MIBGAS

MERCADO IBÉRICO DEL GAS

MIBGAS WEB SERVICES SETUP GUIDE USING SOAP UI

Version 1.0 - 24/05/2023

Index

1. INTRODUCTION AND GLOSSARY

MIBGAS offers Market Participants an API to automate the consultation and exchange of information between market participants and the market. This is a Web Services API. The agents make requests by sending SOAP messages. If the message is correct, the system returns a response with the information in XML format. It is also possible to receive a file attached to the response in other formats such as PDF for some specific requests.

All the information about MIBGAS Web Services can be found in the Help and Documentation section of the MIBGAS agents private web site (www.market.mibgas.es).

ANNEX A contains some basic concepts to understand the Web Services messages.

Glossary of terms

API	Application Programming Interface
SOAP	Simple Object Access Protocol
TLS	Transport Layer Security
WS	Web Services
WSDL	Web Services Description Language
XML	eXtensible Markup Language

2. SOAP UI SETUP

MIBGAS provides an example SOAP-EXAMPLE.zip file containing the basic elements for the Agents to configure the SOAP request invocation using the SOAP-UI application. The main use case of this application is the testing of APIs, being able to use others for this purpose. This one has been chosen because it is commonly used and because of its suitability for testing the MIBGAS API.

Step 0: SOAP UI Installation

The user can invoke the MIBGAS web services with his own client or use the SOAP UI software that has been used as a reference for this guide.

- To install it go to the website https://www.soapui.org/downloads/soapui/

Paso 1: Setup file

The user should modify the **SoapUI-5.7.0.vmoptions** file located in the local configuration folder of the user running the program:

- "%LOCALAPPDATA%\SmartBear\SoapUI-5.7.0\bin\SoapUI-5.7.0.vmoptions"

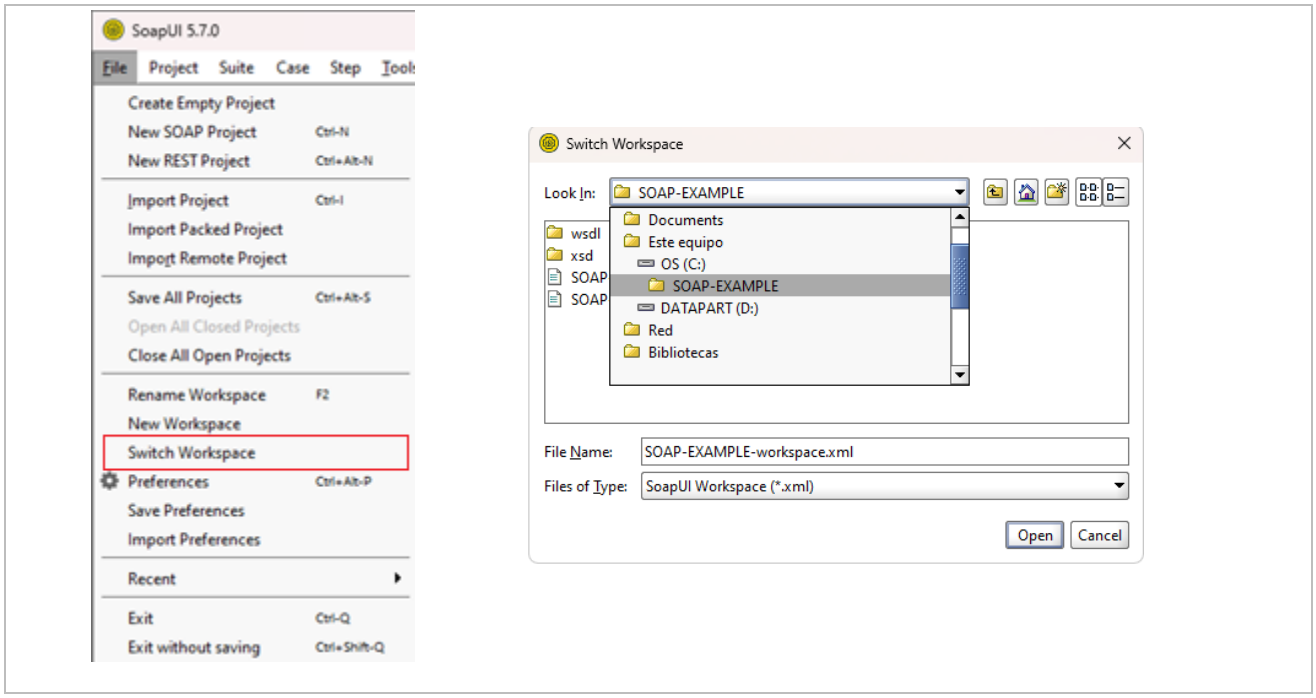
The user has to modify the version of the SOAP UI connectivity TLS protocol. The TLS (Transport Layer Security) protocols allow to ensure encrypted and secure communication between the Agents and MIBGAS, guaranteeing the integrity of the message and the identity of the sender. To do this, open the SoapUI-5.7.0.vmoptions file with a text editor and add the following line at the end of the document:

- -Dsoapui.https.protocols=TLSv1.2

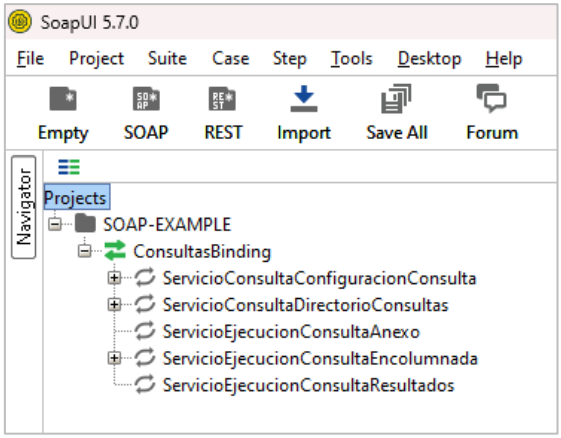
Save the document and launch SOAP UI

Step 2: Workspace Setup

Save and unzip the provided file **SOAP-EXAMPLE.zip** to the path C:/. Launch the SOAP UI program and in the upper menu click on File/Switch Workspace. Find the path C:/SOAP-EXAMPLE with the SOAP browser that appears and select **SOAP-EXAMPLE-workspace.xml**.



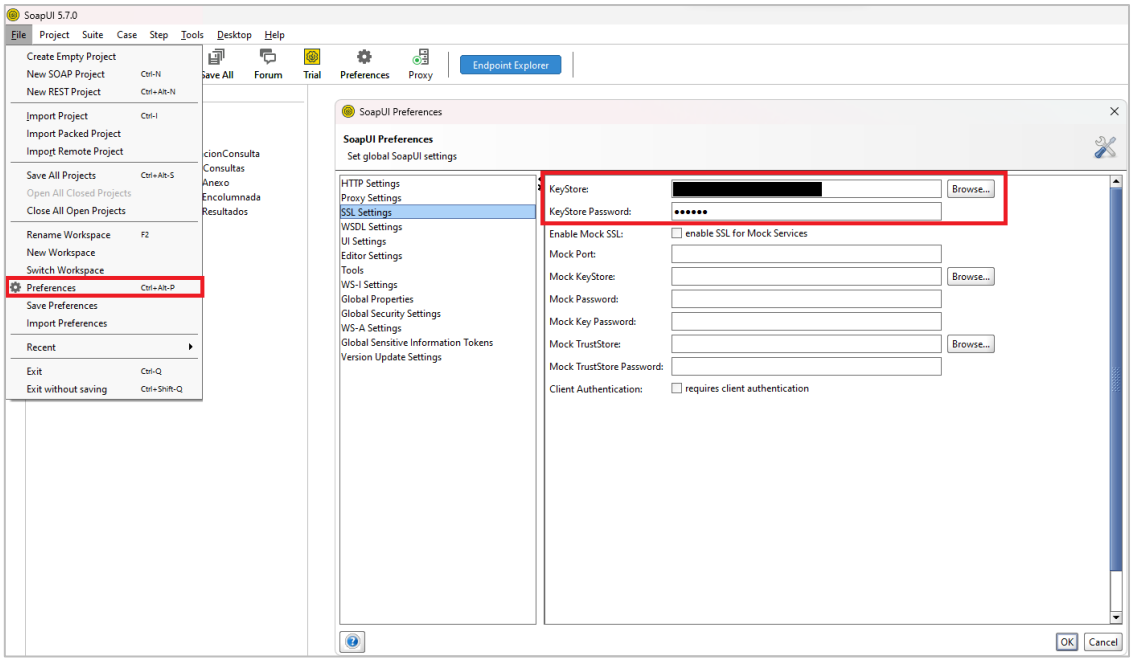
A project with the five basic request types of the MIBGAS Web Service will be shown in the left sidebar under the section ConsultasBinding..



Step 3: Electronic certificate setup

Authentication on the MIBGAS platform is performed via an electronic certificate issued by MIBGAS. Additionally, certain services in which the agent sends information (e.g. UMM messages) require that the message be signed with the electronic certificate. In any case, as mentioned above, for all exchanges via Web Services it is necessary to authenticate using the electronic certificate issued by MIBGAS, which is used to access the market website (www.market.mibgas.es) or the trading module. To configure it in SOAP, click on File/Preferences in the top menu. In the configuration window select the SSL Settings option and in the *KeyStore* field choose the certificate by clicking on the *Browse* button. Once the .p12

certificate issued by MIBGAS has been selected, enter the import password in the KeyStore Password field and save the configuration by clicking the *OK* button.



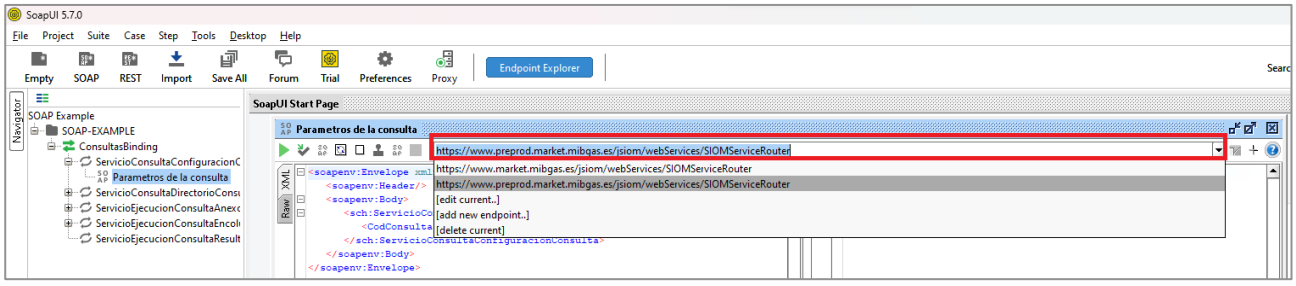
Step 4: Endpoint setup

The endpoint is the path to which the web service points to and to which it makes the request.

Entorno	Endpoint
PRODUCTION (Real)	https://www.market.mibgas.es/jsiom/webServices/SIOMServiceRouter
PREPROD (Testing)	https://www.preprod.market.mibgas.es/jsiom/webServices/SIOMServiceRouter

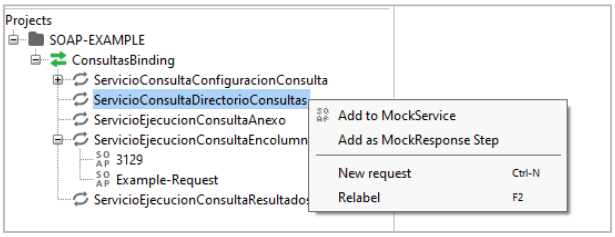
It is advisable to test and configure the requests in the testing environment before making a request to the production environment.

The endpoint can be selected in the top bar of the Request Editor when double clicking on a request.



3. SENDING A REQUEST

To send a Web Service message and make a request, expand one of the General Query Service types in *ConsultasBinding*, right-click on the service and select New Request. A new message will be created with the schema of this service type.



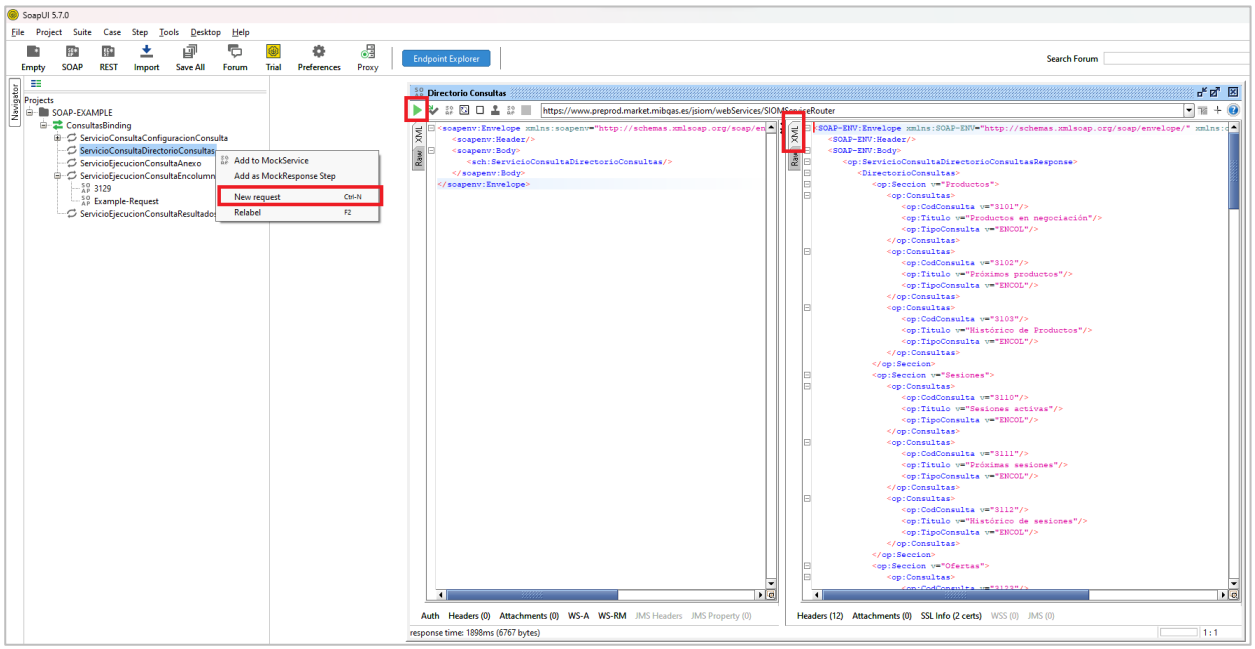
3.1 Query Directory

The **ServicioConsultaDirectorioConsultas** service allows the user to request a list of all the general requests available for invocation by Web Services. It returns a list with the codes and descriptions of all the available requests.

In this case, the SOAP message for this request is very simple and requires no configuration. After creating a new SOAP message (New Request) it is not necessary to modify the message.

To send the request, click on the play button. You will get a response in two formats: Raw and XML By clicking on the XML tab, the user can read the response encoded in this format.

The information of the response appears grouped by section (Products, Sessions, Orders, Prices...) and for each element of the list it gives us the Code of each request, its title, and the type of Request (Columnar, Attached File or Market Results).

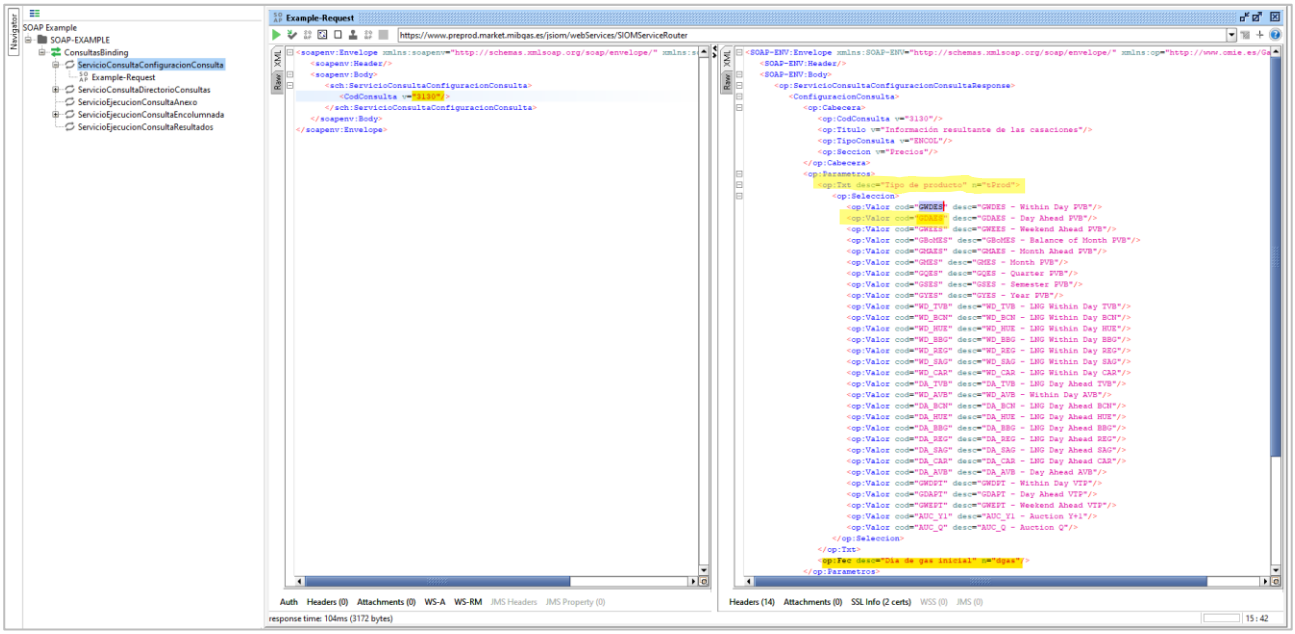


3.2 Characteristics of a query

Once the request to be made and its type are known, it is necessary to retrieve the parameters required by that request. To retrieve the configuration parameters of each request, enter the request code into the **ServicioConsultaConfiguacionConsulta** service**.** Right-click on the service, click on New Request and replace the '?' symbol with the code of the request in the field <CodConsulta v='?'>. Like this:

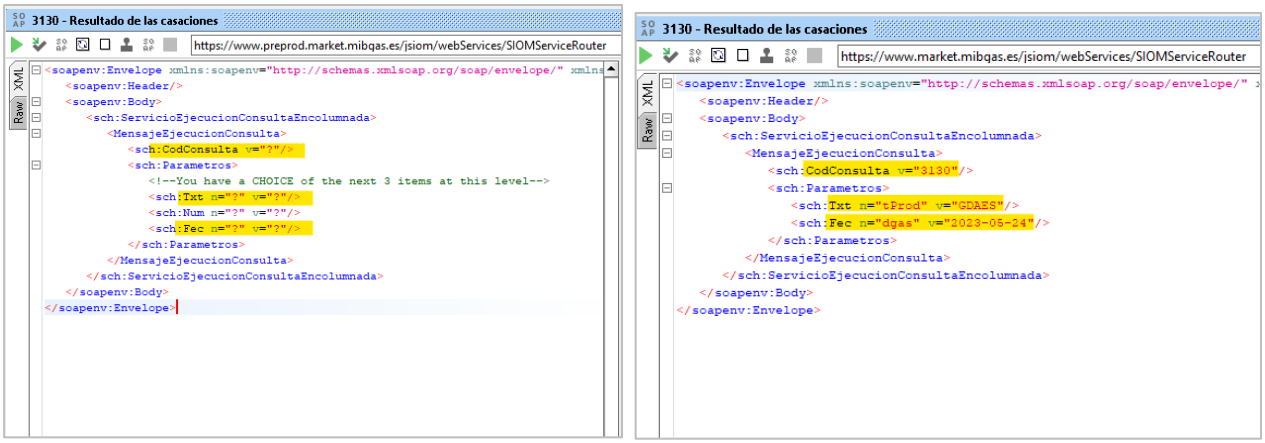
Let's use as an example the request "3130 - Data arising from matching". In the response obtained within the <op: Parameters> field there are two sections, one for the product type and one for the initial Gas Day. For each product type it tells us that this parameter is a "Txt" text field whose "Value" will take the code value "cod" of the product we enter. And for the initial gas day, it has to be a date field "Fec".

Request response:

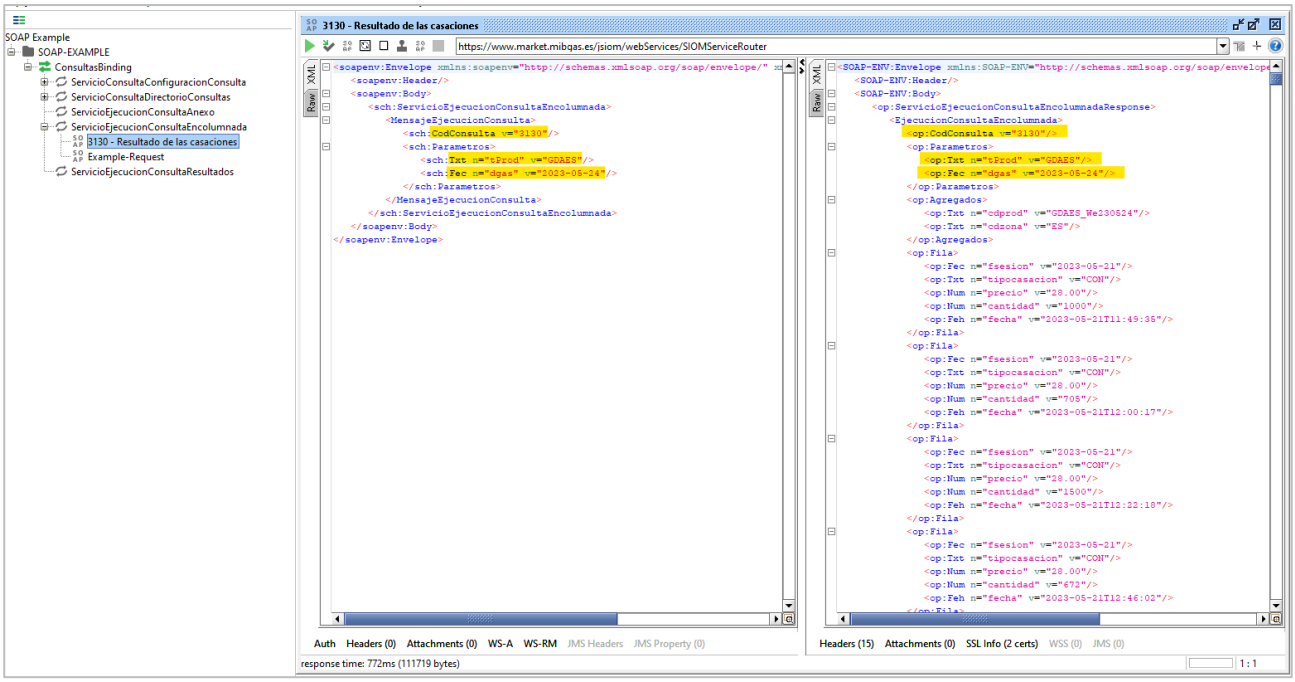


3.3 Columnar query

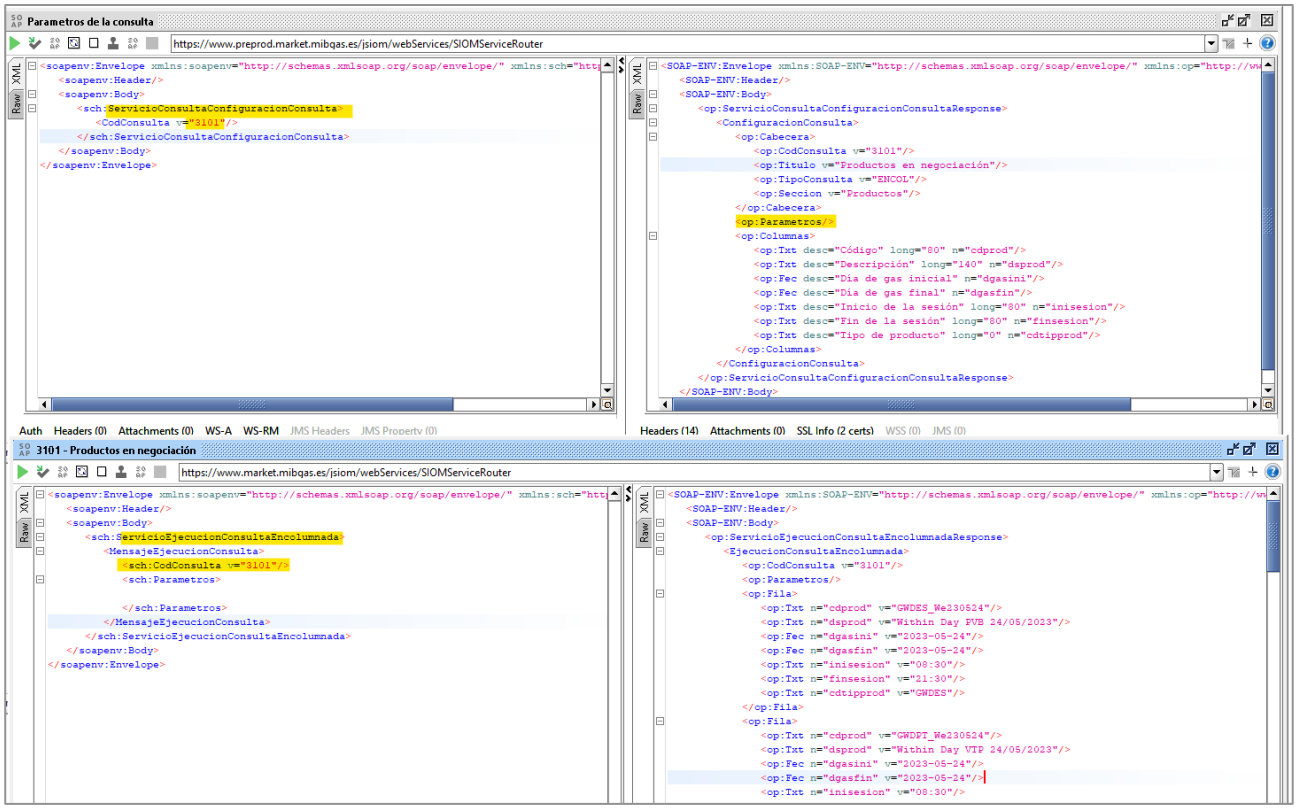
Now that we know the values of the request 3130, we will create it by right-clicking on the **ServicioEjecucionConsultaEncolumnada** service, *New Request. We will get a SOAP message that follows the scheme of the columnar Queries and we will have to fill in the codes and parameters that are blank filled with a '?' symbol. In this case we will fill in the v (value) of the CodConsulta with the code '3130'; in the TXT parameter we have to enter in the field n (name) 'tProd' to indicate that this is the parameter that we are going to configure and in the value v, the type of product 'GDAES' that we want to query. For the Fec field parameter, we fill in the n (name) field with 'dgas' to indicate the gas day and in the v field we fill in the date in the YYYYY-MM-DD date format '2023-04-25'. The parameter with numeric field Num can be removed because it is not necessary in this request.*



Request response:



A simpler example would be the request 3101 - Products under negotiation, this request does not require entering any parameters. So we can remove that part of the XML message.

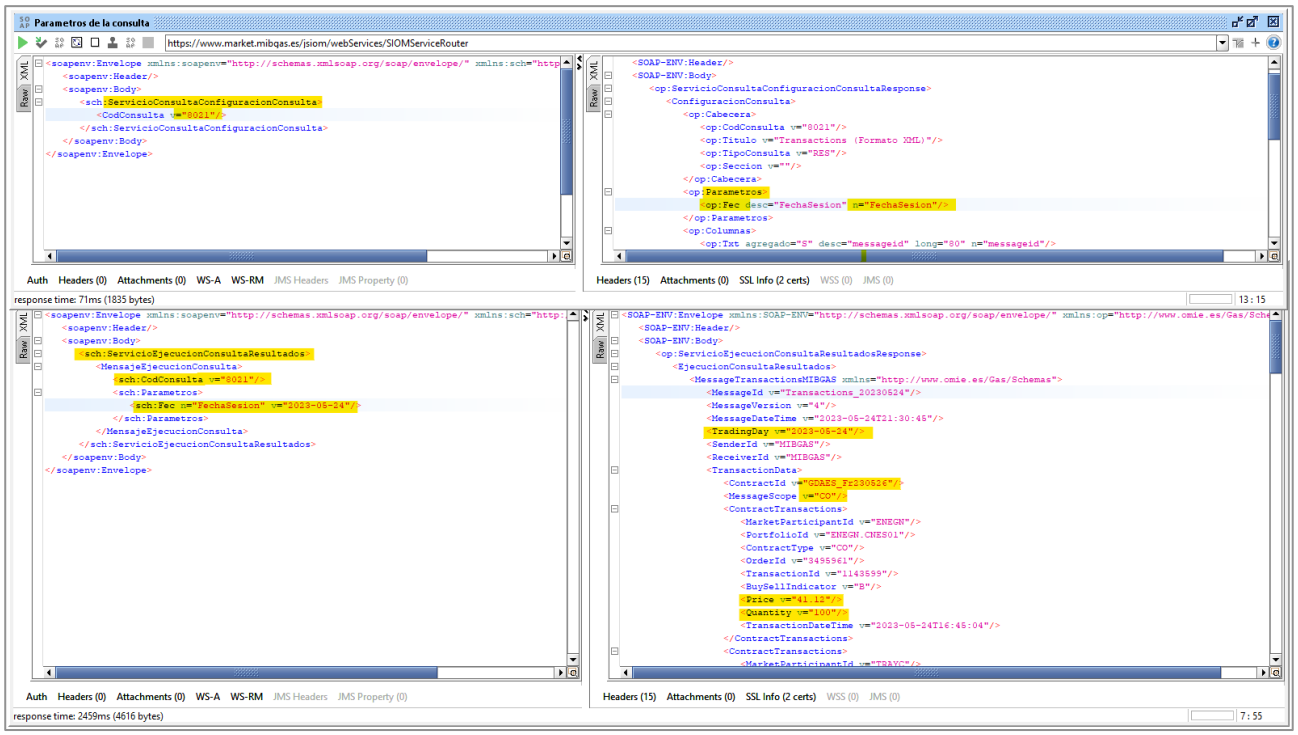


3.4 Queries in the Market Results format

There are a series of requests that do not have "columnar" format (ENCOL), the files with "market results" format (RES), are those in which the result is in XML format, but does not correspond to the format of rows and columns found in the market platform, but is based on one of the formats used in MIBGAS for market results (such as prices, trades and annotations). These requests are the ones available from the Automatic and manual download of files in XML format.

Let's use as an example the request "8021 - Trades (XML Format)". As in the rest of the cases, first of all, the parameters of this request must be queried with the **ServicioConsultaConfiguracionConsulta** request, by entering the code 8021 in the CodConsulta. The response tells us that the only parameter to enter is a Date parameter (Fec) with the name FechaSesion.

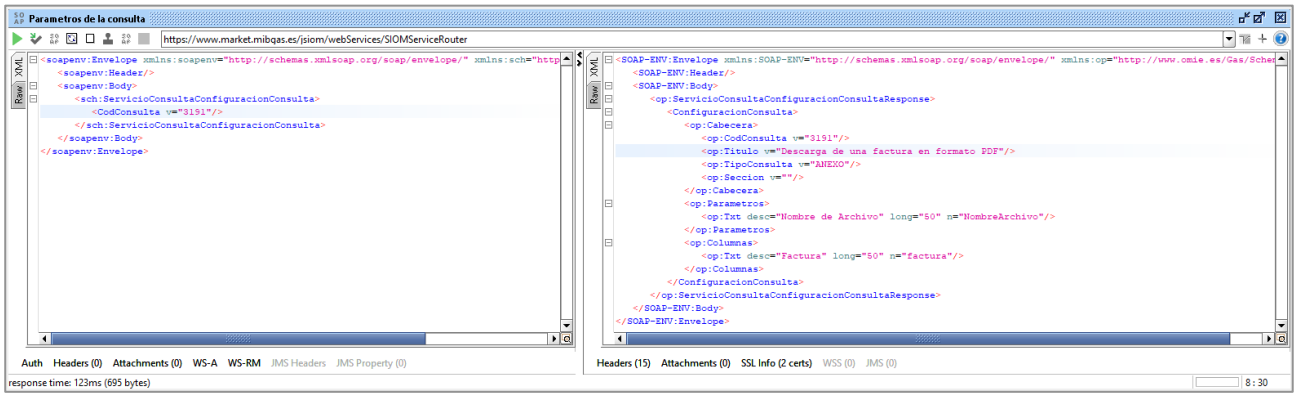
Click with the right button of the mouse on the **ServicoEjecucionConsultaResultados** service in the sidebar and select NewRequest. Enter the CodConsulta and fill in FechaSesion in the n (name) of the Fec parameter, and the v (value) of the date in format YYYYY-MM-DD. You will get a response with all the transactions of the agent that executes the request in each product with price, quantity, type of matching, id of the trade and the order, etc.



3.5 Attached file queries

For certain requests, it is possible to receive a file attached to the XML response in other formats, such as PDF. These requests correspond to some of those available from the Automatic and manual file download.

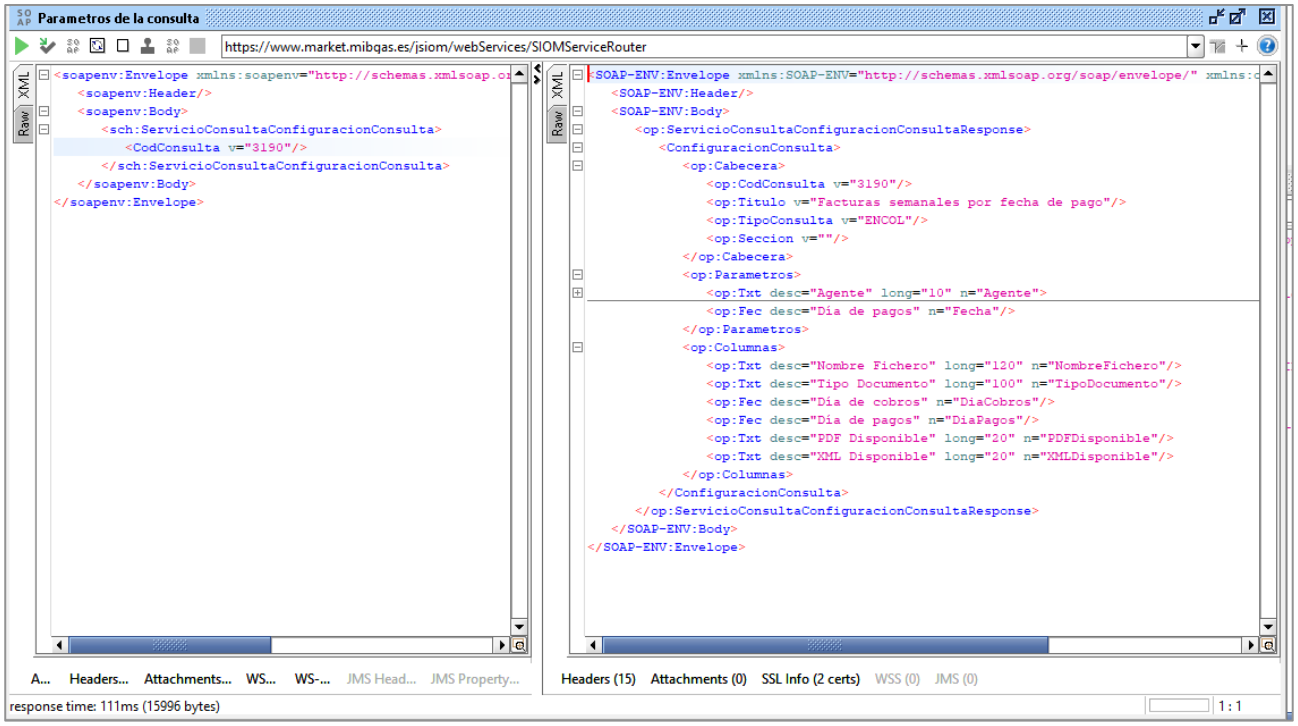
Let's use as an example the request "3191 - Download an invoice in PDF format". First of all, consult the configuration of the request with the **ServicioConsultaConfiguracionConsultas** service**.** The response tells us that the TipoConsulta is "ANEXO" and as for the parameters, we can see that it has only one, "NombreArchivo".



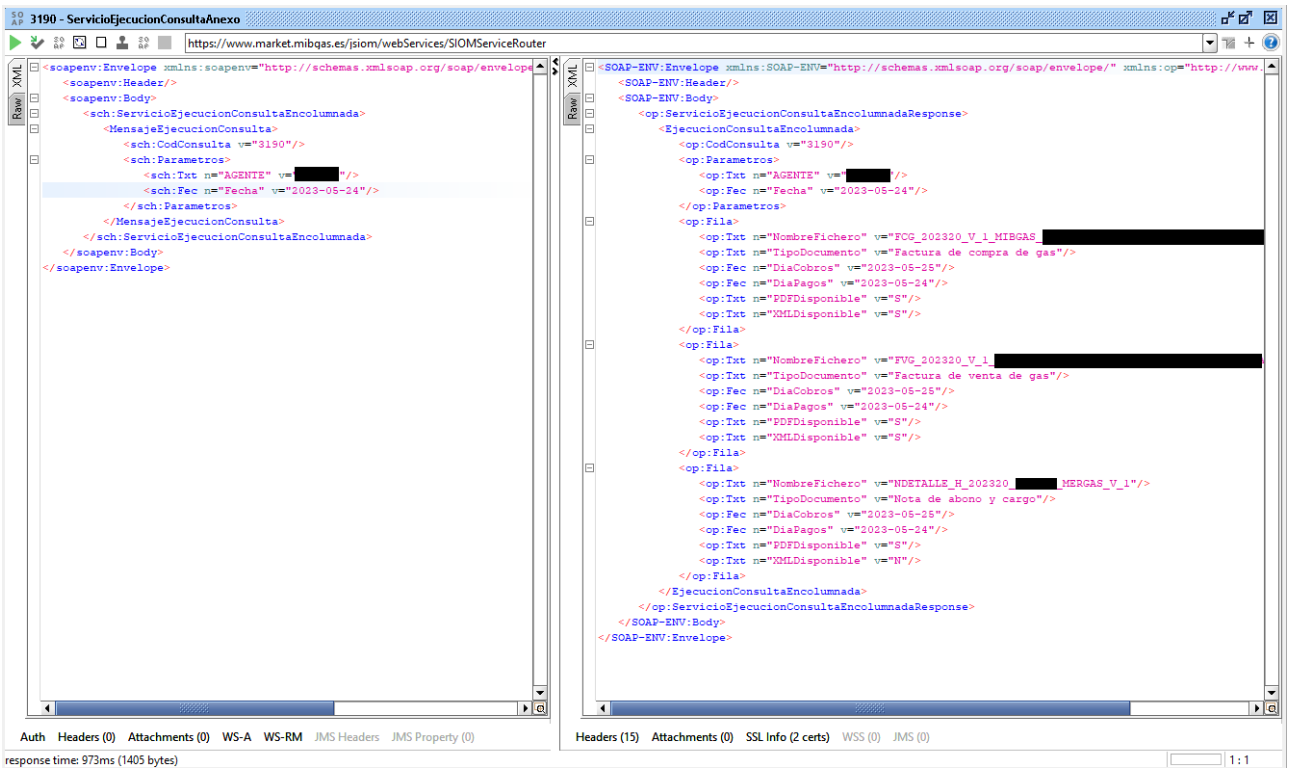
Secondly, we have to identify the file name, to do this we have to make an extra request depending on the billing scope. In this table of the document *Description of the Formats of Information Exchange between Mibgas and Agents* you can find the codes to identify the file names, according to the billing scope.

Scope	Description	Query codes for list of documents
FACTSEM	Weekly electronic invoices	3190
FACTSEMMDER	Weekly electronic invoices MDERSPOT	3198
FACTREMIT	Electronic Invoices for REMIT Services	3196
FSERVMDER	Electronic Invoices for Services REMIT MDER	3197
NOTAINT	Bank commission debit notes	3199

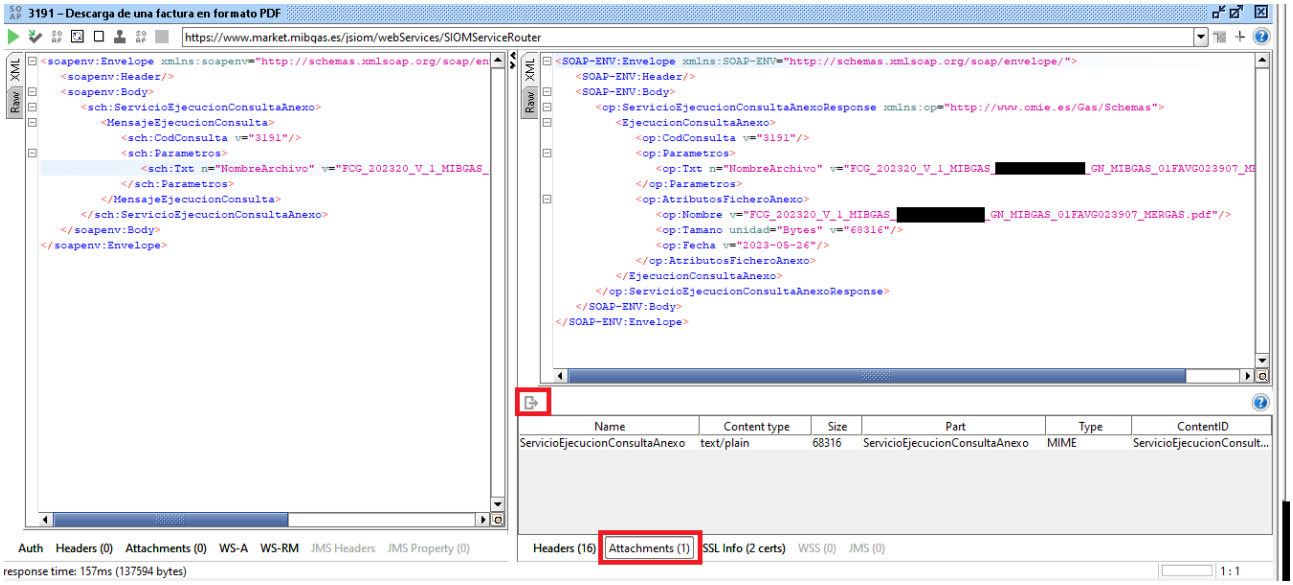
In this case we are going to use as an example the weekly electronic invoices (FACSEM) with the code 3190. We make a request with the **ServicioConsultaConfiguracionConsulta** service to identify the parameters and the type of request for the request "3190 - Weekly Invoices by Payment Date". The request type is columnar (ENCOL). The first parameter is TXT type and is the Agent's code, which must match the certificate holder agent used; the second parameter is the day of payments, type Fec and with the n (name) Date.



Fourthly, as the type of the request 3190 is columnar (ENCOL), use the **ServicioEjecucionConsultaEncolumnada** service filling in the parameters of the agent (which has been anonymized for confidentiality of the information in the example bellow) and Date. We can observe in the answer that there are three files and the first field of each Row is the name of the file "FGC_202320_V1_MIBGAS_..." is the code that we will have to introduce in the v (value) of the "NombreArchivo" (file name) field that we identified at the beginning of this section.



To conclude, we are going to send the request "3191 - Download an invoice in PDF format" using the service **ServicioEjecucionConsultaAnexo,** filling in the parameter of the NombreArchivo (file name) we want to download. We will receive an XML response with the file name and the file size in Bytes. In the Attachments button we can check that there is an attached file “Attachments (1)”.



Clicking on the button we can select the path to save the downloaded file in MIME format encoded in base64, so we will have to perform an additional conversion to convert it to binary, and thus obtain the document in PDF format. The conversion of formats is outside the scope of this guide.

ANEX A: BASIC CONCEPTS OF WEB SERVICES

Web Services messages are used in web applications that use communication standards and protocols compatible with any web browser. This makes them very versatile for exchanging information between different platforms.

They are self-contained, self-describing files encoded in XML.

A SOAP (*Simple Object Access Protocol)* message uses the HTTP/HTTPS protocol to communicate with an endpoint by sending a request and the endpoint returns a response.

SOAP elements

A SOAP is an ordinary XML message containing the following elements:

- The *envelope* element that identifies the XML document as a SOAP message
- The *header* element containing the application-specific information about the SOAP message (such as authentication, payment, etc.)
- The *body* element containing the body of the request and response to the WS service. It contains the information to be retrieved/sent.
- The *fault* element containing the formatting, processing, application incompatibility, etc. errors returned in the SOAP response.

SOAP provides the structure of the message, not how it has to be communicated, this is done with ***SOAP bindings***.

- The bindings allow the message to be correctly exchanged using the protocol.