

**webMethods CloudStreams Provider for Kickbox**

Version 10.11 | January 2022

Contents

[1 Document Change History 3](#_Toc92778028)

[2 What is webMethods CloudStreams Provider for <SaaS Name>? 4](#_Toc92778029)

[3 Steps to create the connector connection 5](#_Toc92778030)

[4 webMethods CloudStreams Provider for <SaaS name> Connector 6](#_Toc92778031)

[4.1 Connector Details 6](#_Toc92778032)

[4.1.1 Supported Resources 6](#_Toc92778033)

[4.1.2 Usage Notes 6](#_Toc92778034)

# Document Change History

|  |  |
| --- | --- |
| **Document revision date** | **Summary of changes** |
| January 2022 | First release of this document. |

# What is webmethod’s Cloud Streams Provider for Kickbox

Kickbox is a white hat service provider, whose team has been building email technology for more than a decade -- from anti-spam to email encryption. This knowledge and experience comprise the foundation on which Kickbox is built and has helped us become the leader in the email verification industry.

Kickbox provides technology that promotes email best practices and improves deliverability. Our mission: ensure customers with opt-in contacts get their message to the inbox and to prevent all the rest from hitting Send.

# Steps to create the connector connection

## Getting an Kickbox API Key

If you haven't already, [sign up](https://app.kickbox.com/signup) for a Kickbox account. It's free to get started.

Once signed up and logged in, click on **API** on the left navigation and then click **Manage Keys**. In the top right, click **Add**.

During the API Key creation process, you will specify which API Endpoints the key has permissions to use.

## Sandbox Mode

When you create a new API key, you'll be asked if you want to create a **Production** key or a **Sandbox** key. Parts of Kickbox can be used in Sandbox mode to test your integration against mock results, without deducting any verification credits from your account. When using your Sandbox API key, all results will return as "deliverable" by default, but you can also [trigger specific responses from the API](https://docs.kickbox.com/docs/sandbox-api).

## Installing Connector

You need to use the previously created Connector package or download the Connector package from

Install the Connector package on **Integration Server**  
To know how to install the package on Integration Server,

## Creating Connection for SaaS Application

To use a Connector, you need to create a Connection for the SaaS application. Before proceeding further, make sure that you have the package and folder structure available in Designer to create the Connection.  
Otherwise, create using the below steps.

**Note**: You can skip the below steps if you already have the package in Designer.

## In Designer, create package and folder structure

## Open **Service Development** perspective in Designer Graphical user interface, text, application, Word Description automatically generated

## Configure IS as shown below **Note**: Provide the credentials as per your local installation. Graphical user interface, application Description automatically generated [IS Admin - Install Package](https://open-source.softwareag.com/Connector-Developer-Program/static/ad83640445c27735b1309a078ef4c077/d79e6/AddIS_Confirm.jpg)

## Create an IS package that holds all the assets required for testing the Connector [IS Admin - Install Package](https://open-source.softwareag.com/Connector-Developer-Program/static/36847f0c3002953d58db5cdce022054f/5c23a/Designer_createPackage.jpg)

## Save the IS package(**KickboxConnectorTest**) with the same name as the Connector for easy identification as shown below: Graphical user interface, text, application Description automatically generated

## Create a folder inside the package with the same name(**KickboxConnectorTest**) to avoid namespace conflicts. The below images show the folder creation

Graphical user interface, application, Word

Description automatically generated

## Create folder **connections** within KickBoxConnectorTest PackageGraphical user interface, application, Word Description automatically generated

## Similarly, you must create folder services to save your FlowServices in the further steps Graphical user interface, application Description automatically generated

## Once you have the package and folder ready in Designer, go to Integration Server Administration page > Solutions > CloudStreams Graphical user interface, text, application Description automatically generated

## In the CloudStreams page, you can see the installed Connector enabled in the Providers section Graphical user interface, text, application Description automatically generated

## Click the Connector name to show the option **Configure New Connection** [CloudStreams - CreateConnection](https://open-source.softwareag.com/Connector-Developer-Program/static/2c3162a6e6883a6d86af9be8e42dbe95/d8d09/cloudStreams_configureNewConnection.jpg)

## The **Configure New Connection** opens the Connection creation wizard. Enter all the required fields (Copy the Folder name from Step 2.1.6) and save them to create a connection to the backend. The below images shows a sample **Note**:- Click the **Advanced view** option to show all the fields. Graphical user interface, text, application Description automatically generated

2.6. Once the connection is created, you must enable the connection as shown below  
  
  
  
Graphical user interface, application

Description automatically generated

2.7. Copy the connection name that will be used for testing the predefined operation  
  
Graphical user interface, application

Description automatically generated

# Testing the Predefined Operation

3.1. Predefined operation is found in the Provider package in the path shown below:

3.2. Now go to the folder **KickBoxConnectorTest** created in step 2.1.7 and right-click to create a new FlowService **verifyEmail** as shown below:  
  
Graphical user interface, text, application

Description automatically generated

3.3. Drag and drop the predefined operation **verifyEmail** from the Connector package as shown in the below image and save the FlowService  
  
Graphical user interface, application

Description automatically generated

3.4. Select the dropped predefined operation in the FlowService and click the **Pipeline** to show the **parameters** and **$connectionAlias**. Copy the entire **Service In** input parameters as shown below:  
  
Graphical user interface, text, application, email

Description automatically generated

3.6. Run the service as shown below:  
  
Graphical user interface, text, application

Description automatically generated

3.7. Provide the inputs for parameters **emailId**, **$connectionAlias** and click **OK**  
**Note**: **$connectionAlias** should be the copied value from Step 2.7

3.8. Below is the screenshot of a successful run. Click the **Results** tab to verify if the response body, status, status message, and signature are as per the SaaS application  
  
A screenshot of a computer

Description automatically generated with medium confidence

# webMethods CloudStreams Provider for Kickbox Connector

## Connector Details

The connector details include:

* **SaaS Provider**: Kickbox
* **API Version**: NA
* **API Type**: REST
* **Developer**: Vachaspati Diwevedi, Vivek Kumar
* **Group**: Kickbox
* **CloudStreams Minimum Version Compatibility**: 10.11
* **Provider Package Name**: kickBox

### Supported Resources

List down resources supported by the provider:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service: Bundle** | | | | |
|  | **Name** | **Path** | **Method** | **Description** |
| 1 | VerifyEmail | /v2/verify | GET | Verify Single email. |
| 2 | Verify Multiple Email | /v2/verify-batch | PUT | Verify multiple email |
| 3 | CheckJobStatus | /verify-batch/{jobid} | GET | CheckJobStatus of Verify Multiple email |
| 4 | CheckCreditBalance | /v2/balance | GET | Check Credit Balance of your API Account |

### Usage Notes

Any special notes