

API Integration Utility

Admin Configuration Guide

Included Components	2
Installation & Setup	3
Integration Mapping	4
Create API Object Mapping configuration records	4
Create API Field Mapping configuration records	6
Field Mapping Examples	9
Record Syncing - Individual Fields	9
Record Syncing - Related Lists	12
External Data List - Related Lists	13
Field Mapping Reference	18
Configure Lightning Pages	19
API Sync	19
External Data List	20
External Data Record	21
Batch Processing	22
Bulk API Sync	22
Scheduled API Sync	23
Event Processing	26
API Sync by Record ID	26
Direct Update Values	26
Outbound Pushes	27
Special Handling	28
Custom HTTP Headers	28
Custom HTTP Request Transforms	29
Custom HTTP Response Transforms	30
Custom DML Pre-Processing	31
Usage Considerations	32
Assigning Licenses	32
User Permissions	32

Purchasing the App

33

Non-Profits

33

Included Components

API Integration Utility includes the following exposed global components available to perform various integration use cases.

Component Name	Component Type	Component Description
API Object Mapping	Custom Metadata	Object level configuration records to drive API integrations
API Field Mapping	Custom Metadata	Field level configuration records to drive API integrations
API Sync	Lightning Component	Send or receive data from API upon page load of a Salesforce record and refresh the record with any data changes
External Data List	Lightning Component	Display list of records from an API response upon page load with available search filters and pagination
External Data Record	Lightning Component	Display individual record fields from an API response upon page load with ability to display special field types such as images.
API Sync Batch	Apex Class	Schedulable Apex class available to automate configured API operations on a recurring basis and sync lists of records
API Sync Invocable	Apex Class	Invocable Apex action available to automate configured API operations using custom Apex, Process Builder, or Flow

Installation & Setup

1. Install the latest API Integration Utility package from the AppExchange

- Navigate to: <https://appexchange.salesforce.com/appxListingDetail?listingId=a0N3A00000G0x9tUAB>
- Click Get It Now and Install. Leave default selection to Install for Admin Users.

2. Create Named Credential for endpoint and authentication

- Navigate to Named Credentials in the Setup menu
- Create a credential record with the base endpoint for the API you need to integrate with. Currently this package only supports authentication mechanisms supported by Named Credentials (e.g. basic, oauth, JWT, aws sig, etc.), however future releases may introduce support for custom authentication schemes.


SETUP
Named Credentials

Named Credential Edit: ZenPortal

Help for this Page ?

Specify the callout endpoint's URL and the authentication settings that are required for Salesforce to make callouts to the remote system.

Save

Cancel

Label ⓘ

ZenPortal

Name ⓘ

ZenPortal

URL

<https://your.api.com>

▼ Authentication

Certificate

Identity Type

Anonymous ▼

Authentication Protocol

No Authentication ▼

▼ Callout Options

Generate Authorization Header ⓘ

☐

Allow Merge Fields in HTTP Header ⓘ

☐

Allow Merge Fields in HTTP Body ⓘ

☐

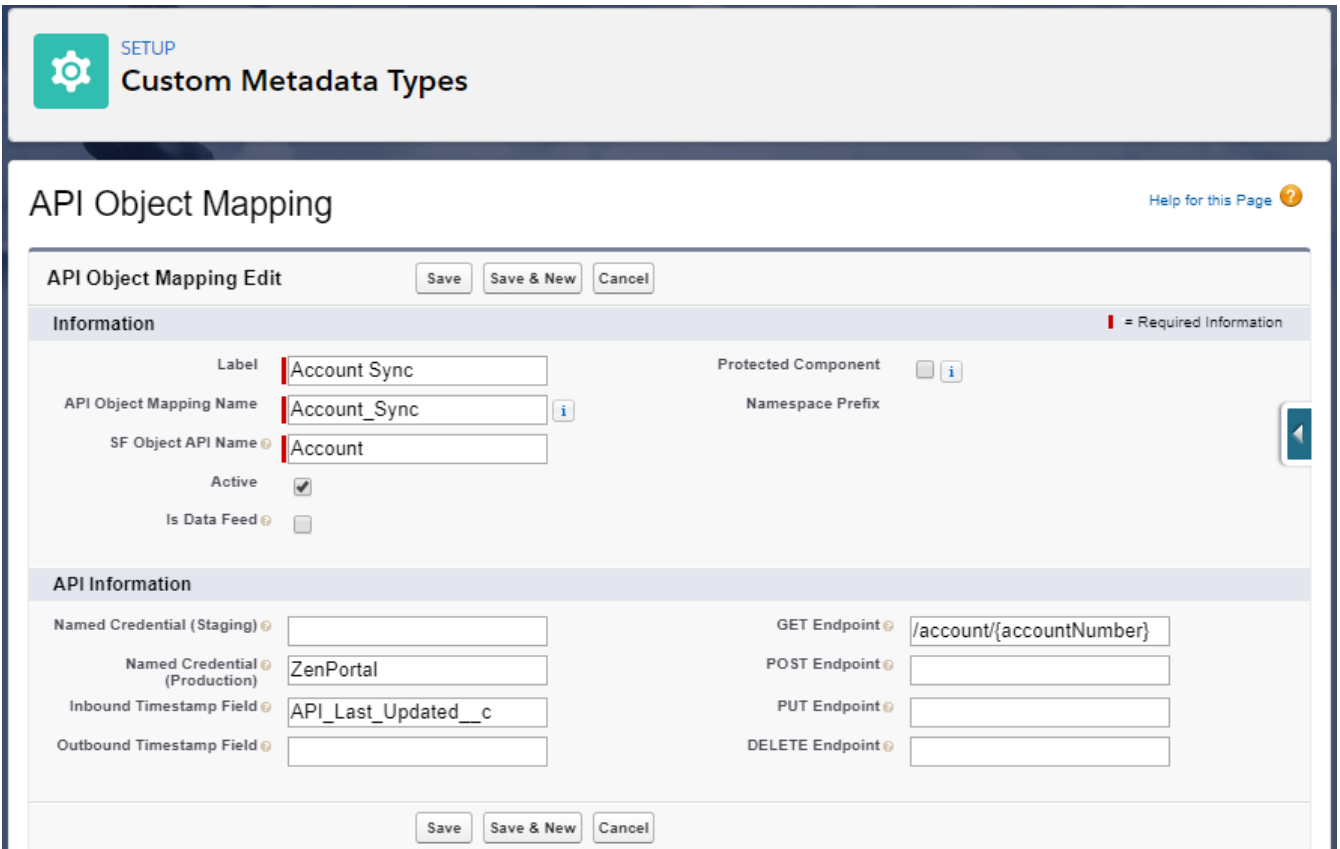
Save

Cancel

Integration Mapping

1. Create API Object Mapping configuration records

- Navigate to Custom Metadata Types in the Setup menu
- Click Manage next to API Object Mapping
- Click “New”
- Define a unique name and populate the Salesforce Object that will be driving the integration.



The screenshot shows the Salesforce Setup interface for Custom Metadata Types. The page title is "Custom Metadata Types" with a "SETUP" link. Below the title is the "API Object Mapping" section. The form is titled "API Object Mapping Edit" and includes "Save", "Save & New", and "Cancel" buttons. The form is divided into two main sections: "Information" and "API Information".

Information Section:

- Label:** Account Sync
- API Object Mapping Name:** Account_Sync (marked as required with a red exclamation mark and an information icon)
- SF Object API Name:** Account (marked as required with a red exclamation mark and an information icon)
- Active:** ☒
- Is Data Feed:** ☐
- Protected Component:** ☐ (with an information icon)
- Namespace Prefix:** (empty field)

API Information Section:

- Named Credential (Staging):** (empty field)
- Named Credential (Production):** ZenPortal
- Inbound Timestamp Field:** API_Last_Updated__c
- Outbound Timestamp Field:** (empty field)
- GET Endpoint:** /account/{accountNumber}
- POST Endpoint:** (empty field)
- PUT Endpoint:** (empty field)
- DELETE Endpoint:** (empty field)

At the bottom of the form are "Save", "Save & New", and "Cancel" buttons. A "Help for this Page" link is visible in the top right corner.

Field	Description
SF Object API Name	Name of the Salesforce standard or custom object that will initiate an API callout.
Active	When checked, this integration can be used. Deselect to quickly deactivate an integration.
Debug Mode	When checked, additional API Interaction Log records are created as type “Info” upon each API Sync invocation to assist in troubleshooting field mapping.
Header Interceptor Class	(Special handling only) Optional name of custom Apex class to specify custom HTTP headers for each API callout. This is particularly used for custom authentication headers unsupported by Named Credentials. If desired you can inject username or password values from the Named

	Credential by using Salesforce supported merge values like {!\$Credential.Username} or {!\$Credential.Password}. Must implement the apiutil.ApiHeaderInterceptor interface.
DML Interceptor Class	(Special handling only) Optional name of custom Apex class to customize record processing after API callout and before final DML transaction. Must implement the apiutil.ApiDatabaseInterceptor interface.
Is Data Feed?	Select this if there will be no data saved to Salesforce from API responses. This is used for data display purposes and works in conjunction with the provided External Data Table component. It also is used for dynamic outbound requests with the API Sync Invocable apex action for Visual Flows (data generated from screen input or calculated without record data).
Named Credential (Staging)	Developer name of the Named Credential record to be used for base API endpoint and authorization. The API Utility will always reference the Staging credential in Sandbox type environments.
Named Credential (Production)	Developer name of the Named Credential record to be used for base API endpoint and authorization. The API Utility will always reference the Production credential in Production type environments.
Inbound Timestamp Field	Populate with a Datetime type field API Name that exists on the object set as the SF Object API Name. This field will get updated on records whenever data changes are detected and applied from an API response.
Outbound Timestamp Field	Populate with a Datetime type field API Name that exists on the object set as the SF Object API Name. This field will get updated on records whenever data is sent outbound in an API request (POST or PUT).
GET Endpoint	Relative endpoint (must begin with a forward slash "/") of the API resource to integrate with for HTTP GET requests. This will typically be the main resource used for getting information from an API about a record. The relative endpoint may contain URI parameters enclosed in curly braces. Any URI parameter must also be configured as a related API Field Mapping (see next section).
POST Endpoint	Relative endpoint (must begin with a forward slash "/") of the API resource to integrate with for HTTP POST requests. This will typically be the resource used for pushing new information to an API about a record. The relative endpoint may contain URI parameters enclosed in curly braces. Any URI parameter must also be configured as a related API Field Mapping (see next section).
PUT Endpoint	Relative endpoint (must begin with a forward slash "/") of the API resource to integrate with for HTTP PUT requests. This will typically be the resource used for pushing updated information to an API about a record. The relative endpoint may contain URI parameters enclosed in curly braces. Any URI parameter must also be configured as a related API Field Mapping (see next section).
DELETE Endpoint	Relative endpoint (must begin with a forward slash "/") of the API resource to integrate with for HTTP POST requests. This will be a resource used for deleting information from an API about a record. The relative endpoint may contain URI

	parameters enclosed in curly braces. Any URI parameter must also be configured as a related API Field Mapping (see next section).
--	---

2. Create API Field Mapping configuration records

- Navigate to Custom Metadata Types in the Setup menu
- Click Manage next to API Field Mapping
- Click "New"
- Define a unique name and populate the Object Configuration that will be driving the integration.
- Define field mapping rules and properties.

Field	Description
Object Configuration	Name of the API Object Mapping that will initiate an API callout.
Active	When checked this field mapping will be used in the integration. Deselect to quickly deactivate a field in an integration.
Is External ID	Select this for at most one field mapping record per Salesforce object type. Map an External ID type field that can be used for inserting or updating an object type. It is not required (but is recommended) to have an External ID field on a parent record mapping. It is required to have this set and mapped to External ID on any child objects if mapping related list record fields.
API Field Name	Populate with the name of the attribute you are mapping to in the HTTP request URL or body. For simple JSON payloads this will usually be a one-word string of text. For JSON payloads that contain inner nested objects or lists, the "." operator is used to separate layers of attributes.
SF API Name	Populate the API Name of the Salesforce Field to map values from/to. For lookup/master-detail mappings, specify the relation name and API Name of External ID field on the related object separated by "." operator. For example, a lookup field name of Preferred_Product__c looks up to a Product object type which has an External ID field External_ID__c. The field mapping would be Preferred_Product__r.External_ID__c. For related list mappings, use a similar format but instead use the Child Relationship Name listed on the child's Lookup/master-detail field definition.
Mapping Type	Define the field mapping to be used in HTTP Request or HTTP Response.
Merge Type	Define the mapping rule to apply for this field. Flat Value will be used in most cases unless mapping a related record, related list, or need to transform the value's text.
Is HTTP Header	Select to merge the API Field Name and value as an outbound HTTP request header.
Is Query Parameter	Select to merge the API Field Name and value appended to the API Object Mapping defined relative endpoints.

Is URI Parameter	Select to merge this value into the API Object Mapping defined relative endpoints containing the API Field Name value in curly brackets (See step 1). If used, the API Object Mapping endpoint value must include the value set in API Field Name in curly brackets. (e.g. <code>"/account/{accountNumber}"</code> with value of <code>"accountNumber"</code> in API Field Name)
GET	Select to use this mapping in HTTP GET Requests.
POST	Select to use this mapping in HTTP POST Requests (values parsed in/out of the request body when not marked as query or URI parameter).
PUT	Select to use this mapping in HTTP PUT Requests (values parsed in/out of the request body when not marked as query or URI parameter).
DELETE	Select to use this mapping in HTTP DELETE Requests.
Include In Datatable	Select to display the mapped value returned from the API as a column in a data table.
Is Dynamic Value	Select if the API Field Name is populated by something other than a Salesforce record (e.g. when defining a Search Filter field or Pagination field).
Datatable Index	Populate with a number for ordering columns displayed in the External Data Table component. This only applies when populated along the "Include In Datatable" checkbox. In addition, this can be populated to perform ordered insert/update DML for child objects mapped for API Sync. This only applies when populated along with "Is External Id" and when Merge Type equals "Related Records (List)".
Datatable Initial Width	Optional initial width in pixels loaded for field column in table.
Datatable Pagination	Map a pagination property to available API attributes. For pagination to work, the external API must support server side pagination via page size and page number as query or uri parameters.
Datatable Search Filter	Select this along with enabling "Is Dynamic Value" to display in the Search Filters section of the component. The Apply Filters button in the component will send HTTP requests based on configured mappings and reload the table with response results.
Default Value	Populate on a field marked as Datatable Search Filter and set default value for a Search Filter. This field supports time merge formula format <code>{!NOW}</code> . Optionally apply <code>+</code> / <code>-</code> operators and <code>Day</code> , <code>Hour</code> , <code>Minute</code> time units (e.g. <code>{!NOW-1H}</code>)
Display Type	For External Data Table component row fields, set a specific display data type (API Utility detects most types by default). This also includes special handling actions for Button to invoke nested record callouts, or download File documents.
Related Object Name	Populate with Object API name of related object when mapping a Related Record or Related List.

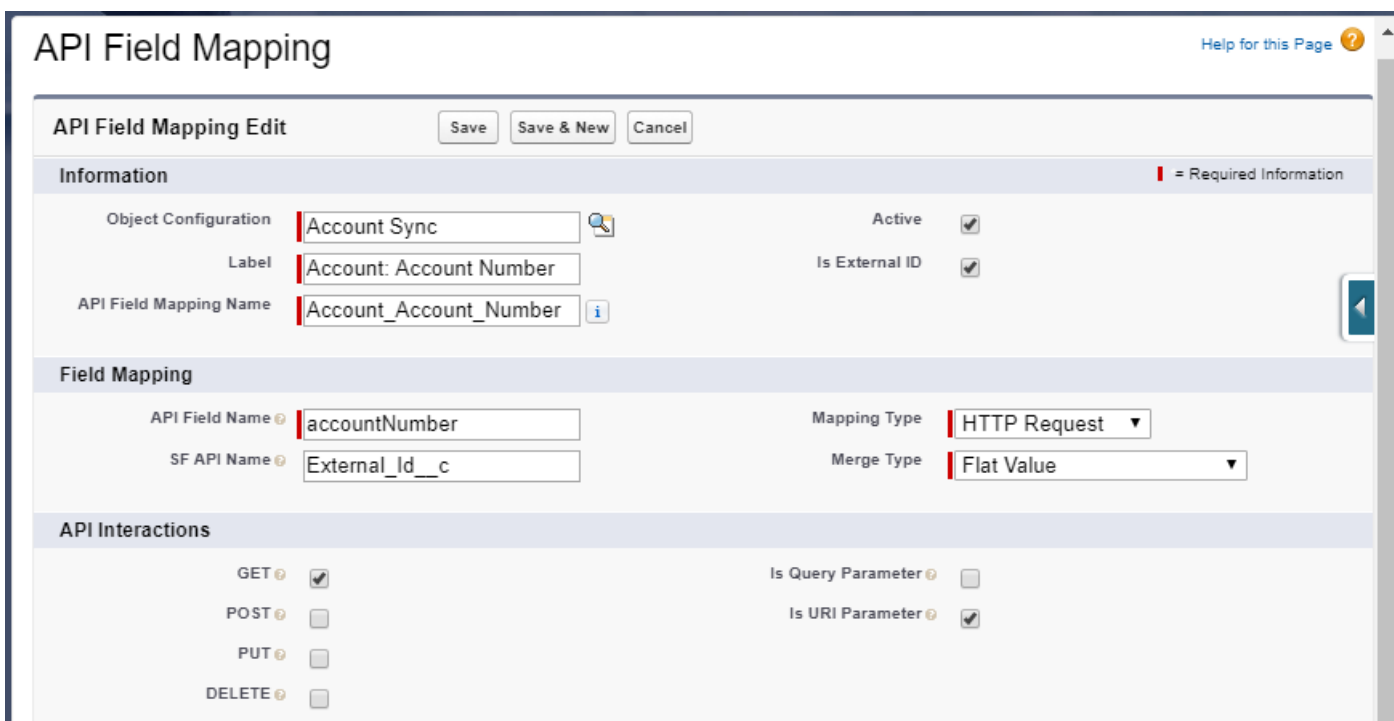
Junction Related Object Name	Special Handling, populate related Object API Name on field mapping of child record when Is External ID is selected.
Junction Parent Field Name	Special Handling, populate related Object's parent field API Name on field mapping of child record when Is External ID is selected.
Junction External ID Field Name	Special Handling: populate with External ID field on child object when mapping Junction objects vs. a standard related list.

Field Mapping Examples

Record Syncing - Individual Fields

1. Request parameters for configured endpoint

Create a field mapping record with a Mapping Type of HTTP Request for each parameter required to make the request. For the API Sync component, a GET request is required and typically needs a Query or URI parameter passed to the API with the identifier for the record to retrieve. The example below is a mapping for sending a custom field value as a URI parameter to the GET Endpoint on the Account Sync configuration. For URI parameters, the API Field Name value must match what is mapped on the GET Endpoint of the API Object Configuration. If set as a Query Parameter, the API Field Name will be passed in the request as the query parameter name. The Is External ID checkbox is optional but can only be selected on one mapping that represents an external id field on the Salesforce object.



The screenshot shows the 'API Field Mapping' configuration page. It includes a title bar with 'API Field Mapping Edit', 'Save', 'Save & New', and 'Cancel' buttons. The main content is divided into three sections: 'Information', 'Field Mapping', and 'API Interactions'. In the 'Information' section, 'Object Configuration' is set to 'Account Sync', 'Label' is 'Account: Account Number', and 'API Field Mapping Name' is 'Account_Account_Number'. 'Active' and 'Is External ID' are both checked. The 'Field Mapping' section shows 'API Field Name' as 'accountNumber' and 'SF API Name' as 'External_Id__c'. 'Mapping Type' is set to 'HTTP Request' and 'Merge Type' is 'Flat Value'. The 'API Interactions' section shows 'GET' as the selected method, with 'Is Query Parameter' unchecked and 'Is URI Parameter' checked. Other methods (POST, PUT, DELETE) are unchecked.

2. Simple field response mapping (Flat Value)

An HTTP Response mapping can be created for each field (i.e. JSON key) that is returned in the external API response. Each response mapping must include the desired field attribute in the API Field Name. For syncing to Salesforce records, populate the SF API Name of the field on the Salesforce object being mapped (set on the API Object Configuration).

API Field Mapping Help for this Page ?

API Field Mapping Edit
Save Save & New Cancel

Information = Required Information

Object Configuration 
Active ☒

Label
Is External ID ☐

API Field Mapping Name 

Field Mapping

API Field Name
Mapping Type

SF API Name
Merge Type

API Interactions

GET ☒
Is Query Parameter ☐

POST ☐
Is URI Parameter ☐

PUT ☐

DELETE ☐

3. Nested JSON response mapping

For nested JSON field attributes, a dot notation can be used to reference inner values.

API Field Mapping Help for this Page ?

API Field Mapping Edit
Save Save & New Cancel

Information = Required Information

Object Configuration 
Active ☒

Label
Is External ID ☐

API Field Mapping Name 

Field Mapping

API Field Name
Mapping Type

SF API Name
Merge Type

API Interactions

GET ☒
Is Query Parameter ☐

POST ☐
Is URI Parameter ☐

PUT ☐

DELETE ☐

4. Related record lookup or master-detail mapping

Note: Master-Detail updates only work if the master-detail field allows reparenting of child objects.

API Field Mapping

API Field Mapping Detail

[Edit](#) [Delete](#) [Clone](#)

Object Configuration [Account Sync](#)

Label Account: Preferred Product

API Field Mapping Name Account_Preferred_Product

▼ Field Mapping

API Field Name [fav_product_id](#)

Mapping Type HTTP Response

SF API Name [Preferred_Product__r.External_ID__c](#)

Merge Type Related Record (Object)

▼ API Interactions

GET ☒

Is Query Parameter ☐

POST ☐

Is URI Parameter ☐

PUT ☐

DELETE ☐

▼ External Data Table

Include In Datatable ☐

Is Dynamic Value ☐

▼ Special Handling

Related Object Name [Product2](#)

Lookup/master-detail field API Name (replace __c with __r) + API Name of external id field on the related object

Related Object Name must be populated with the API Name of the lookup / master object

Record Syncing - Related Lists

5. Related list record mapping (External id)

API Field Mapping

[Help for this Page](#)

API Field Mapping Detail

Edit
Delete
Clone

Object Configuration	<u>Account Sync</u>		Active	☒
Label	Account: Contacts: Id		Is External ID	☒
API Field Mapping Name	Account_Contacts_Id			

Field Mapping

API Field Name	contacts.id	Mapping Type	HTTP Response
SF API Name	Contacts.External_ID__c	Merge Type	Related Records (List)

API Interactions

GET	☒	Parameter	
POST	☐	Parameter	
PUT	☐		
DELETE	☐		

External Data Table

Include In Datatable	☐	Datatable Index	
Is Dynamic Value	☐	Datatable Initial Width	

Special Handling

Related Object Name	Contact
Junction Related Object Name	Account
Junction Parent Field Name	AccountId

Related list records must have a unique external id field

Child relationship name

Parent object name

Parent field name

6. Related list record mapping (Fields)

API Field Mapping

[Help for this Page](#)

API Field Mapping Detail

[Edit](#) [Delete](#) [Clone](#)

Object Configuration	<u>Account Sync</u>	Active	☒
Label	Account: Contacts: Last Name	Is External ID	☐
API Field Mapping Name	Account_Contacts_Last_Name		

▼ Field Mapping

API Field Name	contacts.lastName	Mapping Type	HTTP Response
SF API Name	Contacts.LastName	Merge Type	Related Records (List)

▼ API Interactions

GET	☒	Is Query Parameter	☐
POST	☐	Is URI Parameter	☐
PUT	☐		
DELETE	☐		

▼ External Data Table

Include In Datable	☐	Datable Index	
Is Dynamic Value	☐	Datable Initial Width	
		Datable Search Filter	

▼ Special Handling

Related Object Name	Contact
---------------------	---------

External Data List - Related Lists

Paginated lists of data can be visualized in the External Data List component via the following configuration.

Create an API Object Mapping configuration with **Is Data Feed** selected for data visualization. Create any HTTP Request API Field Mappings to fulfill the API request.

API Object Mapping

[Help for this Page](#)
[API Field Mappings \[23\]](#)

API Object Mapping Detail

[Edit](#) [Delete](#) [Clone](#)

Label	Account Transactions	Protected Component	☐
API Object Mapping Name	Account_Transactions	Namespace Prefix	apiutil
SF Object API Name	Account	Debug Mode	☐
Active	☒		
Is Data Feed	☒		
Is Bulk Sync	☐		

▼ API Information

Named Credential (Staging)	Zen_Test	GET Endpoint	/account/{accountNumber}/transactions
Named Credential (Production)	Zen_Test	POST Endpoint	
Inbound Timestamp Field		PUT Endpoint	
Outbound Timestamp Field		DELETE Endpoint	

Example JSON response of list data:

```
{
  "hasNextPage": true,
  "transactions": [
    {
      "amount": -208.51,
      "description": "Hair of the Dog Brewery",
      "time": "2019-08-27T17:25:15.360Z",
      "transaction_id": "11111111-1111-1111-1111-111111164194",
      "account_id": 1
    }
  ]
}
```

Create API Field Mapping for the list of data to display in the data table. If the API response is already a JSON list and not an object with the list, this field mapping should not be created.

API Field Mapping



API Field Mapping Detail
Edit Delete Clone

Object Configuration	<u>Account Transactions</u>	Active	☒
Label	Transaction List	Is External ID	☐
API Field Mapping Name	Transaction_List		

▼ Field Mapping

API Field Name ⓘ	transactions	Mapping Type	HTTP Response
SF API Name ⓘ		Merge Type	Related Records (List)

If the API response provides a true/false value on if there are more records through a paginated call, map the field the following with the **Datatable Pagination** property set:

API Field Mapping Detail

[Edit](#) [Delete](#) [Clone](#)

Object Configuration	<u>Account Transactions</u>	Active	☒
Label	Transaction Next Page	Is External ID	☐
API Field Mapping Name	Transaction_NextPageAvail		

▼ Field Mapping

API Field Name	hasNextPage	Mapping Type	HTTP Response
SF API Name		Merge Type	Flat Value

▼ API Interactions

GET	☐	Is Query Parameter	☐
POST	☐	Is URI Parameter	☐
PUT	☐	Is HTTP Header	☐
DELETE	☐		

▼ External Data Table

Include In Datatable	☐	Datatable Index	
Is Dynamic Value	☐	Datatable Initial Width	
		Datatable Search Filter	
		Datatable Pagination	Has Next Page

Create API Field Mapping for sending subsequent paginated calls with a Page Number

API Field Mapping Detail

[Edit](#) [Delete](#) [Clone](#)

Object Configuration	<u>Account Transactions</u>	Active	☒
Label	Transaction Page Number	Is External ID	☐
API Field Mapping Name	Transaction_PageNumber		

▼ Field Mapping

API Field Name	pageNumber	Mapping Type	HTTP Request
SF API Name		Merge Type	Flat Value

▼ API Interactions

GET	☒	Is Query Parameter	☒
POST	☐	Is URI Parameter	☐
PUT	☐	Is HTTP Header	☐
DELETE	☐		

▼ External Data Table

Include In Datatable	☐	Datatable Index	
Is Dynamic Value	☒	Datatable Initial Width	
		Datatable Search Filter	
		Datatable Pagination	Page Number

Next, map each table column field to be displayed for the list of records using the Datatable Index for column ordering:

API Field Mapping

API Field Mapping Detail
Edit
Delete
Clone

Object Configuration	<u>Account Transactions</u>	Active	☒
Label	Amount	Is External ID	☐
API Field Mapping Name	Transaction_Amount		

▼ Field Mapping

API Field Name ⓘ	amount	Mapping Type	HTTP Response
SF API Name ⓘ		Merge Type	Flat Value

▼ API Interactions

GET ⓘ	☒	Is Query Parameter ⓘ	☐
POST ⓘ	☐	Is URI Parameter ⓘ	☐
PUT ⓘ	☐	Is HTTP Header ⓘ	☐
DELETE ⓘ	☐		

▼ External Data Table

Include In Datatable ⓘ	☒	Datatable Index ⓘ	4
Is Dynamic Value ⓘ	☐	Datatable Initial Width ⓘ	
		Datatable Search Filter ⓘ	
		Datatable Pagination ⓘ	

Example view:

Transactions

> Search Filters

View	Date	Description	Transaction Id	Amount
	Aug 27, 2019	Hair of the Dog Brewery	11111111-1111-1111-1111-111111164194	-\$208.51
	Aug 26, 2019	Deschutes Brewery	11111111-1111-1111-1111-1111111514124	-\$94.86
	Aug 19, 2019	Baerlic Brewing Company	11111111-1111-1111-1111-1111111572728	-\$171.64
	Aug 19, 2019	Serra Dispensary Downtown	11111111-1111-1111-1111-1111111397492	-\$257.37
	Aug 17, 2019	Subway	11111111-1111-1111-1111-1111111328563	-\$400.39
	Aug 17, 2019	Raley's	11111111-1111-1111-1111-1111111618919	-\$139.71
	Aug 15, 2019	Hair of the Dog Brewery	11111111-1111-1111-1111-1111111948386	-\$482.50
	Aug 14, 2019	Serra Dispensary		

Page 1

⬆ First

< Previous

Next >

Field Mapping Reference

Below are example API Field Mapping configurations and API Field Name notation to generate or process a particular JSON structure in an HTTP Request or HTTP Response.

Expected JSON	API Field Name
<pre>{ "name": "Acme Inc." }</pre>	name
<pre>{ "address": { "street": "123 Test Dr.", "city": "Dallas" } }</pre>	address.city
<pre>{ "phones": [{ "type": "Mobile", "value": "111-222-3333" }] }</pre>	phones[type=Mobile].value

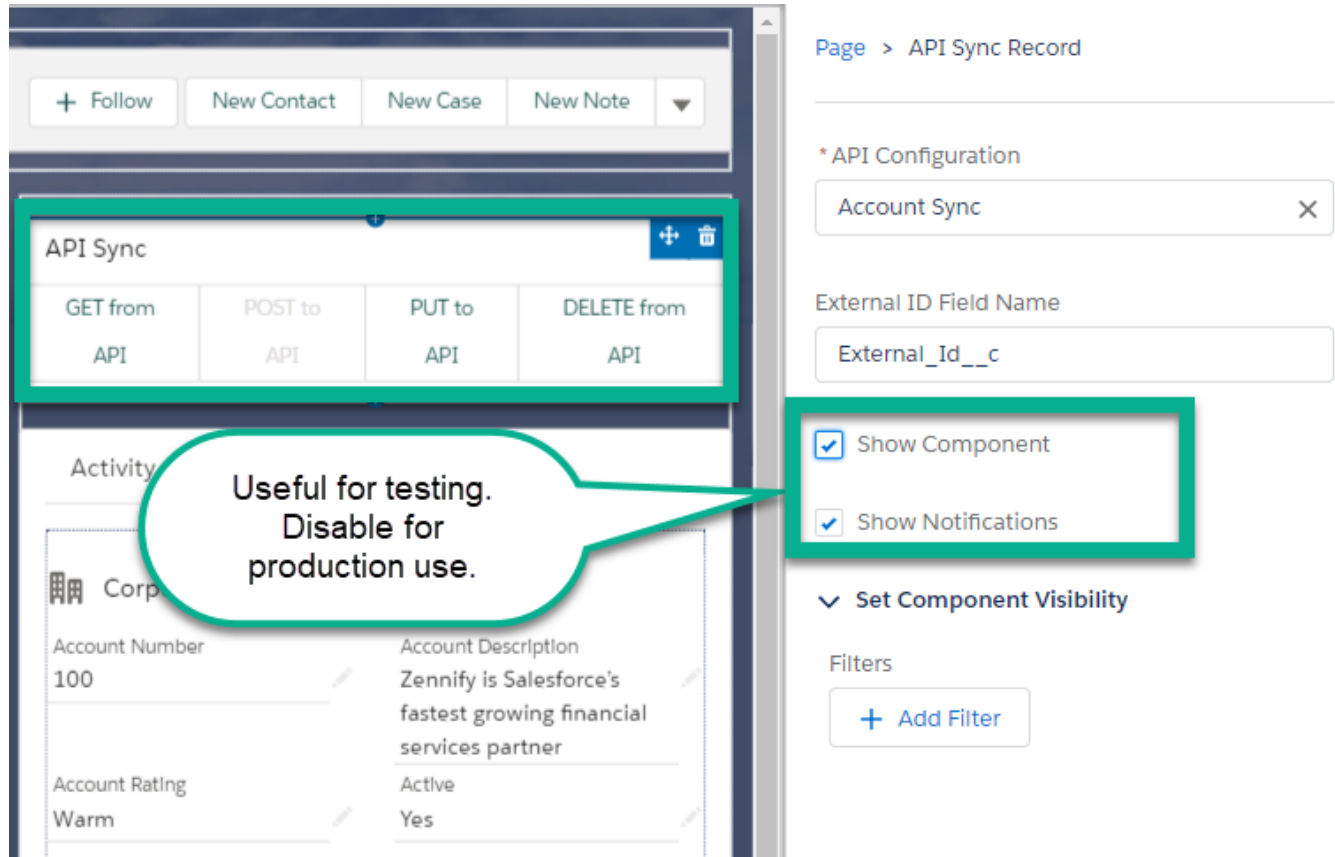
Configure Lightning Pages

Add the included lightning components to desired Lightning Record Pages.

API Sync

The API Sync component is used for syncing data from a data source using configured HTTP GET requests upon page load of a record. The component will provide a list of available API Object Mapping records that are Active and do not have Is Data Feed selected.

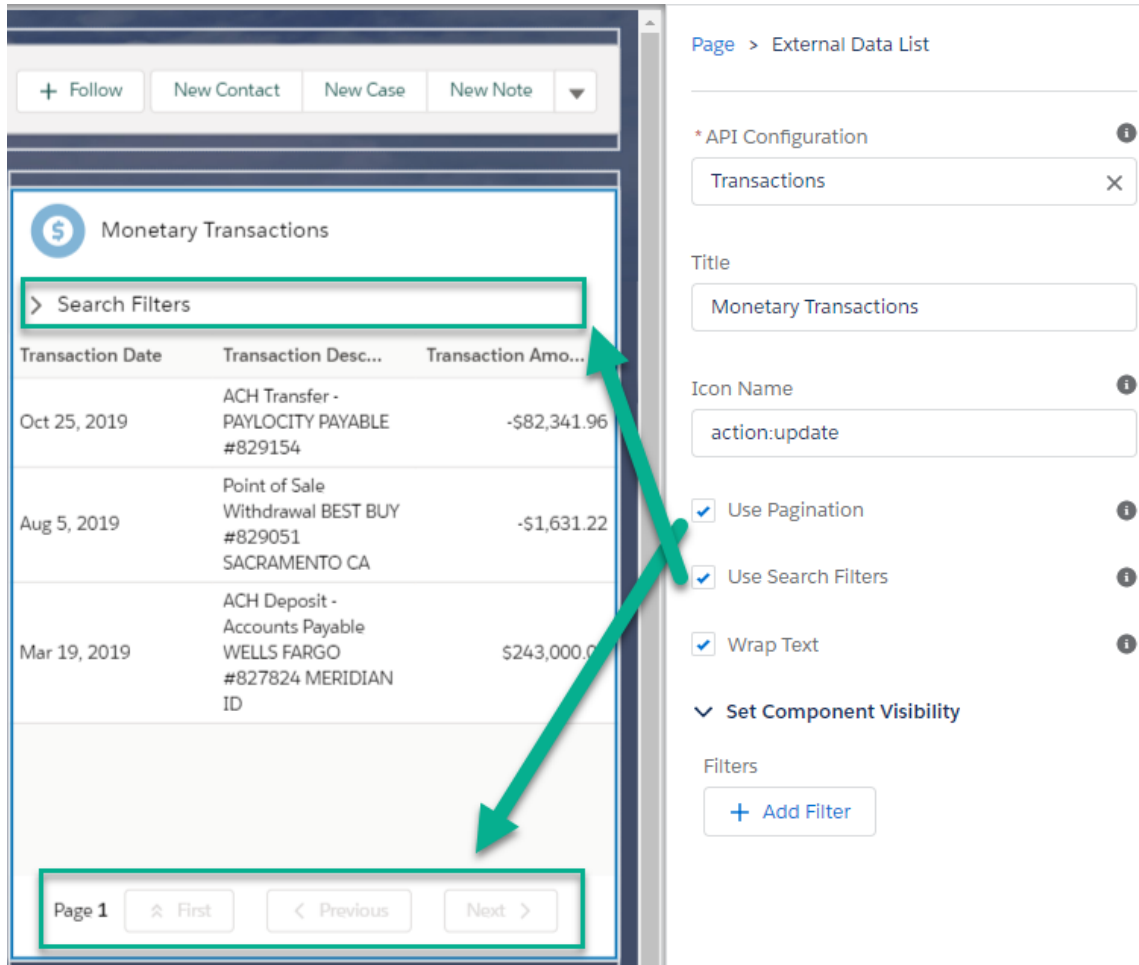
- **API Configuration:** Select desired API Object Mapping configuration. You can have multiple config mappings setup for a given Object type and display different instances of the API Sync component on different record type Lightning Pages or use Component Visibility rules to conditionally run different integrations on page load.
- **External ID Field Name:** Populate the field API Name of the External ID field that drives the integration. This is typically the id of the record in the external system and the component looks to see if this field is populated before calling out to the configured API. This field usually matches one External ID marked API Field Mapping record which is used for passing the value in a query or URI parameter to the API GET Endpoint.
- **Debug Component:** Selecting this will display the API Sync component buttons which can be used to manually test GET, POST, PUT, and DELETE requests. This is only recommended for testing purposes and should be hidden in production use.
- **Show Loading Icon and Message:** Enable to display loading spinner until API Sync has completed loading a response.
- **Loading Message:** Text to display when component is loading
- **Show Notifications:** Selecting this will display success or error toast message notifications when an API response is received. This is also designed for testing purposes and should be left on until ready to deploy integration to production users.
- **Enable Polling:** Select to repeat API invocation on specified time interval while record is open and browser tab is in focus
- **Polling Frequency Seconds:** Number of seconds until next API invocation when polling is enabled



External Data List

The External Data List component displays a list of external records without storage in Salesforce and supports configurable features around pagination and search filters.

- **API Configuration:** Select desired API Object Mapping configuration. You can have multiple config mappings setup for a given Object type and display different instances of the component on different record type Lightning Pages or use Component Visibility rules to conditionally run different integrations on page load. Only mappings with the Is Data Feed checkbox selected will be available.
- **Title:** Optional title text for the component.
- **Icon Name:** Optional icon name for the component title (See <https://lightningdesignsystem.com/icons/> for options). Format is lowercase <category>:<name> e.g. category "Standard" and icon name "account" would be "standard:account".
- **Use Pagination:** Select to enable the pagination component. Must have API Field Mapping records configured for each pagination field and API must support server side pagination via query or URI parameters.
- **Use Search Filters:** Select to enable search filter component. Must have API Field Mapping records configured for each filter field and API must support server side filtering via query or URI parameters.
- **Wrap Text:** Select to force wrap of long text in the data table columns.



Page > External Data List

* API Configuration ⓘ

Transactions ×

Title

Monetary Transactions

Icon Name ⓘ

action:update

☒ Use Pagination ⓘ

☒ Use Search Filters ⓘ

☒ Wrap Text ⓘ

▼ Set Component Visibility

Filters

+ Add Filter

Transaction Date	Transaction Desc...	Transaction Amo...
Oct 25, 2019	ACH Transfer - PAYLOCITY PAYABLE #829154	-\$82,341.96
Aug 5, 2019	Point of Sale Withdrawal BEST BUY #829051 SACRAMENTO CA	-\$1,631.22
Mar 19, 2019	ACH Deposit - Accounts Payable WELLS FARGO #827824 MERIDIAN ID	\$243,000.00

Page 1 ⬆ First < Previous Next >

External Data Record

The External Data Record component displays an individual external record with configured fields without storage in Salesforce.

- **API Configuration:** Select desired API Object Mapping configuration. You can have multiple config mappings setup for a given Object type and display different instances of the component on different record type Lightning Pages or use Component Visibility rules to conditionally run different integrations on page load. Only mappings with the Is Data Feed checkbox selected will be available.
- **Title:** Optional title text for the component.
- **Icon Name:** Optional icon name for the component title (See <https://lightningdesignsystem.com/icons/> for options). Format is lowercase <category>:<name> e.g. category "Standard" and icon name "account" would be "standard:account".

Batch Processing

API Integration Utility supports syncing records in bulk through the use of the provided batch Apex class. The provided Apex below can be run ad-hoc or scheduled to run on a recurring basis. This feature currently requires anonymous Apex code to be executed by an Administrator in the Salesforce organization. Code examples are provided below and can be executed from the Developer Console (Setup -> Developer Console -> Debug -> Open Execute Anonymous Window).

Bulk API Sync

To invoke a configured API Sync integration on a large volume of records that exist in Salesforce, invoke the `apiutil.ApiSyncBatch` Apex class. The included batch Apex will call out to the configured API for each existing record processed.

Example Sync of all records that have their external id populated for given API Object Mapping configuration:

```
//Config developer name
String apiObjectMapping = 'Customer';
//Sync all records (outbound API callout for each individual record)
Database.executeBatch(new apiutil.ApiSyncBatch(apiObjectMapping),
Limits.getLimitCallouts());
```

Example Sync of specific records queried (e.g. today's records):

```
//Config developer name
String apiObjectMapping = 'Customer';
//get desired record ids
List<Id> recordIds = new List<Id>(new Map<Id, Account>([SELECT Id FROM Account
WHERE CreatedDate = TODAY])).keySet();
//Sync specified records (outbound API callout for each individual record)
Database.executeBatch(new apiutil.ApiSyncBatch(apiObjectMapping, recordIds),
Limits.getLimitCallouts());
```

Scheduled API Sync

API Integration Utility can be scheduled to synchronize records in batches. To invoke a configured API Sync integration on a scheduled basis, schedule the `apiutil.ApiSyncBatch` Apex class through an anonymous Apex.

USE CASE #1: Schedule syncing on records that already exist in Salesforce with an external id:

Example scheduling of record sync all records created each hour on a 1 hour interval:

```
String apiObjectMapping = 'Products';
String jobName = 'API Sync Existing Products - Every Hour';
String cronExp = '0 0 * * * ? *'; //top of the hour
System.schedule(jobName, cronExp, new apiutil.ApiSyncBatch(apiObjectMapping,
'CreatedDate > {!NOW-1H}'));
```

NOTE: Api Sync Batch supports custom where clauses with datetime merge fields passed as a custom where clause in the format of `{!NOW}`. Optionally apply `+`/`-` operators and `Day`, `Hour`, `Minute` time units (e.g. `'CreatedDate > {!NOW-1H}'`).

NOTE: An additional method is available on `ApiSyncBatch` to set custom batch size used on scheduled job invocation. By default a batch size of 100 is used per `Limits.getLimitCallouts()`, however this can be overridden by invoking the `setBatchSize` method:

```
apiutil.ApiSyncBatch batchJob = new apiutil.ApiSyncBatch(apiObjectMapping,
'CreatedDate > {!NOW-1H}');
batchJob.setBatchSize(1);
System.schedule(jobName, cronExp, batchJob);
```

USE CASE #2: Schedule syncing of records in bulk that may or may not exist in Salesforce where API returns a list of records to be inserted or updated:

Example scheduling of bulk Product sync on a 1 hour interval:

```
String apiObjectMapping = 'Product_Batch';
String jobName = 'API Sync New or Updated Products - Every Hour';
String cronExp = '0 0 * * * ? *'; //top of the hour
System.schedule(jobName, cronExp, new
apiutil.ApiSyncBatch(apiObjectMapping));
```

NOTE: This use case requires the API Object Mapping configuration to have the **Bulk API Sync** checkbox selected, see example use case below.

Example API response payload:

```
{
  "nextPage": "/products?pageNumber=2",
  "records": [
    {
```

```

    "description": "In eleifend quam a odio. In hac habitasse platea dictumst.",
    "name": "Scallops - 20/30",
    "isactive": true,
    "id": "000d137a-f665-400b-8b91-a03282da5473"
  },
  {
    "description": "In hac habitasse platea dictumst.",
    "name": "Blouse / Shirt / Sweater",
    "isactive": false,
    "id": "000f3e89-2133-4c07-a7bf-8c99e650cb20"
  },
  {
    "description": "Morbi non lectus. Aliquam sit amet diam.",
    "name": "Lobster - Canned Premium",
    "isactive": false,
    "id": "0079b7b3-e61d-44d1-b573-9becfda013b3"
  }
]
}

```

Mapping Example:

API Object Mapping Detail

Label	Product Batch
API Object Mapping Name	Product_Batch
SF Object API Name	Product2
Active	☒
Is Data Feed	☐
Is Bulk Sync	☒

The API Field Mapping Records can be configured for extracting a list and if pagination is supported by the API to get all results.

API Field Mappings							
Action	Label	API Field Mapping Name	Mapping Type	Active	API Field Name	SF API Name	GET
Edit Del	Product Batch Description	Product_Batch_Description	HTTP Response	☒	description	Description	☒
Edit Del	Product Batch External Id	Product_Batch_External_Id	HTTP Response	☒	id	apiutil__External_Id__c	☒
Edit Del	Product Batch IsActive	Product_Batch_IsActive	HTTP Response	☒	isactive	IsActive	☒
Edit Del	Product Batch List	Product_Batch_List	HTTP Response	☒	records		☒
Edit Del	Product Batch List Next Page	Product_Batch_List_Next_Page	HTTP Response	☒	nextPage		☒
Edit Del	Product Batch Name	Product_Batch_Name	HTTP Response	☒	name	Name	☒

To extract a list out of an object structure like this example, add a field mapping of merge type Related Records (List). If the response is already a list than this is not needed:

▼ Field Mapping

API Field Name?	records	Mapping Type	HTTP Response
SF API Name?		Merge Type	Related Records (List)

▼ API Interactions

GET?

☒

Is Query?

☐

Parameter

To support paging of batch records, additionally configure field mapping for the next page endpoint. This can be a relative or full URL of the next page of results, including query parameters, but must be on the same host configured in the Named Credential of the integration.

▼ Field Mapping

API Field Name?	nextPage	Mapping Type	HTTP Response
SF API Name?		Merge Type	Flat Value

▼ API Interactions

GET?

☒

Is Query?

☐

Parameter

POST?

☐

Is URI?

☐

Parameter

PUT?

☐

DELETE?

☐

▼ External Data Table

Include In Datable?	☐	Datatable Index?	
Is Dynamic Value?	☐	Datatable Initial Width	
		Datatable Search Filter	N/A
		Datatable Pagination	Has Next Page

Event Processing

API Integration Utility supports syncing records through publishing of the included Platform Event object **API Sync Event**. Platform Events can be published from within Salesforce (e.g. Apex, Flow), or via outside integrations through use of preferred Salesforce API (i.e. REST API, SOAP API, Bulk API 2.0).

- https://developer.salesforce.com/docs/atlas.en-us.platform_events.meta/platform_events/platform_events_publish.htm

API Sync by Record ID

Creation of API Sync Event with the following fields populated will trigger API Sync configuration:

- Configuration Name: Name of API Object Mapping configuration to invoke API Sync process for an individual record.
- Record ID: Salesforce Record ID to execute API Sync process on

Example (Apex):

```
EventBus.publish(new List<apiutil__API_Sync_Event__e> {  
    new apiutil__API_Sync_Event__e(  
        apiutil__Configuration_Name__c = 'Customer',  
        apiutil__Record_ID__c = '0018c00002G0uS8AAL'  
    )  
});
```

Direct Update Values

Creation of API Sync Event with the following fields populated will trigger direct updates to record using API Sync configuration mappings:

- Configuration Name: Name of API Object Mapping configuration to invoke API Sync process for an individual record.
- Record ID: Salesforce Record ID to execute updates on
- Data: JSON string of field updates, where field name keys match the API Field Mapping API Name values set on specified Configuration Name. The values in the Data field will be directly applied to the record using active API Field Mappings. No API callout is performed and Data values are updated on the record.

```
EventBus.publish(new List<apiutil__API_Sync_Event__e> {  
    new apiutil__API_Sync_Event__e(  
        apiutil__Configuration_Name__c = 'Customer',  
        apiutil__Record_ID__c = '0018c00002G0uS8AAL',  
        apiutil__Data__c = '{"lastName": "New Last Name"}'  
    )  
});
```

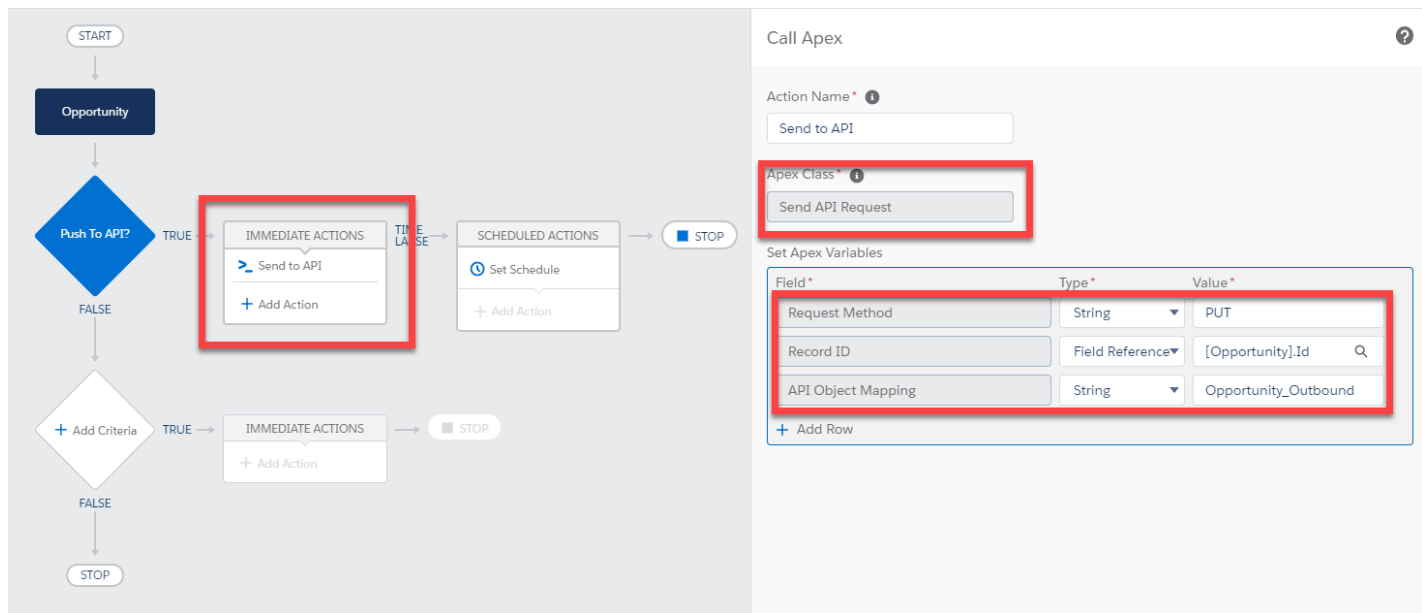
Outbound Pushes

API Integration Utility supports pushing data out of Salesforce to external systems through the use of provided invocable Apex actions that can be launched from Process Builder or Flow.

Push Record Updates

To push record changes outbound to an external source, create a Process Builder or Flow with desired record change criteria to act as the trigger point. The included Apex Action called “Send API Request” can be invoked.

- **Apex Class:** “Send API Request” (i.e. apiutil.ApiSyncInvocable)
- **Request Method:** GET, POST, PUT, or DELETE operation that needs to be performed
- **Record ID:** The Salesforce ID of the record to send data from
- **API Object Mapping:** The developer name of the API configuration record



Push Form Data

The included Apex Action can also be used to publish form data from a Visual Flow to a configured destination. There are two additional input variables supported called “dynamicKeys” and “dynamicValues” which are lists that can be populated with Flow variables. The **dynamicKeys** list must be populated with the API Field Names configured on the API Field Mapping records and the **dynamicValues** list then is populated in order with the values for those fields. Currently the invocable action returns the status code and raw string data of the API response, which must be handled within Flow as desired. If needed, a custom Invocable Apex class can be written with input/output parameters of your choice and can act as a wrapper to invoke the provided **apiutil.ApiSyncInvocable** Apex class.

Special Handling

Define your own custom extensions for specialized integration use cases.

Custom HTTP Headers

If Named Credentials do not support the authentication mechanism required by the external API, or if additional headers are required for access, custom authentication headers can be provided. Define custom HTTP headers through use of a custom Apex class implementing the provided `apiutil.ApiHeaderInterceptor` interface.

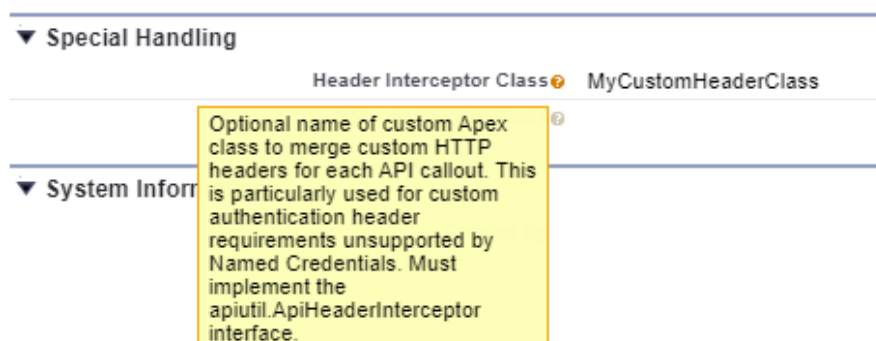
Example: An API requires a custom authentication header in the following format:

X-API-KEY: <secret key>

The API Integration Utility can support custom headers by implementing a custom Apex class like follows:

```
global class MyCustomHeaderClass implements apiutil.ApiHeaderInterceptor {
    /**
     * Implementation for custom headers
     */
    public Map<String,String> getRequestHeaders() {
        return new Map<String,String> {
            'X-API-KEY' => '{{$Credential.Password}}'
        };
    }
}
```

Salesforce HTTP merge fields are supported via Named Credential record (e.g. username or password fields). The custom. The Apex class must be **global** and the “getRequestHeaders” method implemented in order to be executed. The class name must be populated on an API Object Mapping metadata configuration record for usage:



Custom HTTP Request Transforms

Request data for outbound API callouts can be manipulated or transformed via Apex code provided to API Integration Utility via implementing the provided `apiutil.ApiRequestInterceptor` interface. The Apex class must be **global** in order to be executed.

The API Integration Utility can support custom request transformations by implementing a custom Apex class like follows:

```
global class MyCustomRequestTransform implements apiutil.ApiRequestInterceptor {

    /**
     * Implementation for HTTP request transformations
     */
    public HttpRequest transformRequest(HttpRequest request) {
        System.debug('MyCustomRequestTransform --
\n\nInfo: '+request.toString()+'\nData:\n'+request.getBody());

        //get mutable object from json string
        Object requestObj = JSON.deserializeUntyped(request.getBody());
        if (requestObj instanceof Map<String,Object>) {
            Map<String,Object> requestMap = (Map<String,Object>)requestObj;
            //transform request data
            requestMap.put('name', requestMap.get('name')+' - OVERRIDE');
            //set new request data
            request.setBody(JSON.serialize(requestMap));
        }

        return request;
    }
}
```

The class name must be populated on an API Object Mapping metadata configuration record for usage:

▼ Special Handling	
Header Interceptor Class	
HTTP Request Interceptor Class	MyCustomRequestTransform
Optional name of custom Apex class to manipulate or transform raw HTTP request data prior to sending. Must implement the <code>apiutil.ApiRequestInterceptor</code> interface.	
▼ System Infor	

Custom HTTP Response Transforms

Response data from API callouts can be manipulated or transformed via Apex code provided to API Integration Utility via implementing the provided `apiutil.ApiResponseInterceptor` interface. The Apex class must be **global** in order to be executed.

The API Integration Utility can support custom response transformations by implementing a custom Apex class like follows:

```
global class MyCustomResponseTransform implements apiutil.ApiResponseInterceptor {

    /**
     * Implementation for HTTP response transformations
     */
    public HttpResponse transformResponse(HttpResponse response) {
        System.debug('MyCustomResponseTransform --
\n\nStatus: '+response.getStatusCode()+'\nData:\n'+response.getBody());

        //get mutable object from json string
        Object responseObj = JSON.deserializeUntyped(response.getBody());
        if (responseObj instanceof Map<String,Object>) {
            Map<String,Object> responseMap = (Map<String,Object>)responseObj;
            //transform response data
            responseMap.put('name', responseMap.get('name')+' - OVERRIDE');
            //set new response data
            response.setBody(JSON.serialize(responseMap));
        }

        return response;
    }
}
```

The class name must be populated on an API Object Mapping metadata configuration record for usage:

HTTP Response Interceptor Class 🚩	MyCustomResponseTransform
Optional name of custom Apex class to manipulate or transform raw HTTP response data. Must implement the <code>apiutil.ApiResponseInterceptor</code> interface.	

Custom DML Pre-Processing

SObjects and Fields that are mapped by API Integration Utility can be manipulated or transformed prior to DML statements and after HTTP response data has been received and mapped to SObjects. An Apex class implementing the `apiutil.ApiDatabaseInterceptor` interface can be provided to execute custom logic.

The API Integration Utility can support custom response transformations by implementing a custom Apex class like follows:

```
global inherited sharing class DynamicDatabaseOverride implements
ApiDatabaseInterceptor {

    /**
     * Implementation of custom DML preprocessor logic
     */
    public List<SObject> preprocessDML(List<SObject> recordsForDML) {
        //Manipulate SObjects prior to insert/update

        return recordsForDML;
    }
}
```

The class name must be populated on an API Object Mapping metadata configuration record for usage:

▼ Special Handling

Header Interceptor Class

DML Interceptor Class
DynamicDatabaseOverride

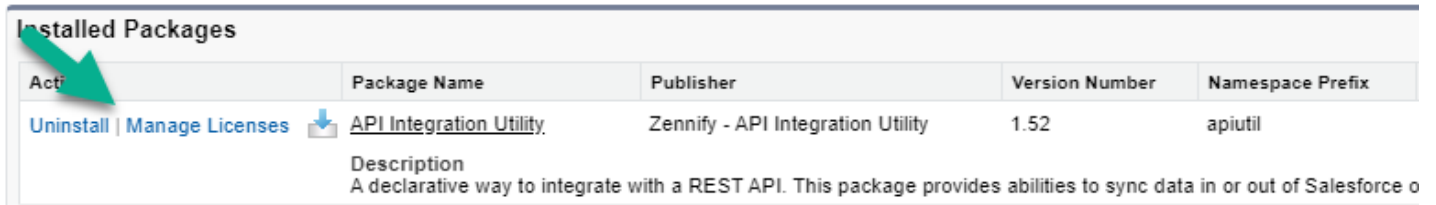
▼ System Info

Optional name of custom Apex class to extend record processing before DML transaction. Must implement the `apiutil.ApiDatabaseInterceptor` interface.

Usage Considerations

Assigning Licenses

Most installations include user licenses. Unless you have been provisioned a site wide license, you will have user licenses that must be allocated to each individual Salesforce user that will use this application. This can be managed from the Installed Packages area of the Salesforce Setup menu. Navigate to the API Integration Utility package. If you have the “Manage Licenses” link available, from there you have user licenses that must be assigned.



Active	Package Name	Publisher	Version Number	Namespace Prefix
Uninstall Manage Licenses	API Integration Utility	Zennify - API Integration Utility	1.52	apiutil
Description A declarative way to integrate with a REST API. This package provides abilities to sync data in or out of Salesforce o				

User Permissions

API Integration Utility will automatically update fields on records based on its configuration and trigger criteria set. Field updates will occur on behalf of the running user that invoked the component. API Integration Utility will update records based on configured HTTP Response field mappings and what data is returned by the API. The API Sync component, when used, executes upon page load on behalf of the running User. This will enable any User with access to the record to invoke the API call and allow the system to update the record with changes received regardless of Field Level Security. The running user must have Object level edit access rights for record page syncing to work. The Last Modified By audit field will record the user that triggered any changes and Field History Tracking will record tracked data changes invoked by the running User. The intention with API Sync is to establish trust with the external system and only map data that you rely on from the external system as a source of truth. A simple example is viewing the updated balance of a bank account. The balance is usually stored in an external banking core and updates to Salesforce for any user that has read access. Best practice will be to maintain Read Only for fields written by API Sync.

Apex Class Access

The following Apex Classes must be associated with a Permission Set or Profile assigned to Salesforce Users for access to the application components.

Name	Purpose
apiutil.ApiSyncComponentController	Aura Enabled Apex for the API Sync lightning component
apiutil.ApiExternalDataComponentController	Aura Enabled Apex for the External Data Table lightning component
apiutil.ApiSyncInvocable	Invocable Apex action that can be used by Flow or custom Apex run by users

Purchasing the App

To purchase the app you will need to login to the AppExchange and install the package in your production org. Once installed in production, you can re-visit the AppExchange listing and you will be presented with a Buy Now option to initiate the AppExchange Checkout process.

Permissions Required: Checkout Enabled (User checkbox) and Manage Billing (System permission).

Non-Profits

You may need to open a ticket with Salesforce support to enable AppExchange Checkout permissions. You must be able to see the Manage Billing system permission on the Profile for your user and see the Checkout Enabled checkbox on your User record.