



МойОфис[®]
**Комплект Средств
Разработки (SDK)**

Руководство программиста

АВТОНОМНЫЙ МОДУЛЬ РЕДАКТИРОВАНИЯ

ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
«МОЙОФИС КОМПЛЕКТ СРЕДСТВ РАЗРАБОТКИ (SDK)»**

АВТОНОМНЫЙ МОДУЛЬ РЕДАКТИРОВАНИЯ

РУКОВОДСТВО ПРОГРАММИСТА

2020.02

На 17 листах

Москва

2020

Все упомянутые в этом документе названия продуктов, логотипы, торговые марки и товарные знаки принадлежат их владельцам.

Товарные знаки «МойОфис» и «MyOffice» принадлежат ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ».

Ни при каких обстоятельствах нельзя истолковывать любое содержимое настоящего документа как прямое или косвенное предоставление лицензии или права на использование товарных знаков, логотипов или знаков обслуживания, приведенных в нем. Любое несанкционированное использование этих товарных знаков, логотипов или знаков обслуживания без письменного разрешения их правообладателя строго запрещено.

СОДЕРЖАНИЕ

Перечень сокращений	5
1 Общие сведения	6
1.1 Назначение программы	6
1.2 Автономный Модуль Редактирования.....	6
1.3 Уровень подготовки пользователя	6
1.4 Системные требования	7
1.5 Ограничения	8
2 Установка	9
2.1 Дистрибутив	9
2.2 Установка.....	10
2.3 Особенности использования	11
2.4 Проверка работоспособности	12
3 Пример использования	14
4 API методы и уведомления	15
4.1 API методы	15
4.1.1 Метод handshake.....	15
4.1.2 Метод openDocument	16
4.1.3 Метод saveDocument.....	16
4.2 API уведомления	16
4.2.1 Уведомление changeDocument.....	16
4.2.2 Уведомление error	17
4.2.3 Уведомление pageReloadRequested	17

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

В настоящем документе используются следующие сокращения (см. Таблица 1):

Таблица 1 – Сокращения и расшифровки

Сокращение	Расшифровка
ОС	Операционная система
ПО МойОфис	Программное обеспечение «МойОфис Комплект Средств Разработки (SDK). Автономный Модуль Редактирования»
API	Application Programming Interface
SDK	Software Development Kit

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Назначение программы

Программное обеспечение «МойОфис Комплект Средств Разработки (SDK)» это набор средств взаимодействия прикладных систем с приложениями МойОфис.

Программное обеспечение «МойОфис Комплект Средств Разработки (SDK). Автономный Модуль Редактирования» предназначено для встраивания в веб-приложения в качестве компонента для просмотра или редактирования текстовых или табличных документов.

1.2 Автономный Модуль Редактирования

Автономный Модуль Редактирования используется для встраивания в веб-приложения сторонних производителей, в качестве компонента для просмотра и редактирования текстовых или табличных документов.

Автономный Модуль Редактирования поддерживает просмотр и редактирование документов следующих форматов:

- PlainText – используется для работы с файлами TXT;
- DSV – используется для работы с табличными данными в текстовой форме (CSV, DSV). Строка текста содержит одно или несколько полей данных, разделенных запятыми или иным разделителем;
- OXML – используется для работы с текстовыми (DOCX) или табличными (XLSX) документами в формате Open Office XML;
- ODF – используется для работы с текстовыми (ODT, XODT) или табличными (ODS, XODS) документами формата Open Document Format (ГОСТ Р ИСО/МЭК 26300-2010).

Сохранение изменений в редактируемых документах возможно только в форматах OXML и ODF.

1.3 Уровень подготовки пользователя

Пользователем ПО «МойОфис Комплект Средств Разработки (SDK). Автономный Модуль Редактирования» является веб-разработчик, интегрирующий компоненты просмотра или редактирования в свое приложение.

Требования к квалификации пользователя ПО МойОфис:

1. уверенное знание современных технологий разработки Single Page Applications: Javascript, Typescript, HTML, CSS;
2. знание API межоконного взаимодействия и их технических особенностей;
3. знание систем управления пакетами javascript (yarn, npm);
4. знание систем сборки веб-приложений (webpack);
5. навыки настройки веб-сервера.

1.4 Системные требования

Использование ПО «МойОфис Комплект Средств Разработки (SDK). Автономный Модуль Редактирования» возможно в браузерах: Chrome, Opera, Яндекс браузер, Mozilla FireFox, Microsoft Edge (Chromium). Версии браузеров для используемой ОС приведены в документе «МойОфис Комплект Средств Разработки (SDK). Автономный Модуль Редактирования. Системные требования».

Для установки ПО «МойОфис Комплект Средств Разработки (SDK). Автономный Модуль Редактирования» должно быть установлено следующее программное обеспечение для используемой ОС:

- командный интерпретатор bash;
- пакетный менеджер для JavaScript Yarn;
- программное обеспечение для автоматизации развертывания и управления приложениями Docker.

Для обращения к методам Автономного Модуля Редактирования должна использоваться библиотека frame-RPC v3.1.0.

Полный перечень требований к программному и аппаратному обеспечению приведен в документе «МойОфис Комплект Средств Разработки (SDK). Автономный Модуль Редактирования. Системные требования».

1.5 Ограничения

ПО МойОфис имеет следующие ограничения:

1. отсутствует поддержка работы с браузерами Apple Safari и Microsoft Internet Explorer;
2. отсутствует поддержка работы с двоичными файлами документов Microsoft (doc, xls);
3. для повторного открытия редактора документа элемент iframe HTML должен быть загружен заново.

2 УСТАНОВКА

2.1 Дистрибутив

Дистрибутив ПО МойОфис поставляется в виде архивного файла **MyOffice_AMP_2020.02_<build number>.zip**, где **<build number>** – номер сборки программы.

Архивный файл содержит следующие данные:

- правовые уведомления;
- папка **wte/dist** с ресурсами для развертывания;
- папка **demo** с демонстрационным приложением.

2.2 Установка

Для установки ПО МойОфис выполните следующие действия:

1. создайте каталог установки, например, папку **AMP**;
2. извлеките содержимое архивного файла дистрибутива (см. раздел 2.1) в каталог установки;
3. перейдите в папку **wte/dist** каталога установки и в командной строке последовательно запустите следующие команды:

```
yarn install  
yarn build  
docker build -t wte-demo .  
docker run -it -p 8000:80 wte-demo
```

2.3 Особенности использования

При использовании ПО МойОфис необходимо учесть следующие особенности:

- содержимое каталога **wte/dist** должно раздаваться веб-сервером в режиме статики;
- для файла **index.html** необходимо запретить кеширование;
- для всех типов файлов должны отдаваться правильные значения заголовков **content-type**;
- в качестве внешнего адреса статических ресурсов подходит как корень домена, так и вложенная директория.

2.4 Проверка работоспособности

Для проверки работоспособности ПО МойОфис выполните следующие действия:

1. перейдите в папку **demo** каталога установки ПО МойОфис и в командной строке последовательно запустите следующие команды:

```
yarn install  
yarn start:dist
```

2. запустите браузер из числа поддерживаемых, например, Mozilla FireFox;
3. в браузере перейдите по адресу:

```
http://localhost:8000
```

4. на экране откроется окно демонстрационного примера использования ПО МойОфис.

ПО МойОфис считается работоспособным, если:

1. содержимое окна демонстрационного примера использования ПО МойОфис на экране аналогично приведенному на Рисунок 1.

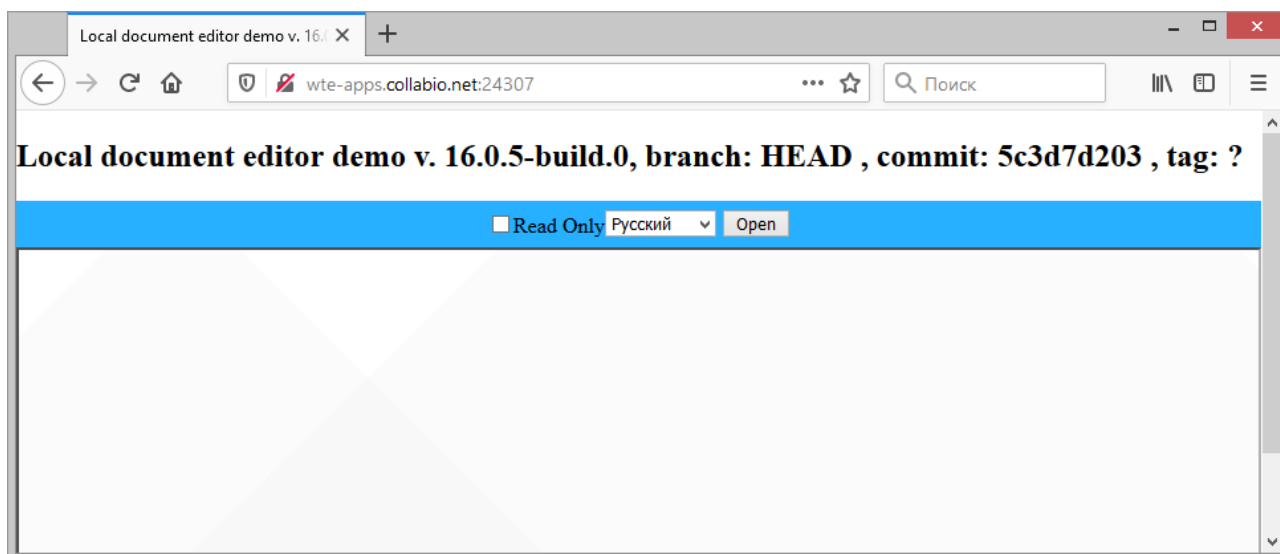


Рисунок 1 – Окно демонстрационного примера использования ПО МойОфис

2. при нажатии кнопки **Open** (см. Рисунок 1) и после подтверждения выбора текстового документа, произошла успешная загрузка документа в окно редактирования (например, см. Рисунок 2).

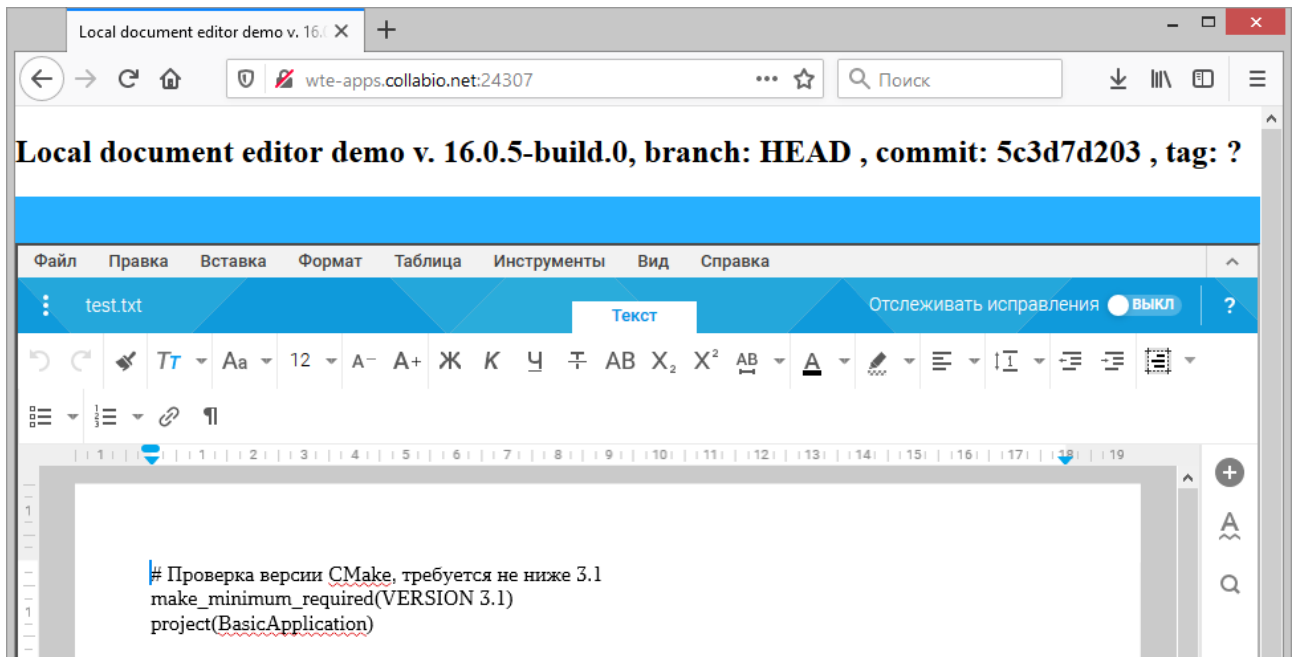


Рисунок 2 – Пример окна редактирования текстового файла

3 ПРИМЕР ИСПОЛЬЗОВАНИЯ

Пример использования API методов и уведомлений при разработке приложений.

```
// Инициализация iframe в HTML
<iframe id="editor" src="https://editor-app.devoffice.ru/">

// Открытие документа в js code (es6)

import RPC from 'frame-rpc';

const frame = document.getElementById('editor');
const documentData = new UintArray(...);
frame.onload = () => {
  const editorAPI = RPC(window, frame.contentWindow, 'https://editor-
app.devoffice.ru', {
    changeDocument: ({hasChanges}) => console.log(`document is ${hasChanges
? 'changed' : 'saved'}`),
    pageReloadRequested: () => window.location.reload(),
    error: ({errorInfo}) => console.log('error', errorInfo),
  });

  editorAPI.call('handshake', { lang: 'ru-RU', userName: 'Ivan Ivanov',
readonly: false }, result => {
    if (result) {
      console.log('error while init editor', result.errorInfo);
      return;
    }

    editorAPI.call(
      'openDocument', {
        name: 'my document.docx',
        type: 'document',
        format: 'OXML',
        content: documentData,
      },
      () => console.log('Document ready for editing')
    );
  })
}

// Редактирование документа пользователем

// Сохранение изменений в документе

editorAPI.call('saveDocument', { format: 'OXML' }, (documentData) =>
console.log('Document saved', documentData));
```

4 API МЕТОДЫ И УВЕДОМЛЕНИЯ

4.1 API методы

4.1.1 Метод handshake

Метод **handshake** предназначен для инициализации редактора с заданием одного из поддерживаемых языков, имени пользователя и специального флага, который позволяет или запрещает редактору вносить изменения в документ.

```
handshake(lang: string, userName: string, readonly: boolean): ErrorDescription
```

Возвращаемый тип: ErrorDescription.

Метод возвращает значение null, если инициализация завершена успешно. В случае ошибки метод возвращает структуру:

```
type ErrorDescription = {
  errorInfo: string; // Сообщение об ошибке
}
```

Параметры:

- lang – обозначение языка редактора.

Поддерживаемые редактором языки:

```
{
  'ru-RU': 'Русский',
  'en-US': 'English',
  'es-PA': 'Español',
  'fr-FR': 'Français',
  'tt-RU': 'Татарча',
  'ba-RU': 'Башкорт',
  'pt-BR': 'Português',
  'de-DE': 'Deutsch',
  'it-IT': 'Italiano',
  'be-BY': 'Беларуская',
  'kk-KZ': 'Қазақша (Kazakh)',
  'ky-KG': 'Кыргызча (Kyrgyz)',
  'hy-AM': 'Հայերեն (Armenian)'
```

- userName – логин, имя или идентификатор пользователя, используется для отслеживания изменений и комментариев;
- readonly – значение типа boolean, позволяет или запрещает редактору вносить изменения в документ.

4.1.2 Метод `openDocument`

Метод **`openDocument`** предназначен для открытия документа в редакторе.

```
openDocument(name: string, type: string, format: string, content: Uint8Array): void
```

Параметры:

- `name` – наименование документа (отображается в заголовке редактора)
- `type` – тип документа (текстовый или табличный);
- `format` – поддерживаемый формат редактируемого документа (см. раздел 1.2);
- `content` – содержимое документа (массив символов).

4.1.3 Метод `saveDocument`

Метод **`saveDocument`** предназначен для сохранения редактируемого документа в выбранном формате.

После сохранения документ не закрывается и не блокируется, поэтому пользователь может продолжать работу с документом.

```
saveDocument(format: string): Uint8Array
```

Возвращаемый тип: `Uint8Array`.

Метод возвращает содержимое сохраненного документа.

Параметр: `format` – поддерживаемый формат сохранения документа (см. раздел 1.2).

4.2 API уведомления

4.2.1 Уведомление `changeDocument`

```
changeDocument({hasChanges: boolean})
```

Уведомляет об изменении флага состояния документа "имеет несохраненные изменения". Например, пользователь может редактировать документ (флаг будет установлен в `true`), затем отменить изменения (флаг будет установлен в `false`), а затем внести другие изменения (флаг будет установлен в `true`).

После вызова метода **`saveDocument`** флагу состояния документа "имеет несохраненные изменения" будет присвоено значение `false`.

4.2.2 Уведомление error

```
error (info: Error Description)
```

Оповещает о фатальной ошибке при открытии, редактировании или сохранении документа.

В случае возникновения ошибки, работа редактора останавливается и на экране отображается диалоговое окно с подробной информацией об ошибке и с кнопкой перезагрузки. Нажатие кнопки перезагрузки отслеживается с помощью уведомления **pageReloadRequested** (см. раздел 4.2.3) и требует корректной обработки. В настоящее время отсутствуют автосохранение или автоматическое восстановление. Все несохраненные изменения будут потеряны.

4.2.3 Уведомление pageReloadRequested

```
pageReloadRequested()
```

Уведомляет о нажатии пользователем кнопки перезагрузки в окне ошибки при остановке редактора.