

## Создание приложения для работы с камерой с помощью библиотеки NW.js

*Кизянов Антон Олегович*

*Приамурский государственный университет имени Шолом-Алейхема  
Студент*

### **Аннотация**

В данной статье будет написано приложения для работы с камерой на языке программирования JavaScript. Программа будет использовать библиотеку NW.js для построения интерфейсов в системе Windows. В результате будет написано приложения способное делать скриншоты с камеры в реальном времени и опробовано на себе.

**Ключевые слова:** NW.js, JavaScript

### **Creating a camera application using the NW.js library**

*Kizyanov Anton Olegovich*

*Sholom-Aleichem Priamursky State University  
student*

### **Abstract**

This article will tell you what NW.js is, where to get it and create an application for working with the camera.

**Keywords:** NW.js, JavaScript

Web технологии уже давно вышли за рамки всем привычного web'а. После создания JavaScript движка Node.js стало доступно выполнять JavaScript не только в браузерах, а и на самой операционной системе. Это стало бумом развития всяческих библиотек и фреймворков работающих на данном движке, одним из которых и является NW.js.

NW.js позволяет разрабатывать настольные графические приложения с использованием веб-технологий: он сочетает в себе механизм рендеринга Chromium и среду выполнения Node.js.[1]

Цель исследования – написание приложения для работы с камерой с помощью библиотеки NW.js.

Ранее этим вопросом интересовались К.Р. Мальцев, А.Н. Брежнева развивали тему «Разработка программного комплекса для обмена сообщениями внутри предприятия» [2] в которой рассматривается разработка программного комплекса для обмена сообщениями между работниками одного из предприятий. Данный комплекс использует фреймворк Node.js в качестве сервера, и фреймворк NW.js в качестве клиента. Данный комплекс должен поддерживать контакты, отправку

изображений и файлов, моментальные уведомления о новых сообщениях, а также соединение без прямой видимости сервера. А.А. Крюкова, В.А. Почебут с темой «Особенности тестирования, внедрения и разработки десктопных приложений» [3], а подробнее про внедрение путей решения проблемы, на основе детального изучения алгоритма внедрения и анализе платформ: nw.js, electron.js, cuba. Выделены преимущества данных приложений над различными web-приложениями. М.В. Маревская и И.Б. Государев опубликовали статью «Особенности использования веб-фреймворков при разработке десктопных приложений» [4] рассказали про фреймворки, позволяющие создавать десктопные приложения на основе веб-технологий, рассмотрены фреймворки Electron и NW.js, их общие черты, специфика их использования, различия между ними.

NW.js позволяет создавать настольные приложения на чистом JavaScript, предоставляя среду выполнения со встроенными в систему API-интерфейсами.

Перед началом работы нужно установить Node.js[5] среду выполнения JavaScript кода. Она устанавливается вместе с NPM(пакетный менеджер), он позволяет устанавливать библиотеки одной командой.

Для начала нужно ввести команду **npm init** в директории будущего проекта. Она создаст файл package.json в котором будут находиться зависимости.

Следующей командой будет **npm install nw**, npm установит nw.js в конкретную директорию.

Теперь нужно отредактировать файл package.json, в него нужно добавить поля main и scripts, они определяют главный файл и сразу прописывают команду на запуск приложения. Должно выглядеть так-же как показано ниже.

```
{
  "name": "Camera",
  "main": "app.html",
  "window": {
    "show": false,
    "frame": true,
    "position": "center",
    "width": 500,
    "height": 500
  },
  "scripts": {
    "start": "nw ."
  },
  "dependencies": {
    "nw": "^0.40.0"
  }
}
```

Нужно создать файл app.js рядом с файлом package.json и поместить в него следующий код.

```
var gui = require('nw.gui');

navigator.webkitGetUserMedia({video: true}, onSuccess, onFail);

function onSuccess(stream) {
    document.getElementById('camFeed').srcObject = stream;
}

function onFail() {
    alert('Не могу соединиться с камерой');
}

function takePhoto() {
    var c = document.getElementById('photo');
    var v = document.getElementById('camFeed');
    c.getContext('2d').drawImage(v, 0, 0, 320, 240);
}

gui.Window.get().show();
```

Этот код формирует главное окно приложения, описывает его размеры и цвет, а потом загружает файл app.html.

Файл app.html это главная страница приложения. Нужно создать его и поместить в него следующее содержимое.

```
<!DOCTYPE html>
<html>
<head>
  <script src="app.js"></script>
  <meta charset="UTF-8">
  <title>Тест камеры</title>
</head>
<body>
<video id="camFeed" width="320" height="240" autoplay></video>
<canvas id="photo" width="320" height="240"></canvas>
<input type="button" value="Take Photo" onclick="takePhoto();">
</body>
</html>
```

В файле app.html определены 3 тега для отображения видео, отображения картинки и кнопка создания картинки.

На данном этапе приложение готово, чтобы запустить нужно открыть нужно в консоли запустить команду **npm start** находясь в этой директории. Результат представлен на рисунках 1 и 2.

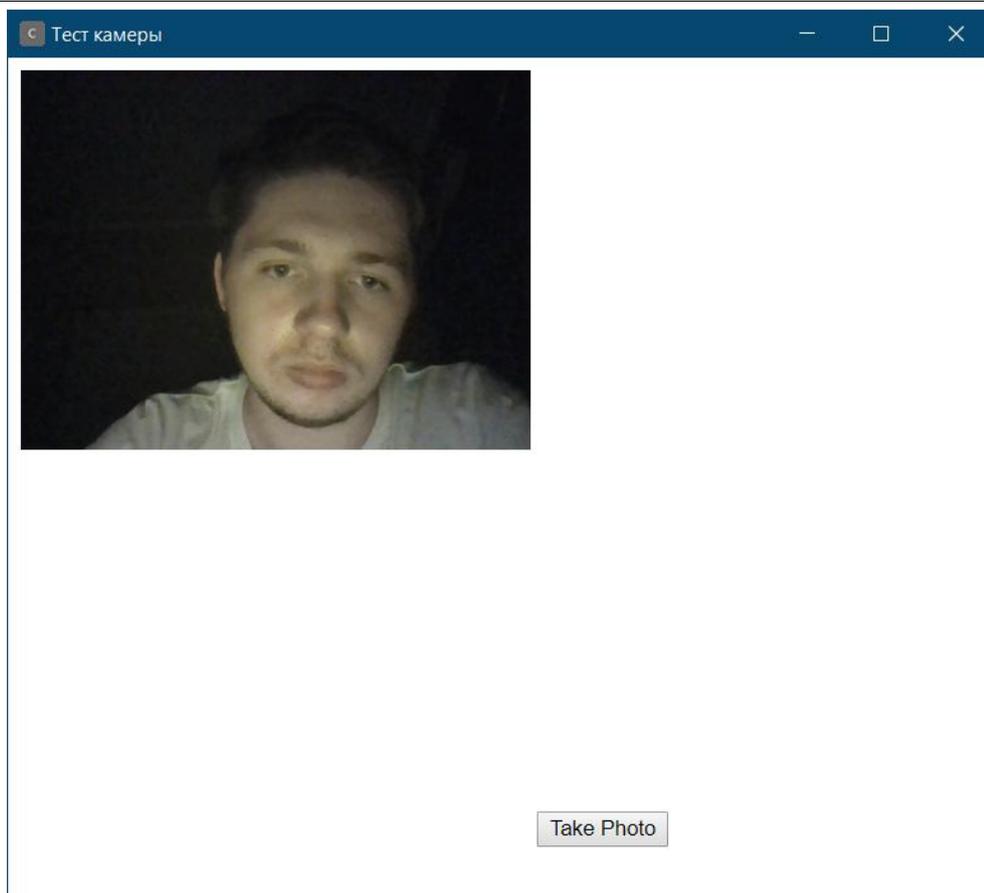


Рис. 1 Запущенное приложение только с камерой

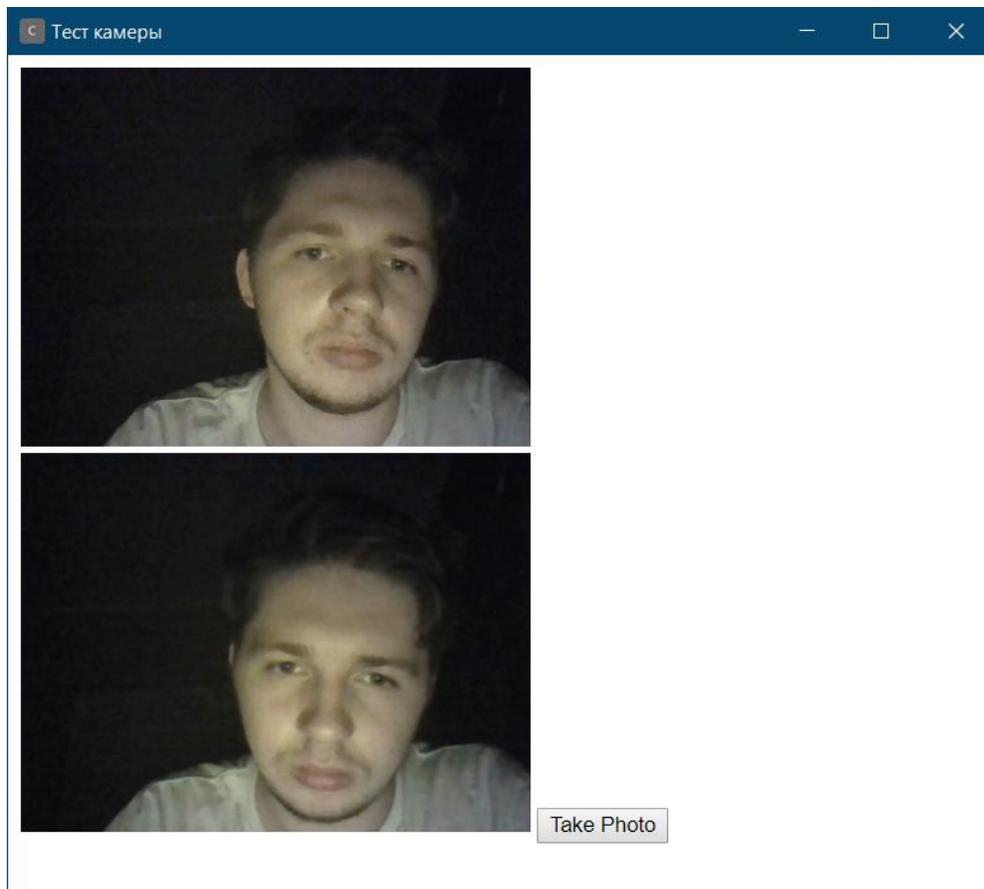


Рис. 2 Запущенное приложение со скриншотом

**Вывод**

В результате получилась программа для создания скриншотов в реальном времени и успешно протестирована. Она может стать хорошим примером работы языка программирования JavaScript и библиотеки NW.js в среде Windows.

**Библиографический список**

1. NW.js URL: <https://nwjs.io/> (Дата обращения: 07.08.2019)
2. Мальцев К.Р., Брежнева А.Н. Разработка программного комплекса для обмена сообщениями внутри предприятия // В сборнике: Программная инженерия: современные тенденции развития и применения сборник материалов 3-й Всероссийской конференции, посвященной 55-летию ЮЗГУ. Курск, 2019. С. 243-246. URL: <https://elibrary.ru/item.asp?id=37117918> (Дата обращения: 16.07.2018)
3. Крюкова А.А., Почебут В.А. Особенности тестирования, внедрения и разработки десктопных приложений // Синергия Наук. 2017. № 12. С. 964-971. URL: <https://elibrary.ru/item.asp?id=29392793> (Дата обращения: 16.07.2018)
4. Маревская М.В., Государев И.Б. Особенности использования веб-фреймворков при разработке десктопных приложений // В сборнике: Альманах научных работ молодых ученых университета ИТМО XLVII научная и учебно-методическая конференция Университета ИТМО по тематикам: экономика; менеджмент, инноватика. 2018. С. 208-210. URL: <https://elibrary.ru/item.asp?id=36987029> (Дата обращения: 07.08.2019)
5. Node.js URL: <https://nodejs.org/en/> (Дата обращения: 07.08.2019)