API GS46 IO

Описание

Evgenii Balakhonov

2021

# Оглавление

[1 Оглавление 1](#_Toc84465531)

[2 Введение 2](#_Toc84465532)

[3 Аутентификация 2](#_Toc84465533)

[4 Описание методов API 3](#_Toc84465534)

[4.1 Получение шаблона 3](#_Toc84465535)

[4.2 Запуск импорта данных 4](#_Toc84465536)

[4.3 Запуск экспорта данных 5](#_Toc84465537)

[4.4 Получение списка обрабатываемых файлов и их статусов 6](#_Toc84465538)

[4.5 Получение файла 7](#_Toc84465539)

[4.6 Коды ошибок 8](#_Toc84465540)

[4.7 Статусы обработки файла 8](#_Toc84465541)

[4.8 Выполнение методов под правами другого пользователя 8](#_Toc84465542)

# Введение

API GS46 IO предназначено для автоматизации обмена данными с каталогом ГС46 в виде файлов Excel, соответствующих заранее определенным шаблонам.

API реализовано на базе протокола SOAP 1.1 и доступно по URL:

<https://srs.gs1ru.org/io/ws>

WSDL файл доступен по URL

<https://srs.gs1ru.org/io/ws/GS46IO.wsdl>

Поддерживаются следующий типы шаблонов:

| **Идентификатор шаблона** | **Описание шаблона** |
| --- | --- |
| MDO | [Шаблон загрузки данных обувных изделий](https://srs.gs1ru.org/io/ui/tpl/MDO) |
| FURS | [Шаблон загрузки данных меховых изделий](https://srs.gs1ru.org/io/ui/tpl/FURS) |
| MDLP | [Шаблон загрузки данных МДЛП](https://srs.gs1ru.org/io/ui/tpl/MDLP) |
| CLOTHING | [Шаблон загрузки данных предметов одежды](https://srs.gs1ru.org/io/ui/tpl/CLOTHING) |
| LINEN | [Шаблон загрузки данных белья](https://srs.gs1ru.org/io/ui/tpl/LINEN) |
| PARFUM | [Шаблон загрузки данных парфюма](https://srs.gs1ru.org/io/ui/tpl/PARFUM) |
| PHOTO | [Шаблон загрузки данных фотокамер](https://srs.gs1ru.org/io/ui/tpl/PHOTO) |
| TIRES | [Шаблон загрузки данных шин](https://srs.gs1ru.org/io/ui/tpl/TIRES) |
| WHEELCHAIRS | [Шаблон загрузки данных кресел-колясок](https://srs.gs1ru.org/io/ui/tpl/WHEELCHAIRS) |
| DAIRY | [Шаблон загрузки данных молочных продуктов](https://srs.gs1ru.org/io/ui/tpl/DAIRY) |
| BICYCLES | [Шаблон загрузки данных велосипедов](https://srs.gs1ru.org/io/ui/tpl/BICYCLES) |
| FRAMES | [Шаблон загрузки данных велосипедных рам](https://srs.gs1ru.org/io/ui/tpl/FRAMES) |
| ALTTOBACCO | [Шаблон загрузки данных альтернативной табачной продукции](https://srs.gs1ru.org/io/ui/tpl/ALTTOBACCO) |
| NICTOBACCO | [Шаблон загрузки данных никотиносодержащей продукции](https://srs.gs1ru.org/io/ui/tpl/NICTOBACCO) |
| CIGARETTES | [Шаблон загрузки данных сигарет](https://srs.gs1ru.org/io/ui/tpl/CIGARETTES) |
| WATER | [Шаблон загрузки данных воды](https://srs.gs1ru.org/io/ui/tpl/WATER) |
| SETS | [Шаблон загрузки данных наборов](https://srs.gs1ru.org/io/ui/tpl/SETS) |
| REREG | [Шаблон перерегистрации продукции](https://srs.gs1ru.org/io/ui/tpl/REREG)  |
| BRANCH | [Шаблон регистрации филиалов](https://srs.gs1ru.org/io/ui/tpl/BRANCH)  |

# Аутентификация

Для использования API необходима аутентификация пользователя. В настоящий момент используется простой метод HTTP Basic.

Пользователь, который должен иметь одну из следующих ролей:

MDLP\_LOAD,MDO\_LOAD,CLOTHING\_LOAD,LINEN\_LOAD,PARFUM\_LOAD,PHOTO\_LOAD,TIRES\_LOAD,WHEELCHAIRS\_LOAD,DAIRY\_LOAD,BICYCLES\_LOAD,FRAMES\_LOAD,ALTTOBACCO\_LOAD,WATER\_LOAD,SETS\_LOAD,FURS\_LOAD,NICTOBACCO\_LOAD,CIGARETTES\_LOAD

Кроме этого пользователь должен иметь общую роль DATA\_UPLOAD\_DOWNLOAD

Для того, чтобы пользователь мог выполнять экспорт/импорт данных под другим пользователем, он должен иметь соответственно роли GS46\_DATA\_READER\_RUNAS или GS46\_DATA\_WRITER\_RUNAS

# Описание методов API

## Получение шаблона

Для получения шаблона в формате Excel используется метод GetTemplate. Пример запроса:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:io="http://gs1ru.org/io">

 <soapenv:Header/>

 <soapenv:Body>

 <io:GetTemplateRequest tpl="MDO"/>

 </soapenv:Body>

</soapenv:Envelope>

Здесь:

* tpl – код шаблона (см. раздел 2).

Пример ответа:

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

 <env:Header/>

 <env:Body>

 <ns2:GetTemplateResponse xmlns:ns2="http://gs1ru.org/io">

 <ns2:tpl>

 <ns2:fileName>MDO.xlsx</ns2:fileName>

 <ns2:fileData>

 <xop:Include href="cid:d644d3b9-2682-4551-a515-988e595e2c7b%40gs1ru.org" xmlns:xop="http://www.w3.org/2004/08/xop/include"/>

 </ns2:fileData>

 </ns2:tpl>

 <ns2:result result="OK"/>

 </ns2:GetTemplateResponse>

 </env:Body>

</env:Envelope>

Здесь:

* fileName – имя файла шаблона
* fileData – данные файла. Для передачи бинарных данных используется [MTOM](https://en.wikipedia.org/wiki/Message_Transmission_Optimization_Mechanism).

## Запуск импорта данных

Для запуска импорта данных файла, оформленному по какому либо из поддерживаемых шаблонов необходимо использовать метод StartImportRequest. Пример запроса:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:io="http://gs1ru.org/io">

 <soapenv:Header/>

 <soapenv:Body>

 <io:StartImportRequest>

 <io:fileData>

 <io:fileName>LOREALREQ.xlsm</io:fileName>

 <io:fileData>cid:LOREALREQ.xlsm</io:fileData>

 </io:fileData>

 </io:StartImportRequest>

 </soapenv:Body>

</soapenv:Envelope>

где:

* fileName – имя загружаемого файла
* fileData – данные файла. Рекомендуется использовать технологию [MTOM](https://en.wikipedia.org/wiki/Message_Transmission_Optimization_Mechanism) при загрузке данных файла! Поддерживается также передача бинарных данных, кодированных в Base64 (не рекомендуется)

Пример ответа:

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

 <env:Header/>

 <env:Body>

 <ns2:StartImportResponse xmlns:ns2="http://gs1ru.org/io">

 <ns2:fileInfo fileName="LOREALREQ.xlsm" fileId="4600007999992\_IMPORT\_LOREALREQ\_211007012742$LOREALREQ.xlsm" templateType="LOREALREQ"/>

 <ns2:result result="OK"/>

 </ns2:StartImportResponse>

 </env:Body>

</env:Envelope>

Здесь:

* fileInfo – данные по файлу, который был загружен на обработку, в данном тэге:
	+ fileName – имя загруженного файла
	+ fileId – уникальный идентификатор файла, который был ему назначен обработкой
	+ templateType – идентификатор распознанного шаблона файла
* result – статус выполнения операции: OK или код ошибки. Коды возможных ошибок приведены в разделе 4.6. В случае ошибки в атрибуте descr будет дано описание ошибки.

## Запуск экспорта данных

Для запуска экспорта данных используется метод StartExportRequest. Пример запроса:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:io="http://gs1ru.org/io">

 <soapenv:Header/>

 <soapenv:Body>

 <io:StartExportRequest templateType="MDO">

 </io:StartExportRequest>

 </soapenv:Body>

</soapenv:Envelope>

Здесь:

* templateType – код шаблона в котором необходимо экспортировать данные (см. раздел 2).

Пример ответа:

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

 <env:Header/>

 <env:Body>

 <ns2:StartExportResponse xmlns:ns2="http://gs1ru.org/io">

 <ns2:file fileName="MDO-20211007014837.xlsx" templateType="MDO" fileId="4600007999992\_EXPORT\_MDO\_211007014837$MDO-20211007014837.xlsx" cDateTime="2021-10-07T01:48:37.727+03:00" status="WAIT" percent="0.0"/>

 <ns2:result result="OK"/>

 </ns2:StartExportResponse>

 </env:Body>

</env:Envelope>

Здесь:

* file – данные экспортируемого файла
	+ fileName – имя файла
	+ templateType – код используемого шаблона (см. раздел 2).
	+ fileId – уникальный идентификатор файла в обработке
	+ cDateTime – дата-время начала обработки
	+ status – статус обработки (см. раздел 4.7)
	+ percent – процент выполнения задачи экспорта
* result – статус выполнения операции: OK или код ошибки. Коды возможных ошибок приведены в разделе 4.6. В случае ошибки в атрибуте descr будет дано описание ошибки.

## Получение списка обрабатываемых файлов и их статусов

Для получения списка обрабатываемых файлов используется метод GetFileListRequest. Пример запроса:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:io="http://gs1ru.org/io">

 <soapenv:Header/>

 <soapenv:Body>

 <io:GetFileListRequest direction="EXPORT" tpl="MDO" />

 </soapenv:Body>

</soapenv:Envelope>

Здесь:

* direction – EXPORT (для экспортируемых данных) или IMPORT (для импортируемых данных)
* tpl – код шаблона для которого нужно получить список обрабатываемых файлов (см. раздел 2).

В ответе возвращается весь перечень обработанных файлов. Пример ответа:

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

 <env:Header/>

 <env:Body>

 <ns2:GetFileListResponse xmlns:ns2="http://gs1ru.org/io">

 <ns2:file fileName="MDO-20211007014837.xlsx" templateType="MDO" fileId="4600007999992\_EXPORT\_MDO\_211007014837$MDO-20211007014837" cDateTime="2021-10-06T22:48:37.000+03:00" eDateTime="2021-10-07T01:48:39.000+03:00" status="COMPLETED" percent="100.0" totalQnt="227"/>

 <ns2:file fileName="MDO-20210725100539.xlsx" templateType="MDO" fileId="4600007999992\_EXPORT\_MDO\_210725100539$MDO-20210725100539" cDateTime="2021-07-25T07:05:39.000+03:00" eDateTime="2021-07-25T10:05:40.000+03:00" status="COMPLETED" percent="100.0" totalQnt="12"/>

 <ns2:result result="OK"/>

 </ns2:GetFileListResponse>

 </env:Body>

</env:Envelope>

Здесь:

* file – блок данных, описывающих конкретный экземпляр файла, в нем:
	+ fileName – имя файла
	+ templateType – идентификатор использованного шаблона (см. раздел 2).
	+ fileId – уникальный идентификатор обрабатываемого файла
	+ cDateTime – дата время начала обработки
	+ eDateTime – дата время конца обработки
	+ status – статус исполнения задачи (см. раздел 2).
	+ percent – процент исполнения задачи
	+ totalQnt – число записей в файле
* result – статус выполнения операции: OK или код ошибки. Коды возможных ошибок приведены в разделе 4.6. В случае ошибки в атрибуте descr будет дано описание ошибки.

## Получение файла

Для получения файла используется метод GetFile. Пример запроса:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:io="http://gs1ru.org/io">

 <soapenv:Header/>

 <soapenv:Body>

 <io:GetFileRequest >

 <io:fileId>4600007999992\_EXPORT\_MDO\_211007014837$MDO-20211007014837</io:fileId>

 </io:GetFileRequest>

 </soapenv:Body>

</soapenv:Envelope>

Здесь:

* fileId – уникальный идентификатор файла

Пример ответа:

<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">

 <env:Header/>

 <env:Body>

 <ns2:GetFileResponse xmlns:ns2="http://gs1ru.org/io">

 <ns2:fileState fileName="MDO-20211007014837" templateType="MDO" fileId="4600007999992\_EXPORT\_MDO\_211007014837$MDO-20211007014837" cDateTime="2021-10-07T01:48:37.000+03:00"/>

 <ns2:file>

 <ns2:fileName>MDO-20211007014837.xlsx</ns2:fileName>

 <ns2:fileData>

 <xop:Include href="cid:400f6911-6825-462e-abf9-6c3748271164%40gs1ru.org" xmlns:xop="http://www.w3.org/2004/08/xop/include"/>

 </ns2:fileData>

 </ns2:file>

 <ns2:result result="OK"/>

 </ns2:GetFileResponse>

 </env:Body>

</env:Envelope>

Здесь:

* file – блок данных, описывающий файл
	+ fileName – имя файла
	+ fileData – данные файла. Для передачи бинарных данных используется [MTOM](https://en.wikipedia.org/wiki/Message_Transmission_Optimization_Mechanism).
* result – статус выполнения операции: OK или код ошибки. Коды возможных ошибок приведены в разделе 4.6. В случае ошибки в атрибуте descr будет дано описание ошибки.

## Коды ошибок

* UNKNOWN – неизвестная ошибка
* WAIT – ожидание ресурсов
* ERROR – возникла ошибка, описание в атрибуте descr
* IN\_PROCESS – задача в обработке
* OK – нет ошибки
* INCORRECT\_PARAMETERS – неверные параметры запроса
* HAS\_SOME\_ERRORS – файл имеет ошибки в отдельных строках
* COMPLETED\_WITH\_SOME\_ERRORS – обработка завершена, но были ошибки
* UNKNOWN\_TEMPLATE\_TYPE – неизвестный тип шаблона
* TEMPLATE\_IS\_BROKEN – шаблон испорчен
* ACCESS\_DENIED – доступ запрещен

## Статусы обработки файла

* UNKNOWN – неизвестный статус
* WAIT – ожидание ресурсов
* ERROR - ошибка
* IN\_PROCESS – в обработке
* TEMPLATE\_IS\_BROKEN – шаблон испорчен
* HAS\_SOME\_ERRORS – в отдельных строках файла имеются ошибки
* COMPLETED – обработка успешно завершена
* COMPLETED\_WITH\_SOME\_ERRORS – обработка завершена при наличии отдельных ошибок

## Выполнение методов под правами другого пользователя

При вызове методов API имеется возможность их запуска под правами другого пользователя. Это полезно, когда сам метод вызывается под аутентификацией технического пользователя, а выполнится он должен от имени реального пользователя другого предприятия.

Для этого пользователь, вызывающий методы, должен иметь роли:

* Для метода StartImportOperation: GS46\_DATA\_WRITER\_RUNAS
* Для метода StartExportOperation: GS46\_DATA\_READER\_RUNAS
* Для методов GetFile, GetFileList: GS46\_DATA\_READER\_RUNAS или GS46\_DATA\_WRITER\_RUNAS

Для указания логина пользователя, от имени которого должен выполниться метод, используется атрибут runAs. Например:

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:io="http://gs1ru.org/io">

 <soapenv:Header/>

 <soapenv:Body>

 <io:StartImportRequest runAs=”4600001998724”>

 <io:fileData>

 <io:fileName>LOREALREQ.xlsm</io:fileName>

 <io:fileData>cid:LOREALREQ.xlsm</io:fileData>

 </io:fileData>

 </io:StartImportRequest>

 </soapenv:Body>

</soapenv:Envelope>