

Composite Resources in Salesforce

May 14, 2020 ypranav

Tired of making multiple REST API calls for a simple task? Then composite resources let you batch up multiple calls in a single call. By using this you can simplify your code, reduce network overhead, and improve your app's performance.

Salesforce provides three composite resources namely composite, batch, tree

composite: Executes a series of REST API requests in a single call. You can use the output of one request as the input to a subsequent request.

composite/batch/: Execute a set of subrequests in a single request. Subrequests are executed independently and information can't be passed between subrequest calls. Up to 25 subrequest in a single request can be executed.

composite/tree/: Creates one or more sObject trees with root records of the specified type. An sObject tree is a collection of nested, parent-child records with a single root record.

Using Of Composite Resource

Use this whenever you need to take an input from one subrequest and use it in other subrequest. Unlike batch resource it will create dependencies between the subrequests. To make a call **/services/data/version/composite/** with a body that contains an array of subrequests. In the each subrequest need to specify the Method(Post,Patch,Get) , Resource Url. For every subrequest , we need to provide a reference Id ,reference Id to pass output from one subrequest to the input of another.

Sample request at below

```
{
  "compositeRequest": [
    {
      "method": "POST",
      "url": "/services/data/v44.0/subjects/Account",
      "referenceId": "newAccount",
      "body": {
        "Name": "Sample Account"
      }
    },
    {
      "method": "POST",
      "url": "/services/data/v44.0/subjects/Contact",
      "referenceId": "newContact",
      "body": {
        "LastName": "New Contact",
        "AccountId": "@{newAccount.id}"
      }
    }
  ]
}
```

Here in the above JSON, you can see how Account ID is used while creating the contact by referring the referenceID.

```
Example 2:Querying the Account and passing to contact.
{
  "compositeRequest": [
    {
      "method": "GET",
      "referenceId": "refAccount",
      "url": "/services/data/v41.0/query?q=Select+Id+from+Account+Limit+1"
    },
    {
      "method": "POST",
      "url": "/services/data/v44.0/subjects/Contact",
      "referenceId": "newContact",
      "body": {
        "LastName": "New Contact",
        "AccountId": "@{refAccount.id}"
      }
    }
  ]
}
```

Points to remember:

- Subrequests can be calls to SObject and Query/QueryAll resources.
- you can have up to 25 subrequests in a single call. Up to 10 of these subrequests can be query operations.

Using of Composite Batch

Use the batch resource when you've got a bunch of independent REST API calls that don't depend on each other . To make a batch resource call, issue a POST to instance **/services/data/API version/composite/batch/** with a request body that contains an array of REST API subrequests.

Subrequests execute serially in their order in the request body. When a subrequest executes successfully, it commits its data. Commits are reflected in the output of later subrequests. If a subrequest fails, commits made by previous subrequests are not rolled back. In the each subrequest need to specify the Method(Post,Patch,Get) , Resource Url. You'll get back a response that contains an array of subrequest results.

Sample Request

```
{
  "batchRequests" : [
    {
      "method" : "PATCH",
      "url" : "v44.0/subjects/account/001xxxxxxxx",
      "richText" : { "Name" : "Update Name" }
    }, {
      "method" : "GET",
      "url" : "v44.0/subjects/account/001XXXXXXXXX?fields=Name,BillingPostalCode"
    }
  ]
}
```

Points to remember:

- You can make up to 25 subrequests in a single batch call.
- While the calls can’t depend on each other, they are called in the order specified in the request data.
- You can’t use different API headers for different subrequests. You can only use API headers that apply to the overall batch call.
- If a particular subrequest takes longer than 10 minutes to execute, the entire batch call times out and subsequent subrequests are not executed.

Using Of Composite Tree

Creates one or more sObject trees with root records of the specified type in a single call. To make a tree resource call, issue a POST to instance **/services/data/API version/composite/tree/SObject Name/** with a request body that contains one or more SObject record trees. A sObject tree is a collection of nested, parent-child records with a single root record.

Sample Request: Creating Multiple Account records.

```
POST Method : /Services/data/v44.0/composite/tree/Account/
{
  "records": [
    {
      "attributes": {
        "type": "Account",
        "referenceId": "ref1"
      },
      "name": "SampleAccount1",
      "numberOfEmployees": "100",
      "industry": "Banking"
    },
    {
      "attributes": {
        "type": "Account",
        "referenceId": "ref2"
      },
      "name": "SampleAccount2",
      "numberOfEmployees": "250",
      "industry": "Banking"
    }
  ]
}
```

And here’s an example request body that uses one SObject tree to create a single Account record along with one child Contact record.

```
POST Method : /Services/data/v44.0/composite/tree/Account/
{
  "records": [
    {
      "attributes": {
        "type": "Account",
        "referenceId": "ref1"
      },
      "name": "SampleAccountWithContacts",
      "phone": "9123458899",
      "Contacts": {
        "records": [
          {
            "attributes": {
              "type": "Contact",
              "referenceId": "ref2"
            },
            "lastname": "Pranav",
            "email": "pranav@sample.com"
          }
        ]
      }
    }
  ]
}
```

Here the response will give you an array of created recordId along with the associate reference Id. Maintain Refrence id as an unique through out the transaction which helps identify the correct record id.

Points to remember:

- The root record of each SObject tree must be same type you specify in the resource call. So, for example if you make a tree resource call using **/services/data/API version/composite/tree/Account/**, all SObject trees you specify in the request body must start with an Account record.
- You can create Up to a total of 200 records across all trees in a single call.
- SObject trees can be a maximum of 5 levels deep, so at most a single “root” record and 4 levels of nested child records.
- Across all your SObject trees in a given request, you can use a maximum of 5 different types of objects