

## Создание web-калькулятора с использованием Electron

*Вавилов Егор Дмитриевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### **Аннотация**

В данной статье описан процесс создание web-калькулятора. При создании использовался проект с исходным кодом, для создания приложений Electron. Итоговым результатом является рабочий калькулятор.

**Ключевые слова:** Electron, Приложение, Разработка

## Building a web calculator using Electron

*Vavilov Yegor Dmitrievich*

*Sholom-Aleichem Priamursky State University*

*student*

### **Abstract**

This article walks you through the process of creating a web calculator. When creating, a project with source code was used to create Electron applications. The final result is a working calculator.

**Keywords:** Electron, App, Development

Electron - это проект с открытым исходным кодом, который позволяет создавать приложения с использованием HTML, CSS и JavaScript. С Electron можно использовать web-технологии для создания приложений.

Цель исследования – разработать web-калькулятор с использованием Electron.

Исследованиями в данной теме занимались следующие авторы. А.К.Борисов разработал web-приложение позволяющее командированному сотруднику получить унифицированный доступ к приложениям IT-инфраструктурам [1]. Д.А.Викулина, С.Н.Макаров рассмотрели в своей работе передовые средства и технологии для разработки настольных приложений, провели их анализ и сравнительную характеристику [2]. Т.И.Тимофеев, В.В.Козлов произвели сравнительный обзор фреймворков для настольных приложений по ряду критериев [3]. Е.Д.Зайцев, Д.М.Зайцев раскрыли вопросы эффективности автоматизации тестирования мобильных приложений и web [4].

Для использования Electron необходимо иметь установленный «Node.js».

Electron не заставляет иметь определенную структуру каталогов при создании приложения. Поэтому создаем корневой каталог приложения и

переходим в него с помощью node. Далее создадим необходимые файлы (рис.1).

```
1 | touch app/index.html
2 | touch app/renderer.js
3 | touch app/styles.css
4 | touch app/app.js
```

Рисунок 1 – создание каталогов

Это код создаст папку с именем «app» внутри папки калькулятора. В этой папке приложения будут:

- index.html: домашняя страница приложения Electron.
- styles.css: стили CSS для приложения Electron.
- app.js: этот файл содержит сведения о конфигурации приложения Electron.
- renderer.js: этот файл javascript содержит все функции приложения.

Далее настроим файл «package.json». Для этого выполним команду «npm init». Она настроит пакет узла, создав файл «package.json». Команда также предложит ввести некоторые основные сведения о приложении (рис.2).

```
D:\Electron\calculator>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>' afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (calculator)
version: (1.0.0)
description: Simple Calculator
entry point: (index.js) ./app/app.js
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to D:\Electron\calculator\package.json:
{
  "name": "calculator",
  "version": "1.0.0",
  "description": "Simple Calculator",
  "main": "./app/app.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}
Is this OK? (yes)
D:\Electron\calculator>
```

Рисунок 2 – настройка json

Чтобы загрузить и установить Electron, выполним приведенную команду из корневого каталога приложения (Рис.3).

```
1 | npm install electron --save-dev
```

Рисунок 3 – установка Electron

Этот код загрузит и установит последнюю версию Electron в каталоге «node\_modules» (рис.4).

```
D:\Electron\calculator>npm install electron --save-dev
> core-js@3.6.5 postinstall D:\Electron\calculator\node_modules\core-js
> node -e "try{require('./postinstall')}catch(e){}"
Thank you for using core-js ( https://github.com/zloirock/core-js ) for polyfilling JavaScript standard library!
The project needs your help! Please consider supporting of core-js on Open Collective or Patreon:
> https://opencollective.com/core-js
> https://www.patreon.com/zloirock
Also, the author of core-js ( https://github.com/zloirock ) is looking for a good job ->

> electron@8.2.5 postinstall D:\Electron\calculator\node_modules\electron
> node install.js

npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN calculator@1.0.0 No repository field.

+ electron@8.2.5
added 87 packages from 98 contributors and audited 104 packages in 53.373s

2 packages are looking for funding
  run 'npm fund' for details

found 0 vulnerabilities

D:\Electron\calculator>
```

Рисунок 4 – установка Electron

Теперь необходимо обновить файл «.json», для этого добавим значение «"start": "electron"». Файл «.json» должен выглядеть следующим образом (рис.5).

```
1  {
2    "name": "calculator",
3    "version": "1.0.0",
4    "description": "Simple Calculator",
5    "main": "./app/app.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1",
8      "start" : "electron ."
9    },
10   "author": "Egor",
11   "license": "ISC",
12   "devDependencies": {
13     "electron": "^8.2.5"
14   }
15 }
```

Рисунок 5 – Файл JSON

Откроем файл «app.js». Этот файл является отправной точкой приложения «Electron». Как только выполняется команда «npm start», то сразу выполняется содержимое этого файла. Этот файл считается основным процессом приложения Electron (рис.6).

```
1  const { app, BrowserWindow } = require('electron')
2
3  function createWindow () {
4    let mainWindow = new BrowserWindow({
5      width: 300,
6      height: 460,
7      icon: 'app/icon.png',
8      webPreferences: {
9        nodeIntegration: true
10     }
11   });
12   mainWindow.loadFile('./app/index.html');
13   mainWindow.setMenuBarVisibility(false);
14   mainWindow.setResizable(false);
15 }
16 app.whenReady().then(createWindow);
```

Рисунок 6 – файл app.js

Добавим содержимое в файл «index.html». Этот файл содержит HTML-контент для приложения-калькулятора Electron (рис.7).

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="Content-Security-Policy" content="default-src 'self';script-src 'self' 'unsafe-inline';connect-src *">
6 <meta name="viewport" content="width=device-width,initial-scale=1">
7 <title>web-Калькулятор</title>
8 <link rel="stylesheet" href="/styles.css" type="text/css">
9 </head>
10 <body>
11 <div class="container">
12 <div class="calculator">
13 <div class="calculator_display">0</div>
14 <div class="calculator_keys">
15 <button class="key-operator">+</button>
16 <button class="key-operator">-</button>
17 <button class="key-operator">x</button>
18 <button class="key-operator">/</button>
19 <button>7</button>
20 <button>8</button>
21 <button>9</button>
22 <button>4</button>
23 <button>5</button>
24 <button>6</button>
25 <button>1</button>
26 <button>2</button>
27 <button>3</button>
28 <button>0</button>
29 <button>.</button>
30 <button>AC</button>
31 <button class="key-equal">=</button>
32 </div>
33 </div>
34 </div>
35 <script>
36 require('./renderer');
37 </script>
38 </body>
39 </html>
```

Рисунок 7 – файл html

Здесь используется функция «require» для загрузки файла «javascript», который содержит функции калькулятора, с помощью тега <script>.

Далее добавим скрипт выполнения сценария(рис.8-10).

```
1 class Calculator {
2   constructor(field1, field2, isSecondVal) {
3     this.field1 = field1;
4     this.field2 = field2;
5     this.isSecondVal = isSecondVal;
6     this.operation = '+';
7   }
8 }
9 var isNumber = false;
10 const calculator = new Calculator('', '', false);
11 const buttons = document.querySelectorAll('button');
12 const display = document.querySelector('.calculator_display');
13
14 for (var i = 0; i < buttons.length; i++) {
15   buttons[i].addEventListener('click', function() {
16     checkNumber(this.innerHTML);
17     if(isNumber || ('.' === this.innerHTML)){
18       if(calculator.isSecondVal === false) {
19         if(calculator.field1 === '0'){
20           calculator.field1 = ('.' === this.innerHTML) ? ('0' + this.innerHTML) : this.innerHTML;
21         } else if(calculator.field1.length < 8){
22           calculator.field1 = calculator.field1 + this.innerHTML;
23         }
24         display.innerText = calculator.field1;
25       } else {
26         if(calculator.field2 === '0') {
27           calculator.field2 = this.innerHTML;
28         }
29         else if(calculator.field2.length < 8){
30           calculator.field2 = calculator.field2 + this.innerHTML;
31         }
32         display.innerText = calculator.field2;
33       }
34     } else {
35       switch(this.innerHTML) {
36         case '+':
37         case '-':
38         case 'x':
39         case '/':
40           calculator.operation = this.innerHTML;
```

Рисунок 8 – файл renderer.js

```
40     calculator.operation = this.innerHTML;
41     calculator.isSecondVal = true;
42     break;
43     case 'AC':
44         clear();
45         display.innerHTML = 0;
46         break;
47     default:
48         calculate();
49     }
50 }
51 });
52 }
53
54 function calculate() {
55     var value1 = parseFloat(calculator.field1);
56     var value2 = parseFloat(calculator.field2);
57     var result;
58     switch (calculator.operation) {
59         case '-':
60             result = value1 - value2;
61             break;
62         case 'x':
63             result = value1 * value2;
64             break;
65         case '/':
66             result = value1 / value2;
67             break;
68         default:
69             result = value1 + value2;
70             break;
71     }
72     parseResult(result);
73     clear();
74 }
75
76 function parseResult(val) {
77     if (isInteger(val)) {
78         display.innerHTML = val;
79     } else {
```

Рисунок 9 – файл renderer.js

```
79     } else {
80         display.innerHTML = val.toFixed(3);
81     }
82 }
83
84 function clear() {
85     calculator.field1 = '0';
86     calculator.field2 = '0';
87     calculator.isSecondVal = false;
88 }
89
90 function isInteger(value) {
91     return /^\d+$/.test(value);
92 }
93
94 function checkNumber(number) {
95     var regex = /^\d+$/;
96     isNumber = regex.test(number);
97 };
98
```

Рисунок 10 – файл renderer.js

Здесь прописан класс «Calculator», который содержит входные значения калькулятора. Он также содержит индикатор второго входного значения и операцию по умолчанию. Переменная «isNumber» используется для определения, является ли нажатая кнопка числом или нет. Так же используется селектор запросов «javascript» для выбора всех элементов кнопки и элемента отображения, который отображает ввод и значение результата калькулятора. Добавлено событие щелчка на любой из кнопок и выполнение действия в зависимости от ввода пользователя. Функция «clear()» очищает все поля калькулятора и инициализирует калькулятор. Функция «checkNumber()» используется, чтобы определить, является ли текст кнопки числом или нет. Функция «parseResult()» анализирует результат с плавающей запятой, установив десятичную точку результата равной 3.

Чтобы запустить приложение, выполним следующую команду из корневой папки приложения «npm start» (рис.11).

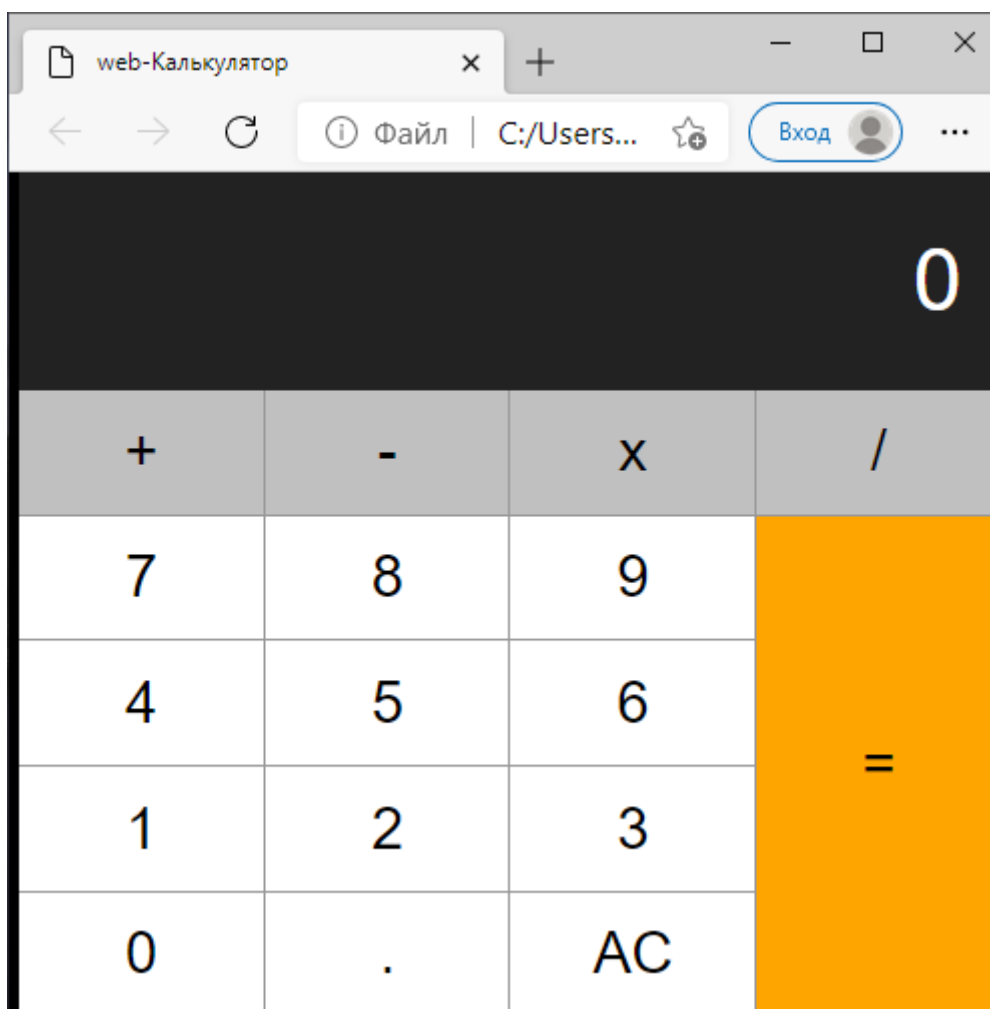


Рисунок 11 – Тест приложения

В данной статье было разобрано, как создавать web-приложение калькулятор с использованием Electron, HTML, CSS и JavaScript.



**Библиографический список**

1. Борисов А.К. Sun secure global desktop все ваши приложения в окне браузера // Системный администратор. 2009. №9(82). С. 48-52.
2. Викулина Д.А., Макаров С.Н. Современные технологии создания desktop-приложений // Наука и современность 2012. №18. С. 180-186.
3. Тимофеев Т.И., Козлов В.В. Обзор современных средств создания интерфейса пользователя для desktop приложений // Традиции и инновации в строительстве и архитектуре. Строительные технологии. 2017. №1. С.585-588.
4. Зайцев Е.Д., Зайцев Д.М. К вопросу об эффективности автоматизации тестирования web-, desktop- и мобильных приложений // Проблемы инфокоммуникаций 2018. №2(8). С. 56-63.