

# I. Web API

## Overview

DBSync Cloud Replication can run replication processes using Web API thus allowing usage of Web Service calls to initiate the process of replication. Users can use the URL thus generated to run or execute the commands or check status of an existing command by using the POST or GET call methods respectively. A process ID is provided for the user to check the status of the command that was run. It supports the four functions of Cloud replication: Clean copy, Update Schema, Source to Database and Database to Source.

## Problem Statement

With DBSync Cloud replication you get an exe. or a zip file based on your environment and you run the application. Once the server starts you basically go to the browser to run the various functionalities of Cloud replication by using the User Interface. This process requires manual intervention for running the various commands, check logs etc. This is a swagger interface for replication API. DBSync Cloud Replication Web API allows the user to run these functions programmatically. The process of Asynchronous calling is applied where you can run an existing process or pass all parameters in real time.

## Setting Up of Web API

To set Web API:

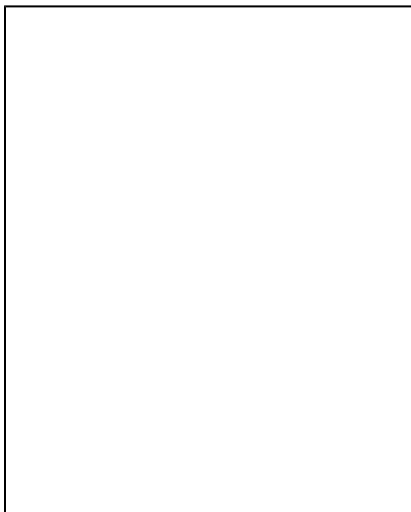
- Login to DBSync Cloud Replication instance



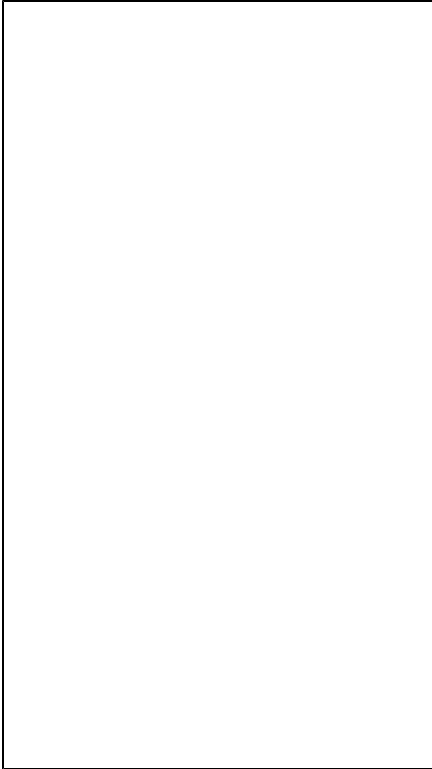
- From the left panel click on the

icon

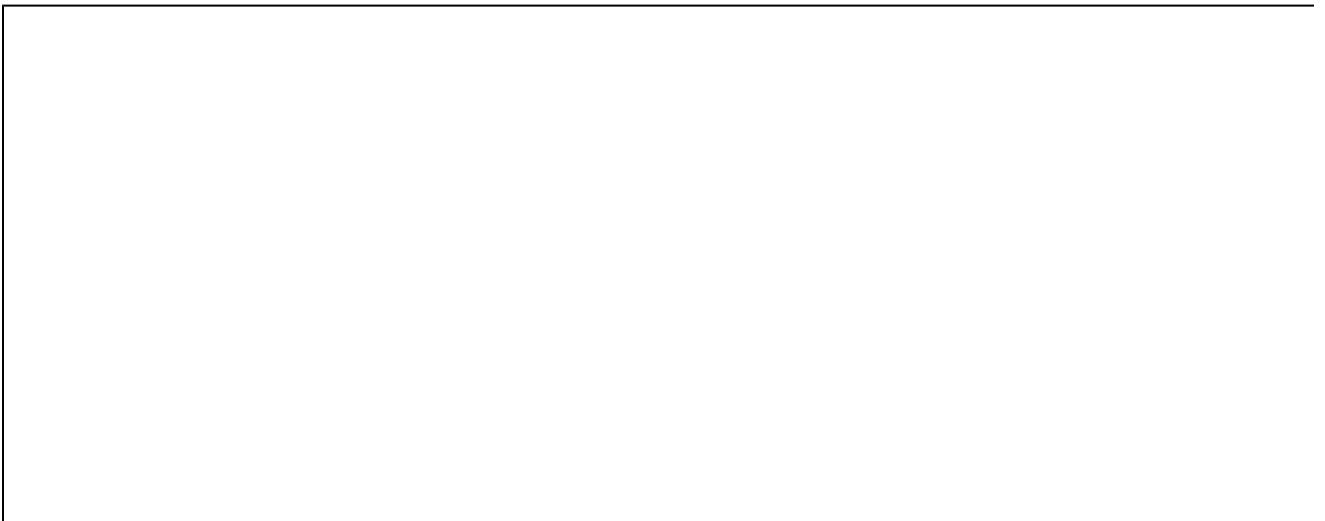
- From the left navigation toolbar click on Administration



- From the drop-down menu select API



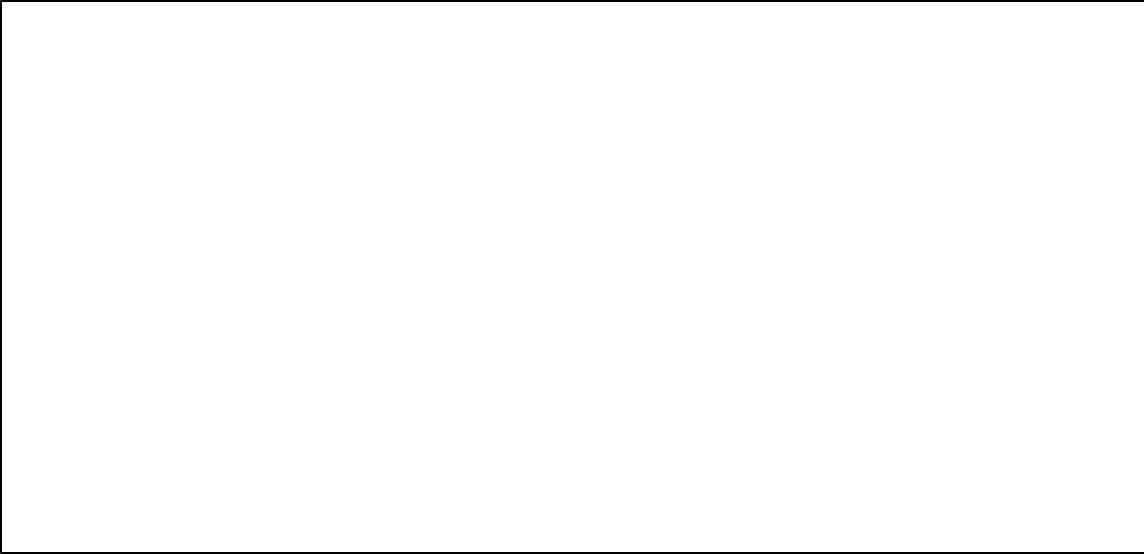
- You will be redirected to a DBSync swagger interface for our replication API



- Click on Authorize

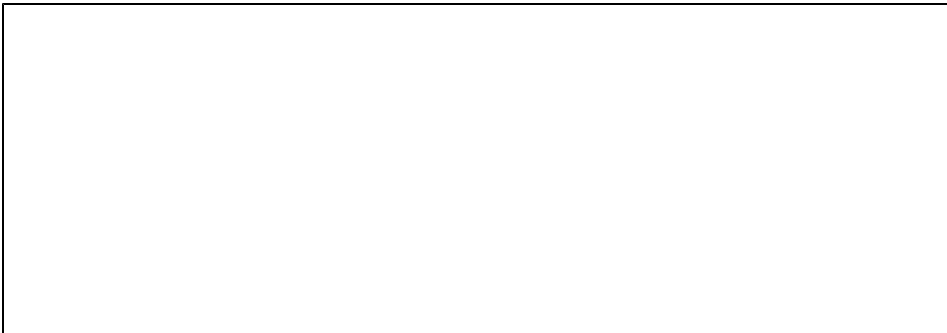


- Provide the User name Password for your profile. (will match the DBSync login credentials)



- Click Authorize
- A pop-up confirming authorization as AUTHORIZED will appear on the screen

- This is done for authentication purposes so only the specific User can Call the API and no one else can on the User's behalf. Only when it is authenticated the user can run the requests.
- The URL should be passed to run the replication using Web API. There are 2 options for sending :



**a. Replicate : This is to trigger the replication process**

- Click on this to configure the API call to trigger a replication process.
- This will include details of : Salesforce, Database, Objects, Logs, Callback API, Run command option
- There is an example of a payload provided for the user to configure the API calls:

- Replace the sample values with actual values and execute the command.
- Click on TRY IT OUT

- You will get an editable payload format where you can enter all the details and configure the API Call.
- You can execute the API call by Profile name or configure all the settings for a profile and run the API calls.

**For example:**

- Let us first configure the Salesforce Instance:

The same details which are in the UI will be entered as values in the JSON. In place of String replace with actual text and enter True or false where applicable.

For this example consider the below image from the UI for filling in the Salesforce details:

Sample Payload	Actual Payload
----------------	----------------

**Salesforce**

```
{
  "salesforce": {
    "soapEndpoint": "string",
    "bulkApiEnable": true,
    "password": "string",
    "username": "string"
  },
}
```

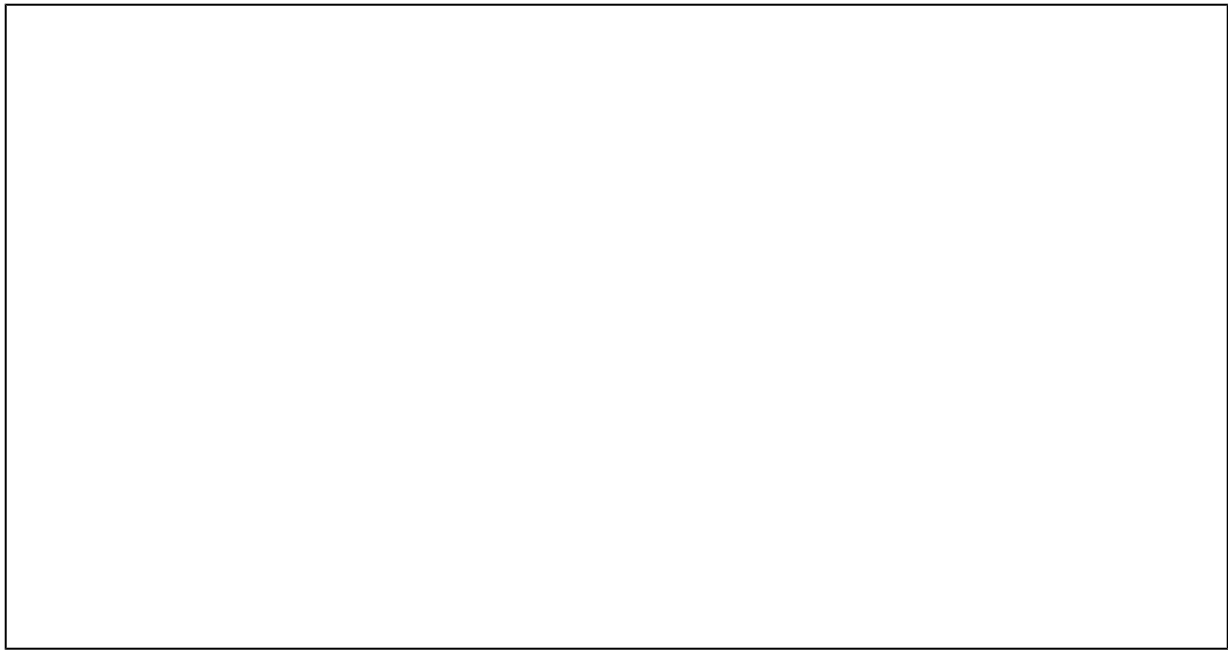
**Salesforce**

```
{
  "salesforce": {
    "soapEndpoint": "https://login.salesforce.com/services/Soap/u/47.0",
    "bulkApiEnable": false,
    "password": "xxxxxxx",
    "username": "dbsync.replicationtest@gmail.com"
  },
}
```

This will set up the Salesforce instance for the above mentioned Profile and for this specific User Name. The URL for the soap endpoint can be taken from the UI. In this instance we do not want Bulk API so we enter as false. Provide the Username and password for the Salesforce instance

**Database**

**Database**



Will split this into 3 parts:

```
"database": {  
  "fileReplication": true,  
  "fileVersion": true,  
  "filepath": "string",  
  
  "s3": true,  
  "awsAccessKey": "string",  
  "awsBucketName": "string",  
  "awsRegion": "string",  
  "consumerSecret": "string",  
  
  "database": "string",  
  "host": "string",  
  "password": "string",  
  "port": "string",  
  "schema": "string",  
  "type": "string",  
  "url": "string",  
  "username": "string"  
},
```

```
"database": {  
  "fileReplication": true,  
  "fileVersion": true,  
  "filepath": "string",  
  
  "s3": true,  
  "awsAccessKey": "string",  
  "awsBucketName": "string",  
  "awsRegion": "string",  
  "consumerSecret": "string",  
  
  "database": "xxxxxxxxxx",  
  "host": "localhost",  
  "password": "xxxxxxxxxx",  
  "port": "0000",  
  "schema": "public",  
  "type": "mysql",  
  "url": "jdbc:mysql://localhost:3306/performance_schema",  
  "username": "xxx"  
},
```

PART 1:

```
"database": {  
  "fileReplication": true,  
  "fileVersion": true,  
  "filepath": "string",
```

```
"database": {  
  "fileReplication": true,  
  "fileVersion": true,  
  "filepath": "C:/DBSync_CloudCDM/files",
```

**NOTE:** This is a file based data type. Salesforce has file based objects. There are attachments, documents and content base. If you want data as a physical file instead of a database use PART 1. You can get the raw data file in your local system.

**File replication:** true

If you are using attachment as an object mark this as TRUE.

If you want to download a local file

**File Path:** Enter the details of the file path location of where you want to store the data

**File Version:** This function enables the user to have the same file name stored multiple times. It will not override. If you want to have this versioning then select true if not false.

In the UI you can see these 3 fields under ADVANCED SETTINGS. Refer image below:

Download as local file: File replication

Download files to Directory: File Path

Keep multiple Version of Files: File Version



**PART 2:**

"s3": true,  
"awsAccessKey": "string",  
"awsBucketName": "string",  
"awsRegion": "string",  
"consumerSecret": "string",

"s3": true,  
"awsAccessKey": "string",  
"awsBucketName": "string",  
"awsRegion": "string",  
"consumerSecret": "string",

**NOTE:** For databases such as AWS REDSHIFT the transfer of data is slow. Hence S3 bucket is used to transfer data as it is much faster and easier. Any data from Salesforce would be dumped into the S3 bucket and from there the transfer is done. So ONLY IF THE DATABASE IS AWS REDSHIFT fill these columns:

Provide details of :  
AWS Access Key  
AWS bucket  
AWS region  
AWS secret

Also note if you are using AWS S3 then you do not enter details of PART 3.

**PART 3:**



<pre>"database": "string", "host": "string", "password": "string", "port": "string", "schema": "string", "type": "string", "url": "string", "username": "string" },</pre>	<pre>"database": "xxxxxxx", "host": "localhost", "password": "xxxxxx", "port": "0000", "schema": "public", "type": "mysql", "url": "jdbc:mysql://localhost:3306/performance_schema", "username": "xxxx" },</pre>
<p><b>NOTE:</b> Enter details of the database in this section Database: Name of database Type: Type of the database URL: Copy it from the UI Schema: the specific schema for this database can be entered. If nothing specific enter as public</p> <p>If you are not using AWS S3 enter PART 3</p>	
<b>Object</b>	<b>Object</b>
<pre>"objects": [ { "excludes": [ "string" ], "filter": "string", "name": "string" } ],</pre>	<pre>"objects": [ { "excludes": [ "xxx" ], "filter": "name='Test_2021'", "name": "xxxx" } ],</pre>

**NOTE:**

The objects section is an Array. Any number of data can be stored. Multiple entries can be passed in this section as required by the user.

**Parameters:**

**Name:** the Name of the object you want to store/Sync. For eg: Account

**Filter:** This can be found in the section Filter Criteria in the UI. Any filter can be applied as required by the user. For eg: If a user wants to display Account object records with name= 'Test\_2021'. All those records in Test 2021 will get synced. This is the filter applied. Enter name='Test\_2021' in the filter tab.

**Exclude:** In Salesforce there may be a number of fields. Enter all those fields which you do not want to Sync. You can choose the list from UI as show below:

**For eg:** If you do not wish to sync FAX then you can enter this in the exclude column.

```
"excludes": [
```

```
  "Fax"
```

If you want to add more than one field to exclude For eg: phone number:

```
"excludes": [
```

```
  "Fax", "Phone"
```

Also note that this is for one OBJECT.

For Account you have added one filter and excluded 2 fields.

You can repeat the same for any number of objects that you wish to sync:

```
{
  "name": "Attachment"
}
```

To add another Object named Attachment.

Logs	Logs
<pre>"log": {   "emailError": true,   "database": true,   "dir": "string",   "emailRecipients": [     "string"   ],   "emailSender": "string", }</pre>	<pre>"log": {   "emailError": true,   "database": true,   "dir": "C:/xxx_xxxM/xxxx_logs_dir",   "emailRecipients": [     "xxxxxx@gmail.com"   ],   "emailSender": "xxxxxt@gmail.com", }</pre>

**NOTE:**

**Email Error:** If the user wants to get notified only in case of an error during a sync then this can be entered. It is a true/False field

**"EmailSender": "string",** : User can choose who should be the sender(email ID of sender)

This is an optional field. By default it is DBSync. This is to allow permissions for sending the calls.

**"EmailRecipients": [** : User can choose who should be the recipient (email ID of receiver)

**"string"**

**"dir": "string",** : this is alternative directory that the user can create to store the log files in their systme

**Database:** If the user wishes to store all information in the database and not in any logs files in the system then enter as true. The logs will be stored as a separate table in the database row-wise.

Callback URL	Callback URL
	If you wish to run the Call from the server enter the URL in this section.
"callbackURL": "string"	"callbackURL": "xxxxxxx@dddd.com"
Run	Run
"run": {  "command": "string",  "profile": "string"  }	"run": {  "command": "cleancopy",  }  

**NOTE:**

**Profile section :** Use this if you want to run a sync from the profile name.

If this option is used the above mentioned section of Salesforce , Database, Objects etc need not be entered., The sync can be run from the profile name itself.

You can enter the desired profile name once configured.

**Run:** Enter the command or function you wish to do.

**For eg:** Clean Copy, Update Schema, Source to DB, DB to Source

The below is an example showing how to run Clean Copy functionality for the profile DBSyncctest

```
{  
  
  "run": {  
  
    "command": "cleancopy",  
  
    "profile": "xxxxxx"  
  
  }  
  
}
```

- Click Execute
- The responses will appear as CURL
- A request URL is given in the REQUEST URL BOX
- Response Body  
Status, Submit time, Process ID and CallbackURL can be found in this section.  
You can also see the Status as submitted if successfully submitted or Failed if any error.
- Process ID:  
  
The process ID is used to check STATUS of a running process or can be used as a trigger.
- b. Check Status(POST) :** This is to check the status of the replication process  
  
**Enter the Process ID acquired in the 1st option to get the status and Click Execute**
- The status will show as in-progress or complete.