Сайт: r7-consult.ru

R7ConsultLibs: Работа с Excel (.xlsx) файлами в JavaScript

R7ConsultLibs — это мощный инструмент для работы с Excel-файлами в JavaScript. Он предоставляет разработчикам простой и интуитивно понятный API, основанный на библиотеке ExcelJS, для обработки .xlsx-документов в табличном редакторе Р7-Офис.

Перечень текущий функций:

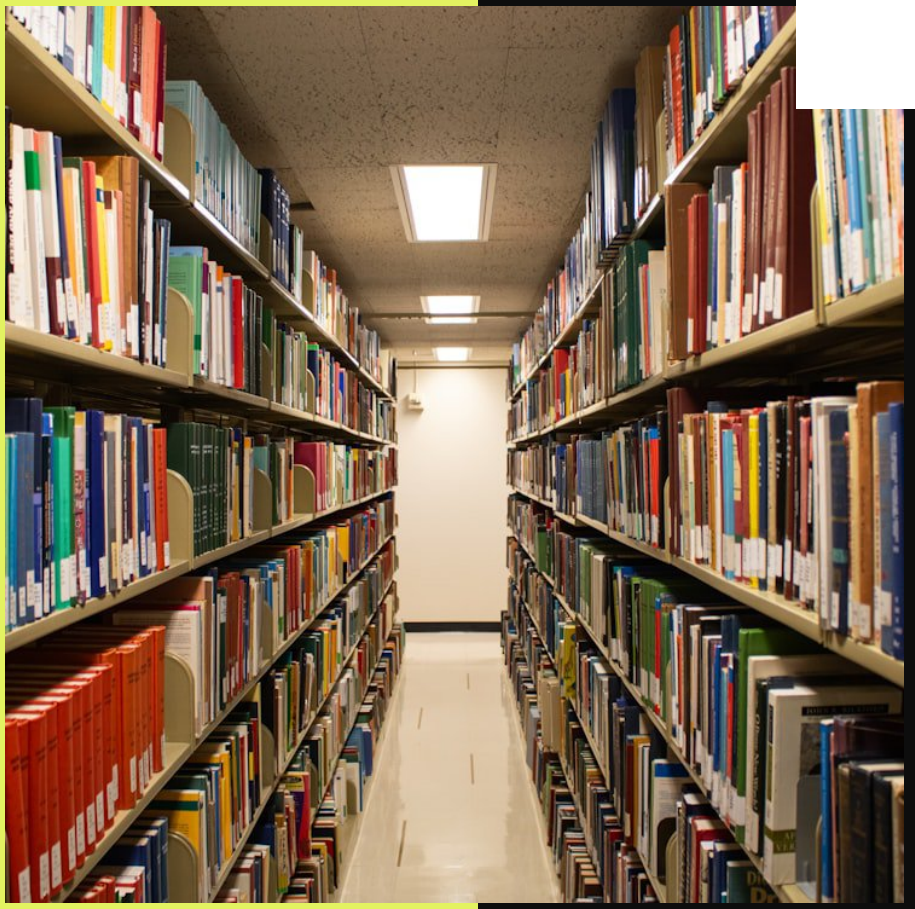
- **Загрузка файлов** — импорт .xlsx-документов во внутреннюю память для последующей обработки.
- **Чтение данных** — извлечение содержимого листов и отдельных ячеек
- **Запись данных** — изменение и добавление новых данных в Excel-файл.
- **Сохранение изменений** — сохранение всех правок в обновлённый файл.
- **Управление памятью** — освобождение ресурсов после завершения работы с файлом.

* Бесплатно для некоммерческого использования

Сайт: r7-consult.ru

Какие проблемы решает R7ConsultLibs?

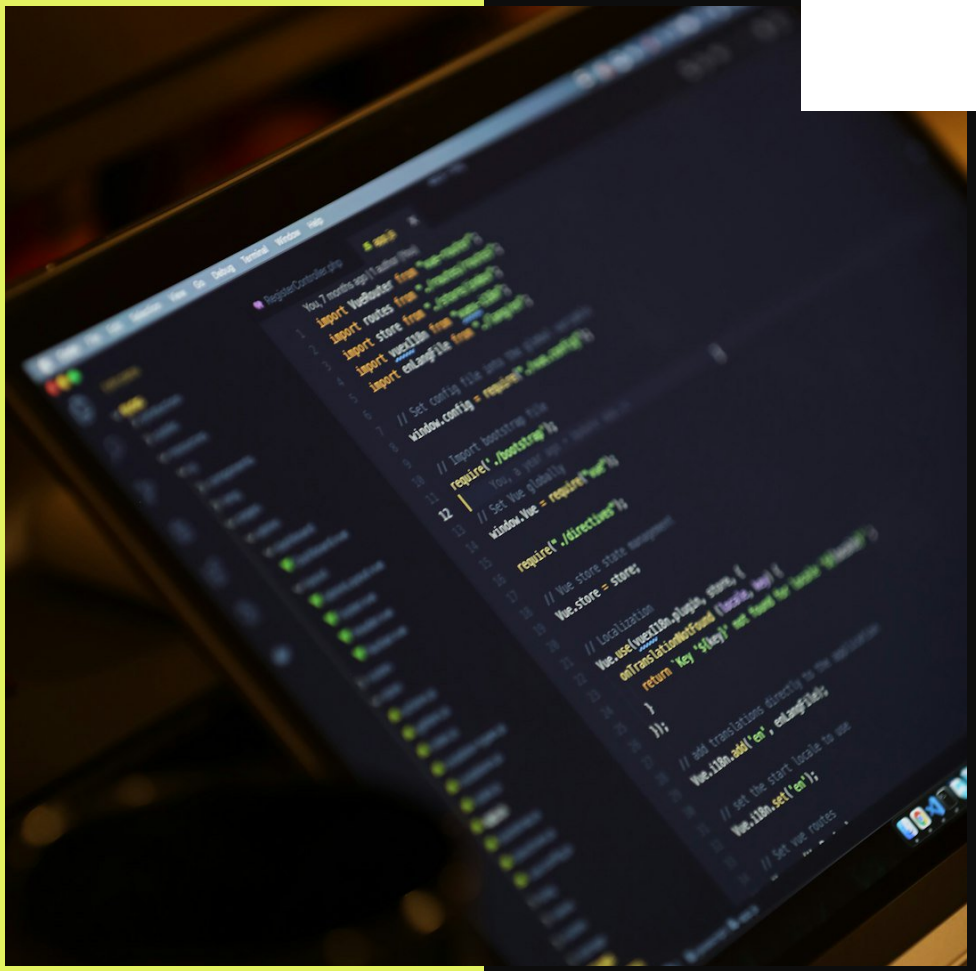
- Использование макросов для чтения и парсинга данных внешних источников Excel из Р7-Офис
- Как дальнейшее развитие продукта
  - Работа с произвольными внешними источниками данных для их обработки и правки через макросы Р7-Офис
  - Интеграция любых внешних библиотек для их использованияв макросах Р7-Офис



Сайт: r7-consult.ru

R7ConsultLibs «под капотом»

- **R7ConsultLibs** — это библиотека для работы с внешними источниками данных в формате Excel (.xlsx) напрямую из макросов редактора Р7-Офис.
- В настоящее время решение реализовано как JavaScript-объект, предназначенный для работы с Excel-файлами в табличном редакторе **Р7-Офис.**
- Технически функциональность **R7ConsultLibs** построена на базе open-source библиотеки ExcelJS, что обеспечивает широкие возможности чтения, записи и обработки Excel-документов в JavaScript-среде.



Сайт: r7-consult.ru

Начало работы с R7ConsultLibs

Перед использованием любых функций объекта ExternalXLSXApi, его необходимо инициализировать. Это обязательный шаг, с которого начинается работа с API.

init();

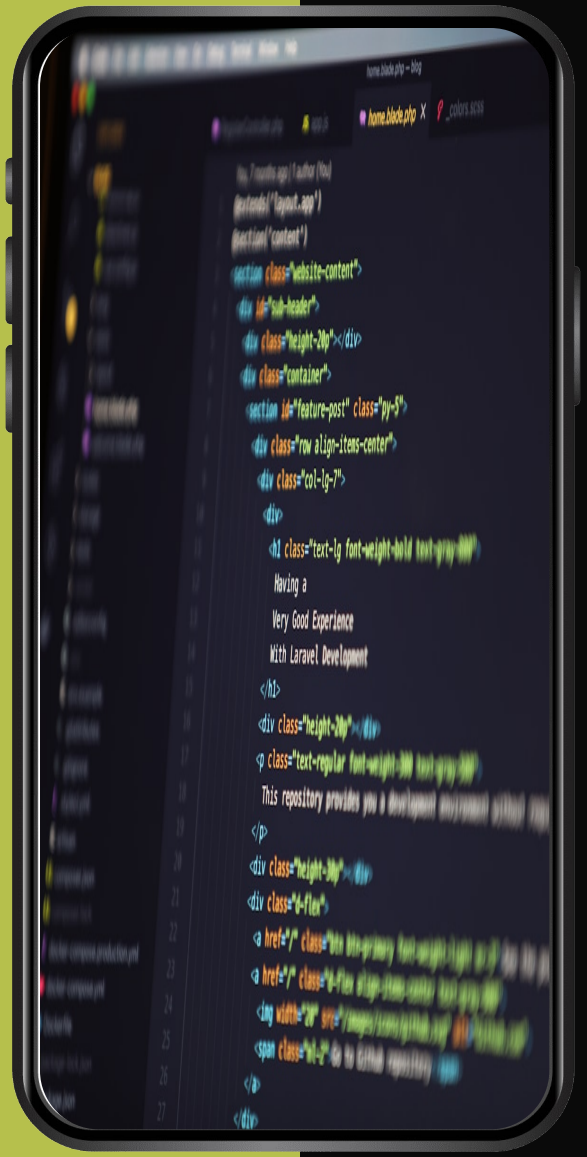
Эта команда запускает API и подготавливает его к работе. В макросе её следует вызывать первой, до выполнения любых других операций

Пример:

Макросы

```
(function(){
    if(Common.R7ConsultLibs)
        console.info("We have R7ConsultLibs library!");
        if(Common.R7ConsultLibs.ExternalXLSXApi){
            let extApi=Common.R7ConsultLibs.ExternalXLSXApi;
            extApi.init();
```

Сайт: r7-consult.ru

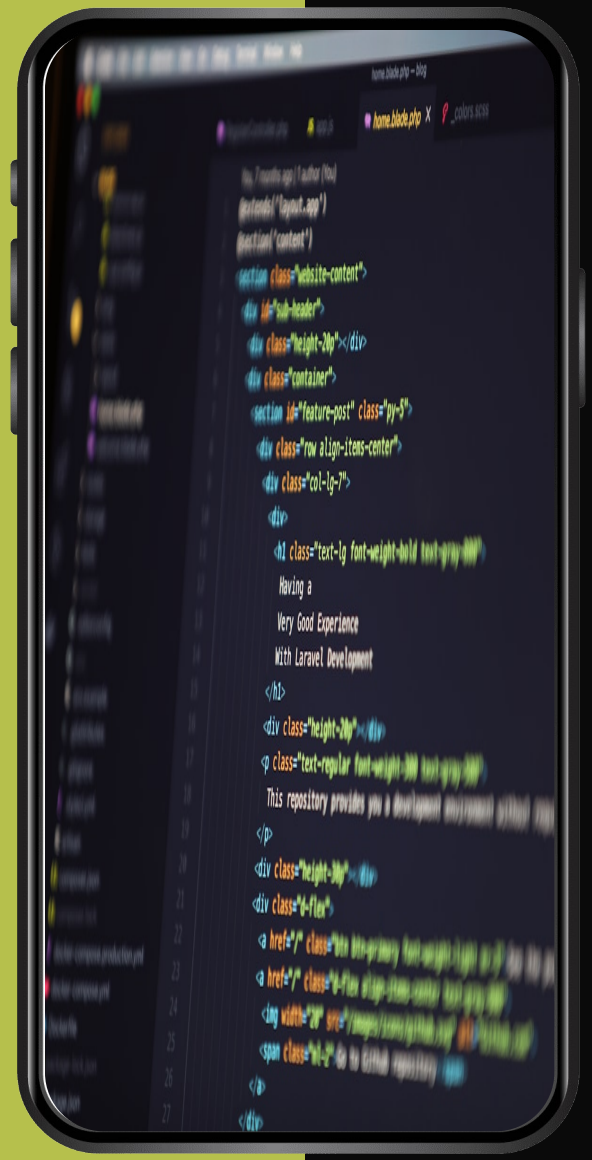


Загрузка файлов Excel: Обзор

Прежде чем читать или записывать данные, необходимо загрузить Excel-файл. Загруженные файлы сохраняются API во внутренней памяти для дальнейшей работы. Существует два способа загрузки:

- **loadWorkbook(filePath, fileData)** - загружает файл Excel из локальных данных (например, из <input type="file"> или других источников)
  - filePath — уникальное имя или путь к файлу
  - fileData — содержимое файла (обычно ArrayBuffer или строка).
- **loadWorkbookFromUrl(filePath, url)** - асинхронно загружает файл Excel по указанному URL
  - filePath — имя файла в памяти.
  - url — ссылка на Excel-файл.

Сайт: r7-consult.ru



Загрузка файлов Excel: Примеры

loadWorkbook:

```
//Загрузка файла из имеющихся данных: `loadWorkbook()
// Предположим, fileArrayBuffer содержит данные вашего .xlsx файла

ExternalXLSXApi.loadWorkbook('filename.xlsx', fileArrayBuffer);
```

loadWorkbookFromUrl:

```
//Загрузка файлов: `loadWorkbookFromUrl`
//Асинхронно загружает файл Excel с указанного URL с помощью `$ajax`.
//Синтаксис: `loadWorkbookFromUrl(filePath, url)`
//`filePath`: Уникальное имя/путь для файла.
//`url`: Веб-адрес файла .xlsx.
//Пример (асинхронная функция):

// Асинхронная функция для загрузки и использования
async function loadFile() {
  try {
    await ExternalXLSXApi.loadWorkbookFromUrl('remotefile_name.xlsx', 'http://example.com/data/spreadsheet.xlsx');
    console.log('Файл загружен!');
    // Теперь можно читать/записывать данные
    const sheetData = ExternalXLSXApi.readSheet('remotefile_name.xlsx', 'Sheet1');
    console.log(sheetData);
  } catch (error) {
    console.error('Ошибка загрузки:', error);
  }
}
loadFile(); // Вызываем функцию
```

Чтение данных: Обзор

После загрузки файла вы можете получать данные из листов и ячеек различными способами — от чтения всего листа до отдельных диапазонов и ячеек.

- **readSheet(filePath, sheetIdentifier)** - читает все данные с указанного листа. Возвращает массив массивов (двумерный массив), представляющий строки и ячейки.
  - filePath — имя файла, использованное при загрузке
  - sheetIdentifier — имя листа (например, 'Sheet1') или его номер
- **readRange(filePath, sheetIdentifier, range)** - читает данные из указанного диапазона ячеек. Возвращает массив массивов с данными из заданного диапазона.
  - range — диапазон ячеек, например 'A1:C5'.
- **readCell(filePath, sheetIdentifier, cellAddress)** - читает значение одной конкретной ячейки. Возвращает значение указанной ячейки
  - cellAddress— диапазон ячеек, например 'A1:C5'.

Сайт: r7-consult.ru

Чтение данных: Примеры

readSheet:

```
const allData = ExternalXLSXApi.readSheet('remote_file_name.xlsx', 'Sheet1');
console.log(allData);
```

readRange:

```
const dataRange = ExternalXLSXApi.readRange('remote_file_name.xlsx', 'Sheet1', 'B2:D4');
console.log(dataRange);
```

readCell:

```
const cellValue = ExternalXLSXApi.readCell('remote_file_name.xlsx', 'Sheet1', 'A1');
console.log(cellValue);
```

Сайт: r7-consult.ru

Сохранение и запись данных в текущем файле: Обзор

После чтения данных из внешнего файла вы можете изменять его содержимое и сохранять свои правки. Работа делится на два этапа: запись и сохранение.

- **writeData(filePath, sheetIdentifier, startCellAddress, dataArray) -** записывает данные в указанный лист, начиная с определённой ячейки.
  - filePath — имя загруженного файла.
  - sheetIdentifier — имя или номер листа.
  - startCellAddress — адрес начальной ячейки (например, 'A1').
  - dataArray — массив массивов с данными (двумерный массив).
    ⚠ Важно: изменения сохраняются только во внутренней памяти — для записи в файл используйте saveWorkbook.
- **saveWorkbook(filePath) -** преобразует изменённый файл из внутренней памяти обратно в формат ArrayBuffer, чтобы вы могли скачать или сохранить его.ArrayBuffer или Promise<ArrayBuffer> — бинарные данные файла
  - filePath — имя файла.

Сайт: r7-consult.ru

Сохранение и запись данных: Примеры

writeData:

```
const newData = [['Hello', 'World'], [123, 456]];
ExternalXLSXApi.writeData('remote_file_name.xlsx', 'Sheet1', 'E1', newData);
console.log('Данные записаны во внутреннюю структуру.');
```

saveWorkbook:

```
async function saveFile() {
  try {
    const обновленныеДанныеФайла = await ExternalXLSXApi.saveWorkbook('remote_file_name.xlsx');
     //Здесь код для сохранения updatedFileData в файл
    console.log('Данные файла получены для сохранения.');
  } catch (error) {
    console.error('Ошибка сохранения:', error);
  }
}
saveFile();
```

Управление памятью: Обзор

Если вы закончили работать с файлом, и он больше не нужен во внутренней памяти, его можно выгрузить из памяти

- **unloadWorkbook(filePath)** - удаляет рабочую книгу из внутренней памяти API.
  - filePath: имя файла, который нужно выгрузить.

Пример:

```
ExternalXLSXApi.unloadWorkbook('remote_file_name.xlsx');
console.log('Файл выгружен из памяти.');
```

Сайт: r7-consult.ru

Пример макроса, использующего R7ConsultLibs

- Макрос запрашивает имя файла и считывает его данные в текущую книгу, начиная с ячейки A1.
- После считывания выдается сообщение, пример которого приведен ниже. В сообщении выводится содержимое ячейки A1.

```
(function(){
    if(Common.R7ConsultLibs)
        console.info("We have R7ConsultLibs library!");
        if(Common.R7ConsultLibs.ExternalXLSXApi){
            let extApi=Common.R7ConsultLibs.ExternalXLSXApi;
            extApi.init();

            var myfile = AscDesktopEditor.OpenFilenameDialog("Excel(*xlsx)"
                ,false, function(_file) {
                // Если файл выбрали
                var file = _file;
                if (Array.isArray(file))
                    file = file[0]; //Если выбрали несколько берем первый,
                        надо потом сделать возможность выбра нескольких
                if (!file)
                    return; // Если не выбран файл закрыть макрос

                file = file.replace(/\\/g,"/"); //Замена Бэкслешей на слэши
                let idFile='test_file';
                loadAndProcessFile(extApi,idFile,file);
            });
        }
})();
```

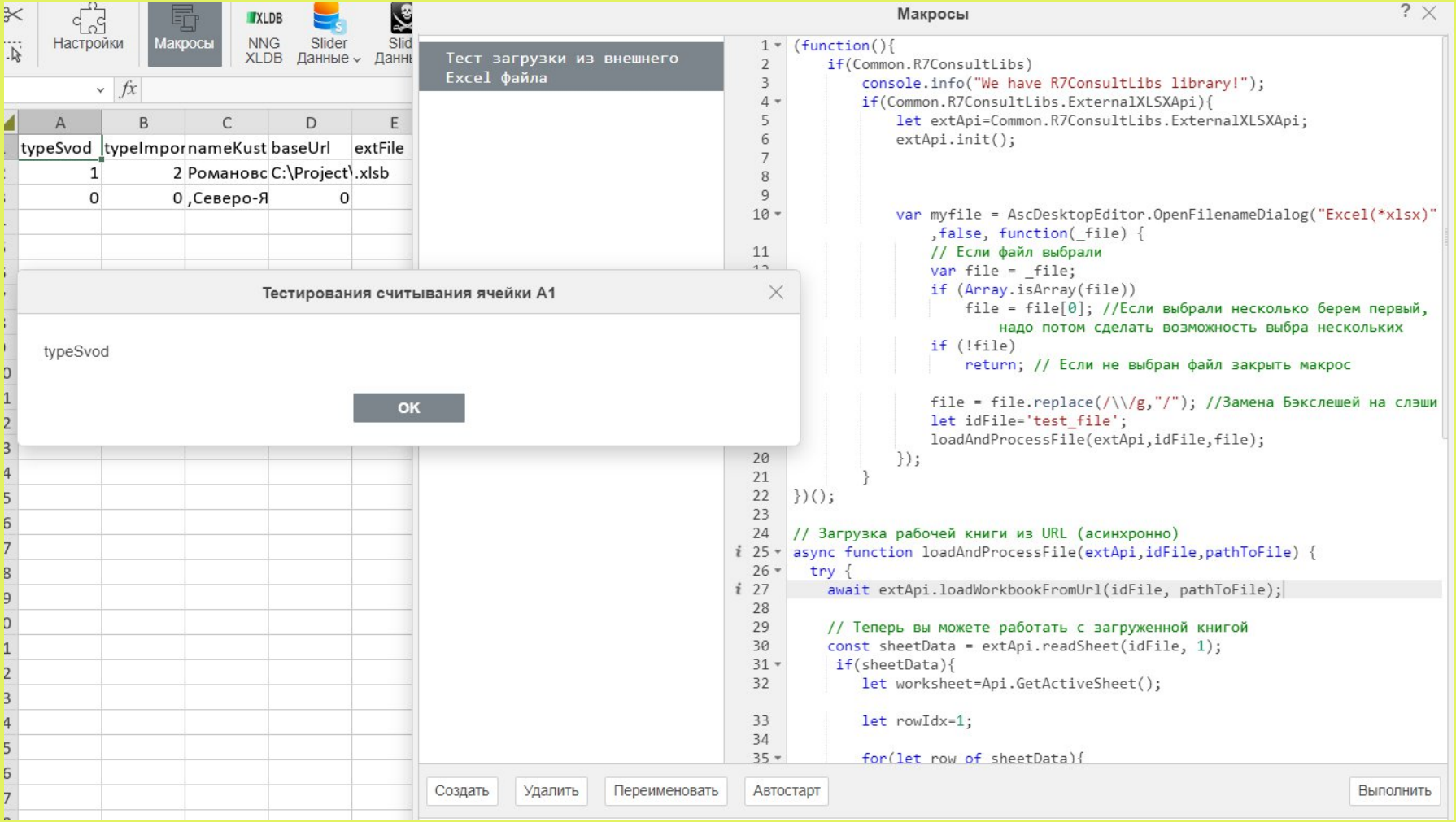
Сайт: r7-consult.ru

Пример макроса, использующего R7ConsultLibs

```
// Загрузка рабочей книги из URL (асинхронно)
async function loadAndProcessFile(extApi,idFile,pathToFile) {
  try {
    await extApi.loadWorkbookFromUrl(idFile, pathToFile);

    // Теперь вы можете работать с загруженной книгой
    const sheetData = extApi.readSheet(idFile, 1);
     if(sheetData){
        let worksheet=Api.GetActiveSheet();

        let rowIdx=1;

        for(let row of sheetData){
            if(row){
                for(let colIdx=1;colIdx<=row.length;colIdx++){
                    worksheet.GetCells(rowIdx,colIdx).SetValue(row[colIdx
                        -1]);
                }
            }
            rowIdx++;
        }

        let cellA1=extApi.readCell(idFile, 1,'A1');
        messageWindow("Тестирования считывания ячейки A1",cellA1);

        Api.asc_calculate(Asc.c_oAscCalculateType.ActiveSheet);
        }

    extApi.unloadWorkbook(idFile); // Выгрузить из кеша после использования

  } catch (error) {
    console.error("Ошибка при загрузке или обработке файла:", error);
  }
}

function messageWindow(title,textMessage){
    Common.UI.alert(
            {
                title: title,
                msg: textMessage,
                width: 600,
                closable: true
            });
}
```

Сайт: r7-consult.ru

Пример макроса, использующего R7ConsultLibs - результат




```
(function(){
    if(Common.R7ConsultLibs)
        console.info("We have R7ConsultLibs library!");
        if(Common.R7ConsultLibs.ExternalXLSXApi){
            let extApi=Common.R7ConsultLibs.ExternalXLSXApi;
            extApi.init();



            var myfile = AscDesktopEditor.OpenFilenameDialog("Excel(*xlsx)"
                ,false, function(_file) {
                // Если файл выбрали
                var file = _file;
                if (Array.isArray(file))
                    file = file[0]; //Если выбрали несколько берем первый,
                        надо потом сделать возможность выбра нескольких
                if (!file)
                    return; // Если не выбран файл закрыть макрос

                file = file.replace(/\\/g,"/"); //Замена Бэкслешей на слэши
                let idFile='test_file';
                loadAndProcessFile(extApi,idFile,file);
            });
        }
})();

// Загрузка рабочей книги из URL (асинхронно)
async function loadAndProcessFile(extApi,idFile,pathToFile) {
  try {
    await extApi.loadWorkbookFromUrl(idFile, pathToFile);

    // Теперь вы можете работать с загруженной книгой
    const sheetData = extApi.readSheet(idFile, 1);
     if(sheetData){
        let worksheet=Api.GetActiveSheet();

        let rowIdx=1;

        for(let row of sheetData){
```

Вопросы? / Контакты

Сайт: r7-consult.ru
Email: er@exceldb.pro
Телефон: +7 915 258-0371
Telegram: https://t.me/r7_js
GitHub:Репозиторий R7ConsultLibs