



МойОфис Комплект Средств Разработки (SDK)

Руководство программиста

СЕРВИС ПРОСМОТРА ДОКУМЕНТОВ

ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»

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

СЕРВИС ПРОСМОТРА ДОКУМЕНТОВ

РУКОВОДСТВО ПРОГРАММИСТА

3.1

Версия 1

На 15 листах

Дата публикации: 26.08.2024

**Москва
2024**

МойОфис

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

1	Общие сведения	6
1.1	Назначение	6
1.2	Возможности	6
1.3	Уровень подготовки пользователя	6
1.4	Системные требования	7
2	Установка	8
2.1	Дистрибутив	8
2.2	Запуск демо приложения	8
2.3	Особенности использования	9
2.4	Проверка работоспособности с использованием демо приложения	10
3	Методы и уведомления AMR Readonly API	12
3.1	Методы AMR Readonly API	12
3.1.1	Метод openDocument	12
3.1.2	Метод blur	14
3.2	Уведомления AMR Readonly API	14
3.2.1	Уведомление onError	14
3.2.2	Уведомление onPageReloadRequested	14
4	Пример использования AMR Readonly API	15

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

В настоящем документе используются следующие сокращения (см. таблицу 1).

Таблица 1 - Сокращения и расшифровки

Сокращение	Расшифровка
AMP	Автономный Модуль Редактирования
DV	Сервис просмотра документов (Document Viewer)
ОС	Операционная система
ПО	Программное обеспечение
ПО МойОфис	Программное обеспечение «МойОфис Комплект Средств Разработки (SDK). Автономный Модуль Редактирования»
API	Application Programming Interface (программный интерфейс приложения)
SDK	Software Development Kit (комплект для разработки программного обеспечения)

МойОфис

1 Общие сведения

1.1 Назначение

Сервис просмотра документов предназначено для встраивания в прикладные системы сторонних производителей в качестве компонента для просмотра текстовых, табличных документов, а также презентаций. В состав продукта входит специальная версия веб-приложений для просмотра текста, таблиц и презентаций МойОфис, предназначенная для исполнения в среде веб-браузера в монопольном режиме.

1.2 Возможности

Сервис просмотра документов используется для встраивания в веб-приложения сторонних производителей в качестве компонента для просмотра текстовых, табличных документов или презентаций.

Пользователю доступны следующие возможности:

1. Открытие электронных текстовых, табличных документов, презентаций в форматах, приведенных в таблице 2).

Таблица 2 - Список поддерживаемых форматов

Функция	Текстовый редактор	Табличный редактор	Редактор презентаций
Открытие для просмотра	xodt, xott, docx, odt, txt, dotx, ott, docm, pdf	xods, xots, xlsx, ods, ots, xltx, csv, scsv, tsv, tab, xlsx, pdf	xodp, pptx, odp, potx, pptm, pdf
Экспорт	xodt, docx, odt, pdf (PDF-a), pdf	xods, xlsx, ods, pdf (PDF-a), pdf	xodp, pptx, odp, pdf (PDF-a), pdf

2. Просмотр содержимого документов, включая текстовые и табличные данные, презентации, диаграммы, изображения и др.

1.3 Уровень подготовки пользователя

Пользователем ПО МойОфис 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 возможно в браузерах: Chrome, Яндекс.Браузер, Mozilla FireFox, Microsoft Edge (Chromium). Версии браузеров для используемой ОС приведены в документе «МойОфис Комплект Средств Разработки (SDK). Сервис просмотра документов. Системные требования».

Для интеграции ПО МойОфис необходим установленный веб-сервер.

Полный перечень требований к программному и аппаратному обеспечению приведен в документе «МойОфис Комплект Средств Разработки (SDK). Сервис просмотра документов. Системные требования».

2 Установка

2.1 Дистрибутив

Дистрибутив ПО МойОфис поставляется в виде архивного файла **MyOffice_DVT_SDK_<release_name>_<build number>.zip**, где **<release_name>** – название релиза, а **<build number>** – номер сборки программы.

Архивный файл содержит следующие данные:

- правовые уведомления;
- папка **wte/dist** с ресурсами для развертывания;
- папка **amr-api/dist** с JavaScript файлом `amrApi.js` – скрипт для взаимодействия внешнего приложения с SDK;
- папка **demo** с демонстрационным приложением для нескольких фреймворков.

2.2 Запуск демо приложения

В папке **demo** находятся демо-приложения для следующих фреймворков:

- Angular;
- Electron;
- React;
- Vite;
- Vue.

Для запуска демо - приложения ПО МойОфис выполните следующие действия:

1. Создайте каталог установки, например, папку **DVT**.
2. Извлеките содержимое архивного файла дистрибутива (см. раздел [Дистрибутив](#)) в каталог установки.
3. Перейдите в папку **demo** каталога установки, далее в папку необходимого фреймворка, и в командной строке последовательно запустите следующие команды:

```
yarn install
yarn build
yarn start
```



Дополнительные сведения о настройке окружения и запуске приложения содержатся в файле README.md.

В случае использования ранних версий AMP (до 2.4) демо приложение запускается в контейнере Docker.

Для этого необходимо предварительно установленное ПО Docker.



В этом случае для установки демо приложения необходимо выполнить следующую последовательность команд:

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

2.3 Особенности использования

Текущие ограничения использования:

1. Документы большого объема могут открываться медленно.
2. Бинарные форматы документов Microsoft (doc, xls, etc) не поддерживаются.
3. Копирование больших объемов текста может занять некоторое время.
4. Для повторного открытия документа необходима перезагрузка iframe.
5. При настройке шаблона для табличных документов необходимо прописать достаточное для пользователя количество строк и столбцов (например, 300).

Для правильной и быстрой работы приложения нужно реализовать следующее:

1. Правильно настроить заголовки для типа файлов WebAssembly (application/wasm для wasm файлов).
2. Подключить правильные заголовки кеширования на уровне HTTP (<https://developer.mozilla.org/ru/docs/Web/HTTP/Headers/Cache-Control>, <https://developer.mozilla.org/en-US/docs/Web/HTTP/Caching>).
3. При перемещении внутри таблиц возможно выполнение встроенных браузерных жестов («назад» и «вперед»). Чтобы избежать срабатывания браузерных жестов навигации, необходимо в элементах <html/> и <body/> страницы интегратора добавить стиль `overscroll-behavior-x: none`.
4. Необходимо предоставить доступ iframe к Clipboard API: `<iframe src="amr_url" allow="clipboard-read; clipboard-write"></iframe>`.
5. Необходимо дождаться события 'load' у iframe, в котором мы запускаем приложение.

6. Подключить статическое сжатие контента (рекомендуется Brotli) (<https://www.smashingmagazine.com/2021/01/front-end-performance-assets-optimizations/#assets-optimizations>). Запрос клиента должен содержать в заголовке `accept-encoding: gzip, br`, сервер должен поддерживать сжатие.

2.4 Проверка работоспособности с использованием демо приложения

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
```

2. Запустите браузер из числа поддерживаемых, например, Mozilla FireFox.

3. В браузере перейдите по адресу:

```
http://localhost:8080
```

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 рисунке 1.

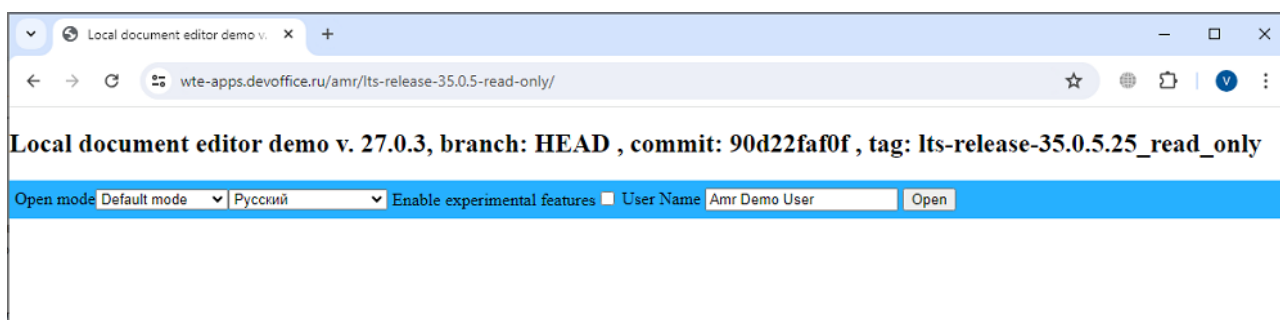


Рисунок 1 – Окно демонстрационного примера использования ПО МойОфис

2. При установке режима **OpenMode=«Read only mode»** и последующем нажатии кнопки **Open** (см. Рисунок 1) и после подтверждения выбора документа произошла успешная загрузка документа в окно просмотра в режиме **read-only** (см. Рисунок 2).

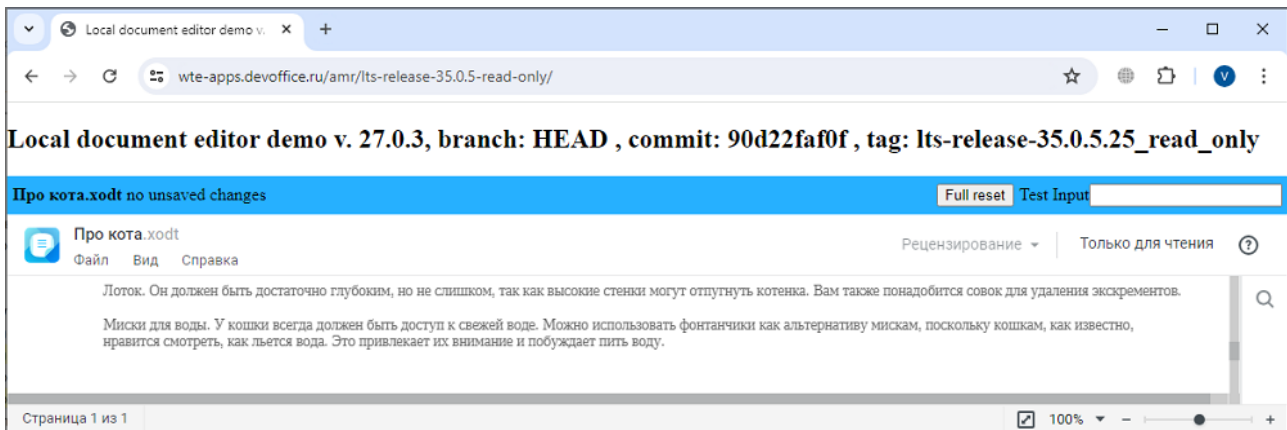


Рисунок 2 – Пример окна редактирования текстового документа

3 Методы и уведомления AMR Readonly API

Содержит методы для открытия документов в режиме просмотра.

3.1 Методы AMR Readonly API

3.1.1 Метод openDocument

Метод openDocument предназначен для открытия документа в режиме просмотра.

```
openDocument = ({
  content: ArrayBuffer,
  filename: string,
  lang: string,
  userName: string,
  workerInitTimeout?: number,
  coreInitTimeout?: number,
}) => DocumentDescription | ErrorDescription
```

Параметры:

- content – содержимое документа (массив байт данных файла);
- filename – название документа, отображается в заголовке редактора;
- lang – язык интерфейса редактора;
 - 'ru-RU' – русский;
 - 'en-US' – английский;
 - 'es-PA' – испанский;
 - 'fr-FR' – французский;
 - 'tt-RU' – татарский;
 - 'ba-RU' – башкирский;
 - 'pt-BR' – португальский;
 - 'de-DE' – немецкий;
 - 'it-IT' – итальянский;
 - 'be-BY' – белорусский;
 - 'kk-KZ' – казахский;
 - 'ky-KG' – киргизский;
 - 'hy-AM' – армянский;

МойОфис

- `userName` – имя пользователя (строка, значение по умолчанию - 'Local User'), которое будет отображено в комментариях и отслеживаемых изменениях;
- `workerInitTimeout` – таймаут загрузки потока ядра, необязательный параметр;
- `coreInitTimeout` – таймаут инициализации ядра, необязательный параметр.

Метод возвращает структуру `DocumentDescription`, если инициализация была успешно выполнена.

```
type DocumentDescription = {
    type: TYPES; // тип документа
    format: OPEN_FORMATS; // формат документа
}

enum TYPES = {
    document = 'document', // текстовые документы
    spreadsheet = 'spreadsheet', // табличные документы
    presentation = 'presentation', // презентации
    pdf = 'pdf' // документы PDF
}

enum OPEN_FORMATS = {
    oxml = 'OXML', // формат Microsoft Office Open XML (docx, xlsx, pptx, dotx, xltx, potx, docm, xlsm, pptm)
    odf = 'ODF', // формат OpenDocument (odt, ods, odp, ott, ots), также используется для открытия файлов XO (xodt, xott, xods, xots, xodp), так как базируется на ODF
    plain = 'PlainText', // текстовый формат
    dsv = 'DSV', // текстовый формат с разделителями, используется для хранения табличных данных (csv, scsv, tsv)
    pdf = 'PDF', // межплатформенный открытый формат электронных документов
    pdfa = 'PDF/A' // ISO-стандартизированный формат PDF/A для долгосрочного архивного хранения электронных документов
}
```

В случае ошибки метод возвращает структуру `ErrorDescription`, содержащую сообщение об ошибке:

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

3.1.2 Метод blur

Метод blur отключает сохранение фокуса в просмотрщике.

```
blur = () => Promise<void>
```

Метод возвращает объект Promise, при этом нет необходимости дожидаться окончания выполнения.

3.2 Уведомления AMR Readonly API

3.2.1 Уведомление onError

```
onError (errorDescription: string)
```

Оповещает о фатальной ошибке при открытии документа.

В случае возникновения ошибки на экране отображается диалоговое окно с подробной информацией об ошибке и с кнопкой запроса перезагрузки. Нажатие кнопки перезагрузки отслеживается с помощью уведомления onPageReloadRequested и требует корректной обработки.

Работа с защищенным паролем документом в настоящее время не поддерживается. При открытии такого документа хост-приложение получит уведомление об ошибке, содержащей строку **"PASSWORD_PROTECTED_DOCUMENT_ERROR"**.

3.2.2 Уведомление onPageReloadRequested

```
onPageReloadRequested()
```

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

4 Пример использования AMR Readonly API

Пример использования методов и уведомлений **AMR Readonly API** при разработке приложений.

```
// создание iframe
<iframe id="viewer" src={http://some.site.name}>

// инициализация API, подключение скрипта для работы с SDK
import AmrApi from 'AmrApi.js';
const origin = window.location.origin;

const amrIframe = document.getElementById('amr');
const amrApi = new AmrApi(viewerIframe, origin);

function onError(errorMessage) {
  console.log('Ошибка открытия документа', err);
}
// обработка ошибки
viewerApi.onError(onError);
viewerApi.onPageReloadRequested(() => window.location.reload());

// открытие документа
const documentData = new UintArray(...);
const docInfo = await editorApi.openDocument({
  content: documentData,
  filename: 'filename',
  lang: 'ru-RU',
});

window.console.log('opened', docInfo);
```