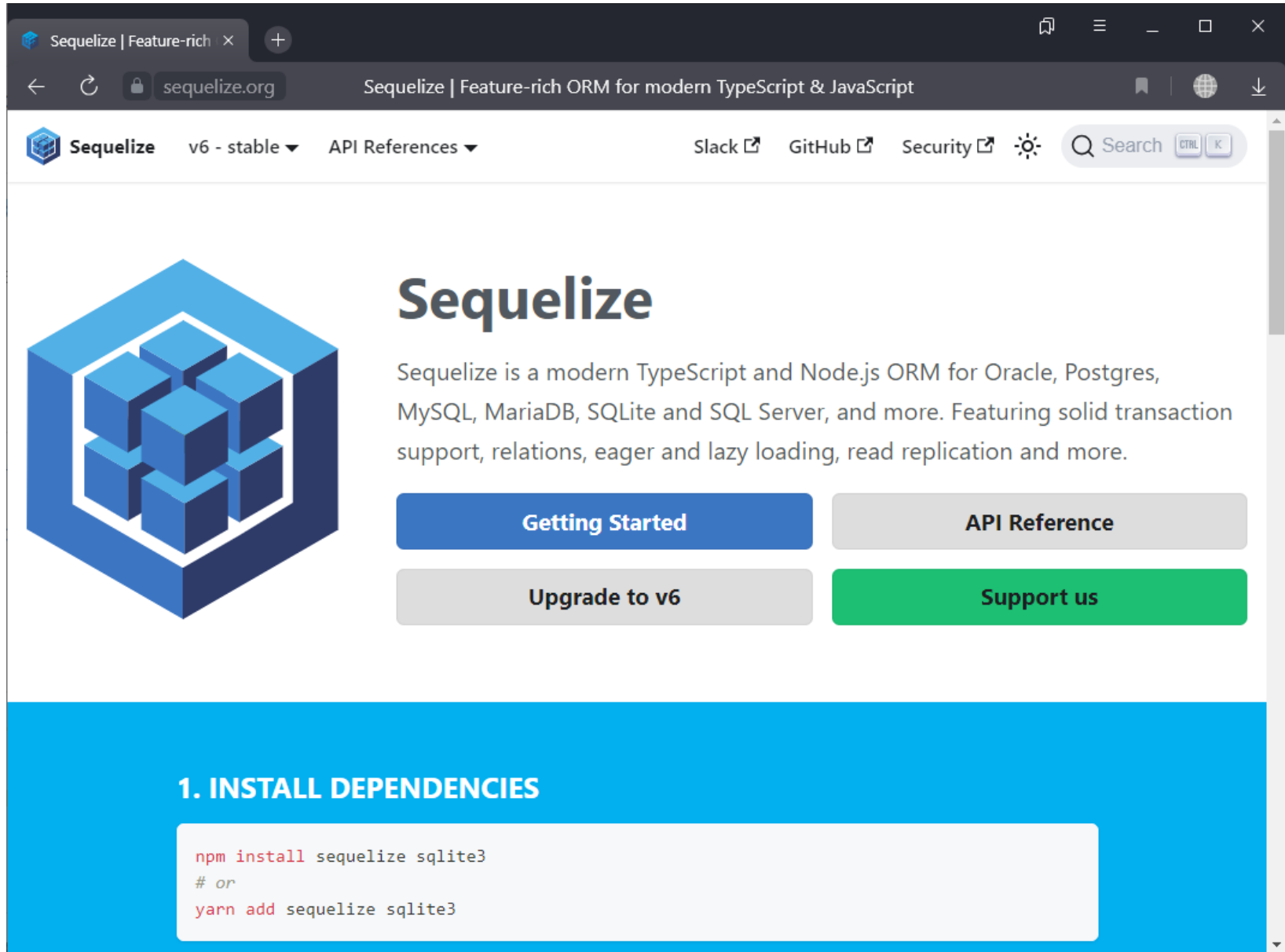
```
server.get("/secured", authenticationMiddleware(), (req, res) => { res.render("secured", {}) })
```

// Страница ошибки

```
server.get("/error", (req, res) => { res.render("error", {}) })
server.listen(3000)
```

Object Relational Mapping

83

A screenshot of the Sequelize website in a web browser. The browser's address bar shows 'sequelize.org'. The website header includes the Sequelize logo, version 'v6 - stable', and links to 'API References', 'Slack', 'GitHub', and 'Security'. A search bar is also present. The main content area features a large blue isometric cube logo on the left. To its right, the word 'Sequelize' is displayed in a large, bold font. Below the name, a paragraph describes it as a modern TypeScript and Node.js ORM for various databases. Four buttons are arranged in a 2x2 grid: 'Getting Started' (blue), 'API Reference' (gray), 'Upgrade to v6' (gray), and 'Support us' (green). At the bottom, a blue banner contains the heading '1. INSTALL DEPENDENCIES' and a code block with installation commands for npm and yarn.

Sequelize | Feature-rich ORM for modern TypeScript & JavaScript

Sequelize v6 - stable API References Slack GitHub Security Search

Sequelize

Sequelize is a modern TypeScript and Node.js ORM for Oracle, Postgres, MySQL, MariaDB, SQLite and SQL Server, and more. Featuring solid transaction support, relations, eager and lazy loading, read replication and more.

[Getting Started](#) [API Reference](#)

[Upgrade to v6](#) [Support us](#)

1. INSTALL DEPENDENCIES

```
npm install sequelize sqlite3
# or
yarn add sequelize sqlite3
```

<https://sequelize.org/>

Sequelize ORM

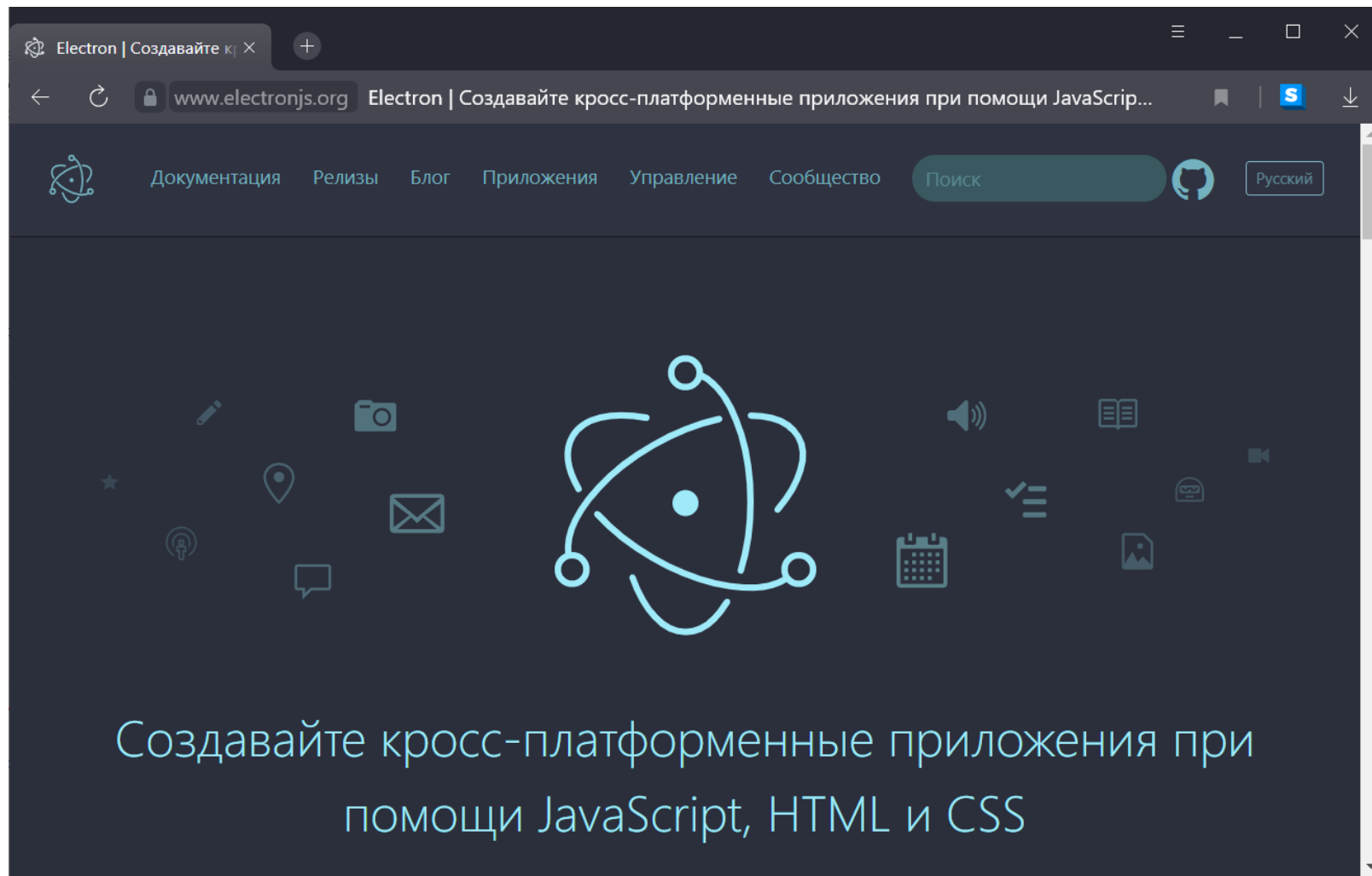
- ORM – Object-Relational Mapping
 - Замена прямых SQL-запросов к БД
- Sequelize – библиотека на Node.js
 - **`npm install --save sequelize`**
- Поддерживаемые СУБД
 - **Postgres**
 - **MySQL**
 - **MariaDB**
 - **SQLite**
 - **Microsoft SQL Server**
- Поддерживает транзакции, связи, ленивую загрузку, репликации и др.

Пример использования Sequelize

85

```
const { Sequelize, Model, DataTypes } = require('sequelize');
const sequelize = new Sequelize('sqlite::memory:');
class User extends Model {}
User.init({
  username: DataTypes.STRING,
  birthday: DataTypes.DATE
}, { sequelize, modelName: 'user' });
sequelize.sync()
  .then(() => User.create({
    username: 'janedoe',
    birthday: new Date(1980, 6, 20)
  }))
  .then(jane => { console.log(jane.toJSON()); });
```

Electron — фреймворк для кроссплатформенной разработки настольных приложений



<https://www.electronjs.org/>

- ECMAScript 2016

```
// includes
const arr = [1,2,3]
console.log(arr.indexOf(3)) // 2
console.log(arr.includes(3)) // true
console.log(arr.indexOf(NaN)) // -1
console.log(arr.includes(NaN)) // true
// **
console.log(2**3) // 8
```

Много нововведений по регулярным выражениям, по работам с асинхронным и параллельным кодом

- ECMAScript 2017

```
// Object.values()
const cars = {leti: 1, itmo: 2}
console.log(Object.values(cars)) // [ 1, 2 ]
// Object.entries()
console.log(Object.entries(cars)) // [ [ 'leti', 1 ], [ 'itmo', 2 ] ]
// String.prototype.padStart(), String.prototype.padEnd()
console.log("5".padStart(5)) // '    5'
console.log("5".padEnd(5)) // '5    '
// async / await
```

Airbnb JavaScript Style Guide

88

The screenshot shows the GitHub repository for the Airbnb JavaScript Style Guide. The repository is named 'javascript' and is public. It has 139k stars, 26.4k forks, and 3.8k watchers. The repository is managed by vmohir and ljharb. The file list shows various files and folders, including .github/workflows, css-in-javascript, linters, packages, react, .editorconfig, .gitignore, .npmrc, and LICENSE.md. The right sidebar shows the repository's about section, including the README, MIT license, Code of conduct, Security policy, Activity, and 139k stars.

airbnb/javascript: JavaScript Style Guide

airbnb / javascript

Code Issues 91 Pull requests 72 Actions Wiki Security Insights

javascript Public

Watch 3.8k Fork 26.4k Star 139k

master 5 branches 102 tags Go to file Add file Code

vmohir and ljharb [eslint config] [b... 11ab371 on Aug 21 1,964 commits

File/Folder	Description	Last Commit
.github/workflows	[Tests] re-enable tests disabled for the eslint ...	2 years ago
css-in-javascript	[guide] [css] Fixed Italic subtitle in css-in-js R...	6 years ago
linters	[Docs] HTTP => HTTPS	2 years ago
packages	[eslint config] [base] [patch] Add a message ...	3 months ago
react	[readme] update eslint plugin repo URLs	6 months ago
.editorconfig	[eslint config] [*] [tests] use ecint instead ...	4 years ago
.gitignore	Only apps should have lockfiles.	6 years ago
.npmrc	Only apps should have lockfiles.	6 years ago
LICENSE.md	[guide] Update license year	6 years ago

About

JavaScript Style Guide

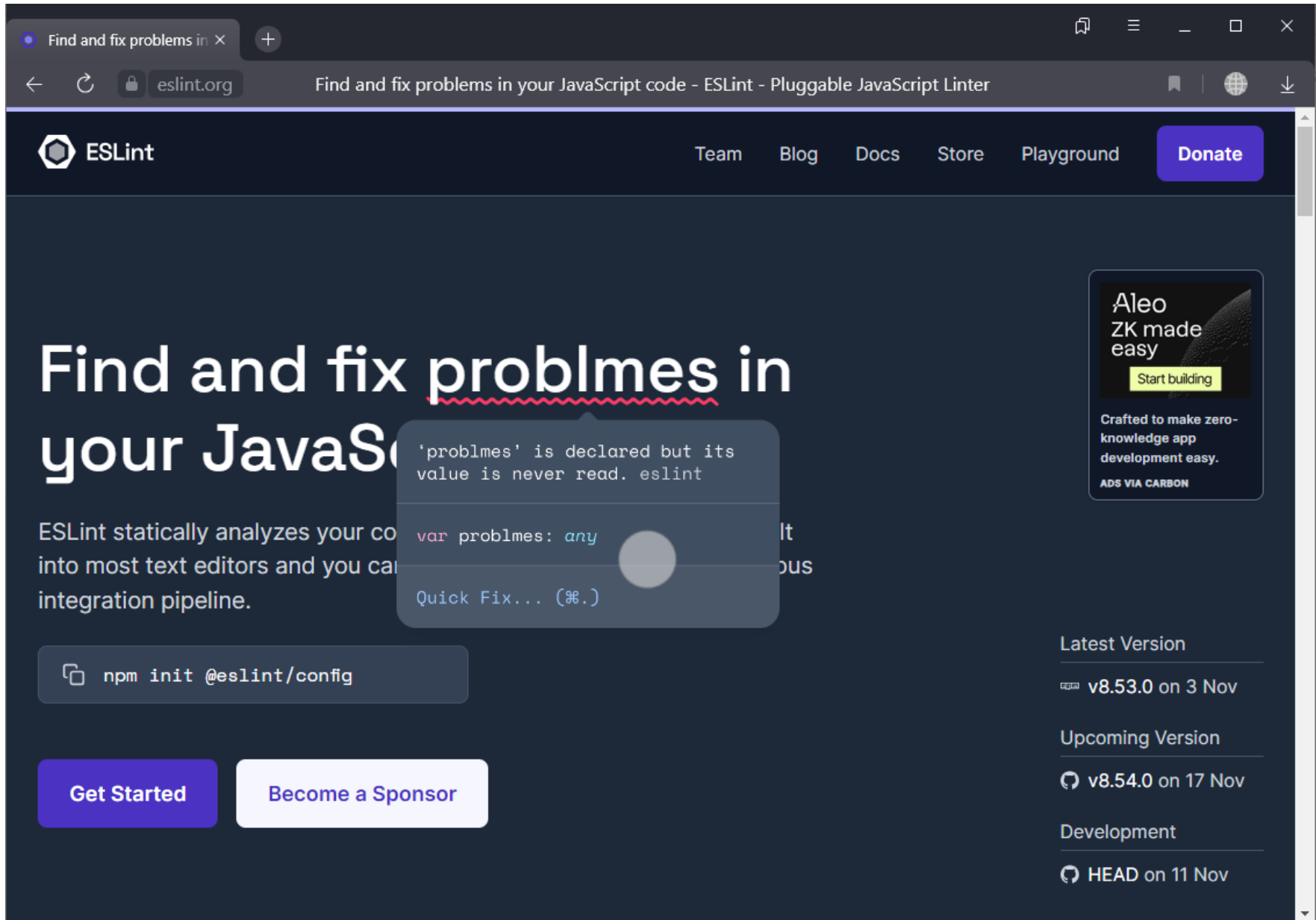
javascript styleguide eslint es6 style-linter linting style-guide es2015 naming-conventions arrow-functions es2017 es2016 tc39 es2018

Readme MIT license Code of conduct Security policy Activity 139k stars 3.8k watching 26.4k forks Report repository

<https://github.com/airbnb/javascript>

ESLint – статический анализатор кода

89



The screenshot shows the ESLint website homepage. The browser's address bar displays 'eslint.org'. The website header includes the ESLint logo and navigation links: Team, Blog, Docs, Store, Playground, and a purple 'Donate' button. The main heading reads 'Find and fix problems in your JavaScript code'. Below this, a text block states: 'ESLint statically analyzes your code to find and fix problems in your JavaScript code. It integrates into most text editors and you can use it in your CI/CD integration pipeline.' A code editor overlay is positioned over the word 'problems' in the heading, displaying an ESLint error: `'problems' is declared but its value is never read. eslint`. The editor also shows the variable declaration `var problems: any` and a 'Quick Fix...' button. A terminal-like box below the text contains the command `npm init @eslint/config`. At the bottom left are two buttons: 'Get Started' and 'Become a Sponsor'. On the right side, there is an advertisement for 'Aleo ZK made easy' and a section titled 'Latest Version' showing 'v8.53.0 on 3 Nov', followed by 'Upcoming Version' showing 'v8.54.0 on 17 Nov', and a 'Development' section showing 'HEAD on 11 Nov'.

Find and fix problems in your JavaScript code

ESLint statically analyzes your code to find and fix problems in your JavaScript code. It integrates into most text editors and you can use it in your CI/CD integration pipeline.

```
npm init @eslint/config
```

Get Started [Become a Sponsor](#)

Latest Version
v8.53.0 on 3 Nov

Upcoming Version
v8.54.0 on 17 Nov

Development
HEAD on 11 Nov

<https://eslint.org/>

js-green-licenses – проверка лицензионной чистоты модулей

google/js-green-licenses: JavaScript package.json License Checker

google / js-green-licenses

Code Issues 8 Pull requests 15 Actions Security Insights

js-green-licenses Public

Watch 5 Fork 34 Star 169

main 23 branches 18 tags

Go to file Add file <> Code

release-please[bot] and bcoe chore(main): release 4.0.0 (#201) 6bedc88 On Jan 9 174 commits

.github	chore(deps): update actions/checkout action...	10 months ago
src	fix(deps): upgrade to typescript 4.4.4 (#180)	2 years ago
test	feat!: use allowlist for allowed licenses (#153)	2 years ago
.eslintignore	build!: drop support for node.js 8 (#124)	3 years ago
.eslintrc.json	build!: drop support for node.js 8 (#124)	3 years ago
.gitignore	chore: upgrade typescript to 2.7.1 (#30)	5 years ago
.prettierrignore	build!: drop support for node.js 8 (#124)	3 years ago
.prettierrc.js	build!: drop support for node.js 8 (#124)	3 years ago
CHANGELOG.md	chore(main): release 4.0.0 (#201)	10 months ago

About

JavaScript package.json License Checker

- Readme
- Apache-2.0 license
- Code of conduct
- Security policy
- Activity
- 169 stars
- 5 watching
- 34 forks

Report repository

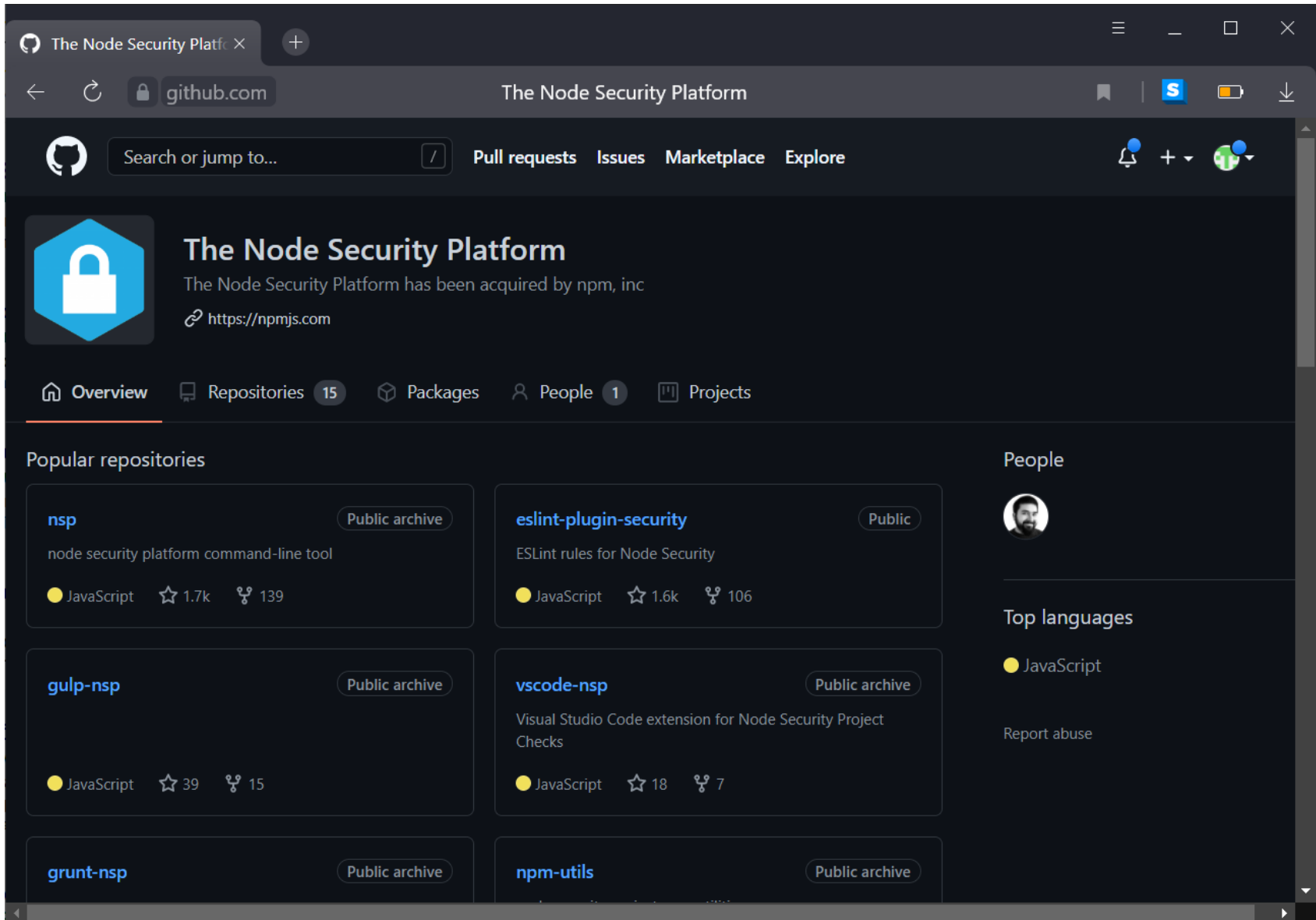
Releases 17

v4.0.0 Latest

<https://github.com/google/js-green-licenses>

NSP – Node security platform

91



The screenshot shows the GitHub page for 'The Node Security Platform'. The page header includes the GitHub logo, a search bar, and navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The main content area features the repository name 'The Node Security Platform' with a blue padlock icon. Below the name, it states 'The Node Security Platform has been acquired by npm, inc' and provides a link to 'https://npmjs.com'. A navigation bar below the header shows 'Overview' (selected), 'Repositories 15', 'Packages', 'People 1', and 'Projects'. The 'Popular repositories' section displays six repositories in a grid:

Repository Name	Language	Stars	Forks	License
nsp	JavaScript	1.7k	139	Public archive
eslint-plugin-security	JavaScript	1.6k	106	Public
gulp-nsp	JavaScript	39	15	Public archive
vscode-nsp	JavaScript	18	7	Public archive
grunt-nsp	JavaScript	-	-	Public archive
npm-utils	JavaScript	-	-	Public archive

On the right side, there is a 'People' section with a profile picture and a 'Top languages' section showing 'JavaScript' as the primary language. A 'Report abuse' link is also visible.

<https://github.com/nodesecurity>

SonarQube – статический анализатор кода

Clean Code: Writing Clear, Readable, Understandable & Reliable Quality Code | Sonar

sonar Solutions Products Resources Company

START FOR FREE **EXPLORE PRICING**

GOODBYE RE-WORK. HELLO INNOVATION.

home of {clean code}

Trusted by 7 million developers, Sonar™ helps you build responsible, secure, high-quality code quickly and systematically.

START FOR FREE --> **WHAT IS SONAR? ▶**

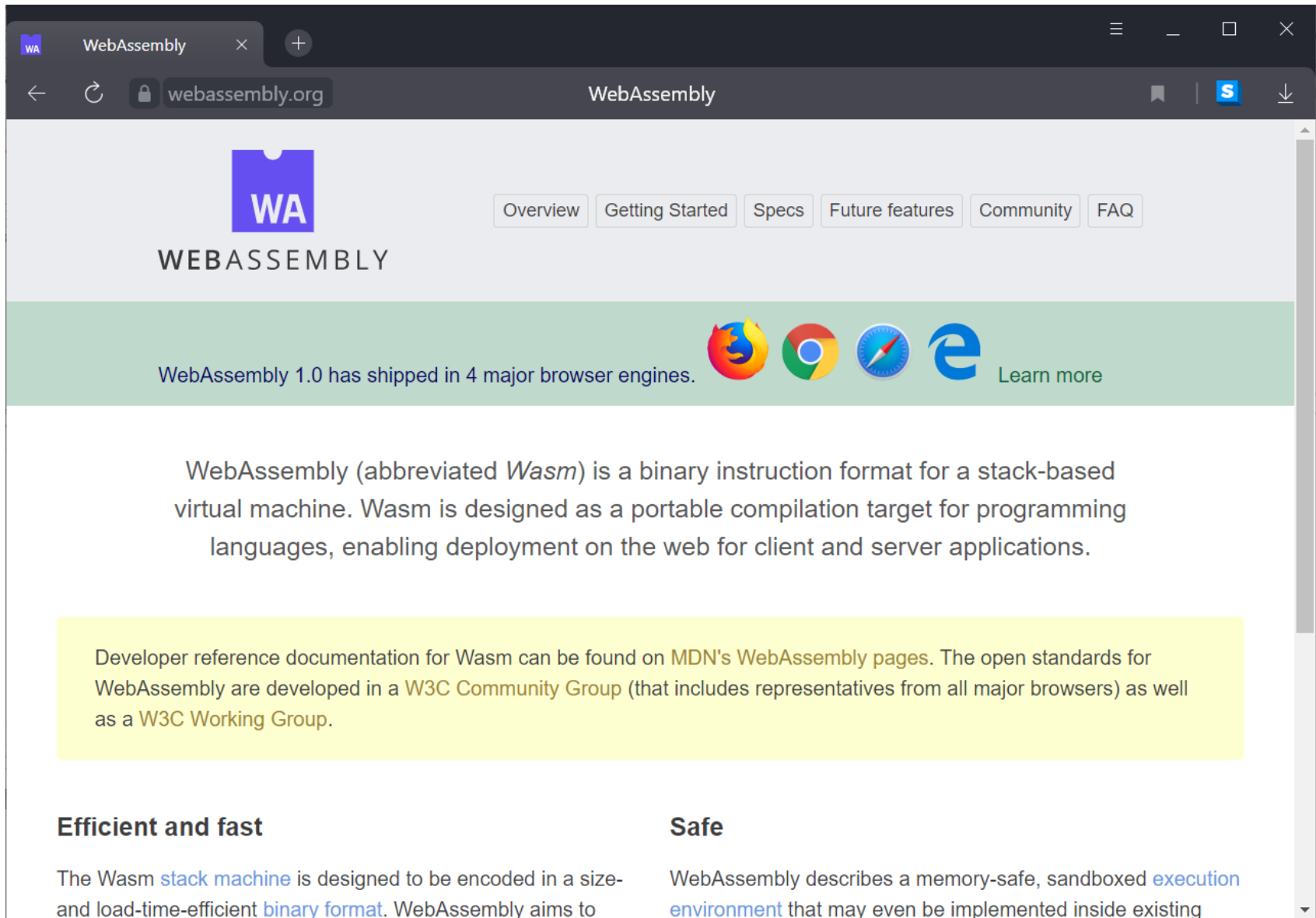
Quality Gate **Passed** New Code Overall Code
New Code: since about 1 month ago

Reliability 23 Bugs	Maintainability 574 Code Smells
Security 0 Vulnerabilities	Security Review 0 Security Hotspots
Coverage 76% Coverage On 21k New Lines to cover	Duplications 0.0% Duplications On 46k New Lines

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

WebAssembly (WASM)





93



The screenshot shows the WebAssembly website (webassembly.org) in a browser window. The page features a dark header with the 'WA' logo and navigation links: Overview, Getting Started, Specs, Future features, Community, and FAQ. Below the header, a green banner announces 'WebAssembly 1.0 has shipped in 4 major browser engines.' with logos for Firefox, Chrome, Safari, and Edge, followed by a 'Learn more' link. The main content area describes WebAssembly as a binary instruction format for a stack-based virtual machine, designed as a portable compilation target for programming languages. A yellow box highlights that developer reference documentation is available on MDN's WebAssembly pages and that standards are developed in W3C Community and Working Groups. The page also features two columns: 'Efficient and fast' and 'Safe', each with a brief description of WebAssembly's design goals.

WebAssembly

Overview Getting Started Specs Future features Community FAQ

WebAssembly 1.0 has shipped in 4 major browser engines.     [Learn more](#)

WebAssembly (abbreviated *Wasm*) is a binary instruction format for a stack-based virtual machine. Wasm is designed as a portable compilation target for programming languages, enabling deployment on the web for client and server applications.

Developer reference documentation for Wasm can be found on [MDN's WebAssembly pages](#). The open standards for WebAssembly are developed in a [W3C Community Group](#) (that includes representatives from all major browsers) as well as a [W3C Working Group](#).

Efficient and fast

The Wasm [stack machine](#) is designed to be encoded in a size- and load-time-efficient [binary format](#). WebAssembly aims to

Safe

WebAssembly describes a memory-safe, sandboxed [execution environment](#) that may even be implemented inside existing

<https://webassembly.org/>

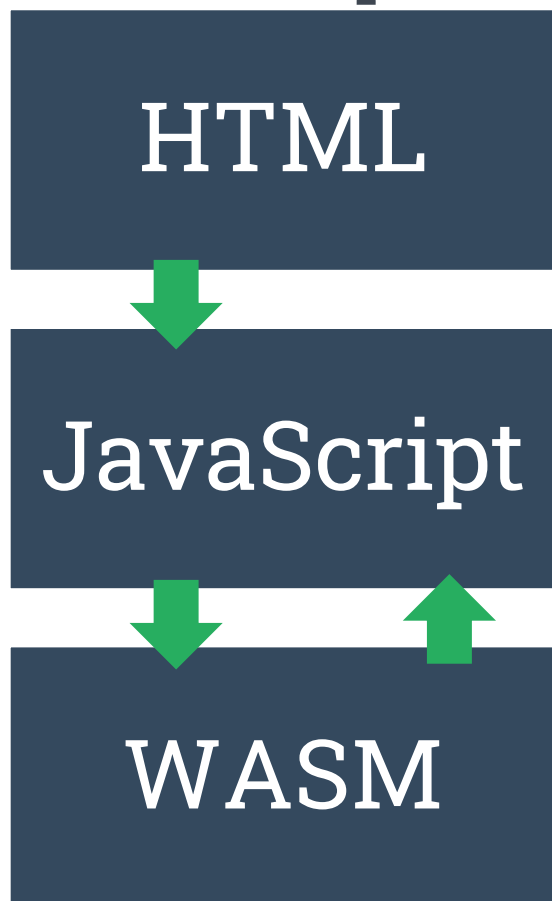
- **WebAssembly** — бинарный формат, запускаемый в браузере, виртуальная машина, и результат компиляции с языка высокого уровня
- Не язык программирования
- Не связан с веб
 - Работает и вне браузера
 - **WebAssembly System Interface (WASI)**
 - <https://github.com/WebAssembly/WASI>
 - <https://wasi.dev/>

Создание модулей с языков:

- C/C++
- Rust
- AssemblyScript
- C#
- F#
- Go
- Kotlin
- Swift
- D
- Pascal
- Zig
- Grain

WASM и JavaScript

- Wasm загружается и запускается только из JavaScript




Пример вызова WASM:

```
fetch('module.wasm').then(response =>
  response.arrayBuffer()
).then(bytes =>
  WebAssembly.instantiate(bytes, importObject)
).then(results => {
  // Что делать с результатом работы WebAssembly
})
```

Виртуальная машина WASM

96



- Introduction
- Structure
- Validation
- Execution
- Binary Format
- Text Format
- Appendix
- Index of Types
- Index of Instructions
- Index of Semantic Rules
- Index
- Download as PDF

Index of Instructions

Instruction	Binary Opcode	Type	Validation	Execution
unreachable	0x00	$[t_1^*] \rightarrow [t_2^*]$	validation	execution
nop	0x01	$[] \rightarrow []$	validation	execution
block <i>bt</i>	0x02	$[t_1^*] \rightarrow [t_2^*]$	validation	execution
loop <i>bt</i>	0x03	$[t_1^*] \rightarrow [t_2^*]$	validation	execution
if <i>bt</i>	0x04	$[t_1^* \text{ i32}] \rightarrow [t_2^*]$	validation	execution
else	0x05			
(reserved)	0x06			
(reserved)	0x07			
(reserved)	0x08			
(reserved)	0x09			
(reserved)	0x0A			
end	0x0B			
br <i>l</i>	0x0C	$[t_1^* \ t^*] \rightarrow [t_2^*]$	validation	execution
br_if <i>l</i>	0x0D	$[t^* \text{ i32}] \rightarrow [t^*]$	validation	execution

<https://webassembly.github.io/spec/core/appendix/index-instructions.html>

- **WebAssembly**
 - <https://webassembly.org/>
- **WebAssembly Studio**
 - <https://webassembly.studio/>
- **WasmFiddle**
 - <https://wasdk.github.io/WasmFiddle/>
- **The WebAssembly Binary Toolkit**
 - <https://github.com/WebAssembly/wabt>

Вопросы для самопроверки

98

- Чем отличаются GULP и Webpack?
- Какие задачи можно решать с использованием GULP?
- Как можно описать задачу, которую будет выполнять GULP?
- Есть ли в GULP ограничения на имя файла?
- Можно ли в GULP использовать ключевое слово `import`?
- Какая связь между `entry` и `output` в Webpack? Как результат называется по умолчанию и куда попадёт?
- Каких типов бывают загрузчики (loader) в Webpack? Для чего используются?
- В каком порядке выполняются загрузчики (loader) при описании в одном `use`?
- Какие команды загрузки модулей поддерживает Webpack?
- Для чего в Webpack используется `devtool`? Какие бывают?
- Что такое Web-сообщения/Worker/SharedWorker? Для чего используются?
- Как создать web-сокеты? Как с ним взаимодействовать? Из других языков можно?
- Можно ли в истории (history) сохранить выполнение AJAX-запросов или внутренних событий web-приложения?
- Как работать с `sessionStorage`/`localStorage`? Чем они отличаются?
- Что нужно для создания просто авторизации на основе PassportJS?
- Что такое ORM? Для чего нужны? Как использовать в JS?
- Для чего нужен WebAssembly?