ATS External Integration Guide

Introduction

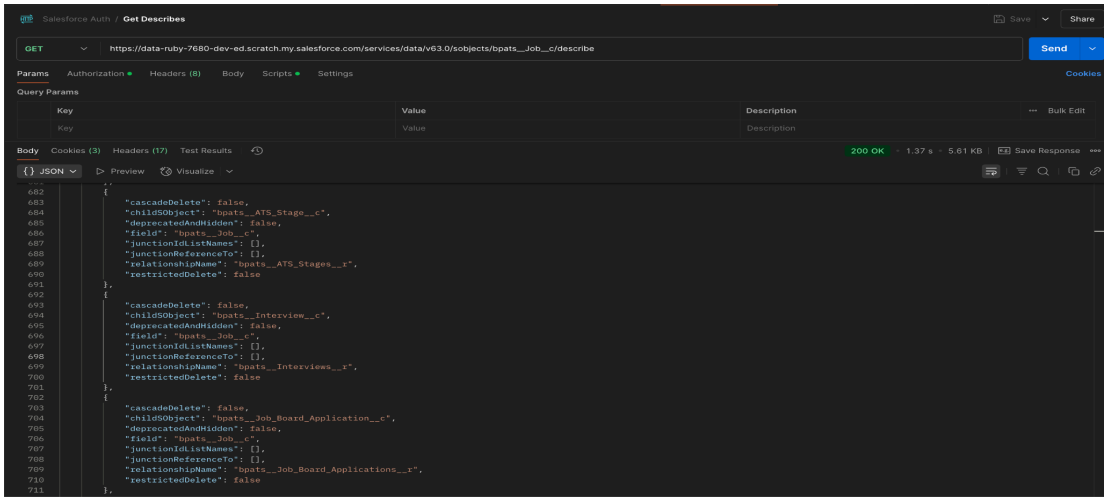
This reference guide provides general information about each Applicant Tracking System (ATS) and specific information about each API operation and supported object type. This document provides an overview of how external systems can integrate with our ATS. It includes methods for establishing connections, retrieving job listings, and submitting job applications through various integration points.

Objects

Inthe ATS, the term object refers to a specific type of record used within the system. Examples of these object include Job, Contact, Placement etc. Each object represents a fundamental concept in ATS, helping to structure how staffing data is stored, managed, and processed according to the system's rules and logic.

The external system can get the object by using the salesforce standard API.

Endpoint:
https://<your-instance>.salesforce.com/services/data/vXX.0/sobjects/<your-object-api-name>/describe



Note: Before accessing objects through the API, make sure the external system is properly authenticated with your Salesforce org.

JSON

JSON (JavaScript Object Notation) is a lightweight, text-based data format used for storing and exchanging data between systems—especially in web applications. It is easy for humans to read and write, and easy for machines to parse and generate.

Salesforce Versions appendix

In the below examples replace the 'vXX.0' placeholder for the version of the Salesforce API you want to use — for example:

- v59.0 = Spring '24 release
- v60.0 = Summer '24 release
- v61.0 = Winter '25 release

Use the latest version your org supports to:

- Access the latest features and field types
- Avoid deprecated functionality
- Improve compatibility with newer Salesforce objects or APIs

API Information

Base URL : https://<your-instance>.salesforce.com/services/data/vXX.0/

- Replace <your-instance> with your Salesforce domain.
- Replace vXX.0 with your Salesforce API version (e.g., v60.0 for Spring '24).

Common Request Headers

Header	Type	Required	Example	Description
Authorization	String	Yes	Bearer <Access_Token>	OAuth 2.0 Bearer Token

Content-Type	String	Yes	application/json	JSON body expected
Accept	String	Yes	*/*	Expected response content type

Access Token can be obtained <TBD>

Request Parameters

Parameter Location	Name	Type	Required	Example	Description
Query (URL)	q (for SOQL query)	String	Yes (for query endpoint)	SELECT Name FROM bpats__Job__c	SOQL statement for record fetching
Path	Object_API_Name	String	Yes	bpats__Job__c	The Salesforce API name of the object you're targeting
Path	recordId	String (18-char Salesforce ID)	Yes (for single record updates/deletes)	a01ABC1234xyz123	Unique ID of the record

Request Body (Payload)

For POST, PATCH, and Composite endpoints:

- Format: application/json
- Structure: JSON object
- Field Validation Rules:

Field Property	Rule
Data Types	String, Number, Boolean, Array (for composite bulk ops)
Maximum Length	Depends on field metadata in Salesforce (commonly 255 characters for text fields)
Required Fields	Must be filled if Salesforce marks them as Required in Object Schema
Optional Fields	Can be omitted if not mandatory
Lookup Fields	Provide 18-digit Salesforce Record ID for references

asymbl.

Status Codes and Error Responses

HTTP response code	Description
200	"OK" success code, for GET, HEAD, and some PATCH requests.
201	"Created" success code, for POST requests and some PATCH requests.
204	"No Content" success code, for DELETE requests and some PATCH requests.
300	The value returned when an external ID exists in more than one record. The response body contains the list of matching records.
304	The request content hasn't changed since a specified date and time. The date and time is provided in a If-Modified-Since header. See Get Object Metatdata Changes for an example.
400	The request couldn't be understood, usually because the JSON or XML body contains an error.
401	The session ID or OAuth token used has expired or is invalid. The response body contains the message and errorCode.
403	The request has been refused. Verify that the logged-in user has appropriate permissions. If the error code is REQUEST_LIMIT_EXCEEDED, you've exceeded API request limits in your org.
404	The requested resource couldn't be found. Check the URI for errors, and verify that there are no sharing issues.
405	The method specified in the Request-Line isn't allowed for the resource specified in the URI.
409	The request couldn't be completed due to a conflict with the current state of the resource. Check that the API version is compatible with the resource you're requesting.
410	The requested resource has been retired or removed. Delete or update any references to the resource.
412	The request wasn't executed because one or more of the preconditions that the client specified in the request headers wasn't satisfied. For example, the request includes an If-Unmodified-Since header, but the data was modified after the specified date.
414	The length of the URI exceeds the 16,384-byte limit.
415	The entity in the request is in a format that's not supported by the specified method.

420	Salesforce Edge doesn't have routing information available for this request host. Contact Salesforce Customer Support.
428	The request wasn't executed because it wasn't conditional. Add one of the Conditional Request Headers, such as If-Match, to the request and resubmit it.
431	The combined length of the URI and headers exceeds the 16,384-byte limit.
500	An error has occurred within Lightning Platform, so the request couldn't be completed. Contact Salesforce Customer Support.
502	Salesforce Edge wasn't able to communicate successfully with the Salesforce instance.
503	The server is unavailable to handle the request. Typically this issue occurs if the server is down for maintenance or is overloaded.

Response Body Schemas

For successful responses:

Field	Type	Description
id (for POST)	String	The Salesforce ID of the created record.
success (for POST/Composite)	Boolean	Indicates if the operation was successful.
errors (for POST/Composite)	Array	List of errors, if any.
records (for Query)	Array	Array of records matching query criteria.

Example: Successful POST Response (Create Record)

```javascript
{
   "hasErrors": false,
   "results": [
      {
         "referenceId": "ref1",
         "id": "a0LEm000007fZDxMAM"
      },
      {
```

```
        "referenceId": "ref2",
        "id": "a0LEm000007fZDyMAM"
      }
    ]
}
```

Example: Successful GET Query Response

```
JavaScript
{
   "totalSize": 1,
   "done": true,
   "records": [
      {
         "attributes": {
            "type": "bpats__Job__c",
            "url": "/services/data/v57.0/sobjects/bpats__Job__c/a0MEm000004fFnBMAU"
         },
         "Name": "Salesforce Developer Test",
         "bpats__End_Date__c": null
      }
   ]
}
```

Error Responses

If something goes wrong, Salesforce returns a standard error format.

Field	Type	Description
message	String	Human-readable error message
errorCode	String	Salesforce error code
fields	Array	List of fields causing the error (optional)

Example: 400 Bad Request

```javascript
{
   "hasErrors": true,
   "results": [
      {
         "referenceId": "ref3",
         "errors": [
            {
               "statusCode": "REQUIRED_FIELD_MISSING",
               "message": "Required fields are missing: [LastName]",
               "fields": [
                  "LastName"
               ]
            }
         ]
      }
   ]
}
```

Possible Causes:

- Your API request body is missing a **required field** (in this case, LastName).
- Validation rules on the Salesforce object require this field.

How to Handle:

- Before sending the request, ensure that all mandatory fields are populated.
- Implement client-side or server-side validation to check required fields before making the API call.

Example: 401 Unauthorized (Invalid Token)

```javascript
[
 {
  "message": "Session expired or invalid",
  "errorCode": "INVALID_SESSION_ID"
 }
]
```

Possible Causes:

- The session (OAuth token or session ID) is **expired**, **revoked**, or **invalid**.
- The user was logged out, or the token has timed out.

How to Handle:

- Reauthenticate and obtain a new access token/session ID.
- Implement automatic token refresh logic if using OAuth.
- Check your session timeout settings and ensure sessions are still valid before making requests.

Example: 400 Not Found

```javascript
[
 {
  "message": "The requested resource does not exist",
  "errorCode": "NOT_FOUND"
 }
]
```

Possible Causes:

- You're trying to access a record, object, or endpoint that doesn't exist.
- You might have a typo in the URL or the record ID provided is wrong.
- The resource might have been deleted.

How to Handle:

- Verify that the URL is correct.
- Ensure the record or resource actually exists before requesting it.
- Implement error handling to gracefully show "resource not found" messages.

Example: 400 REQUIRED FIELD MISSING

```javascript
{
   "hasErrors": true,
   "results": [
      {
         "referenceId": "ref3",
         "errors": [
            {
               "statusCode": "REQUIRED_FIELD_MISSING",
               "message": "Required fields are missing: [LastName]",
               "fields": [
                  "LastName"
               ]
            }
         ]
      }
   ]
}
```

Possible Causes:

- Your API request body is missing a required field (in this case, LastName).
- Validation rules on the Salesforce object require this field.

How to Handle:

- Before sending the request, ensure that all mandatory fields are populated.
- Implement client-side or server-side validation to check required fields before making the API call.

Example: 404 INVALID_OR_NULL_FOR_RESTRICTED_PICKLIST

```javascript
[
  {
    "message": "Status: bad value for restricted picklist field: Test",
    "errorCode": "INVALID_OR_NULL_FOR_RESTRICTED_PICKLIST",
    "fields": [
      "bpats__Status__c"
    ]
  }
]
```

Possible Causes:

- You're trying to set a picklist field (bpats__Status__c) to a value (Test) that isn't allowed.
- The picklist is **restricted**, meaning only predefined values are accepted.

How to Handle:

- Fetch the allowed picklist values from Salesforce metadata or manually maintain a list.
- Validate picklist inputs before sending them to the server.

- Show users only valid choices in UI dropdowns.

Example: 400 STRING_TOO_LONG

```JavaScript
{
  "hasErrors": true,
  "results": [
    {
      "referenceId": "ref3",
      "errors": [
        {
          "statusCode": "STRING_TOO_LONG",
          "message": "First Name: data value too large: candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate test candidate (max length=40)",
          "fields": [
            "FirstName"
          ]
        }
      ]
    }
  ]
}
```

Possible Causes:

- The value you're trying to insert into the FirstName field exceeds the maximum allowed characters (40 characters in this case).

How to Handle:

- Limit field input length based on Salesforce field metadata (e.g., maxlength in input fields).
- Truncate or validate strings before sending the request.
- Show the user a clear message like "First Name must be 40 characters or fewer."

asymbl.

API Request Limits and Allocations

Salesforce Edition	API Calls Per License Type Per 24-Hour Period	Total Calls Per 24-Hour Period
Developer Edition	N/A	15,000
Enterprise Edition Professional Edition with API access enabled	Salesforce: 1,000 Salesforce Platform: 1,000 Lightning Platform - One App: 200 Customer Community: 0 Customer Community Login: 0 Customer Community Plus: 200 Customer Community Plus Login: 10 External Identity 25,000: 70,000 External Identity 250,000: 750,000 External Identity 1,000,000: 4,000,000 Partner Community: 200 Partner Community Login: 10 Lightning Platform Starter: 200 per member for Enterprise Edition orgs Lightning Platform Plus: 1000 per member for Enterprise Edition orgs	100,000 + (number of licenses x calls per license type) + purchased API Call Add-Ons
Unlimited Edition Performance Edition	Salesforce: 5,000 Salesforce Platform: 5,000 Lightning Platform - One App: 200 Customer Community: 0 Customer Community Login: 0 Customer Community Plus: 200 Customer Community Plus Login: 10 External Identity 25,000: 70,000 External Identity 250,000: 750,000 External Identity 1,000,000: 4,000,000 Partner Community: 200 Partner Community Login: 10 Lightning Platform Starter: 200 per member for Unlimited and Performance Edition orgs Lightning Platform Plus: 5,000 per	100,000 + (number of licenses x calls per license type) + purchased API Call Add-Ons

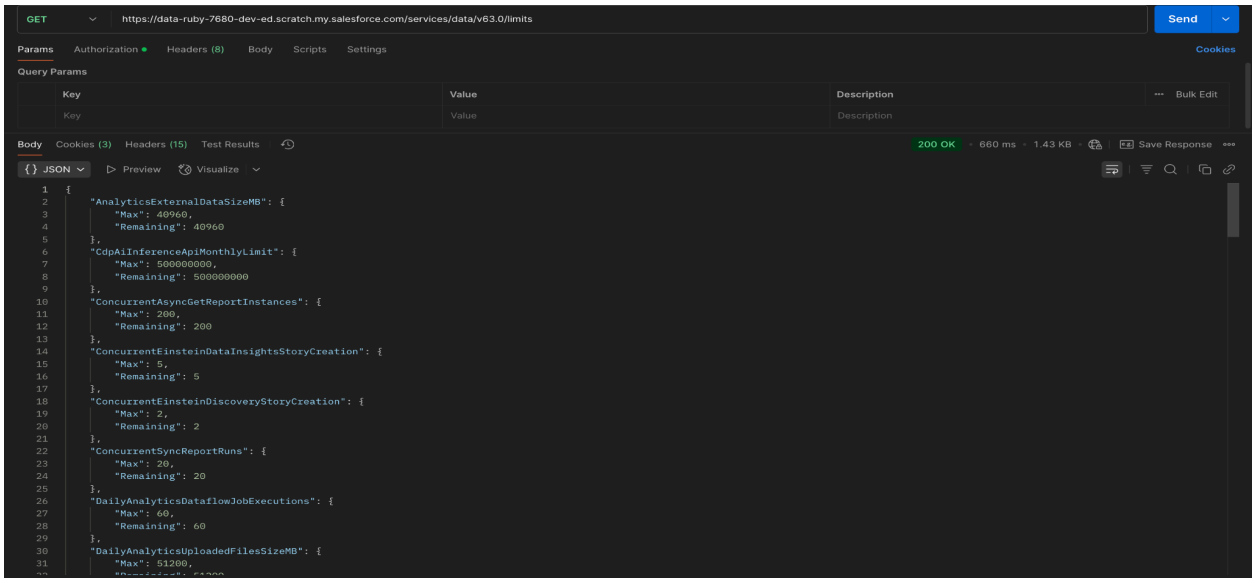
	member for Unlimited and Performance Edition orgs	
Full Sandbox	N/A	5,000,000This limit applies only to Full Sandboxes that aren't created from a template. For any sandbox created from a template, values in the template determine the limits. For more information, visit Salesforce Help: Sandbox Types and Templates.

Getting API Limits

HTTP method: GET

You can use the Salesforce Limits API to programmatically retrieve your current API usage and limits.

- Endpoint: https://<your-instance>.salesforce.com/services/data/vXX.0/limits
- Replace `vXX.X` with your API version.



Permission Set Overview

- **Asymbl ATS Job Portal Permission Set:** This permission set is designed for external-facing portals.

  Object Access & Allowed Operations

Object	Read	Create	Edit	Delete	View All	Modify All
Job Board Application	✅	✅	❌	❌	❌	❌
Job	✅	❌	❌	❌	❌	❌

- **External Interview Feedback Permission Set:** Permission set is designed to submit interview feedback

  Object Access & Allowed Operations

Object	Read	Create	Edit	Delete	View All	Modify All
Interview Feedback	✅	✅	❌	❌	❌	❌
Interview Topic Feedback	✅	✅	❌	❌	❌	❌
Interview Topic	✅	✅	❌	❌	❌	❌
Interview	✅	✅	❌	❌	❌	❌
Interview Template Detail	✅	❌	❌	❌	❌	❌
Interview Template	✅	❌	❌	❌	❌	❌
Contact	✅	❌	❌	❌	❌	❌

- Asymbl ATS User Permission Set: This permission set grants users access to nearly all object permissions within the system, including Read, Create, and Edit capabilities. However, users assigned to this permission set do not have Delete access to any objects.

Object	Read	Create	Edit	Delete	View All	Modify All
ATS Action Link	✅	❌	❌	❌	❌	❌
ATS Action	✅	❌	❌	❌	❌	❌
ATS Applicant	✅	✅	✅	❌	❌	❌
ATS List Filter	✅	✅	✅	❌	❌	❌
ATS Log	✅	✅	✅	❌	❌	❌
ATS Stage	✅	✅	✅	❌	❌	❌
ATS Template Filter	✅	✅	✅	❌	❌	❌
ATS Template Stage	✅	❌	❌	❌	❌	❌
ATS Template	✅	❌	❌	❌	❌	❌
ATS Timeline Child	✅	✅	✅	❌	❌	❌
ATS Timeline Configuration	✅	✅	✅	❌	❌	❌
Account	✅	✅	✅	❌	❌	❌
Branch	✅	✅	✅	❌	❌	❌
Contact	✅	✅	✅	❌	❌	❌
Contact List	✅	✅	✅	❌	❌	❌
Interview Feedback	✅	✅	✅	✅	✅	✅
Interview Template Detail	✅	❌	❌	❌	❌	❌
Interview Template	✅	❌	❌	❌	❌	❌
Interview Topic Feedback	✅	✅	✅	❌	❌	❌
Interview Topic	✅	✅	✅	❌	❌	❌
Interview	✅	✅	✅	❌	❌	❌
Job Board Application	✅	✅	✅	❌	❌	❌
Job	✅	✅	✅	❌	❌	❌
Offer	✅	✅	✅	❌	❌	❌
Placement Credit Template Entry	✅	✅	✅	❌	❌	❌
Placement Credit Template	✅	✅	✅	❌	❌	❌

Placement Credit	✅	✅	✅	❌	❌	❌
Placement	✅	✅	✅	❌	❌	❌
Purchase Order	✅	✅	✅	❌	❌	❌
Retained Milestone	✅	✅	✅	❌	❌	❌
Work Site Location	✅	✅	✅	❌	❌	❌

- Asymbl ATS Admin Permission Set: This permission set provides full administrative access within the Asymbl ATS system. Users with this role have Read, Create, Edit, and Delete permissions across all standard and custom objects.

Object	Read	Create	Edit	Delete	View All	Modify All
ATS Action Link	✅	✅	✅	✅	✅	✅
ATS Action	✅	✅	✅	✅	✅	✅
ATS Applicant	✅	✅	✅	✅	✅	✅
ATS List Filter	✅	✅	✅	✅	✅	✅
ATS Log	✅	✅	✅	✅	✅	✅
ATS Stage	✅	✅	✅	✅	✅	✅
ATS Template Filter	✅	✅	✅	✅	✅	✅
ATS_Template Stage	✅	✅	✅	✅	✅	✅
ATS Template	✅	✅	✅	✅	✅	✅
ATS Timeline Child	✅	✅	✅	✅	✅	✅
ATS Timeline Configuration	✅	✅	✅	✅	✅	✅
Account	✅	✅	✅	✅	✅	✅
Branch	✅	✅	✅	✅	✅	✅
Contact	✅	✅	✅	✅	✅	✅
Contact List	✅	✅	✅	✅	✅	✅
Interview Feedback	✅	✅	✅	✅	✅	✅
Interview Template Detail	✅	✅	✅	✅	✅	✅
Interview Template	✅	✅	✅	✅	✅	✅

Interview Topic Feedback	✅	✅	✅	✅	✅	✅
Interview Topic	✅	✅	✅	✅	✅	✅
Interview	✅	✅	✅	✅	✅	✅
Job Board Application	✅	✅	✅	✅	✅	✅
Job	✅	✅	✅	✅	✅	✅
Offer	✅	✅	✅	✅	✅	✅
Placement Credit Template Entry	✅	✅	✅	✅	✅	✅
Placement Credit Template	✅	✅	✅	✅	✅	✅
Placement Credit	✅	✅	✅	✅	✅	✅
Placement	✅	✅	✅	✅	✅	✅
Purchase Order	✅	✅	✅	✅	✅	✅
Retained Milestone	✅	✅	✅	✅	✅	✅
Work Site Location	✅	✅	✅	✅	✅	✅

1. Authentication

Authentication in REST APIs is the process of verifying the identity of a client making a request. It's essential to ensure that only authorized users or systems can access specific resources. Our ATS is built on Salesforce, which supports secure and scalable integrations via **Connected Apps**. Authentication ensures secure communication and helps enforce access control in RESTful services.

Method: Connected App in Salesforce

To connect an external system:

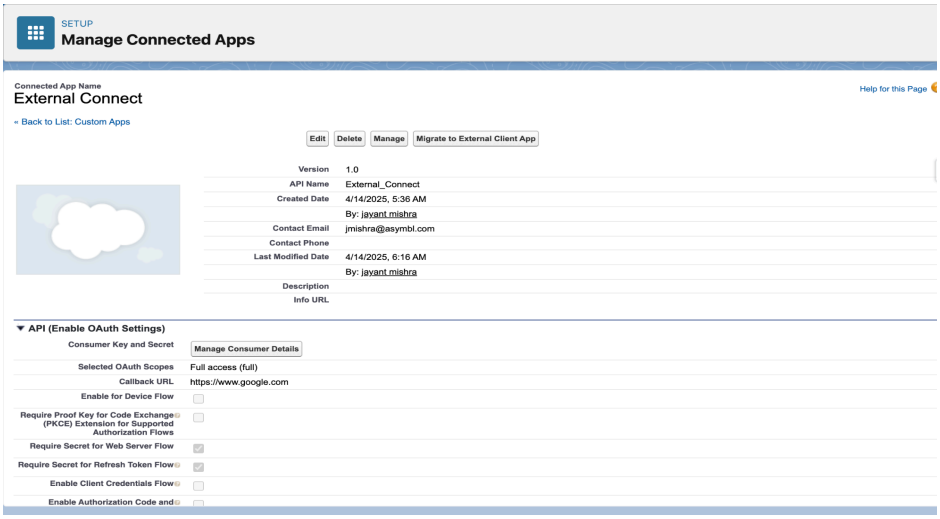
1. **Create a Connected App** in Salesforce:
    - Go to **Setup** > **App Manager** > **New Connected App**.
    - Define the app name and enable **OAuth Settings**.
2. **Generate Client Credentials** (Client ID and Client Secret).

3. Authenticate using Salesforce OAuth 2.0 flow.

4. Use the access token to call Salesforce REST APIs to read/write data.

Security Note: All API interactions are performed under the context of the authenticated user. This means access to data and functionality is governed by that user's assigned permissions.

Tip: Ensure the external system supports OAuth 2.0 and is configured to handle Salesforce-style authentication tokens.



2. Record Operations

External systems can retrieve and manage records using the various supported Salesforce API methods.

Get Records: Salesforce Standard API – GET Records

HTTP method: GET

The GET request method is used to **retrieve data** from a server at a specified resource URL. It is one of the most common HTTP methods and is **read-only**, meaning it does not change or modify any data on the server.

External systems can use the Salesforce REST API to **query job records** using SOQL.

- Endpoint:
  https://<your-instance>.salesforce.com/services/data/vXX.0/query/?q=SELECT+<field-api-name>,+<field-api-name>+FROM+<Object_API_Name>

- Replace <your-instance> with your actual Salesforce domain.
- Replace vXX.0 with your specific version number.
- Replace <Object_API_Name> with the API name of the target Salesforce object.

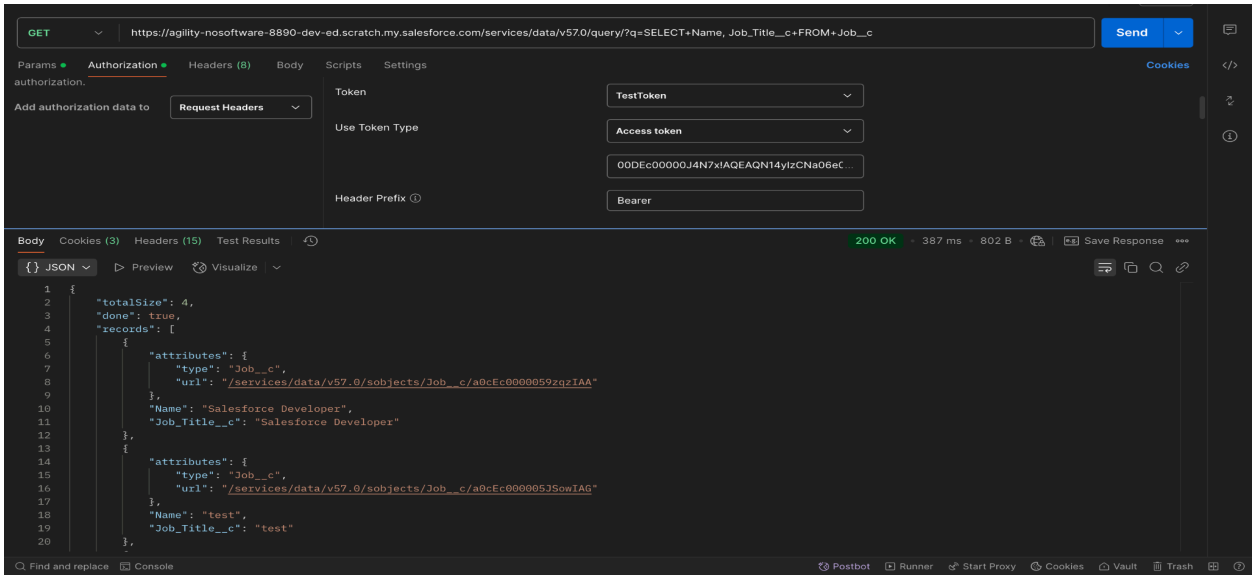
The query string after q= is a URL-encoded SOQL query. Here's how it works:

- SELECT+Name,+bpats__Job_Title__c+FROM+bpats__Job__c is equivalent to the raw SOQL query: SELECT Name, bpats__Job_Title__c FROM bpats__Job__c

Example

- Query for Job Records: SELECT+Name,+bpats__Job_Title__c+FROM+bpats__Job__c
- Query for Job Board Application Records: SELECT+Name,+bpats__Job_c+FROM+bpats__Job_Board_Application__c

Note: The screenshots below depict examples of integrating the ATS with Postman. These are provided for demonstration purposes to show how API requests



Create Records: Salesforce Standard API

HTTP method: POST

The POST method is used to create new records in Salesforce. This method targets a resource collection (i.e., a Salesforce object) and adds a new subordinate resource (i.e., a record) to it. External systems can use this method to create records for any custom or standard Salesforce object via the REST API.

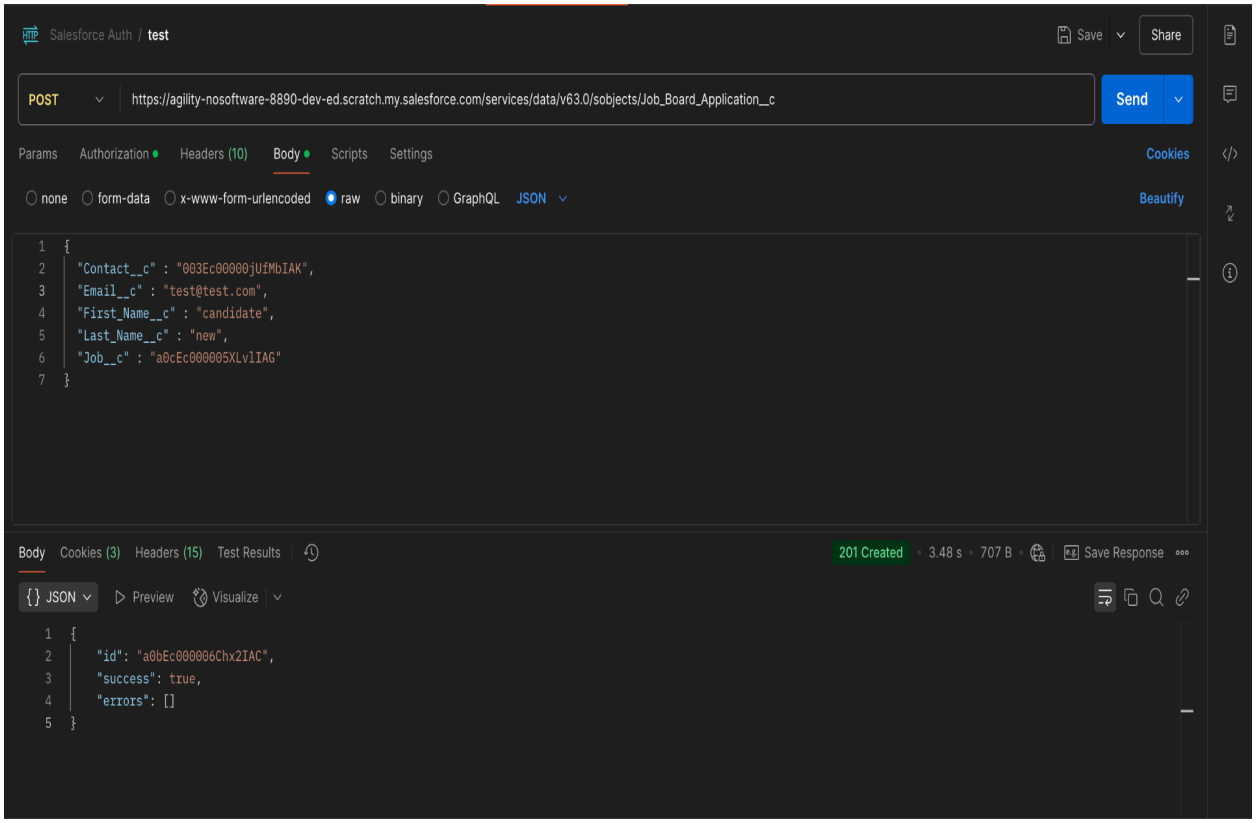
- Endpoint:
- https://<your-instance>.salesforce.com/services/data/vXX.0/sobjects/<Object_API_Name>
- Replace <your-instance> with your actual Salesforce domain.
- Replace vXX.0 with your specific version number.
- Replace <Object_API_Name> with the API name of the target Salesforce object (e.g., Account, Contact, bpats__Job_Board_Application__c, etc.).

Example

- Object API Name: bpats__Job_Board_Application__c
- Required Fields:
    - bpats__Contact__c (Reference to Candidate (e.g., Contact Id))
    - bpats__Job__c (Reference to Job (e.g., Job Id))
    - bpats__Email__c (Candidate Email)
    - bpats__First_Name__c (Candidate First Name)
    - bpats__Last_Name__c (Candidate Last Name)

Authentication: Via Connected App

Note: The screenshots below depict examples of integrating the ATS with Postman. These are provided for demonstration purposes to show how API requests.



Create Records in Bulk Record: Salesforce Standard API

HTTP method: POST

The **Composite Tree API** allows external systems to create **multiple records** in a single API call. This is especially useful for creating a set of related or independent records efficiently.

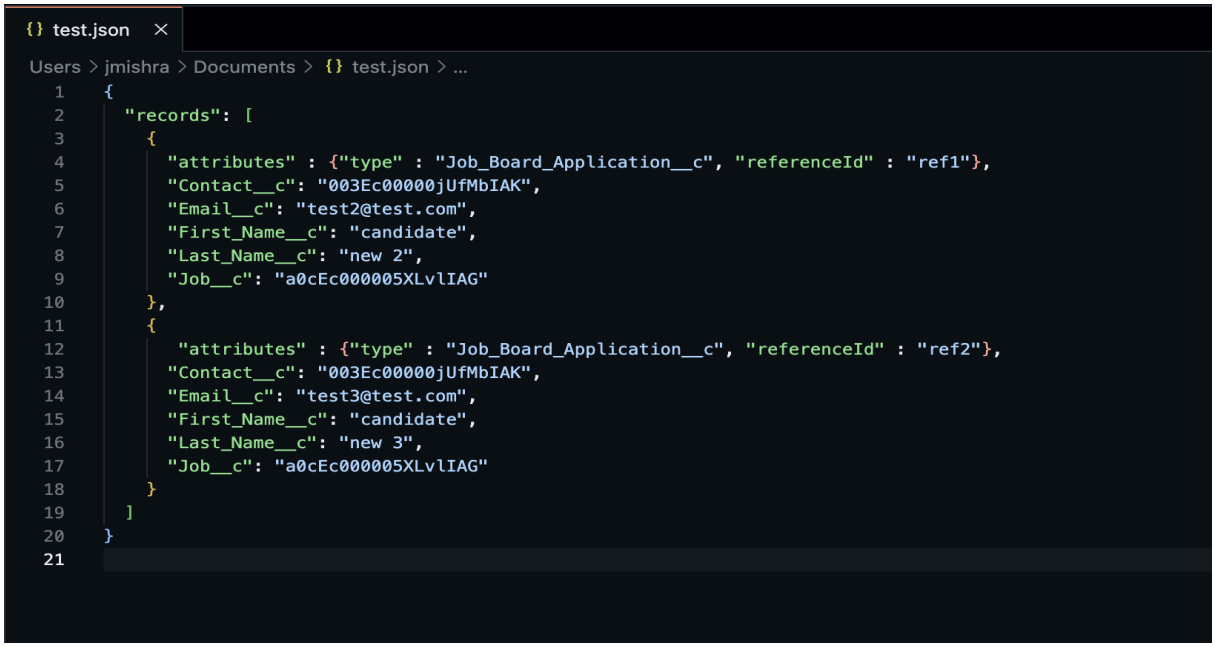
- Endpoint: https://<your-instance>.salesforce.com/services/data/vXX.0/composite/tree/<Object_API_Name>
- Replace vXX.0 with the appropriate API version (e.g., v60.0).
- Replace <Object_API_Name> with the API name of the object you're creating (e.g., Account, Contact, bpats__Job_Board_Application__c, bpats__Job__c, etc.).

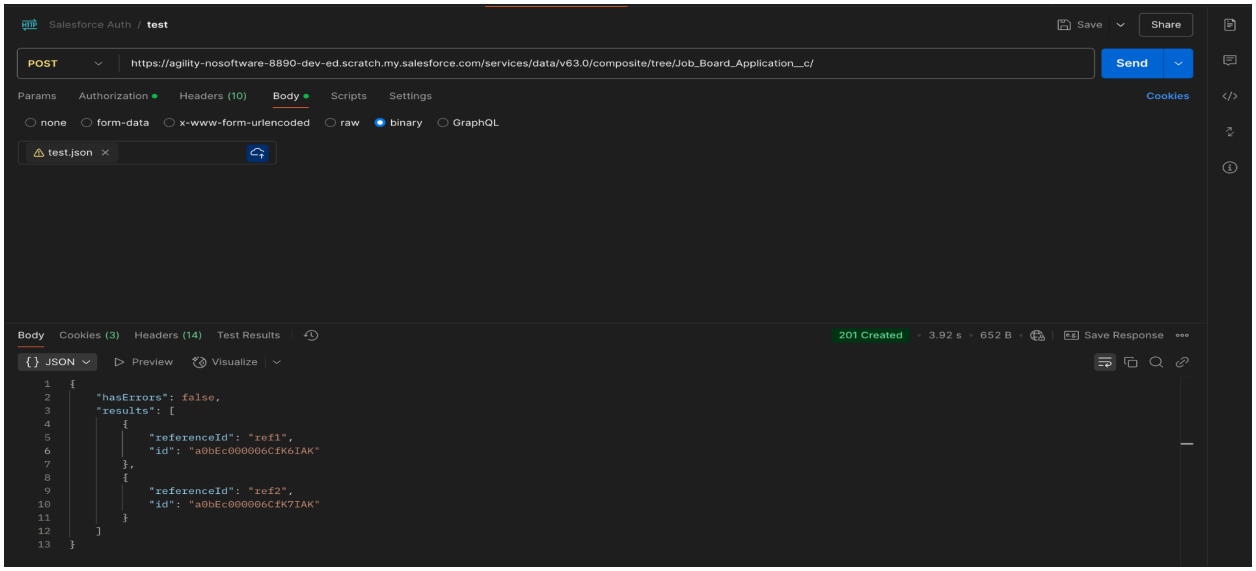
The body must include a records array with JSON objects, each representing a record to be created. You can assign a referenceId to each record for easy identification in the response.

Example

- Object API Name: bpats__Job_Board_Application__c
- Required Fields: You must include all **required fields** for the target object. These are defined in the object schema

Sample JSON Payload:





Update Single Records: Salesforce Standard API

HTTP method: PATCH

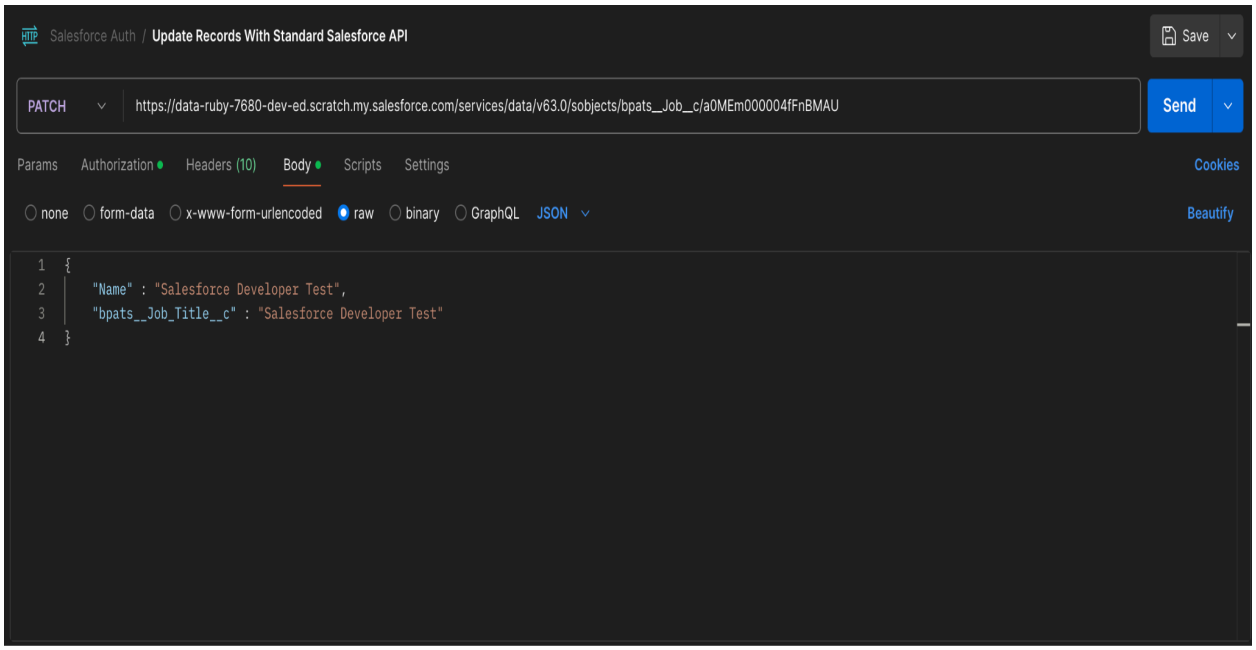
The PATCH request method is used to modify the values of the resource properties. The PATCH method requires a request body. The body of the request must contain representation of the JSON Patch operations that you want to perform on the resource.

External systems can update records using the standard Salesforce object API.

- Endpoint: https://<your-instance>.salesforce.com/services/data/vXX.0/sobjects/<Object_API_Name>/{recordId}
- Replace <your-instance> with your actual Salesforce domain.
- Replace vXX.0 with your specific version number.
- Replace <Object_API_Name> with the API name of the object you're creating.
- Replace {recordId} with the Salesforce ID of the record you want to update.

Example

- Object API Name: bpats__Job_c
- Required Fields: You must include all **required fields** for the target object. These are defined in the object schema

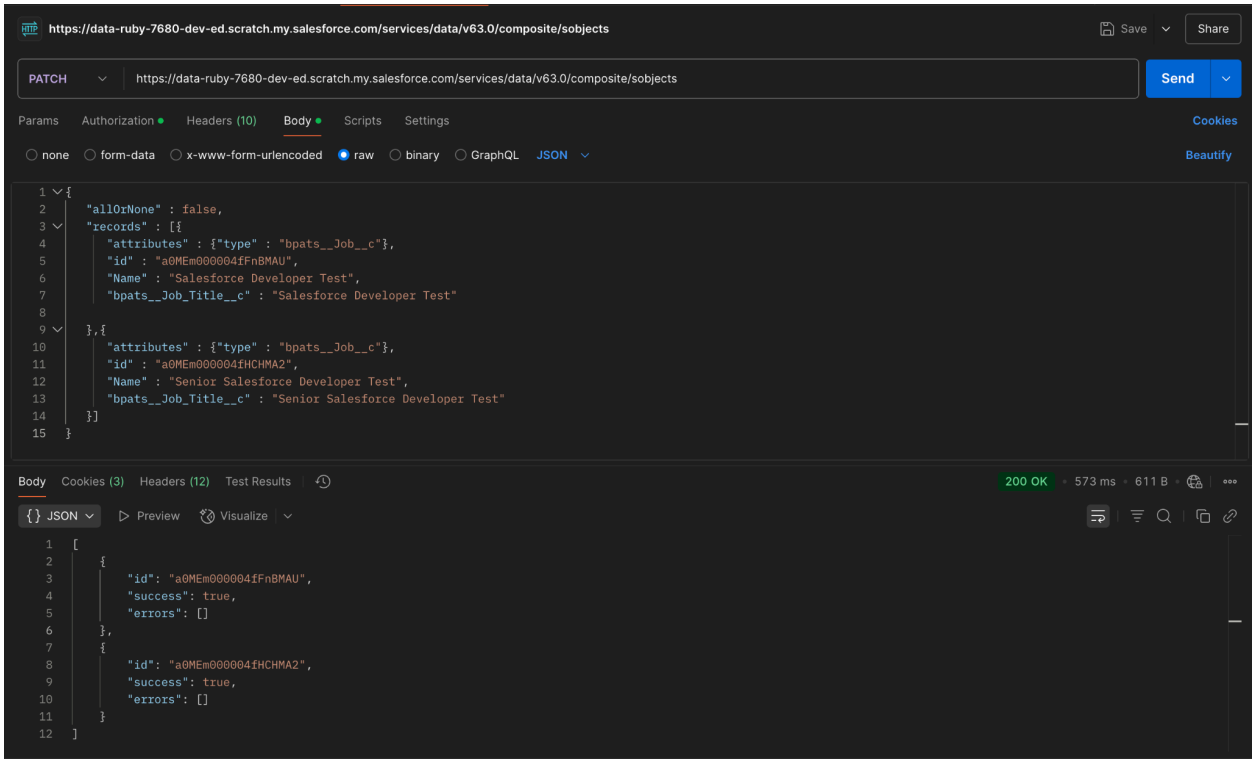


Update Bulk Records: Salesforce Standard API

HTTP method: PATCH

External systems can update records in Bulk using the Standard Salesforce object API.

- Endpoint: https://<your-instance>.salesforce.com/services/data/vXX.0/composite/sobjects
- Replace <your-instance> with your actual Salesforce domain.
- Replace vXX.0 with your specific version number.



Delete Records: Salesforce Standard API

HTTP method: DELETE

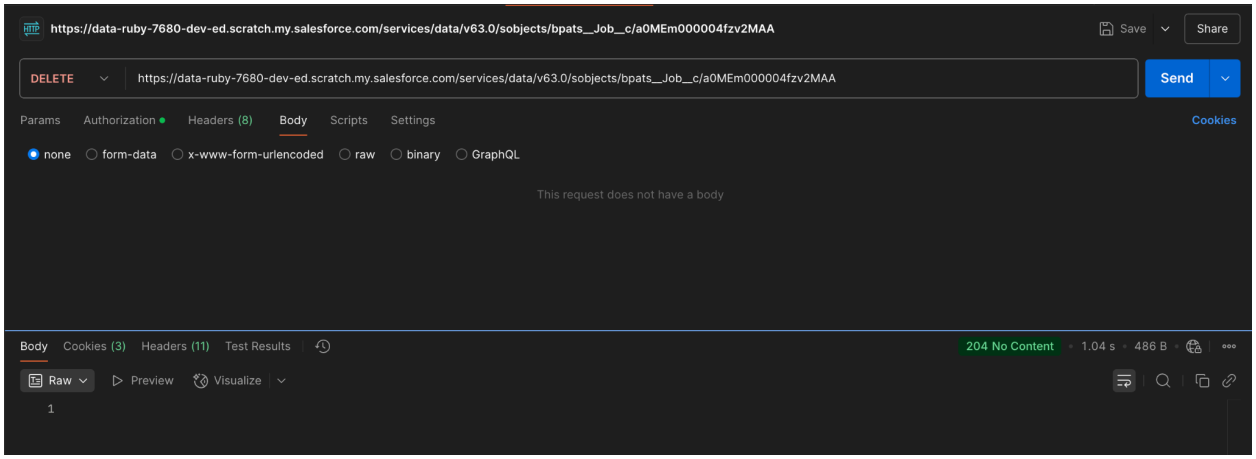
The DELETE request is used to delete a specific resource from your Salesforce instance. External systems can delete a specific record from Salesforce by using the Standard Salesforce object API.

- Endpoint: https://<your-instance>.salesforce.com/services/data/vXX.0/sobjects/<Object_API_Name>/{recordId}
- Replace <your-instance> with your actual Salesforce domain.
- Replace vXX.0 with your specific version number.

- Replace <Object_API_Name> with the API name of the object you're creating
- Replace {recordId} with the Salesforce ID of the record you want to delete.

Example

- Object API Name: bpats__Job_c
- Required Fields: You must include all **required fields** for the target object. These are defined in the object schema.

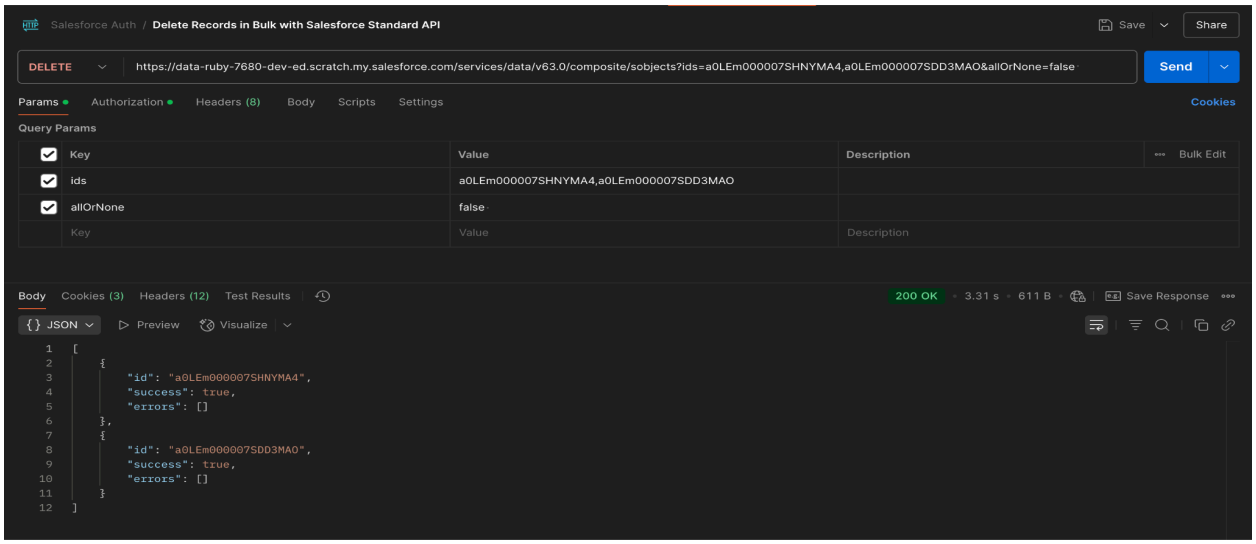


Delete Records in Bulk: Salesforce Standard API

HTTP method: DELETE

The DELETE request can also be used to delete multiple records from your Salesforce instance. External systems can delete records in Bulk from Salesforce by using the Standard Salesforce object API.

- Endpoint: https://<your-instance>.salesforce.com/services/data/vXX.0/composite/sobjects?ids=<comma-separated-record-ids>&allOrNone=false
- Replace <your-instance> with your actual Salesforce domain.
- Replace vXX.0 with your specific version number.


Salesforce Auth / Delete Records in Bulk with Salesforce Standard API
Save
Share
DELETE
https://data-ruby-7680-dev-ed.scratch.my.salesforce.com/services/data/v63.0/composite/sobjects?ids=a0LEm000007SHNYMA4,a0LEm000007SDD3MAO&allOrNone=false
Send
Params
Authorization
Headers (8)
Body
Scripts
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
ids
a0LEm000007SHNYMA4,a0LEm000007SDD3MAO
allOrNone
false
Key
Value
Description
Body
Cookies (3)
Headers (12)
Test Results
200 OK
3.31 s
611 B
Save Response
{ } JSON
Preview
Visualize
[
{
"id": "a0LEm000007SHNYMA4",
"success": true,
"errors": []
},
{
"id": "a0LEm000007SDD3MAO",
"success": true,
"errors": []
}
]