Context Service in Unified CVP

Context Service

Cisco Context Service is a cloud-based omnichannel solution for Cisco Contact Center Express and Contact Center Enterprise. It enables you to capture your customer's interaction history by providing flexible storage of customer-interaction data across any channel.

Context Service works with Cisco Customer Collaboration products. Context Service also provides an SDK interface for integration with your own applications or third-party applications to capture end-to-end customer-interaction data.

For more information about Context Service and to check service availability, see https://cisco.com/go/contextservice.

Context Service Setup

For information about registering Unified CVP with Context Service, deregistering Context Service in Unified CVP, and configuring Context Service connection data, see the *Context Service Setup* section in Administration Guide for Cisco Unified Customer Voice Portal.

Integrate Context Service through Unified CVP

You can integrate Context Service into an existing contact center through Unified CVP by using several built-in Call Studio elements. For example, you can create a Piece of Data (POD) by using `POD_Add` element when a customer calls the contact center. A POD is a data element that can store customer interaction data, except for the media (such as audio recording). You can create, read, or update PODs and lookup customers by using the following Call Studio elements available under **Cisco** > **Context Service**.

- `POD_Add`

- `POD_Read`
- `POD_Update`
- `Customer_Lookup`

For more information about configuring Call Studio elements, see Element Specifications for Cisco Unified CVP VXML Server and Cisco Unified Call Studio.

Parsing JSON Data

The output of the Customer_Lookup and Pod_Read element can be in JSON format. The JSON output will have only the field set values along with POD/Customer ID.

For example:

```
[{"Context_Street_Address_1":"123 Sesame Street""
Context_State":"MI""Context_Mobile_Phone":"555-555-5555""Context_ZIP":"90210""Context_First_Name":"John"
"Context_City":"Detroit""Context_Work_Email":"api@test.com""Context_Country":"IND""Context_Last_Name":"Mayer"
"Customer Key":"41452dd0-b2e0-11e6-902b-dbb3691e8436"}{"Context_Street_Address_1":"123
Sesame Street"
"Context_State":"MI""Context_Mobile_Phone":"555-555-5555""Context_ZIP":"90210""Context_First_Name":"John"
"Context_City":"Detroit""Context_Work_Email":"api@test.com""Context_Country":"IND""Context_Last_Name":"Philps"
"Customer Key":"41112580-b2e0-11e6-b233-d7b3aba55b39"}]
```

This value is available in {Data.Element.Customer_Lookup_01.search_result_as_json}.

While parsing the JSON data to get a specific data value use, "Set_Value element's" Java Script option. For more details on "Set Value Element", refer to section "Local Variables" *Element Specifications for Cisco Unified CVP VXML Server and Cisco Unified Call Studio Guide*

The output of the Customer lookup and Pod Read element is fed as an input to the Set Variable element.

For example, this sample Java Script gets the value of Customer's first name(Context_First_Name):

```
importPackage(com.audium.server.cvpUtil);
var csdate = JSON.parse({Data.Element.Customer_Lookup_01.search_result_as_json});
var CustLastName = csdate[0].Context_Last_Name;
String(CustLastName);
```

Result

The Set variable gives the value as "Mayer" as in the Context Service.

POD.ID ECC Variable

You can define the `POD.ID` ECC variable to carry POD ID at a solution level. `POD.ID` ECC variable is used by a CVP VXML Server to send the POD ID to Unified ICM.

After creating the POD.ID ECC variable, you must add it to an ECC payload on ICM before you can use it.

When you configure a `POD_Add` element in a call flow, the element interacts with the Context Service for POD creation. The Context Service sends the POD ID to the `POD_Add` element. If the POD creation is successful, the `POD_Add` element stores the POD ID in the session as `PodId` session data. Now, if you configure a `CVP Subdialog Return` element in the call flow, after the subdialog returns, the VRU populates the `POD.ID` ECC variable with `PodId`. The Call Server subsequently sends the `POD.ID` ECC variable to Unified ICM.

Similarly, Unified ICM can send the `POD.ID` ECC variable to Unified CVP. During the subsequent VXML application execution, the POD ID is available as a session data on the CVP VXML Server where the key is `PodId` and the value is the POD ID sent in the ECC variable.

For more information about setting up the `POD.ID` ECC variable, see Configuration Guide for Cisco Unified Customer Voice Portal.