

**ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»**

**Программное обеспечение  
МойОфис Комплект Средств Разработки (SDK)**

**Автономный модуль редактирования**

**3.6**

**Руководство программиста**

**На 30 листах**

**Версия документа: 1**

**Дата публикации: 25.11.2025**

**Москва  
2025**

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

<b>1 Общие сведения .....</b>	<b>6</b>
1.1 Назначение .....	6
1.2 Возможности .....	6
1.3 Уровень подготовки пользователя .....	7
1.4 Системные требования .....	7
<b>2 Установка .....</b>	<b>8</b>
2.1 Дистрибутив .....	8
2.2 Запуск демо приложения .....	8
2.3 Особенности использования .....	9
2.4 Проверка работоспособности с использованием демо приложения .....	11
<b>3 Методы и уведомления AMR API .....</b>	<b>13</b>
3.1 Методы AMR API .....	13
3.1.1 Метод openDocument .....	13
3.1.2 Метод saveDocument .....	15
3.1.3 Метод blur .....	16
3.1.4 Метод getScale .....	16
3.1.5 Метод setScale .....	16
3.1.6 Метод increaseScale .....	17
3.1.7 Метод decreaseScale .....	17
3.1.8 Метод fitToWidth .....	17
3.1.9 Метод getActivePage .....	17
3.1.10 Метод getPagesCount .....	18
3.1.11 Метод goToPage .....	18
3.1.12 Метод getActiveSheetId .....	18
3.1.13 Метод getSheetList .....	18
3.1.14 Метод setActiveSheetId .....	18
3.2 Уведомления AMR API .....	19
3.2.1 Уведомление onChange .....	19
3.2.2 Уведомление onDocumentSaved .....	19

3.2.3 Уведомление onError .....	19
3.2.4 Уведомление onPageReloadRequested .....	20
3.2.5 Уведомление onScaleChanged .....	20
3.2.6 Уведомление onPageActivePageChanged .....	20
3.2.7 Уведомление onPageCountChanged .....	20
3.2.8 Уведомление onPageActiveSheetIdChanged .....	20
3.2.9 Уведомление onSheetListChanged .....	20
<b>4 Пример использования AMR API .....</b>	<b>21</b>
<b>5 Параметры URL-запросов .....</b>	<b>24</b>
5.1 Управление интерфейсом редактора .....	24
5.1.1 Иконка приложения .....	24
5.1.2 Имя файла .....	24
5.1.3 Пункт меню Справка .....	25
5.1.4 Верхняя панель .....	26
5.1.5 Боковая правая панель .....	26
5.1.6 Нижняя панель со статусом и навигацией .....	27
5.1.7 Индикатор загрузки документа .....	27
5.1.8 Режим рецензирования .....	28
5.2 Навигация по документу .....	28
5.2.1 Переход на страницу документа .....	29
5.2.2 Переход на лист / ячейку .....	29

## 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)

## 1 ОБЩИЕ СВЕДЕНИЯ

### 1.1 Назначение

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

### 1.2 Возможности

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

Пользователю доступны следующие возможности:

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

Таблица 2 - Список поддерживаемых форматов

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

2. Редактирование содержимого документов, включая текстовые и табличные данные, презентации, диаграммы, изображения и др.
3. Поиск и замена фрагмента текста в документе.
4. Запуск макрокоманд.

### **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). Автономный модуль редактирования. 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). Автономный модуль редактирования. Системные требования».

## 2 УСТАНОВКА

### 2.1 Дистрибутив

Дистрибутив ПО МойОфис поставляется в виде архивного файла **MyOffice\_SEK\_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 **wte/dist-brotli** с ресурсами для развертывания, сжатая версия;
- папка **amr-api/dist** с JavaScript файлом **amrApi.js** – скрипт для взаимодействия внешнего приложения с SDK;
- папка **demo** с демонстрационным приложением для нескольких фреймворков.

К дистрибутиву дополнительно добавлены файлы приложения, сжатые с помощью алгоритма **Brotli**.

Для использования файлов приложения, сжатых по алгоритму Brotli, необходимо активировать их поддержку веб-сервером.



Некоторые браузеры не поддерживают алгоритм Brotli. В этом случае необходимо использовать файлы без сжатия из каталога **wte/dist**.

Для ускорения загрузки SDK в браузерах, не поддерживающих алгоритм Brotli, рекомендуется сжимать динамически оригинальные файлы веб-сервером с помощью утилиты **gzip** перед использованием.



С дополнительными сведениями по поддержке алгоритма Brotli в браузерах можно ознакомиться на ресурсе <https://caniuse.com/brotli>

### 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 SEK.
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. После сбоя документ автоматически не сохраняется и редактор не восстанавливается.
3. Бинарные форматы документов Microsoft (doc, xls, etc) не поддерживаются.
4. Копирование больших объемов текста может занять некоторое время.
5. Для повторного открытия документа необходима перезагрузка iframe.
6. 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.
7. Реализована возможность предварительной загрузки ресурсов (механизм [prefetch](#)) для ускорения открытия документов. В файле `dist/prefetches.json` содержится список относительных путей к ресурсам. Необходимо прочитать/запросить этот файл на серверной стороне, либо, если такой возможности нет, на клиентской стороне интегратора. Далее вставить на страницу инструкции `<link rel="prefetch" ... />` для прочитанных ресурсов.

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

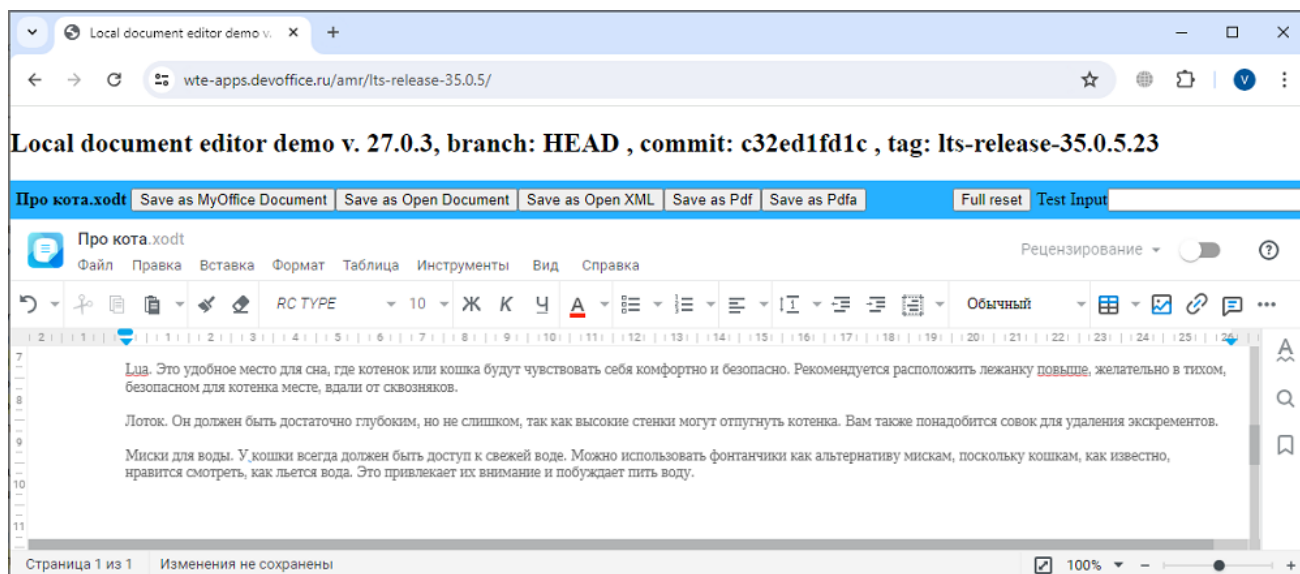


Рисунок 2 – Пример окна редактирования текстового документа

## 3 МЕТОДЫ И УВЕДОМЛЕНИЯ AMR API

Содержит методы для открытия документов в режиме редактирования.

### 3.1 Методы AMR API

#### 3.1.1 Метод openDocument

Метод openDocument предназначен для открытия документа в редакторе.



Перед вызовом данного метода нужно дождаться события load iframe'а редактора.

```
openDocument = ({
  content: ArrayBuffer,
  filename: string,
  lang: string,
  mode: 'review' | 'readonly' | 'default',
  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;
  - 'be-BY' – белорусский;
- mode – параметр, позволяющий выбрать один из следующих режимов редактора:

```
enum Modes {
  // только чтение, без разрешения на редактирование документа
  readonly = "readonly",
```

```
// редактирование документа в режиме рецензирования,  
// без возможности отключить отслеживание изменений  
review = "review",  
// редактирование документа с полными правами  
default = "default"  
}
```

- 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;
- userPermissions – разрешение пользователя, имеет формат `type Partial<Record<Key, boolean>>` (необязательный параметр);
  - canCopy – запрет на копирование из документа;
  - canPrint – запрет на печать;
  - canCopyToExternalClipboard – запрет на копирование из документа в буфер обмена. Value: true (по умолчанию) - разрешено, false - запрещено.

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

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

Где `type` - один из представленных ниже типов документов:

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

И `format` - один из форматов документов:

```
enum OPEN_FORMATS = {  
  // формат Microsoft Office Open XML  
  // (docx, xlsx, pptx, dotx, xltx, potx, docm, xlsx, pptm)
```

```
    oxml = 'OXML',  
    // формат OpenDocument (odt, ods, odp, ott, ots),  
    // также используется для открытия файлов XO  
    // (xodt, xott, xods, xots, xodp), так как базируется на ODF  
    odf = 'ODF',  
    // текстовый формат  
    plain = 'PlainText',  
    // текстовый формат с разделителями,  
    // используется для хранения табличных данных (csv, scsv, tsv)  
    dsv = 'DSV',  
    // межплатформенный открытый формат электронных документов  
    pdf = 'PDF',  
    // ISO-стандартизированный формат PDF/A  
    // для долгосрочного архивного хранения электронных документов  
    pdfa = 'PDFA'  
}
```

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

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

### 3.1.2 Метод `saveDocument`

Метод `saveDocument` предназначен для сохранения редактируемого документа в выбранном формате. После сохранения документ не будет заблокирован или закрыт, пользователь сможет продолжить редактирование документа.

```
saveDocument = (format: SaveDocumentParam) => Uint8Array
```

**Возвращаемый тип:** `Uint8Array`.

Метод возвращает массив байт данных файла.

**Параметр:**

`SaveDocumentParam` - один из форматов документов, возможные значения описаны в `SAVE_FORMATS`:

– XO - формат MyOffice Document. Документ будет сохранён в `*.xodt`, `*.xods`, `*.xodp`;

- OXML - формат OpenDocument. Документ будет сохранен в \*.odt, \*.ods или \*.odp. При сохранении файла необходимо убедиться, что у него правильное расширение или тип MIME;
- ODF - формат Microsoft Office Open XML. Документ будет сохранен в \*.docx, \*.xlsx или \*.pptx;
- PDF - межплатформенный открытый формат электронных документов. Документ будет сохранен в \*.pdf;
- PDF/A - ISO-стандартизированный формат PDF/A для долгосрочного архивного хранения электронных документов. Документ будет сохранен в \*.pdf.

### 3.1.3 Метод blur

Метод blur отключает перехват фокуса редактором.

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

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

### 3.1.4 Метод getScale

Метод getScale возвращает установленный масштаб окна редактора в процентах.

```
getScale = () => Promise<number | null>
```

Метод возвращает объект Promise, который содержит значение в процентах.

### 3.1.5 Метод setScale

Метод setScale устанавливает масштаб окна редактора в процентах.

```
setScale = (percentageScale: number) => Promise<void>
```

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



### 3.1.6 Метод `increaseScale`

Метод `increaseScale` увеличивает значение масштаба окна редактора.

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

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

### 3.1.7 Метод `decreaseScale`

Метод `decreaseScale` уменьшает значение масштаба окна редактора.

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

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

### 3.1.8 Метод `fitToWidth`

Метод `fitToWidth` устанавливает масштаб окна редактора по ширине экрана.

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

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

### 3.1.9 Метод `getActivePage`

Метод `getActivePage` возвращает номер текущей страницы текстового документа.

```
getActivePage = () => Promise<number | null>
```

Метод возвращает объект `Promise`, который может содержать значение ошибки, если открыт документ неподходящего типа.

### 3.1.10 Метод `getPagesCount`

Метод `getPagesCount` возвращает количество страниц текстового документа.

```
getPagesCount = () => Promise<number | null>
```

Метод возвращает объект `Promise`, который может содержать значение ошибки, если открыт документ неподходящего типа.

### 3.1.11 Метод `goToPage`

Метод `goToPage` осуществляет переход к указанной странице текстового документа.

```
goToPage = (page: number) => Promise<void>
```

Метод возвращает объект `Promise`, который может содержать значение ошибки, если открыт документ неподходящего типа.

### 3.1.12 Метод `getActiveSheetId`

Метод `getActiveSheetId` возвращает `id` активного листа табличного документа.

```
getActiveSheetId = () => Promise<number | null>
```

Метод возвращает объект `Promise`, который может содержать значение ошибки, если открыт документ неподходящего типа.

### 3.1.13 Метод `getSheetList`

Метод `getSheetList` возвращает список листов табличного документа.

```
getSheetList = () => Promise<SpreadsheetTabInfo[] | null>
```

Метод возвращает объект `Promise`, который может содержать значение ошибки, если открыт документ неподходящего типа.

### 3.1.14 Метод `setActiveSheetId`

Метод `setActiveSheetId` устанавливает активным указанный лист табличного

документа.

```
setActiveSheetId = (sheetId: number) => Promise<SpreadsheetTabInfo[] | null>
```

Метод возвращает объект Promise, который может содержать значение ошибки, если открыт документ неподходящего типа.

## 3.2 Уведомления AMR API

### 3.2.1 Уведомление onChange

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

Уведомляет о наличии изменений в документе. Например, пользователь может отредактировать документ (флаг будет установлен в true), затем отменить изменения (флаг будет установлен в false), затем внести другие изменения (флаг будет установлен в true).

После вызова метода saveDocument флагу будет присвоено значение false.

### 3.2.2 Уведомление onDocumentSaved

```
onDocumentSaved ()
```

Уведомляет об успешном сохранении документа.

### 3.2.3 Уведомление onError

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

Уведомляет 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 запроса перезагрузки. Нажатие кнопки перезагрузки отслеживается с помощью уведомления [onPageReloadRequested](#) и требует корректной обработки.

Автосохранение/автовосстановление на данный момент не реализовано, все несохраненные изменения будут потеряны.

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”.

#### 3.2.4 Уведомление onPageReloadRequested

```
onPageReloadRequested()
```

Уведомляет о запросе на перезагрузку редактора.

#### 3.2.5 Уведомление onScaleChanged

```
onScaleChanged ({page: number})
```

Уведомляет об изменении масштаба окна редактора.

#### 3.2.6 Уведомление onActivePageChanged

```
onActivePageChanged ({page: number})
```

Уведомляет об изменении текущей активной страницы в текстовом документе.

#### 3.2.7 Уведомление onPagesCountChanged

```
onPagesCountChanged ({pagesCount: number})
```

Уведомляет об изменении количества страниц в текстовом документе.

#### 3.2.8 Уведомление onActiveSheetIdChanged

```
onActiveSheetIdChanged ({sheetId: number})
```

Уведомляет об изменении текущего листа в табличном документе.

#### 3.2.9 Уведомление onSheetListChanged

```
onSheetListChanged ({sheets: SpreadsheetTabInfo[]})
```

Уведомляет об изменении списка листов в табличном документе.

## 4 ПРИМЕР ИСПОЛЬЗОВАНИЯ AMR API

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

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

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

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

function onHasChanges(hasChanges) {
  this.hasChanges = hasChanges;
  if (hasChanges) {
    console.log(DOCUMENT_STATE.changed);
  } else {
    console.log(DOCUMENT_STATE.saved);
  }
}

function onError(errorMessage) {
  console.log('Ошибка открытия документа', err);
}

if (typeof(amrApi.onChange === 'function')) {
  // отслеживание изменений в документе
  amrApi.onChange(({ hasChanges }) => {
    onHasChanges(hasChanges);
  });
}

// обработка ошибки
amrApi.onError(onError);
amrApi.onPageReloadRequested(() => window.location.reload());

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

```
mode: 'default',  
});  
  
window.console.log('opened', docInfo);
```

По окончании редактирования пользователь может получить измененное содержимое документа:

```
const documentData = editorAPI.saveDocument('OXML')  
console.log('Document saved', documentData);
```

Если пользователь не сохранил изменения, то можно предотвратить потерю изменений в документе с помощью предупреждения. Для этого можно использовать событие `onBeforeUnload` и проверку флага `hasChanges`. Значение флага `hasChanges` можно получить из обработчика [onChange](#). Реализуйте обработчик `onBeforeUnload` в `componentDidUpdate`. Пользователь увидит уведомление перед закрытием или перезагрузкой документа с несохраненными изменениями.

Пример установки состояния флага `hasChanges` и добавления/удаления обработчика для события `beforeunload`:

```
if (this.hasChanges) {  
  window.addEventListener("beforeunload", onBeforeUnload, {  
    capture: true,  
  });  
} else {  
  window.removeEventListener("beforeunload", onBeforeUnload, {  
    capture: true,  
  });  
}  
  
function onBeforeUnload(event) {  
  event.preventDefault();  
  return (event.returnValue = 'You have unsaved changes. Are you sure you  
want to leave this page?');  
}
```

После проверки значения `hasChanges`, запрашиваем пользователя о несохраненных изменениях с помощью `confirm`.

```
reload() {  
    if (this.hasChanges) {  
        const result = confirm('You have unsaved changes. Are you sure you  
want to leave this page?');  
        if (result) {  
            window.removeEventListener("beforeunload", this.onBeforeUnload, {  
                capture: true,  
            });  
        } else {  
            return;  
        }  
    }  
    // reload iframe  
}
```

### Ограничения

1. После сбоя документ не сохраняется автоматически, окно редактора не восстанавливается.
2. Бинарные форматы документов (.doc, .xls, .ppt и другие) не поддерживаются.
3. Для повторного открытия документа необходима перезагрузка iframe.

## 5 ПАРАМЕТРЫ URL-ЗАПРОСОВ

### 5.1 Управление интерфейсом редактора

Для управления отображением разных элементов интерфейса редактора реализована поддержка следующих query-параметров.

#### 5.1.1 Иконка приложения

Иконка приложения **appSwitcher**, расположенная на верхней панели слева, может быть скрыта или отображена.

Таблица 1 – Параметр запроса на скрытие иконки приложения

Параметр запроса	Значение параметра	Описание
appSwitcher	<ul style="list-style-type: none"> <li>0</li> <li>false</li> </ul>	иконка приложения скрыта
appSwitcher	<ul style="list-style-type: none"> <li>1 (по умолчанию)</li> <li>любое значение, кроме 0 / false</li> </ul>	иконка приложения отображена

Пример использования **appSwitcher**:

```
<iframe id="amr" src="https://some.site.name/?appSwitcher=0"></iframe>
```

Пример запроса **appSwitcher**:

```
https://some.site.name/?appSwitcher=0
```

#### 5.1.2 Имя файла

Имя файла **fileName**, расположенное на верхней панели слева, над командным меню, может быть скрыто или отображено.

Таблица 2 – Параметр запроса на скрытие имени файла

Параметр запроса	Значение параметра	Описание
fileName	<ul style="list-style-type: none"> <li>0</li> <li>false</li> </ul>	имя файла скрыто



Параметр запроса	Значение параметра	Описание
fileName	<ul style="list-style-type: none"> <li>1 (по умолчанию)</li> <li>любое значение, кроме 0 / false</li> </ul>	имя файла отображено

Пример использования **fileName**:

```
<iframe id="amr" src="https://some.site.name/?fileName=0"></iframe>
```

Пример запроса **fileName**:

```
https://some.site.name/?fileName=0
```

### 5.1.3 Пункт меню Справка

Пункт командного меню "Справка" **help**, иконка "Помощь" могут быть скрыты или отображены.

Таблица 3 – Параметр запроса на скрытие пункта "Справка", иконки "Помощь"

Параметр запроса	Значение параметра	Описание
help	<ul style="list-style-type: none"> <li>0</li> <li>false</li> </ul>	пункт командного меню Справка, иконка Помощь скрыты
help	<ul style="list-style-type: none"> <li>1 (по умолчанию)</li> <li>любое значение, кроме 0 / false</li> </ul>	пункт командного меню Справка, иконка Помощь отображены

Пример использования **help**:

```
<iframe id="amr" src="https://some.site.name/?help=0"></iframe>
```

Пример запроса **help**:

```
https://some.site.name/?help=0
```

### 5.1.4 Верхняя панель

Верхняя панель **topBar** может быть скрыта или отображена.

Таблица 4 – Параметр запроса на скрытие верхней панели

Параметр запроса	Значение параметра	Описание
topBar	<ul style="list-style-type: none"> <li>• 0</li> <li>• false</li> </ul>	верхняя панель скрыта
topBar	<ul style="list-style-type: none"> <li>• 1 (по умолчанию)</li> <li>• любое значение, кроме 0 / false</li> </ul>	верхняя панель отображена

Пример использования **topBar**:

```
<iframe id="amr" src="https://some.site.name/?topBar=0"></iframe>
```

Пример запроса **topBar**:

```
https://some.site.name/?topBar=0
```

### 5.1.5 Боковая правая панель

Боковая правая панель **sideBar** может быть скрыта или отображена.

Таблица 5 – Параметр запроса на скрытие правой боковой панели

Параметр запроса	Значение параметра	Описание
sideBar	<ul style="list-style-type: none"> <li>• 0</li> <li>• false</li> </ul>	правая боковая панель скрыта
sideBar	<ul style="list-style-type: none"> <li>• 1 (по умолчанию)</li> <li>• любое значение, кроме 0 / false</li> </ul>	правая боковая панель отображена

Пример использования **sideBar**:

```
<iframe id="amr" src="https://some.site.name/?sideBar=0"></iframe>
```

Пример запроса **sideBar**:

```
https://some.site.name/?sideBar=0
```

### 5.1.6 Нижняя панель со статусом и навигацией

Нижняя панель со статусом и навигацией **statusBar** может быть скрыта или отображена.

Таблица 6 – Параметр запроса на скрытие нижней панели

Параметр запроса	Значение параметра	Описание
statusBar	<ul style="list-style-type: none"> <li>• 0</li> <li>• false</li> </ul>	нижняя панель скрыта
statusBar	<ul style="list-style-type: none"> <li>• 1 (по умолчанию)</li> <li>• любое значение, кроме 0 / false</li> </ul>	нижняя панель отображена

Пример использования **statusBar**:

```
<iframe id="amr" src="https://some.site.name/?statusBar=0"></iframe>
```

Пример запроса **statusBar**:

```
https://some.site.name/?statusBar=0
```

### 5.1.7 Индикатор загрузки документа

Индикатор загрузки документа **baseLoader** может быть скрыт или отображен.

Таблица 7 – Параметр запроса на скрытие индикатора загрузки документа

Параметр запроса	Значение параметра	Описание
baseLoader	<ul style="list-style-type: none"> <li>• 0</li> <li>• false</li> </ul>	индикатор загрузки документа скрыт
baseLoader	<ul style="list-style-type: none"> <li>• 1 (по умолчанию)</li> <li>• любое значение,</li> </ul>	индикатор загрузки документа отображен

Параметр запроса	Значение параметра	Описание
	кроме 0 / false	

Пример использования **baseLoader**:

```
<iframe id="amr" src="https://some.site.name/?baseLoader=0"></iframe>
```

Пример запроса **baseLoader**:

```
https://some.site.name/?baseLoader=0
```

### 5.1.8 Режим рецензирования

Режим рецензирования **reviewMarkup** может быть скрыт или отображен.

Таблица 8 – Параметр запроса на скрытие режима рецензирования

Параметр запроса	Значение параметра	Описание
reviewMarkup	<ul style="list-style-type: none"> <li>0</li> <li>false</li> </ul>	индикатор загрузки документа скрыт
reviewMarkup	<ul style="list-style-type: none"> <li>1 (по умолчанию)</li> <li>любое значение, кроме 0 / false</li> </ul>	индикатор загрузки документа отображен

Пример использования **reviewMarkup**:

```
<iframe id="amr" src="https://some.site.name/?reviewMarkup=0"></iframe>
```

Пример запроса **reviewMarkup**:

```
https://some.site.name/?reviewMarkup=0
```

## 5.2 Навигация по документу

Для текстовых навигации по документам разного типа реализована поддержка следующих query-параметров.

### 5.2.1 Переход на страницу документа

Для текстовых документов и презентаций реализован запрос **page**, позволяющий перейти на страницу и слайд по порядковому номеру.

Таблица 9 – Параметр запроса перехода на страницу

Параметр запроса	Значение параметра	Описание
page	любое числовое значение, начинающееся с 1	отображена страница с соответствующим порядковым номером

Пример использования **page**:

```
<iframe id="amr" src="https://some.site.name/?page=7"></iframe>
```

Пример запроса **page**:

```
https://some.site.name/?page=7
```

### 5.2.2 Переход на лист / ячейку

Для табличных документов реализованы запросы **sheet** и **cell**, позволяющий перейти на лист или на лист и ячейку.

Таблица 10 – Параметр запроса перехода на лист и ячейку

Параметр запроса	Значение параметра	Описание
sheet	имя листа, соответствует имени листа на нижней панели листов и имени в списке листов	отображен лист с соответствующим названием
cell	адрес ячейки на листе в рабочей области в формате R1C1	курсор установлен на выбранную ячейку

Пример использования **page**:

```
<iframe id="amr" src="https://some.site.name/?sheet=Отчет квартал  
3&cell=D5"></iframe>
```

Пример запроса **page**:

```
https://some.site.name/?sheet=Отчет квартал 3&cell=D5
```