

Salesforce Open API Implementation

Overview

Salesforce has implemented 8 Open API services within the platform. These services are hosted directly on the Salesforce platform and map to internal Salesforce Objects directly. At Salesforce we've taken the approach to only implement those APIs that conform with the Salesforce out of the box Object model. The APIs are exposed using an APEX framework that allows the user to define the mapping using configuration (called Custom Metadata within Salesforce).

FAQ

What is the general design principle Salesforce used to implement the OPEN APIs

Salesforce only mapped resource model concepts that exist as out of the box Objects within Salesforce, and extended fields on the core Objects where appropriate

Why are some of the sub-nodes in the services not mapped within Salesforce

The OPEN API specification relies on a specific resource model (like party roles, product specifications, etc.) - these concepts are not directly implemented in Salesforce and hence not mapped. Salesforce tends to have a more normalised Object model, and while the resource model can be fully implemented in Salesforce - this may not conform to out of the box Object model concepts.

How do IDs work with the Service responses

In all of the services - where there is an ID field, this maps to a Salesforce ID. [Salesforce employs a unique ID model](#) across all records and these are what are returned in all ID fields. This allows an end consumer of the service to use the OPEN APIs in coordination with standard Salesforce APIs (if required).

Why aren't all the HTTP Methods Implemented

Salesforce has strived to support the basic GET, PATCH, POST commands (retrieving data, updating data, and inserting data). The reason PUT commands are not supported is that the way the specification is written it requires an all or nothing update (which Salesforce does not generally support - and would require significant coding). In addition the DELETE method is not supported - as there are various implications when deleting records in Salesforce. In general Salesforce does not recommend deleting records from a consumption Use Case - rather marking records correctly (and only deleting as part of an Data Quality process)

Service Patterns

All services on Salesforce are implemented in the following format:

[**Salesforce base URL**]/services/apexrest/openAPI/[**Open API service**]

Here is an example of calling the **/customerAccount** API:

<https://eu11.salesforce.com/services/apexrest/openAPI/customerAccount>

All service patterns support retrieving all collections, or with an ID. The previous example of **/customerAccount** API will retrieve the collection of customerAccounts, whereas this sample will retrieve a single account:

<https://eu11.salesforce.com/services/apexrest/openAPI/customerAccount/001460000XczF4AAJ>

Salesforce OPEN API Mapping Specs

Salesforce Reference Implementation

As mentioned in the overview this implementation on Salesforce uses a developed Framework. If you would like to get a copy of that framework to deploy to your Salesforce org - you can find it on Github here: <https://github.com/baumannndm-sf/TMForumHackathon>

Postman Collection (Sample Calls)

Here is a collection of example calls that cover all the services as a Postman collection

<https://www.getpostman.com/collections/d003c76b10896c31b7b9>

Specification Documentation

The following table links through to the documentation that shows what fields are mapped to which internal Salesforce Objects and fields.

Service	Link
/customerManagement/customer/	customer API Salesforce Mapping
/customerManagement/customerAccount/	customerAccount API Salesforce Mapping
/agreementmanagement/agreement/	agreement API Salesforce Mapping
/orderManagement/productOrder/	productOrder API Salesforce Mapping
/catalogueManagement/productOffering/	productOffering API Salesforce Mapping
/inventoryApi/product/	product API Salesforce Mapping
/API/troubleTicket/	troubleTicket API Salesforce Mapping
/quoteManagement/quote/	quote API Salesforce Mapping

How to Authenticate to Salesforce

Salesforce has a number of ways to authenticate in order to use the APIs. The basic form is to authenticate to retrieve a session token that is required when calling the API. Salesforce recommends using an OAuth.

You can refer to the following article ([Using OAuth to Authorize External Applications](#)) - this will explain how to authenticate and use services, in addition to some sample code.