

Oracle® Health Sciences mHealth Connector Cloud Service

Cloud-to-Cloud Connector Framework for Data Ingestion from a Third-Party Cloud Service

Release 1.4

F19304-01

June 2019

Contents

- Introduction
- Connector Framework Components
- Typical Workflow
- Create and Configure an HTTP Connector
- On-boarding a Device Cloud
- Documentation Accessibility

Introduction

The Cloud Connector framework allows customers to import data collected and managed by a third-party or device vendor's cloud platform to Oracle mHealth Connector Cloud. This document explains how device cloud vendors and the Oracle services team can use a set of standard connectors, including the standard HTTP connector, which accepts messages in JSON format from an external service.

If you are interested in connecting to mHealth Connector, please contact your Oracle sales representative or email Oracle Health Sciences at: healthsciences_ww_grp@oracle.com.

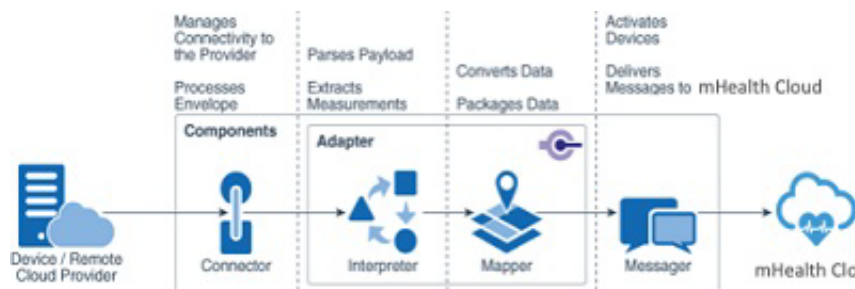
You can also visit the Oracle Health Sciences website at:

<https://www.oracle.com/industries/health-sciences/index.html>

Connector Framework Components

The four components of the connector framework are the connector, the interpreter, the mapper, and the messenger.

Figure 1 mHealth Connector Cloud-to-Cloud Connector framework components



- **Connector**—This provider-specific code manages connection and communication with the provider's devices and services. It's responsible for device discovery, connection management, and communication of data from the external devices or cloud services, including the processing of any provider-specific communication metadata, known as the envelope. The three types of connectors are generic, specialized, and custom.
- **Interpreter**—The interpreter parses device payloads and extracts data and telemetry information. Interpreters can be categorized as JSON or binary. Binary interpreters support additional payload grammars which can be used to parse device-native binary payloads.
- **Mapper**—The device model mapper takes data extracted from incoming device data and processes it into messages that are compatible with the mHealth Connector cloud message format. You can create a mapping between the fields extracted from the incoming data parsed by the interpreter, to the fields of mHealth Connector device model format. The mapping docCloud-to-Cloud Connector Framework for Data Ingestion from a Third-Party Cloud Serviceument associates a grammar and a device model.
- **Messenger**—The messenger communicates information about the device registration, device metadata, and device message to the Oracle Internet of Things service instance embedded within the mHealth Connector platform.

Typical Workflow

In a typical workflow, a device cloud vendor calls a REST endpoint published by the mHealth Connector. The mHealth Cloud Service framework invokes a series of mappings and transformation steps, configured for each clinical study, to create a new data message and stores the message in the mHealth Cloud Service.

Table 1 *Typical workflow*



Task	More Information	Description
Create an HTTP connector and configure it.	Give a name and credentials for authentication.	Create and configure HTTP Connector.
Verify that the connector has started.	Once the connector configuration is saved, it is automatically started by the framework. Review status and messages on the connector screen.	Create and configure HTTP Connector.

Table 1 (Cont.) Typical workflow

Task	More Information	Description
Create/upload device models.	Device models define the structure of a transformed message. Once a message is received, the interpreter transforms or maps incoming messages to a message format defined in the device model.	Create a new device model.
Create an interpreter.	The interpreter transforms or maps a message to an mHealth Connector message format. From the Interpreter screen, configure an interpreter specifying selection criteria, a device model, and the payload processing information. Create an interpreter for each device message format.	Configure interpreter.
Review device registration.	Send sample data from calling published mHealth Connector REST endpoint and verify that the virtual device instance and message created per the structure specified in the mapping screen of the interpreter.	Manage a virtual device.




Create and Configure an HTTP Connector

Connectors manage the connection and communication with devices or with third-party cloud services. They also process the metadata and telemetry data from the devices or the services. HTTP Connector hosts a set of HTTP servers that are front-ended with a load balancer. The HTTP Connector must be configured for basic authentication. For authentication configuration, create a user account with the administrator role in the embedded IoT cloud service console and configure the same in the Connector screen.


1. Open the Internet of Things Cloud Service management console.
2. Click **Menu** .
3. Click **Devices**, and then click **Connectors**.
4. On the Connectors page, click **Create New Connector** .
5. Enter values in the **Name** and **Description** fields.
6. If you need only one instance of the connector, select **1** in the **Scale Factor** field.
7. In the **Type** field, select **Generic**.
8. In the **Telemetry** section, select the appropriate values.
 - **Protocol:** Select **HTTP Server**.

- **Authentication:** Select **Basic** authentication, and then enter your **user name** and **password**. You must have administrative access to perform this step.
 - **Envelope:** Click **Add Grammar**, and add the details in the Grammar window. Use the **Map** field to map the required device metadata fields with the values of the interpreted envelope data.
 - **Payload encoding:** If your payload is in JSON, select **asciixhex**. If your payload is in base64-encoded binary, select **base64**.
9. Click **Save**. When you receive the message, *Saved successfully*, click **Back**.

The generic connector is listed on the Connectors page and the State field displays the status of the connector. The five states are STARTING, STARTED, FAILED, STOPPING, and STOPPED. STARTED indicates that the connector is ready to accept device data.

- To modify your generic connector, click **Edit** .
- To remove the generic connector, click **Delete** .
- To create an instance of your generic connector, click **Start** .

Note: The Edit, Delete, and Start buttons are enabled only when the connector is in STOPPED state.

- To view the device notifications that the connector receives, click **connectors_messageALERT** .

Note: On the Notifications page, click the data under Content to view the device information. You can also create an interpreter from this page when device data is received for the first time. See [Configure an Interpreter](#).


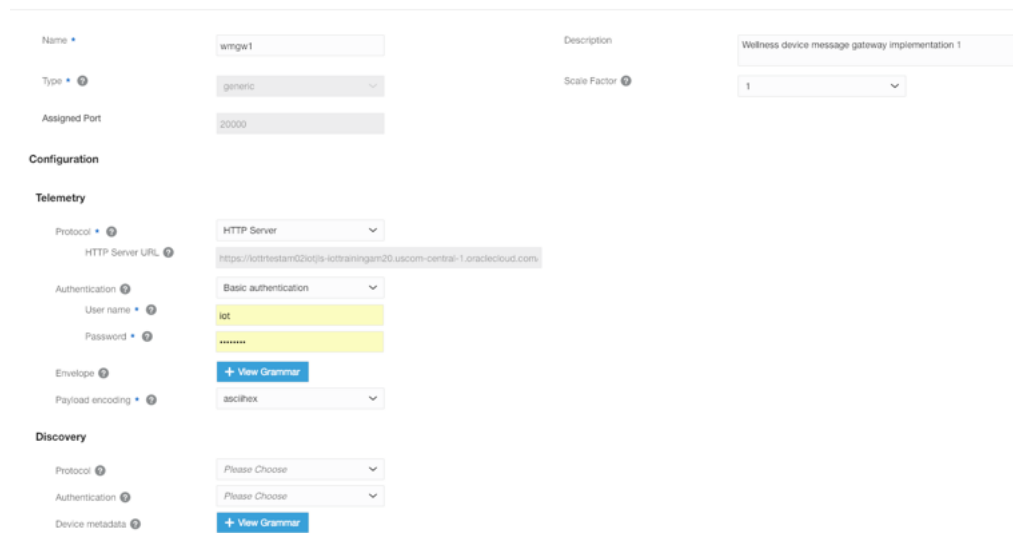
- To view the details of your generic connector, click **View** .

Figure 2 View the Cloud-to-Cloud Connector framework details



The screenshot displays the configuration interface for a generic connector. It includes the following sections and fields:

- Name:** wmgw1
- Description:** Wellness device message gateway implementation 1
- Type:** generic
- Assigned Port:** 20000
- Scale Factor:** 1
- Configuration:**
 - Telemetry:**
 - Protocol:** HTTP Server
 - HTTP Server URL:** https://kotttestam02lotjs-icttrainingam20.uscom-central-1.oraclecloud.com
 - Authentication:** Basic authentication
 - User name:** lot
 - Password:** *****
 - Envelope:** + View Grammar
 - Payload encoding:** asciixhex
 - Discovery:**
 - Protocol:** Please Choose
 - Authentication:** Please Choose
 - Device metadata:** + View Grammar

Create an Interpreter

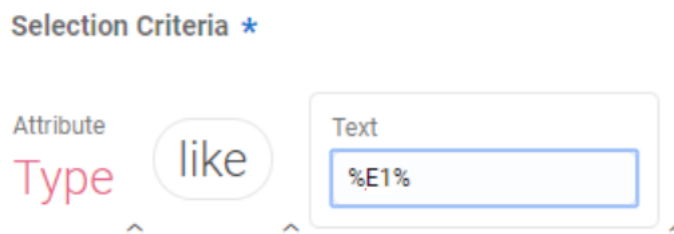
An interpreter parses the payload and extracts the device telemetry data. The framework then uses the data to form a device message in the mHealth Cloud. You create interpreters on the embedded IoT cloud service Devices configuration screen.

To create an interpreter you need the following:

- **Selection criteria:** The framework uses the selection criteria to decide which interpreter(s) to use to process the message received by a connector. If multiple interpreters are created with same selection criteria, the input messages will be processed by all those Interpreters. See [Configure an Interpreter](#) to define selection criteria.

Example: If the "type" attribute in the message payload contains string "E1," the Interpreter associated with this selection criterion will be used to process the message.



Figure 3 Selection criteria for creating an interpreter



- **Device model:** Identify the device model. The device model's attributes are used for mapping the message attributes.
- **Sample payload:** Provide a sample JSON payload data so that the mapping tool can start creating a source message-to-device model attributes mapping.
- **Device metadata:** Map the standard metadata fields such as Name, Hardware ID, or custom fields from parsed sample data (for example, Hardware ID mapped to KEY_UID (as per sample)).

Configure an Interpreter

To create an Interpreter from the management console:

1. Open the Oracle Internet of Things Cloud Service management console.
2. Click **Menu** .
3. Click **Devices**, and then click **Interpreters**.
4. On the Interpreters page, click **Create New Interpreter** .

5. Enter values in the **Name** and **Description** fields.
6. In the **Selection Criteria** field, select an attribute and map it to a pattern or value. The connector uses the selection criteria to select the interpreter.

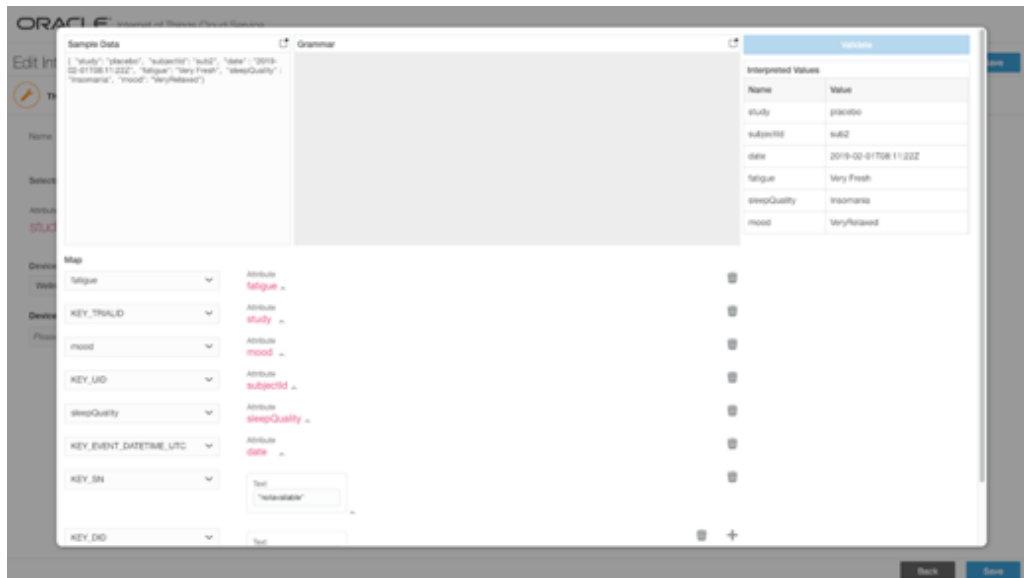
Figure 4 Selection criteria for configuring an interpreter

The screenshot shows a configuration form with the following elements:

- Name**: A text input field containing "wgtw1wellness".
- Description**: A text input field containing "wgtw1wellness".
- Selection Criteria**: A section with a red "Attribute" label and a dropdown menu showing "study". To the right is a "Text" input field containing "'placebo'".
- Device Model**: A dropdown menu showing "Wellness Device".
- Edit Grammar**: A blue button located to the right of the Device Model dropdown.



7. Select the **Device Model**, and click **Add Grammar**.
 - a. Do one of the following:
 - For a **JSON payload**, enter the payload data in Sample Data and click Validate. The interpreted values are listed.
 - For a **binary payload**, enter the payload in Sample Data, enter the binary grammar in Grammar, and click Validate. The interpreted values are listed.
 - b. From the **Map list** box, select an attribute of the device model.
 - c. From the adjacent combo box, select **Property**, **ATTRIBUTE**, and then the field of one of the interpreted values. Values can also be combined or manipulated using the Formula menu.
 - d. To select another attribute, click **Add**.
 - e. Repeat steps c and d to map the attributes of the device model that you need in the device message.

Figure 5 Map the attributes of the device model



8. In the **Device Metadata** section, select a field from the list box and map it to an attribute.
9. Click **Add**, and repeat the step for all the fields that you wish to map. These fields appear as metadata in the device record when the device is registered in Oracle Internet of Things Cloud Service.
10. Click **Save**.
11. When you receive a message, *Saved successfully*, click **Back**.

On the Interpreters page, your interpreter is listed and its Device Model is displayed.

- To modify any of the fields of your interpreter, click **Edit** .
- To remove the interpreter, click **Delete** .

On-boarding a Device Cloud

To send data to mHealth Cloud Service, a device cloud needs POST device messages for JSON format to a HTTP connector REST resource created in the mHealth Cloud Service. The entities within a JSON payload must be accessible using JSONPath notation. Although a data message can have an array of entities, it is expected that every POST message includes the same number of entities.

If one HTTP Connector handles data messages across multiple clinical studies, make sure that each message includes a study identifier field. Similarly, it is assumed that a subject identifier (a non-PII value) and a physical or virtual device identifier and /or serial number is included in every message. For example:

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

