



Celigo + D7SMS Integration

Celigo + D7SMS integration allows you to send SMS messages to mobile numbers using the "REST API" option provided.

Requirements:

1. Celigo account
2. D7SMS account (You can also use trial account without a purchase)
3. Positive balance on D7SMS account

Notes:

- This automation action sends outbound messages only. For tracking, replies, reporting, and billing, please check D7SMS account at <https://app.d7networks.com>
- Messages sent with this automation action have a 700 character limit. This includes field values for any personalization tags you use in the message.
- The recipient's phone number should have a country code prefix. You can also set the default country [here](#), and all your messages will be prefixed with the selected country code..

How the D7SMS app works on Celigo:

- ❖ Generate D7SMS Token
- ❖ Create D7SMS connection at Celigo
- ❖ Configure Authentication
- ❖ Choose D7SMS on flowbuilder
- ❖ Configure message parameters
- ❖ Test it and activate automation

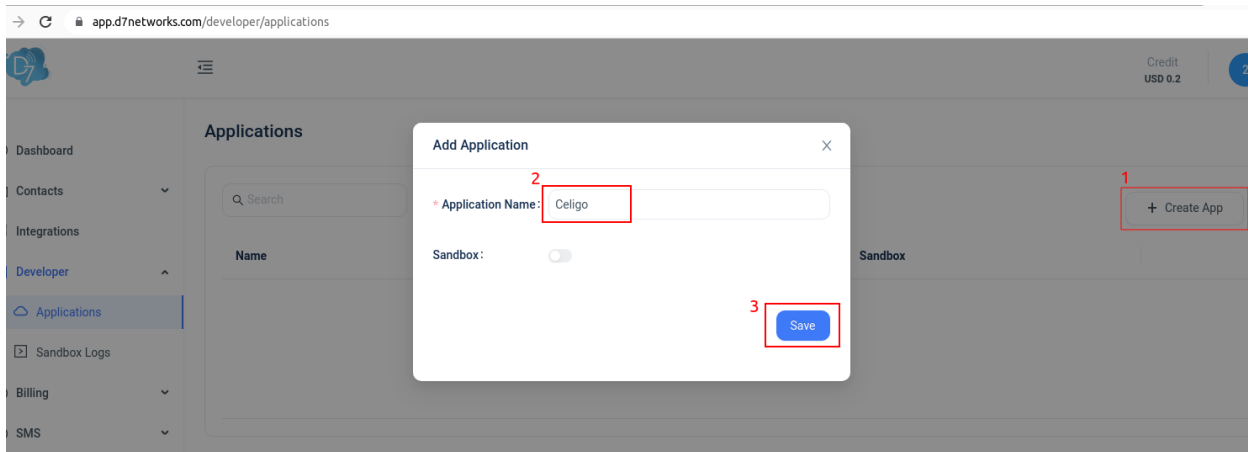


1. Generate D7SMS Token:

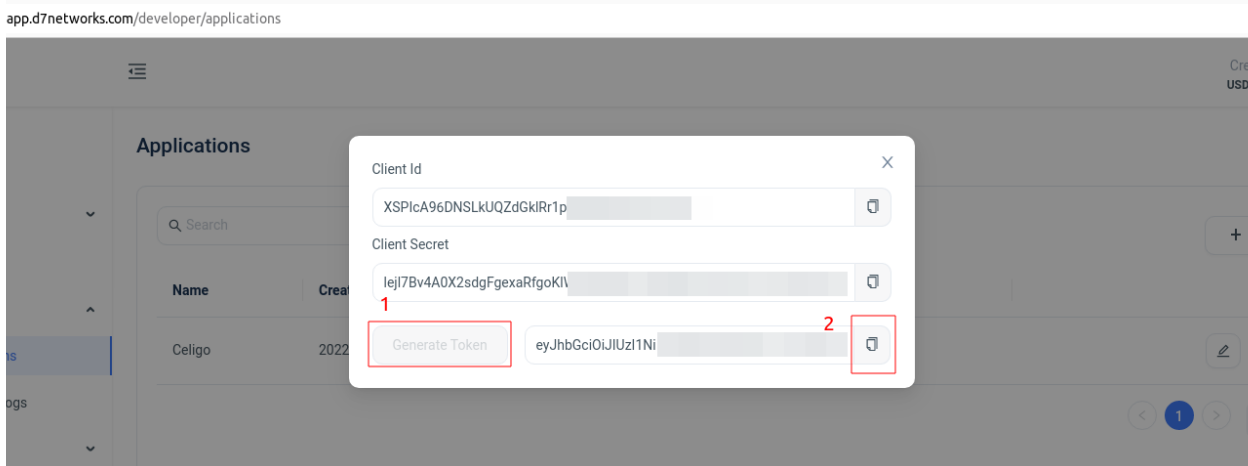
D7SMS connection uses OAuth 2.0 based Bearer token authentication and you can create this token at the developer section of <https://app.d7networks.com/>

Following are the steps by step instructions to create the token:

- Signup with <https://app.d7networks.com/>
- Navigate to <https://app.d7networks.com/developer/applications>
- Click on “**Create App**”
- Enter an **Application Name** and “**Save**”



- On the next window click on “**Generate Token**” and copy the token generated.





2. Create D7SMS connection at Celigo:

- Log in to Celigo account and Navigate to “Resources > Connections”
- Click on “**Create Connection**”

The screenshot shows the Celigo web interface at the URL `integrator.io/connections`. The left sidebar contains navigation options: Home, Dashboard, Tools, Resources, Connections (highlighted with a red box and a '1' badge), Imports, and Exports. The main content area displays a table of connections. A red box with a '2' badge highlights the '+ Create connection' button in the top right corner. The table below shows one connection named 'D7SMS' with the following details:

Name	Status	Type	API	Last updated ↓	Queue size	Actions
D7SMS	● Online	HTTP	<code>https://api.d7networks.com</code>	11/14/2022 10:47:33 am	0	...

c. Select “REST API”

The screenshot shows the 'Create connection' dialog box in Celigo. The 'Application' section is active, displaying a search bar and a list of applications. The 'REST API (HTTP)' option is highlighted with a red box. The list is organized into categories: Databases, Universal connectors, and Connectors.

Application	Icon
Databases	
Amazon Redshift	
DynamoDB	
Google BigQuery	
Microsoft SQL	
MongoDB	
MySQL	
Oracle DB (SQL)	
PostgreSQL	
Snowflake	
Universal connectors	
AS2	
FTP	
GraphQL	
https://	
HTTP	
{REST} API	
REST API (HTTP)	
Wrapper	
Connectors	



d. Add name and Base URI

Name: D7SMS

Base URI :https://api.d7networks.com

Create connection HTTP connection guide <https://>

^ General

Name * ?

D7SMS

Application

REST API (HTTP)

Mode ?

Cloud

On-premise

^ Application details

Base URI * ?

https://api.d7networks.com

Configure HTTP headers ?

name	value
------	-------

Media type * ?

JSON

Override media type for success responses ?

Do not override

Override media type for error responses ?

Do not override



3. Configure authentication:

- a. Configure the following parameters as given below:

Auth type: Select “Token”

Token: Insert the D7SMS token which was generated at step 1

Send token via: Select “HTTP header”

Header name: Select “Authorization”

Header scheme: Select “Bearer”

The screenshot shows the 'Create connection' configuration window for an HTTP connection. The window title is 'Create connection' and it includes a link to the 'HTTP connection guide' and the URL 'https://'. The configuration is organized into several sections:

- Media type:** A dropdown menu set to 'JSON'.
- Override media type for success responses:** A dropdown menu set to 'Do not override'.
- Override media type for error responses:** A dropdown menu set to 'Do not override'.
- Configure authentication:** A section with a dropdown menu for 'Auth type' set to 'Token'.
- Configure token auth:** A section with a text field for 'Token' containing a masked token (dots), a dropdown menu for 'Send token via' set to 'HTTP header', a text field for 'Header name' set to 'Authorization', and a dropdown menu for 'Header scheme' set to 'Bearer'.
- Configure refresh token:** A checkbox labeled 'Configure refresh token' which is currently unchecked.

Red boxes in the original image highlight the 'Auth type', 'Token', 'Header name', and 'Header scheme' fields.



4. Choose D7SMS on flowbuilder:

- a. Open Tools > Flow builder
- b. Select and Configure Sources as your requirement
- c. Click on “**Add destination / lookup**”

The screenshot shows the Celigo Flow Builder interface. The left sidebar contains navigation options: Home, Dashboard, Tools, Flow builder (highlighted), Data loader, Reports, Resources, Help, Celigo university, Marketplace, and Product portal. The main workspace displays a 'New flow' with the title 'Send SMS via webhook'. It features two main sections: 'SOURCES' and 'DESTINATIONS & LOOKUPS'. The 'SOURCES' section contains a 'webhooks Listener' component. The 'DESTINATIONS & LOOKUPS' section contains an 'Add destination / lookup' component, which is highlighted with a red box. A dashed line connects the source to the destination, indicating a flow configuration.

d. Select “REST API”

The screenshot shows the 'Create destination / lookup' dialog box. It features a search bar at the top with the text 'Choose application or start typing to browse 250+ applications'. Below the search bar, there are three categories of applications: 'Databases', 'Universal connectors', and 'Connectors'. The 'REST API (HTTP)' option is highlighted with a red box.

Databases	
	Amazon Redshift
	DynamoDB
	Google BigQuery
	Microsoft SQL
	MongoDB
	MySQL
	Oracle DB (SQL)
	PostgreSQL
	Snowflake

Universal connectors	
	AS2
	FTP
	GraphQL
	HTTP
	REST API (HTTP)
	Wrapper

Connectors	
------------	--



e. Select "Import records into destination application" and Select D7SMS Connection.

Create destination / lookup ✕

Application ?
REST API (HTTP) 🔍

What would you like to do? * ?
Import records into destination application ▾

Connection * ?
D7SMS ● Online ^ + ✎

Please select

D7SMS



5. Configure HTTP Method, URI, and message parameters:

- Provide a **Name** and select “**POST**” as the HTTP method
- Set Relative URI as given below
Relative URI:/messages/v1/send

General

Name * ?
D7SMS - Send SMS

Description ?
D7SMS - Send SMS

Connection ?
D7SMS ● Online ▼ + ✎

One to many ?
 Yes (advanced)
 No

How would you like the records imported?

HTTP method * ?
POST

Relative URI ?
/messages/v1/send {}
Relative to: <https://api.d7networks.com>

Configure HTTP headers ?

name	value
------	-------

Override request media type ?
Do not override



- c. Configure body parameters as follows. You can refer handlebar resources by adding it inside `{{}}`

```
{
  "messages": [
    {
      "channel": "sms",
      "originator": "{{record.originator}}",
      "recipients": ["{{record.recipients}}"],
      "content": "{{record.content}}",
      "data_coding": "text",
      "client_ref": "integrator.io"
    }
  ]
}
```

The screenshot shows the 'Create import' configuration page for a REST client. The interface is split into two main sections: configuration on the left and a preview on the right.

Configuration Section (Left):

- How would you like the records imported?** (Expandable header)
- HTTP method ***: POST
- Relative URI**: /messages/v1/send (Relative to: https://api.d7networks.com)
- Configure HTTP headers**: A table with columns 'name' and 'value'.
- Override request media type**: Do not override
- Number of records per HTTP request**: 1
- HTTP request body**: A text area containing the JSON payload from the code block above, with a red box highlighting the handlebar syntax `{{record.recipients}}`.
- Non-standard API response patterns** (Expandable header)

Preview & send Section (Right):

- Buttons: Edit mock input, Preview, Send
- Preview button: Preview >
- Request URL**: Empty text field
- Response tabs: HTTP request (selected), HTTP response, Parsed output
- Response sub-tabs: Body (selected), Headers, Other



Build HTTP request body

AFE 1.0 AFE 2.0 ?

Handlebars guide Create lookup ? Auto preview Preview

Type your handlebars template here

```
{
  "messages": [
    {
      "channel": "sms",
      "originator": "{{record.originator}}",
      "recipients": ["{{record.recipients}}"],
      "content": "{{record.content}}",
      "data_coding": "text",
      "client_ref": "integrator.io"
    }
  ]
}
```

Resources available for your handlebars template

```
1- {
2-   "record": {
3-     "content": "This is test message from Celigo/integrator.io via webhook at 0514PM",
4-     "recipients": "+971509752655",
5-     "originator": "D7SMS"
6-   },
7-   "connection": {
8-     "name": "D7SMS",
9-     "http": {
10-      "unencrypted": {
11-        "field": "value"
12-      },
13-      "encrypted": "*****"
14-    }
15-   },
16-   "import": {
17-     "name": "SendSMS"
18-   },
19-   "settings": {
20-     "integration": {},
21-     "flow": {},
22-     "flowGrouping": {},
23-     "connection": {},
24-     "import": {}
25-   }
26- }
```

6. Test it and activate automation:

Create import

https://

Name * ?
SendSMS

Description ?

Connection ?
D7SMS ● Online +

One to many ?
 Yes (advanced)
 No

How would you like the records imported?

HTTP method * ?
POST

Relative URI ?
/messages/v1/send
Relative to: https://api.d7networks.com

Preview & send

Edit mock input Preview **Send** 1

2 Success!
Send >
1 Page, 1 Records

Request URL
https://api.d7networks.com/messages/v1/send

HTTP request HTTP response Parsed output

3 Body Headers Other

```
{
  "request_id": "f1d2b79c-ab97-4ff0-b0bf-7c1db0180e47",
  "status": "accepted",
  "created_at": "2022-11-08T15:18:18.879216"
}
```

