

# Web services data dip connector API

**Feature deprecation:** Genesys plans to deprecate the web services data dip connector on June 30, 2020 and the Bridge platform on January 1, 2021. Customers can migrate to the web services data actions integration, which runs as a Genesys Cloud service. For more information, see [Deprecation: Genesys Cloud Bridge integrations \(Resource Center\)](#) and [About the web services data actions integration \(Resource Center\)](#).

This API allows you to extend the functionality of the web services data dip connector. The connector works with Genesys Cloud and Architect and talks to any web services that you write using the API. You are responsible for writing the web services and tying them to your web service provider and your third-party system, for example, a CRM or a database.

You can use default (Web Service Contracts) or custom connector actions to create web services that implement the web services data dip connector API. For a C# example and library, see [webservice-datadip-connector-sample \(GitHub\)](#) and [webservice-datadip-connector-lib \(GitHub\)](#).

The default connector actions (Web Service Contracts) retrieve information about accounts, cases, or contacts from a third-party system. A custom connector action allows you to interact with information from a third-party system through user-defined schemas. For more information, see [Web Service Contracts](#) and [Add a bridge action for connectors \(Resource Center\)](#).

- What is the connector, Genesys Cloud, and Architect responsible for?
- What are you responsible for?
- What does the connector **not** do?
- What language do you need to use?
- Example user-defined schemas
  - Parameters
    - properties object
    - (property) object
  - Example use case with an SQL database
  - Example SQL REST service
    - Parameters
      - contactConfig object
      - accountConfig object

## What is the connector, Genesys Cloud, and Architect responsible for?

The connector, Genesys Cloud, and Architect handle the following tasks:

- Authentication between Genesys Cloud and the connector.
- The creation of connector groups.
- The publication of data actions.
- The routing of requests to the Bridge connector from Architect.

## What are you responsible for?

You are responsible for all aspects of your web service, web service provider, and third-party system (such as a CRM or database), which involve the following tasks:

- Writing the web service that implements the default connector actions (Web Service Contracts) or custom connector actions.
  - For more information, see [Web Service Contracts](#) and [Add a bridge action for connectors \(Resource Center\)](#).
- Tying the web service to your web service provider.
- Responding to requests.
- Receiving information from your third-party system.

## What does the connector **not** do?

The connector does **not** handle the following tasks:

- Authentication between the connector and the web service that you write.
- Authentication to your third-party system.
- Data formatting or manipulation.

## What language do you need to use?

The web service that you are writing does not live in the Bridge infrastructure, which means that you can develop your service using any language.

For more information, see [How the connector works \(Resource Center\)](#).

## Example user-defined schemas

Custom connector actions allow a more flexible exchange of data between your third-party system and the web services data dip connector. When you add a custom connector action in Genesys Cloud, you add user-defined request and response schemas. The schemas allow you to specify the information that you want to request and receive from your web service. For more information, see [Add a bridge action for connectors \(Resource Center\)](#).

**Important:**

- The connector only works with valid JSON schemas and does not validate the request and response bodies against the schemas.
- You are responsible for tying the schemas that you create to your web service and to the connector. For more information about setting up the connector, see [About the web services data dip connector \(Resource Center\)](#).

The following information describes the schemas.

### Parameters

| Name                 | Data type | Description                             | Notes   |
|----------------------|-----------|---|---|
| \$schema             | String    | Version of the schema standard used.    | Default: <a href="http://json-schema.org/draft-04/schema#">http://json-schema.org/draft-04/schema#</a> .<br><br><b>Warning:</b> Do not change.                            |
| title                | String    | Schema title.                           |   |
| description          | String    | Schema description.                     |   |
| type                 | String    | Schema type.                            | Default: object.<br><br><b>Warning:</b> Do not change.  |
| properties           | Object    | See <a href="#">properties object</a> . | These objects appear under <b>Inputs</b> in Architect call flows.<br><br><b>Warning:</b> Do not use nested objects. Schemas with nested objects do not flatten correctly. |
| additionalProperties | Boolean   | Additional properties for the schema.   | Default: true.<br><br><b>Warning:</b> Do not change.  |

### PROPERTIES OBJECT

| Name       | Data type | Description  | Notes   |
|------------|-----------|--|---|
| (property) | Object    | Names of object. See <a href="#">(property) object</a> . | These objects appear under <b>Inputs</b> in Architect call flows.<br><br><b>Warning:</b> Do not use nested objects. Schemas with nested objects do not flatten correctly. |

### {PROPERTY} OBJECT

| Name        | Data type | Description                | Notes  |
|-------------|-----------|----------------------------|--|
| type        | Object    | Type of object.            | Supports all JSON types, except object, null, and nested arrays. |
| description | String    | Description of the object. |  |

**Request Schema**

```

1 - {
2   "schema": "http://json-schema.org/draft-04/schema#",
3   "title": "Code Request Schema",
4   "description": "Schema to send a code for evaluation.",
5   "type": "object",
6   "properties": {
7     "code": {
8       "type": "string",
9       "description": "Your entry code"
10    }
11  },
12  "additionalProperties": true
13 }
```

**Response Schema**

```

1 - {
2   "schema": "http://json-schema.org/draft-04/schema#",
3   "title": "Code Status Response Schema",
4   "description": "The schema for the results of code evaluation",
5   "type": "object",
6   "properties": {
7     "status": {
8       "type": "string",
9       "description": "The status of your code"
10    },
11    "prize": {
12      "type": "string",
13      "description": "The prize won, if any"
14    }
15  },
16  "additionalProperties": true
17 }
```

## Example use case with an SQL database

The following workflow outlines how to use the web services data dip connector and a web service that connects to a SQL database.

- A phone call is routed to the Genesys Cloud Edge, which passes the call to the IVR call flow that you created in Architect.
- The IVR call flow on the Genesys Cloud Edge routes Bridge actions to Genesys Cloud.
- Genesys Cloud passes the Bridge actions to the correct Bridge Server where the connector lives.
- The connector contacts your web service using the API that you implemented.

**Important:**

- You are responsible for creating the web service, for authentication between your web service and your SQL database, and for defining the SQL queries and data transforms that retrieve information from your SQL database as responses to requests from your IVR.
- Your web service must implement the Web Service Contracts or custom connector actions. For more information, see [Web Service Contracts](#) and [Add a bridge action for connectors \(Resource Center\)](#).

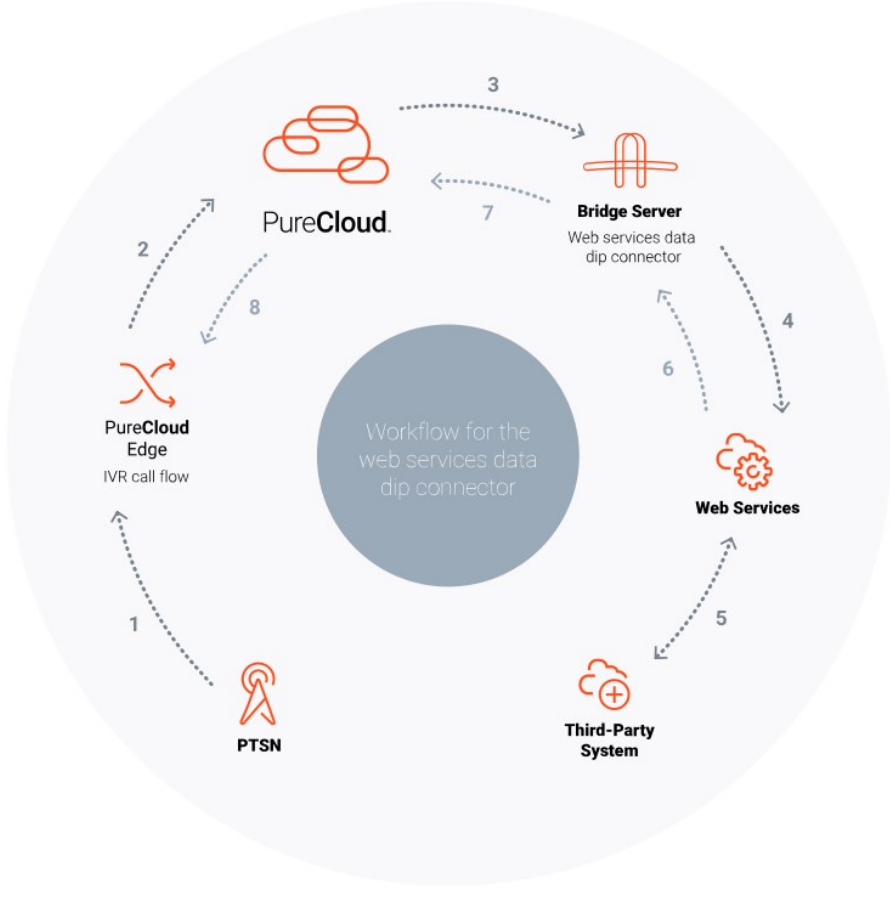
- Your web service contacts your SQL database, which returns information to your web service.

**Important:** You are responsible for this portion of the workflow.

- Your web service returns the information to the connector in the appropriate format.

**Important:** You are responsible for the formatting of data being returned to the connector.

- The connector sends the information back to Genesys Cloud by means of the Bridge Server.
- Genesys Cloud sends this information to the IVR call flow, which performs tasks that you set up in Architect.



For more information, see [How the data dip connector works \(Resource Center\)](#).

## Example SQL REST service

The following information describes an example .NET REST service with a SQL backend that uses the web services data dip connector API. For the source code for this example, see [SqlRestService \(GitHub\)](#).

**Important:**

- The service supports only one address for accounts.
- `getMostRecentOpenCaseByContactId` is not implemented.
- If multiple results are found, queries return only the first result.
- For all queries, the names of the result columns must match the names of the return fields.
- "%1" is used as a placeholder for the input parameter of the request. For example, with `getContactByPhoneNumber`, "%1" is replaced by the phone number.
- The ServiceConfig.json file must be in the same directory as the executable application.

The following information describes elements in the ServiceConfig.json file.

### Parameters

| Name                             | Data type | Description   | Notes  |
|----------------------------------|-----------|---|--|
| URL                              | String    | URL and port of the service at runtime.   | Default: <a href="http://127.0.0.1:8088">http://127.0.0.1:8088</a> , if the service runs on a Bridge Server. |
| connectionString                 | String    | SQL database connection string.   |  |
| getAccountByPhoneNumber          | String    | Query for contact information by phone number.  |  |
| getAccountByPhoneNumber          | String    | Query for account information by phone number.  |  |
| getAccountByAccountNumber        | String    | Query for account information by account number   |  |
| getAccountByContactId            | String    | Query for account information by contact ID.  |  |
| getMostRecentOpenCaseByContactId | String    | Query for case information by contact ID.   |  |
| contactConfig                    | Object    | Instructions for retrieving contact information. See <a href="#">contactConfig object</a> . |  |
| accountConfig                    | Object    | Instructions for retrieving account information. See <a href="#">accountConfig object</a> . |  |

### CONTACTCONFIG OBJECT

| Name          | Data type | Description  | Notes  |
|---------------|-----------|--|--|
| fields        | Array     | List of contact fields to return.  | Values: FirstName, LastName, FullName, Id, CustomAttribute.          |
| getAddress    | Boolean   | The application returns an address (true) or does not return an address (false). |  |
| addressFields | Array     | List of address fields to return.  | Values: Line1, Line2, Line3, City, State, Country, Type, PostalCode. |
| phoneCount    | Integer   | Number of phone numbers to return.   |  |
| phoneMappings | Array     | Ordered list of phone numbers to return.   | The first item in the list is returned as Phone1.                    |
| emailCount    | Integer   | Number of email addresses to return.   |  |
| emailMappings | Array     | Ordered list of email addresses to return.                                       | The first item in the list is returned as Email1.                    |

### ACCOUNTCONFIG OBJECT

| Name          | Data type | Description  | Notes  |
|---------------|-----------|--|--|
| fields        | Array     | List of contact fields to return.  | Values: FirstName, LastName, FullName, Id, CustomAttribute.          |
| getAddress    | Boolean   | The application returns an address (true) or does not return an address (false). |  |
| addressFields | Array     | List of address fields to return.  | Values: Line1, Line2, Line3, City, State, Country, Type, PostalCode. |
| phoneCount    | Integer   | Number of phone numbers to return.   |  |
| phoneMappings | Array     | Ordered list of phone numbers to return.   | The first item in the list is returned as Phone1.                    |
| emailCount    | Integer   | Number of email addresses to return.   |  |
| emailMappings | Array     | Ordered list of email addresses to return.                                       | The first item in the list is returned as Email1.                    |

For more information about the web services data dip connector, see [About the web services data dip connector \(Resource Center\)](#).

Was this page helpful?