

AMODIT WebAPI how-to

How to use AMODIT SOAP web service

Available materials

All information about AMODIT WebAPI (SOAP web service) you can find at:

<https://wiki.amodit.com/pl/?top-category=dla-integrator%c3%b3w>

and subsites:

<https://wiki.amodit.com/pl/baza-wiedzy/wspolpraca-aplikacji-zewnetrznych-z-amodit-poprzez-api/>

<https://wiki.amodit.com/pl/baza-wiedzy/amodit-webservice/>

<https://wiki.amodit.com/pl/baza-wiedzy/amodit-api/>

Connecting to AMODIT WebAPI

The following code samples are written in C# language.

The code below is common for all method calls of AMODIT web service.

```
// AMODIT web service URL. Add a reference to it in your project.  
http://{server_address}/webserviceanon/amodwebapi.asmx?WSDL  
  
// Connection to the AMODIT web service  
AMODWebAPI.AMODWebAPISoapClient client = new AMODWebAPI.AMODWebAPISoapClient();  
AMODWebAPI.AMODWebAPIHeader head = new AMODWebAPI.AMODWebAPIHeader();  
  
// Authentication - you should create a system user account in AMODIT.  
// To do this go to the "Permissions" section for this user and select: "System user. Cannot log in."  
head.Username = {user_login (string)};  
head.Password = {user_password (string)};  
string token = client.AuthenticateUser(head);  
head.AuthenticatedToken = token;
```

Updating a case field

The screenshot shows the AMODIT software interface for updating a case field. At the top, there is a header bar with the text "Invoice: AFX/2021/0036/Astrafox (34)" and "Accounting | Accounting | Invoicing Process". Below the header, the "Caselid" value is displayed as "34". To the right of the Caselid, a red box highlights the "City" field, which contains the value "Warszawa". A red arrow points from this highlighted field to a code snippet in the upper right corner.

To set a value in this field via AMODIT WebAPI you can call the following method in your application code:

```
client.UpdateCaseField(head, 34, "City", "Warszawa");
```

The interface includes tabs for Seller, Buyer, Invoice data, Cost allocation, and Invoice acceptance. The Seller tab is active. Below the tabs, there are several input fields: Name (Astrafox), Address (Poloneza 93), City (highlighted with a red box and containing Warszawa), VAT number (5252171560), Postal Code (02-826), and Seller country (PL). A dropdown arrow is shown next to the Seller country field.

Updating a table column

Invoice: AFX/2021/0036/Astrafox (34)
Accounting | Accounting | Invoicing Process

Repository 1

Caseld of the row

Invoice items

	Row ID	Name	Quantity	Net price	Net value	VAT	Gross value	VAT rate	
1	36	Licencje AMODIT	10	75.00	750.00	172.50	922.50	23	X

To set a value in this field/column via AMODIT WebAPI you can call the following method in your application code:

```
iCaseRowId=40;  
client.UpdateCaseField(head, iCaseRowId, "Net value", 750);
```

VAT summary

	Row ID	Net value	VAT rate	VAT value	Gross value	
1	40	750.00	23	172.50	922.50	X

field/column name

field/column value

The diagram illustrates the mapping between the 'Net value' field in the VAT summary table and the 'Net value' column in the Invoice items table. Red arrows point from the 'Net value' field in both tables to their respective columns in the explanatory text above.

Creating a new case

```
// This method creates a new case.  
client.CreateCase(head, {procedure_title (string)}, {starting_state_title (string)}, {case_title (string)});
```

```
// An example:  
client.CreateCase(head, "Invoicing process", "New document", "An invoice from AstraFox");
```

```
// The above method returns an XML file containing Caseld of the created case.
```



`webapi_createcase_result.xml`

Adding a table row and setting a value in a column

```
// The AddTableRow method adds a row to a table {table_field_name (string)} on the form of the case identified by {caseId (int)}.  
int iCaseRowId = client.AddTableRow(head, {CaseId (int)}, {table_field_name (string)});
```

```
// The UpdateCaseField method updates the field on a case's form named {field/column_name (string)} with the value {field/column_value // (string)}.
```

```
// The example below shows how to set a column value in a table on a form.
```

```
client.UpdateCaseField(head, iCaseRowId, {field/column_name (string)}, {field/column_value (string)});
```

Getting case XML

```
// The GetCase method returns the XML code which defines the field and/or column values for the form associated with {caseId}.

string caseXml = client.GetCase(head, {caseId(int)});
```

```
// The XML structure of the case:
```



webapi_sample_xml.xml

A sample case form XML

```
Code Writer
```

```
webapi_sample_xml.xml
1<AMODCases>
2  <AMODCase>
3    <Property Name="CaseTitle" Type="String" >Faktura od FEDEX na kwotę 1000 (LB-OF/0002)</Property>
4    <Property Name="CaseOwnerId" Type="Int32" >69</Property>
5    <Property Name="CaseCreated" Type="DateTime" >10/05/2021 14:58:18</Property>
6    <Property Name="CaseCreatedById" Type="Int32" >69</Property>
7    <Property Name="CaseModified" Type="DateTime" >17/06/2021 15:56:50</Property>
8    <Property Name="CaseModifiedById" Type="Int32" >69</Property>
9    <Property Name="CaseForwarded" Type="DateTime" >17/06/2021 16:06:52</Property>
10   <Property Name="CaseForwardedById" Type="Int32" >69</Property>
11   <Property Name="CaseEmployeeId" Type="Int32" >0</Property>
12   <Property Name="CaseProcedureId" Type="Int32" >1323</Property>
13   <Property Name="CaseStatusId" Type="Int32" >1959</Property>
14   <Property Name="CaseRead" Type="Boolean" >True</Property>
15   <Property Name="CaseIsClosed" Type="Boolean" >False</Property>
16   <Property Name="CaseEmployeeNumber" Type="String" ></Property>
17   <Property Name="CaseId" Type="Int32" >73445</Property>
18   <Property Name="CaseForwardedCC" Type="List`1" ></Property>
19   <Property Name="CaseContributors" Type="List`1" ></Property>
20   <Property Name="CaseReadText" Type="String" >Read</Property>
21   <Property Name="Sprzedawca" DisplayName="Seller" Type="DefinedDictionary" DBValue="908">FEDEX</Property>
22   <Property Name="Nasz" DisplayName="AFX seller code" Type="Varchar" DBValue="FDX/123">FDX/123</Property>
23   <Property Name="Nabywca" DisplayName="Buyer" Type="Varchar" DBValue="0x AstraF">0x AstraF</Property>
24   <Property Name="Data sprzedaży" DisplayName="Sale date" Type="DateTime" DBValue="17/06/2021 00:00:00">17/06/2021</Property>
25   <Property Name="Data wykonania usługi" DisplayName="Delivery date" Type="DateTime" DBValue="17/06/2021 00:00:00">17/06/2021</Property>
26   <Property Name="Termin płatności" DisplayName="Payment due date" Type="DateTime" DBValue="24/06/2021 00:00:00">24/06/2021</Property>
27   <Property Name="Kwota netto PLN" DisplayName="Net value" Type="Money" DBValue="1000.0000">1,000.00</Property>
28   <Property Name="Nasz numer faktury" DisplayName="Invoice number" Type="Varchar" DBValue="LB-OF/0002">LB-OF/0002</Property>
29   <Property Name="Status" DisplayName="Status" Type="Varchar" DBValue=""></Property>
30   <Property Name="Osoba opisująca merytorycznie" DisplayName="Person describing the invoice" Type="User" DBValue="69">Łukasz Bott (lbott)</Property>
31   <Property Name="Opis" DisplayName="Invoice description" Type="Text" DBValue="How to use AMODIT WebAPI.">How to use AMODIT WebAPI.</Property>
32   <Property Name="Podział kosztów" DisplayName="Cost-sharing table" Type="Table">
33     <Row CaseId="73915" Index="1">
34       <Property Name="Przedmiot lub usługa" DisplayName="Cost item" Type="Varchar" DBValue="AMODIT WebAPI tests">AMODIT WebAPI tests</Property>
35       <Property Name="Kwota netto" DisplayName="Cost item net value" Type="Money" DBValue="500.0000">500.00</Property>
36       <Property Name="Typ projektu" DisplayName="Project type" Type="DefinedDictionary" DBValue="160">Improvement 2</Property>
37       <Property Name="Przedsięwzięcie" DisplayName="Project name" Type="DefinedDictionary" DBValue="164">2CC</Property>
38     </Row>
39     <Row CaseId="73916" Index="2">
40       <Property Name="Przedmiot lub usługa" DisplayName="Cost item" Type="Varchar" DBValue="Documentation update">Documentation update</Property>
41       <Property Name="Kwota netto" DisplayName="Cost item net value" Type="Money" DBValue="300.0000">300.00</Property>
42       <Property Name="Typ projektu" DisplayName="Project type" Type="DefinedDictionary" DBValue="162">n/d</Property>
43       <Property Name="Przedsięwzięcie" DisplayName="Project name" Type="DefinedDictionary" DBValue="167">AC.AL.0213</Property>
44     </Row>
45     <Row CaseId="73917" Index="3">
46       <Property Name="Przedmiot lub usługa" DisplayName="Cost item" Type="Varchar" DBValue="Implementation services">Implementation services</Property>
47       <Property Name="Kwota netto" DisplayName="Cost item net value" Type="Money" DBValue="200.0000">200.00</Property>
48       <Property Name="Typ projektu" DisplayName="Project type" Type="DefinedDictionary" DBValue="163">Project</Property>
49       <Property Name="Przedsięwzięcie" DisplayName="Project name" Type="DefinedDictionary" DBValue="168">AC.BC.0226</Property>
50     </Row>
51   </Property>
52 </AMODCase>
53 </AMODCases>
```

Ready Ln 20, Col 5 Auto CRLF XML

Searching for cases

```
//The GetCasesByFieldValue method returns a comma separated list of cases {caseId (int)} where a field {field_name} matches a value  
//{field_value}.
```

```
string caseIds = client.GetCasesByFieldValue(head, {field_name}, {field_value});  
string[] caseIdSplit = caseIds.Split(',');
```