

CDS Batch Service (CDSBS)

NYS OASAS Web Service

User/Developer Guide

2024 New York State ITS. All Rights Reserved.
Any unauthorized reproduction or distribution is prohibited.

Version 1.0

Date: June 13, 2024

NYS ITS BAS – OASAS

Author: Srikanth

TABLE OF CONTENTS

(CDSBatch) Web Service	2
<i>What is CDS Batch Service ?</i>	2
<i>Available methods</i>	2
<i>Authentication And Method access</i>	3
<i>Technical Requirements Integration Requirements</i>	5
<i>Availability of the Web Service</i>	5
<i>Invoking the Web Service</i>	5
<i>Web Service Error codes</i>	6
<i>Soap Exchange between Web Service Consumer and Provider</i>	6

INTRODUCTION TO CDS BATCH SERVICE

What is CDS Batch Service ?

CDS batch service is a web service that enables provider users to upload a batch file to the system. Which is the alternate to CDS Batch application in UI. Service will have the following methods.

- Upload a file (accepted files format .txt)
- Get File Status (results of the processed file)
- Delete Batch File (Deletes the uploaded file)

Available methods

The following methods are available in the PDA web service:

uploadFile() This method is invoked to save the file in data base. This method takes input as binary file. Accepted file format is .txt

Input – file – binary, providerNo, fileFormat(example v1,v2),fileName

Response - The OUTPUT structure includes the data in an xml string.

getFileStatus() This method is invoked to get the file status results.

Input – filename, providerNo

Response – The OUTPUT structure includes the data in an xml string. Its XML result set data

deleteBatchFile() Method invoked to delete the file

Input – filename, providerNo

Response - The OUTPUT structure includes the data in an xml string.

Authentication And Method access

Authentication is performed by the service access level. The web service is protected under WS-Security. The service must provide the user credentials to access any methods above, it has two layers of security. 1) credentials will be validated against LDAP/OID as they access OASAS application. 2) after #1 additional security context will be created for that provider to minimize the risk of access other provider data. Once validated the system will process the data.

Please see below the example created for each method using the SOAPUI.

1) UploadFile : - Example

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="/wsFileUpload/common/types/">
  <soapenv:Header xmlns:wsa="http://www.w3.org/2005/08/addressing">
    <wsse:Security soapenv:mustUnderstand="1" xmlns:wss="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken-342173C75DBF8FCEFD16957397667041">
        <wsse:Username>test_batch</wsse:Username> --replace with your provider's username
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-
token-profile-1.0#PasswordText">XXXXX</wsse:Password> - Password used to access OASAS application

      <wsse:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-
security-1.0#Base64Binary">rgJuEEPlE17Kci4JpW7rfg==</wsse:Nonce>

      <wsu:Created>2024-01-26T14:49:26.629Z</wsu:Created>

    </wsse:UsernameToken>
  </wsse:Security>
</soapenv:Header>
<soapenv:Body>
<typ:uploadFile>
  <typ:imageBytes>cid:1292911776742</typ:imageBytes> -- It should be binary file
  <typ:providerNo>#####</typ:providerNo> -- replace with provider's ID #
  <typ:fileFormat>V1</typ:fileFormat> -- replace with version of CDS data being uploaded
  <typ:fileName>Sample1.txt</typ:fileName> -- replace with filename of provider file being processed
</typ:uploadFile>
</soapenv:Body>
</soapenv:Envelope>
```

2) How to execute service method example – **getFileStatus**

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="/wsFileUpload/common/types/">
  <soapenv:Header xmlns:wsa="http://www.w3.org/2005/08/addressing">
    <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken-342173C75DBF8FCEFD16957397667041">
        <wsse:Username>test_batch</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-
token-profile-1.0#PasswordText">XXXX</wsse:Password>
        <wsse:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-
message-security-1.0#Base64Binary">rgJuEEPlE17Kci4JpW7rfg==</wsse:Nonce>
        <wsu:Created>2024-01-26T14:49:26.629Z</wsu:Created>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <typ:getFileStatus>
      <typ:fileName>Sample1.txt</typ:fileName>
      <typ:providerNo>#####</typ:providerNo>    - replace with provider's ID #
    </typ:getFileStatus>
  </soapenv:Body>
</soapenv:Envelope>
```

3) **deleteBatchFile ()** - Change the provider no and file name as required.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:typ="/wsFileUpload/common/types/">
  <soapenv:Header xmlns:wsa="http://www.w3.org/2005/08/addressing">
    <wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
      <wsse:UsernameToken wsu:Id="UsernameToken-342173C75DBF8FCEFD16957397667041">
        <wsse:Username>test_batch</wsse:Username>
        <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-
token-profile-1.0#PasswordText">XXXX</wsse:Password>
        <wsse:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-
message-security-1.0#Base64Binary">rgJuEEPlE17Kci4JpW7rfg==</wsse:Nonce>
        <wsu:Created>2024-01-26T14:49:26.629Z</wsu:Created>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body>
    <typ:deleteBatchFile >
```

```
<typ:fileName>Sample1.txt</typ:fileName>  
<typ:providerNo>####</typ:providerNo> - replace with provider's ID #  
</typ:deleteBatchFile>  
</soapenv:Body>  
</soapenv:Envelope>
```

Technical Requirements Integration Requirements

To integrate the CDS Batch web service into your application, you must develop custom code that adapts your application's business logic and data to the service. You can use any programming language that supports HTTP/HTTPS and SOAP communication across secure connections—for example, Java, Visual Basic, C#, etc.

You will need to:

- know how to encode and decode XML documents.
- design a strategy into your software for handling errors
- establish a connection to an approved network from the computer that runs your application.
- create a SOAP wrapper for the Web service request.

The CDS Batch web service is a standard Web service that uses SOAP and WSDL.

The Web Service Description Language (WSDL) for the CDS batch web service can be viewed at the following URL:

<https://custapps.oasas.ny.gov/CDSBatchService/CDSBatchUploadService?WSDL>

Availability of the Web Service

The CDS Batch Service system is available 7 days a week except for periods of scheduled system maintenance or service downtime. Support is limited to normal NYS OASAS business hours.

Invoking the Web Service

An application invokes the CDS Batch web service by initiating HTTPS communication with the server hosting the Service. All interactions with the host server are through the HTTPS protocol. The message is an XML request formatted as a standard SOAP package.

Web Service Error codes

Application when invoked may throw some exceptions and they have the error codes. Please see below the for the error code details.

WSOASAS-01 - Invalid Credentials – User might have entered incorrect credentials.

WSOASAS-02 - Invalid Credentials – Also credentials issue in second layer of security – User might not have access to this functionality.

WSOASAS-03 - Invalid Credentials – Also credentials issue in third layer of security – User have the access but not to the same provider access. Example – provider ##### trying to lookup the data for provider NNNNN

Soap Exchange between Web Service Consumer and Provider

It will be necessary to include a security header in the SOAP call to the web service. Invoking the service without this header will result in a server error.

