

Oracle DB Connector

Last Modified on 12/15/2023 5:50 pm EST

On this page:

[Setting up Oracle DB](#)

[Required Authentication Parameters](#)

[Connector Fact Sheet and Reference](#)

[Connector API Documentation](#)

[Provider Documentation](#)

Setting up Oracle DB

You can create a connection to a Oracle DB in one of the two ways:

- Directly via an IP Address and Port Number, e.g. `123.123.1.123:3306`
- Use [Ground2Cloud](#) service to connect your Database

Connecting Directly via IP Address and Port Number

This method would require a port be exposed so a connection can be made with SAP Open Connectors. When creating an instance, the user would input the IP Address and Port Number exposed publicly.

Connecting via Ground2Cloud

The Ground2Cloud integration consists of two parts: Client and Server. The Ground2Cloud Client creates a tunnel to a public Ground2Cloud Server, and enables requests from the SAP Open Connectors Production Cloud to transparently pass through that tunnel to reach the Client Service.

The Ground2Cloud Client installation program is a self-unpacking executable. Once it finishes running, the Ground2Cloud Client is installed as a Windows Service which constantly runs to keep this tunnel open. You generally don't have to worry about this; once installed, the service automatically restarts in case of failure, or when your Windows machine is rebooted.

The installer also installs a GUI (Graphical User Interface) program, which can be used to monitor and manage the Ground2Cloud Client. When launched, it opens a window with simple dialogs that let you browse logs files, change configuration, and perform other management operations. Details on how to use the GUI is described in the [Ground2Cloud docs](#).

If you are interested in using our Ground2Cloud Service, contact us for details.

Required Authentication Parameters

SAP Open Connectors requires a database installed or any hosted database services like AWS, RDS etc. before you authenticate with the Oracle DB connector. This connector uses a **custom** authentication workflow.

The following is required to create a Oracle DB connector instance:

- Database Host: e.g. `123.123.1.123:3306`
- Database Name
- Database Username
- Database Password
- Database Schema Name
- Database Tables (**OPTIONAL**: Can connect a set of tables i.e. contacts, accounts or prefixed tables, i.e. `data_*` via comma separated list)

You can authenticate a connector instance using the UI or via API calls. For detailed information on how to authenticate an instance, see our related documentation:

- [Authenticate a Connector Instance \(UI\)](#)
- [Authenticate a Connector Instance \(API\)](#)

Sample Configuration JSON

```
"configuration": {
  "db.host": "[myDatabaseHost]",
  "db.name": "[myDatabase]",
  "username": "[myUsername]",
  "password": "[myPassword]",
  "db.schemaname": "[myDatabaseSchemaName]"
}
```

API docs before provisioning an instance with the Oracle DB SAP Open Connectors look like they're shown below.

bulk	
GET	/bulk/{id}/{objectName} Retrieve the results of an asynchronous bulk query.
PUT	/bulk/{id}/cancel Cancel an asynchronous bulk query job.
GET	/bulk/{id}/errors Retrieve the errors of a bulk job.
GET	/bulk/{id}/status Retrieve the status of a bulk job.
POST	/bulk/{objectName} Upload a file of objects to be bulk uploaded to the provider.
GET	/bulk/jobs Fetch all the bulk jobs for an instance
POST	/bulk/query Create an asynchronous bulk query job.

When you successfully authenticate an instance with the Oracle DB connector, SAP Open Connectors makes a connection to Oracle Database, reads the database schema and converts all its tables to APIs. For example, tables such as Countries, Departments, Regions etc. in the database are generated dynamically into APIs by SAP Open Connectors, as shown below.

bulk	
GET	/bulk/{id}/{objectName} Retrieve the results of an asynchronous bulk query.
PUT	/bulk/{id}/cancel Cancel an asynchronous bulk query job.
GET	/bulk/{id}/errors Retrieve the errors of a bulk job.
GET	/bulk/{id}/status Retrieve the status of a bulk job.
POST	/bulk/{objectName} Upload a file of objects to be bulk uploaded to the provider.
GET	/bulk/jobs Fetch all the bulk jobs for an instance
POST	/bulk/query Create an asynchronous bulk query job.

When changes are made to the schema in Oracle database, you need to re-authenticate the connector instance in order to pull whatever changes have been made in Oracle database and for endpoints to get added or removed accordingly.

③ **Note:** If you'd like to see specific tables as APIs, you can provide table names separated by commas under the optional field called Table Names, while you're creating an instance.

HIDE OPTIONAL FIELDS

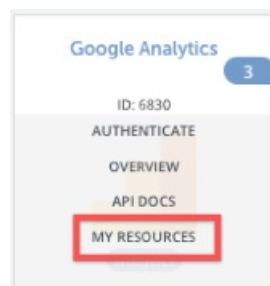
Table Names ?

Connector Fact Sheet and Reference

Authentication and Configuration Parameters

To see all authentication parameters for the Oracle DB connector, follow these steps:

1. Sign in to SAP Open Connectors and navigate to Connectors.
2. Hover over the card for a connector, and then click My Resources.



3. In the top navigation toolbar, click Setup.
4. From the Setup page, you can view a complete list of the connector's authentication and configuration parameters.

Events and Supported Resources

The Oracle DB connector supports events via polling. For detailed information about our Events framework and how to configure a connector instance with events, see our documentation:

- [Authenticate a Connector Instance with Events \(UI\)](#)
- [Authenticate a Connector Instance with Events \(API\)](#)

You can set up events via polling for the `tableName` resource of the Oracle DB connector.

Queryable Endpoints

You can use OCNQL to query the following endpoints of the Oracle DB connector:

- GET /bulk/jobs
- GET /{objectName}
- GET /{objectName}/{objectId}/{childObjectName}

Connector API Documentation

See the [API Documentation](#) for more details.

Provider Documentation

- [Oracle Database REST Documentation](#)
-