

Como Utilizar a API Webservices Multidados 1.0

A API 1.0 foi montada utilizando o padrão SOAP (<https://pt.wikipedia.org/wiki/SOAP>) e pode ser testada utilizando um software como o SOAPUI (<https://www.soapui.org/>)

Existe uma documentação simples no caminho [URL_SISTEMA]/webservices os métodos disponíveis e quais são os campos e seus tipos para envio.



The screenshot displays the 'WebServicesMultidados' interface. On the left, a list of service methods is provided, including 'GetPrestador', 'DadosEmpresa', 'DadosCliente', 'DadosProjeto' (highlighted with a red box and a mouse cursor), 'global_get_cliente', and several 'summary_of...' methods. On the right, the details for the 'DadosProjeto' method are shown. This includes its name, binding, endpoint, SOAP action, and style. The 'input' section lists various parameters such as 'USUARIO_WS', 'SENHA_WS', 'CNPJ_EMPRESA', 'IDENT_CLIENTE', 'NOME_PROJETO', 'COD_PROJETO', 'PROJETO_APELIDO', 'CNPJ_EMPRESA_PROJETO', 'CLASSE', 'TABELA_PRECO', 'DPLNIVELSGACOES', 'DATA_PROJETO', 'COD_PROJETO_PAI', and 'CAMPOS_VARIAVEIS_PROJETO'. The 'output' section indicates the response message 'DadosProjetoResponse' and provides a detailed description of the data returned, including fields like 'SUCESSO', 'ERRO', 'USUARIO_WS', 'SENHA_WS', 'CNPJ_EMPRESA', 'DATA_PROJETO', 'TABELA_PRECO', and 'DPLNIVELSGACOES'.

Basta clicar no nome do método para ter acesso às suas informações.

Programas como o SoapUI conseguem gerar o XML de envio dinamicamente ao importar o WSDL (Documento XML que informa à aplicação quais são os métodos e campos) da seguinte forma:

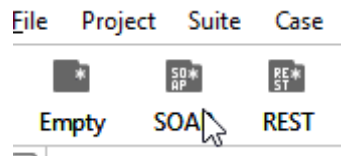
Acesse o WSDL no caminho [URL_SISTEMA]/Webservices/index.php?wsdl ou clique aqui:



This screenshot shows the 'WebServicesMultidados' header at the top. Below it, a blue box contains the text 'View the WSDL for the service. Click on an operation name to view it's deta'. A mouse cursor is pointing at the word 'WSDL'. Below this text, a list of service methods is shown: 'GetPrestador', 'DadosEmpresa', and 'DadosCliente'. To the right of this list, a partial view of the 'DadosProjeto' method details is visible, showing fields like 'Name', 'Binding', 'Endpoint', 'Soap Action', 'Style', 'Input', and 'Output'.

Copie a URL do WSDL para colocar no SoapUI.

Clique para criar um novo projeto SOAP:

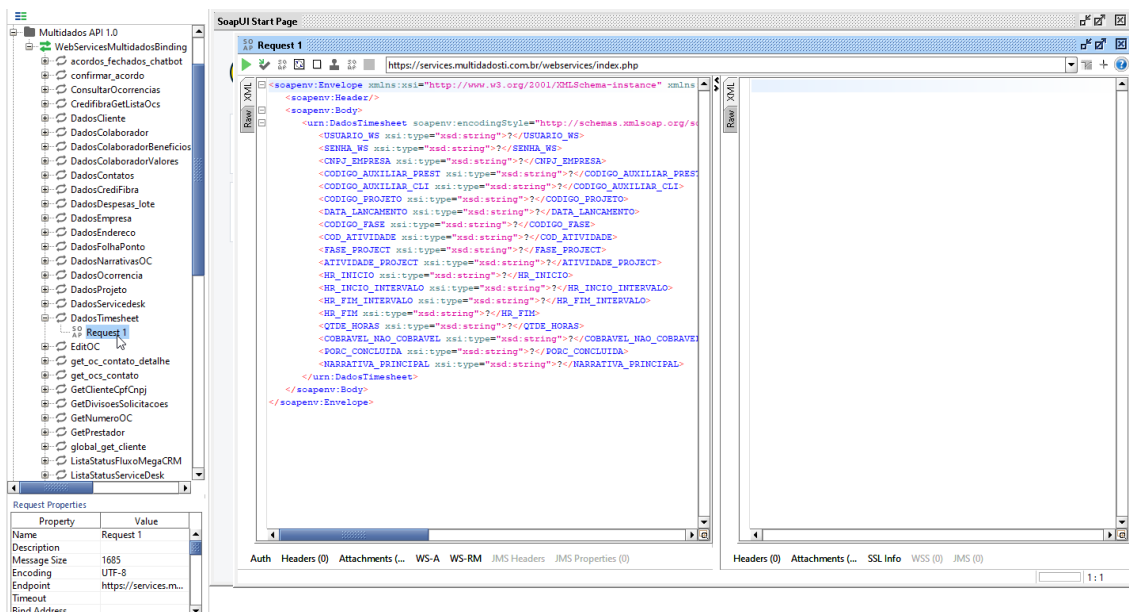


Um pop up surgirá com 2 campos:

“Project Name” é o nome que será dado ao projeto. Fique a vontade para colocar o nome que preferir.

“Initial WSDL” é onde você deve colar o WSDL da sua aplicação:

Deixando a opção de “Create sample requests for all operations?” ligada (padrão), o programa gera o XML de todos os métodos automaticamente, deixando um ponto de interrogação “?” para ser substituído pelo valor a ser enviado.



É comum a todas as integrações preencher os campos “USUARIO_WS” e “SENHA_WS” com um usuário e senha que possuam acesso ao CRM.

Se houver algum campo que não quer passar valor, apenas remova a “?” e deixe o nó do XML vazio.

Nesta API, todos os campos são do tipo STRING para facilitar envio. Campos com valores Booleanos devem ser preenchidos com ‘T’ para TRUE e ‘F’ para FALSE. Isto vale para campos que no CRM são ligados e desligados com uma chave ou SIM ou NÃO.

Campos de data aceitam o formato americano ‘AAAA-MM-DD’. Campos de Timestamp utilizam o mesmo formato acrescido das horas e minutos: ‘AAAA-MM-DD hh:mm’.

Campos COMPLEX TYPE aceitam LISTAS de valores como conjuntos (arrays) multidimensionais ou dicionários (Python). Como SOAP é um protocolo padrão, sua linguagem de programação deve ter uma biblioteca pronta para tratar este tipo de integração como a soaplib do Python (<https://pypi.org/project/soaplib/>), o NuSoap do PHP (<https://github.com/f00b4r/nusoap>) entre outras. Estas bibliotecas conseguem lidar com Arrays Multinível sem problemas. Mas se for necessário criar o XML manualmente, siga o seguinte procedimento:

Abra o XML do WSDL no seu navegador Google Chrome:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<definitions xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tns="urn:server.Multidados" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns="http://schemas.xmlsoap.org/wsdl/" targetNamespace="urn:server.Multidados">
  <types>
    <xsd:schema targetNamespace="urn:server.Multidados">
      <xsd:import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <xsd:import namespace="http://schemas.xmlsoap.org/wsdl/" />
      <xsd:complexType name="CA_ClientesStruct">
        <xsd:all>
          <xsd:element name="codigo" type="xsd:string"/>
          <xsd:element name="valor" type="xsd:string"/>
        </xsd:all>
      </xsd:complexType>
      <xsd:complexType name="CA_Clientes">
        <xsd:complexContent>
          <xsd:restriction base="SOAP-ENC:Array">
            <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:CA_ClientesStruct[]" />
          </xsd:restriction>
        </xsd:complexContent>
      </xsd:complexType>
      <xsd:complexType name="request_data">
        <xsd:all>
          <xsd:element name="endpoint" type="xsd:string"/>
          <xsd:element name="method" type="xsd:string"/>
          <xsd:element name="params" type="xsd:string"/>
          <xsd:element name="ref_id" type="xsd:string"/>
          <xsd:element name="ref_obj" type="xsd:string"/>
        </xsd:all>
      </xsd:complexType>
      <xsd:complexType name="result_call_other_ws">
        <xsd:all>
          <xsd:element name="success" type="xsd:string"/>
          <xsd:element name="error" type="xsd:string"/>
          <xsd:element name="response" type="xsd:string"/>
          <xsd:element name="response_full" type="xsd:string"/>
          <xsd:element name="request_data" type="tns:request_data"/>
        </xsd:all>
      </xsd:complexType>
      <xsd:complexType name="result_oc_adicionar_obs">
        <xsd:all>
          <xsd:element name="success" type="xsd:string"/>
          <xsd:element name="error" type="xsd:string"/>
        </xsd:all>
      </xsd:complexType>
      <xsd:complexType name="BD_ClientesStruct">
        <xsd:all>
          <xsd:element name="codigo" type="xsd:string"/>
          <xsd:element name="valor" type="xsd:string"/>
        </xsd:all>
      </xsd:complexType>
    </xsd:schema>
  </types>
</definitions>
```

Localize o nome do campo COMPLEX TYPE que você precisa inserir. Como exemplo, usaremos CA_CLINTES (Campos adicionais de Clientes).

```
<xsd:complexType name="CA_ClientesStruct">
  <xsd:all>
    <xsd:element name="codigo" type="xsd:string"/>
    <xsd:element name="valor" type="xsd:string"/>
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="CA_Clientes">
  <xsd:complexContent>
    <xsd:restriction base="SOAP-ENC:Array">
      <xsd:attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="tns:CA_ClientesStruct[]" />
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="request_data">
```

Observe que ele possui uma CA_CLIENTESSTRUCT que é composta por um nó (CA_CLIENTES) seguido de outros 2 nós “codigo” e “valor”. No XML ficaria assim:

```
<CAMPOS_VARIAVEIS_CLIENTES xsi:type="urn:CA_Clientes"
SOAP-ENC:arrayType="urn:CA_ClientesStruct[]">
  <campo_adicional>
    <codigo>pre_contatos</codigo>
    <valor>tel_ad</valor>
  </campo_adicional>
  <campo_adicional>
    <codigo>habilita_personalizacao</codigo>
    <valor>sim</valor>
  </campo_adicional>
</CAMPOS_VARIAVEIS_CLIENTES>
```

Veja que criamos em “envelope” campo_adicional para envolver os nós codigo e valor.

Para facilitar a sua visualização, em um objeto JSON, esta estrutura seria:

```
[
```

```
{
  "codigo":"pre_contatos",
  "valor":"tel_ad"
},
{
  "codigo":"habilita_personalizacao",
  "valor":"sim"
}
]
```

A Biblioteca SOAP que for utilizada provavelmente aceitará o conjunto multidimensional que estiver disponível na linguagem, por exemplo no PHP:

```
$complexType = array(
  array(
    "codigo"=>"pre_contatos",
    "valor"=>"tel_ad"
  ), array(
    "codigo"=>" habilita_personalizacao",
    "valor"=>"sim"
  )
);
```

Caso tenha dificuldades com algum tipo de campo, entre em contato com o suporte da Multidados.