GEMBA Note for Salesforce Development Guide

Updated on January 11, 2018
MetaMoJi Corporation

Table of Content

1 Introduction

GEMBA Note for Salesforce is a package that connects Salesforce information and pages in GEMBA Note for Business (http://business.metamoji.com/gemba-note/).
This linkage allows Salesforce users to open a GEMBA Note page containing rich annotations (handwriting, voice, video, etc.) from an instance such as a case or opportunity in Salesforce. The structured data is kept in Salesforce and the unstructured data is saved in GEMBA Note.

The client app for GEMBA Note for Business has been released in the Appstore and Windows Store. The features of GEMBA Note for Salesforce will be added to the business app.

2 GEMBA Note for Business Overview

2.1 What is GEMBA Note for Business

GEMBA Note for Business is a state-of-the-art app that can digitize the paper forms and workbooks used for services, and industries engaged in field based activities. The app makes it possible to record information, create flexible to-do lists and to prioritize tasks and inspection results. Users can transform their notes into sophisticated organizational plans, assessment reports, and action lists. The app combines the flexibility of a pen and paper with digital media, capturing directly from your tablet. Audio recording features enable users to capture an accurate record of conversations, environmental sounds, or just spoken notes.

Information comes in many forms, and GEMBA Note is designed to help you capture it quickly and efficiently. Combine structured and unstructured information, such as forms, spreadsheets, sketches, and free format notes. Forms can be designed within the application or imported from PDF and office documents. GEMBA Note is used in a wide variety of industries, including construction, civil engineering, facilities management, surveying, and servicing.
Teams can co-edit documents in real time and even use the application as an interactive whiteboard. The Share function of the app is a group collaboration tool for dozens of editors or readers to share information, and to visually express their ideas and show

their current circumstances through live interactive sessions.

Feature Overview:

- Use the included form and paper templates, import your existing forms or create new ones
- Maintain a diary with pages automatically timestamped, allowing you to organize and view your records using the calendar view
- Using the spreadsheet component enter data, automatically calculate results, and perform analysis
- Capture audio and movie quickly with voice/video memos. Easily play within your note pages
- Jump to notes taken on a specific day using the calendar view
- Search for target templates or content using tags, enabling to-do lists and much more
- Customize search conditions
- Write, sketch and draw notes with a variety of pens, paper layouts, and graphics. Includes calligraphy pens and special inks from a vast color palette
- Take photographs using the camera or import existing images straight into your documents
- A vector graphics system (as used in CAD applications) means you can fine tune and edit anything you add
- Move, scale and rotate anything on your page without losing quality
- Import PDF documents as pages in a note, annotate and then save as a new PDF file
- Import multiple PDF documents on the same page for side by side information analysis
- Easily capture text from a keyboard
- Enhanced text formatting options include options to add bullets and increase & decrease indents
- Pre-defined basic shapes library for quick use
- Include web page links and visualizations within a note
- Smart cropping tool for photo editing
- Quickly access frequently used tools and commands, customized for different business scenarios

Share Features:

- Create teams and share templates, items, and toolboxes between members
- Convert to a Share Note to allow multiple users to visually interact with the documents and add text, handwriting, photos, new pages, graphics, spreadsheets, and any other content types.
- Tools for managing meetings:
    - ➢ Control a meeting by a presenter. The presenter can turn pages of the current Share Note and zoom in and out on the current editor screen
    - ➢ Easily view planned and ad-hoc future and past meetings

2.2 What GEMBA Note for Salesforce Can Do

GEMBA Note for Salesforce is a bridge between Salesforce and GEMBA Note. The data such as cases and opportunities in Salesforce can be imported into GEMBA Note as pages. The pages are automatically generated using a note template that is customized in GEMBA Note. On GEMBA Note, the user can draw, add a text unit, web unit, photo, spreadsheet, voice, video, and so on. These functions are available even the user is located without network. For example, a field service engineer brings an iPad or iPhone on which case data are stored from Salesforce to GEMBA Note. The engineer can keep records with handwriting, text, photo, voice, and video at customer's office offline.
In addition, the updated data (Salesforce's field values) on GEMBA's pages can be stored in Salesforce. At the time, a custom object defined inside Salesforce is automatically generated.

MetaMoJi

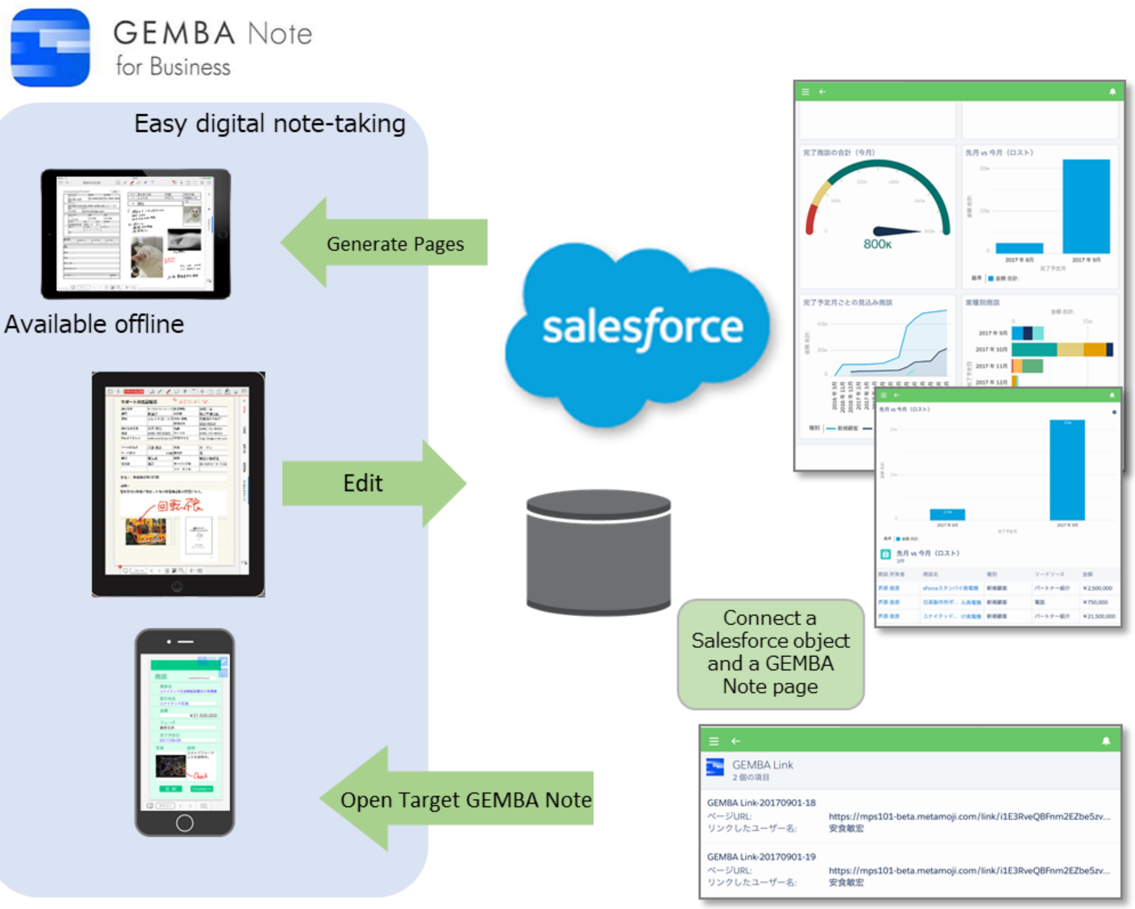
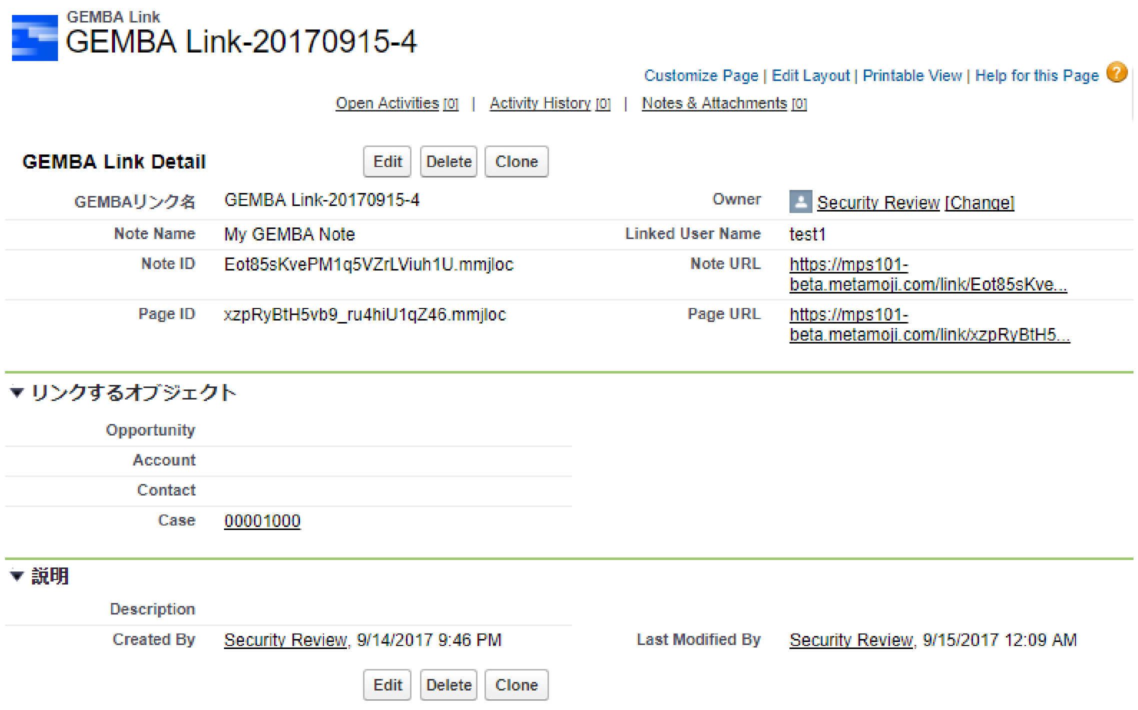


Figure 1: Salesforce and GEMBA Note

The GEMBA Note for Salesforce package includes a custom object named **GEMBA Link**. The object has properties describing the note and page in a GEMBA Note as shown below:



GEMBA Link
GEMBA Link-20170915-4

Customize Page | Edit Layout | Printable View | Help for this Page

Open Activities [0] | Activity History [0] | Notes & Attachments [0]

GEMBA Link Detail

GEMBAリンク名	GEMBA Link-20170915-4	Owner	Security Review [Change]
Note Name	My GEMBA Note	Linked User Name	test1
Note ID	Eot85sKvePM1q5VZrLViuh1U.mmjloc	Note URL	https://mps101-beta.metamoji.com/link/Eot85sKve...
Page ID	xzpRyBtH5vb9_ru4hiU1qZ46.mmjloc	Page URL	https://mps101-beta.metamoji.com/link/xzpRyBtH5...

▼ リンクするオブジェクト

Opportunity	
Account	
Contact	
Case	00001000

▼ 説明

Description			
Created By	Security Review, 9/14/2017 9:46 PM	Last Modified By	Security Review, 9/15/2017 12:09 AM

Figure 2: A sample of GEMBA Link in Salesforce

The custom object includes:

Note Name:	Title of the linked note in GEMBA Note
Linked User Name:	Who linked the note in GEMBA Note (account from GEMBA Note)
Note ID:	Identifier of the note in GEMBA Note
Note URL:	URL of the linked note in GEMBA Note
Page ID:	Identifier of the linked page in GEMBA Note
Page URL:	URL of the linked page in GEMBA Note

When you tap the '*Page URL*' using the Safari app on iOS, the corresponding GEMBA note will open at the correct page.
A GEMBA Link object refers to a target Salesforce object such as case and opportunity. It means that the Salesforce user can open a target GEMBA Note from an object in Salesforce to see the richer annotations such as handwriting, photo, and video.

2.3 How GEMBA Note for Salesforce Works

GEMBA Note for Salesforce works with a web service using REST. When you get Salesforce objects, GEMBA Note submits a GET message with JSON data. When you send field values from GEMBA Note, the app submits a POST message with JSON data. The role of the web service is not only sending or receiving a message, but also adjust naming in schemas between the Salesforce and GEMBA Note API's.

We assume that specific web services will be designed and implemented by system integrators to access data as required for individual installations. At MetaMoJi we have developed the service utilizing node-red (https://nodered.org/).

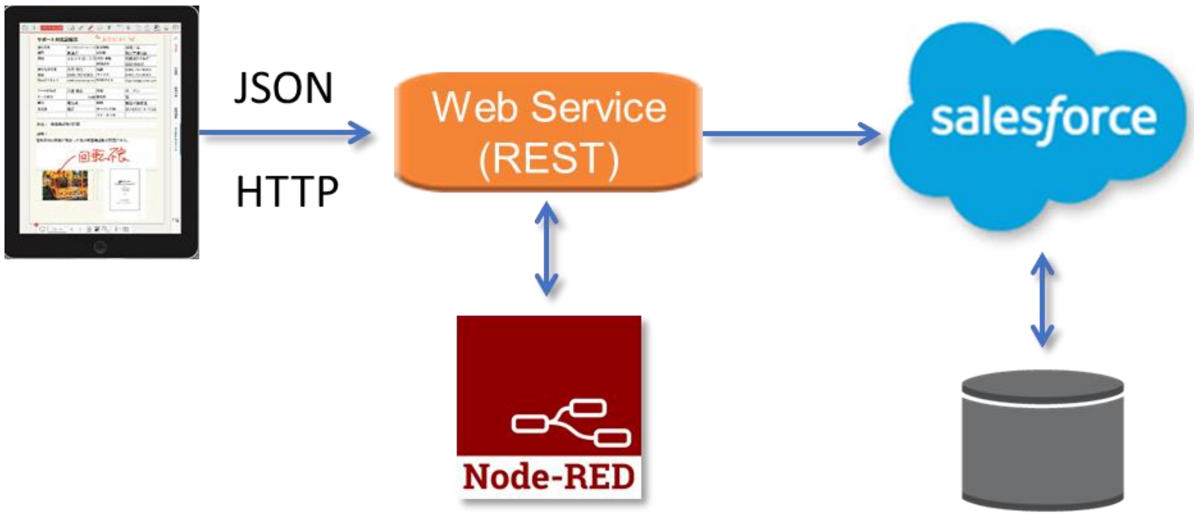


Figure 3: RESTful Service based on Node-RED

3 Customization Overview

This chapter summarizes what components should be customized or developed for both the GEMBA Note client app and Salesforce.

3.1 Components to be Customized

To gain benefits from the GEMBA Note for Salesforce, you need to customize the following components:

Component	Product/Store	Description
Form	GEMBA Note Client app	Form (page template) to be created with the development environment in GEMBA Note
Tag Schema	GEMBA Note Client app	Collection of tags to be defined using the GEMBA Note client app.
GEMBA Link	AppExchange	A custom object in Salesforce. Install the GEMBA Note app from the AppExchange with your Salesforce organization.
Fields & Relations	Salesforce	Create a link relation from your Salesforce objects to GEMBA Link
HTTP Methods	Server	GET and POST methods used for linking GEMBA Note and Salesforce.
Search Settings Files	GEMBA Note Client app	XML files that define how data is retrieved from Salesforce.

3.2 Users and Access Permissions

In the GEMBA Note side, the user type must be "*Developer*" to create a form, to define a tag schema, and to link the form and the schema.
The profile of the user should be "*System Administrator*" in Salesforce.

3.3 Sample Data - Property

To describe steps to customize or develop each component, a sample object named "Property" will be used. A property object is a real estate that represents an apartment, a house, an office. It includes the following information:

Field	Data Type	Description
ID	Text	Identifier used in the Salesforce object (*) This parameter is used for an HTTP POST method.
Property Name	Text	Name of the property
Type	Text	Property type such as Apartment, House, Office, Land Only, and Other.
Price	Currency (Float)	The price of the property
Location	Text	Place (address) of the property.

4 Customization in GEMBA Note

This chapter illustrates tasks within the GEMBA Note client app.
For more details about the form creation, please have a look into:
http://product.metamoji.com/manual/gemba_b3/document/gemba_note3_form_guide_en.pdf

4.1 Creating a Development Team Drive

A form can be created within a development team drive (folder). Please create a team from ＋ on the top bar menu > [Create Package] on the Notes List screen of GEMBA Note.

4.2 Creating a Form

A form is a page that allows you to enter or edit data within a note. On the form, you can put a variety of form controls; checkbox, number input, text input, text-area, etc.

[1] Creating a Note

On the created development team drive, tap ☰+ and select an appropriate note type. A new note will be created.
You can change the paper size and background ☰ > [Paper].

[2] Putting Form Controls

On the page, choose a form control from ＋ > [Add Form Control].

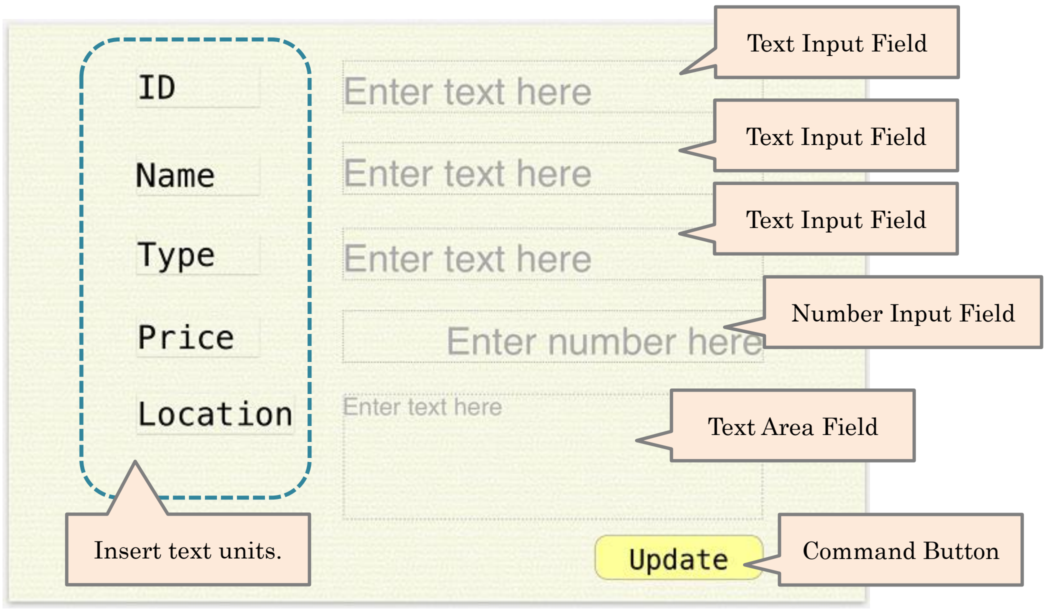


Figure 4: Property Form

You can insert a caption for each form control by tapping ⊤ on the top bar menu and place the text unit.

4.3 Defining a Tag Schema

A tag schema is a collection of attributes that represents a target object. For example, a real estate property includes attributes such as name, type, price, and location as described before.

To define a tag schema, hold down on a development team drive and navigate to [More] > [Tag Schema List] from the context menu popped up.

[1] Creating a Tag Schema

Tap [Import] on the Tag Schema List dialog. Then tap [Add Tag Schema]. You have the Create Tag Schema dialog.

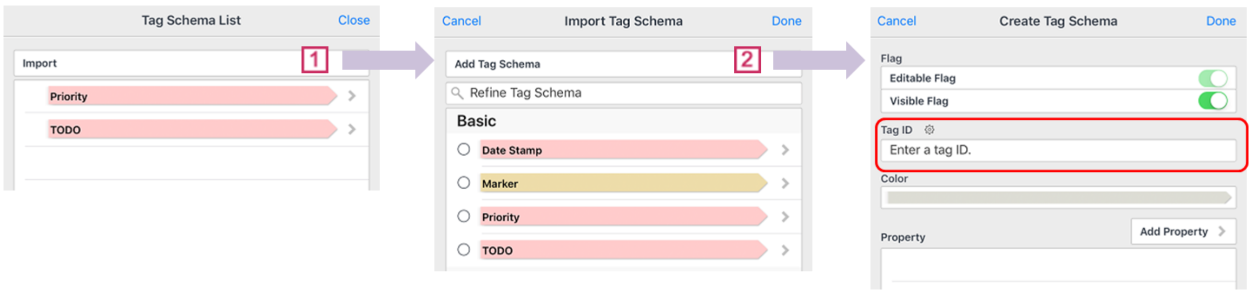


Figure 5: Creating a Tag Schema

Enter an ID of the schema (e.g. Property) on the Create Tag Schema.
When tapping the setting icon next to "Tag ID", you can specify a display name for Japanese and English.

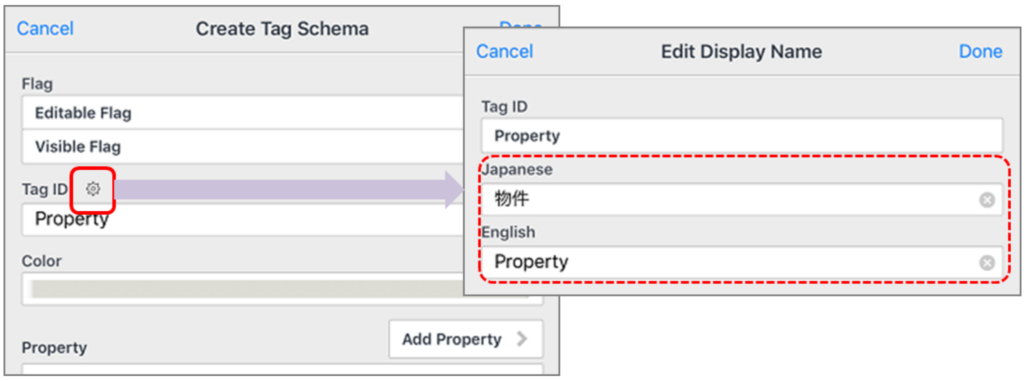


Figure 6: Editing Display Name of a Tag Schema

[2] Creating a Property (Attribute)

Tapping [Add Property] creates a new property.

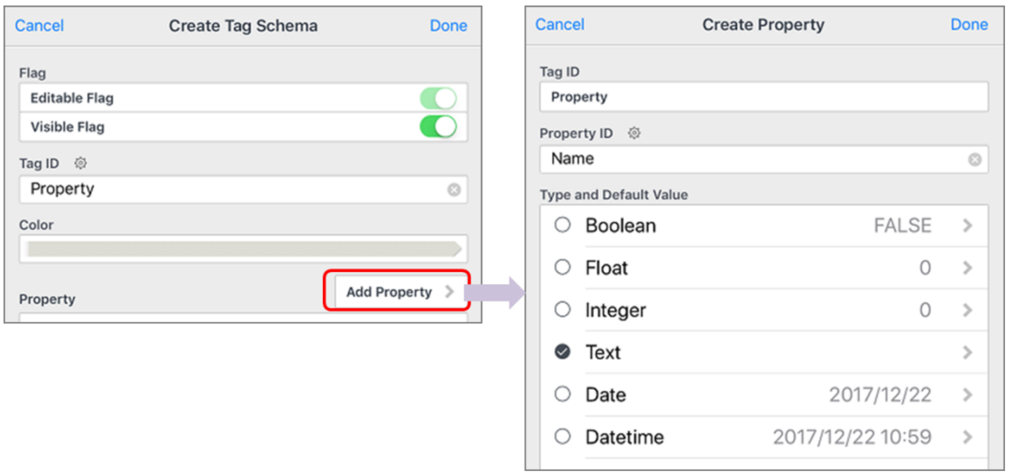
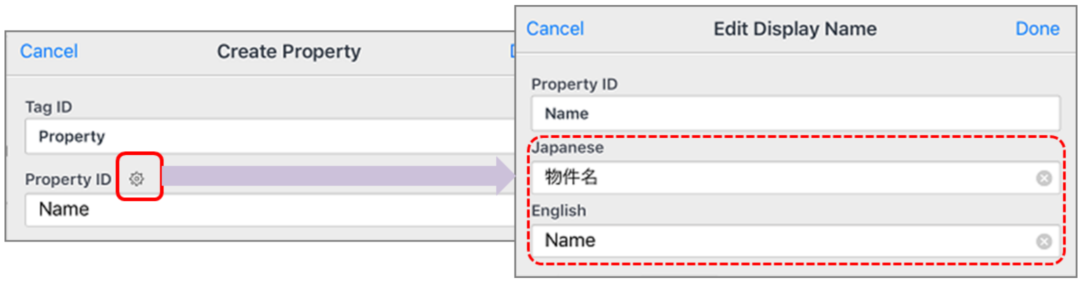


Figure 7: Creating a Property

Enter the property ID and choose one of the data types (Boolean, Float, Integer, Text, Date or Datetime). When tapping the setting icon next to "Property ID", you can specify a display name for Japanese and English.


Cancel Create Property

Tag ID: Property

Property ID: Name

Cancel Edit Display Name Done

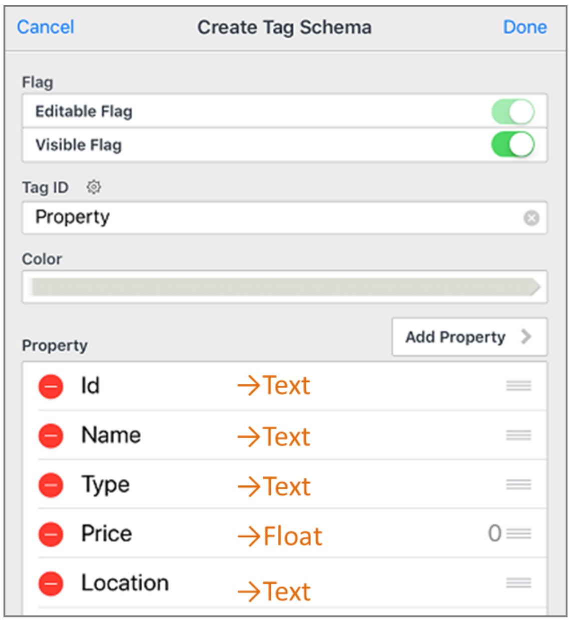
Property ID: Name

Japanese: 物件名

English: Name

Figure 8: Editing Display Name of a Property

The following dialog shows the all available properties of the Property tag schema.


Cancel Create Tag Schema Done

Flag

- Editable Flag
- Visible Flag

Tag ID: Property

Color

Property — Add Property

Id	→Text	
Name	→Text	
Type	→Text	
Price	→Float	0
Location	→Text	

Figure 9: Property Tag Schema

4.4 Linking Form Controls to Tags

The next step is to connect each form control to a tag (a property) of your created tag schema.

[1] Assigning a Tag Schema to a Form

Open your note which includes the Property form again, and touch the book icon 📖 to open the list of pages (Pages List). Then hold down on the page. You have the context menu.

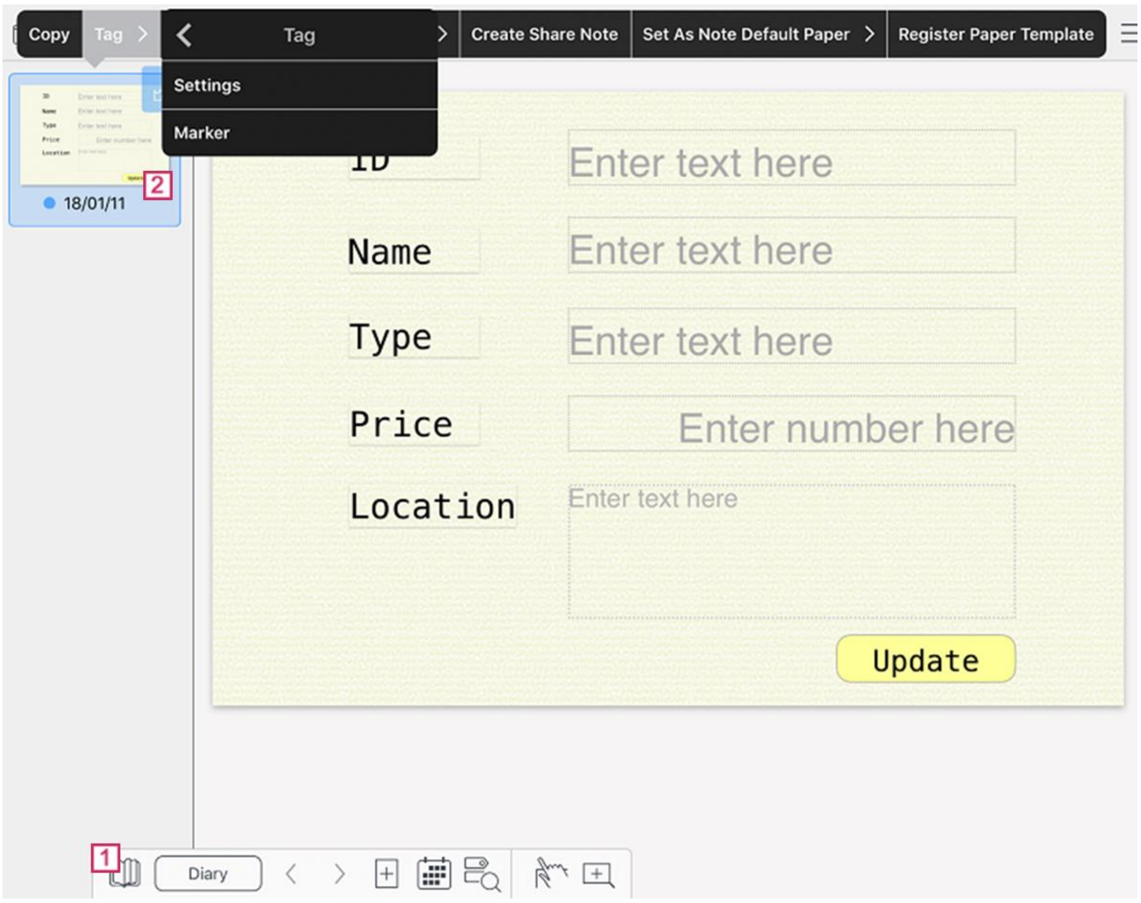


Figure 10: Tag Schema Settings

Select [Tag] > [Settings] from the context menu. You have the Tags dialog. Tapping [Add Tag] brings the Tag Schema List dialog. Check the Property item in the list and tap the Done button.

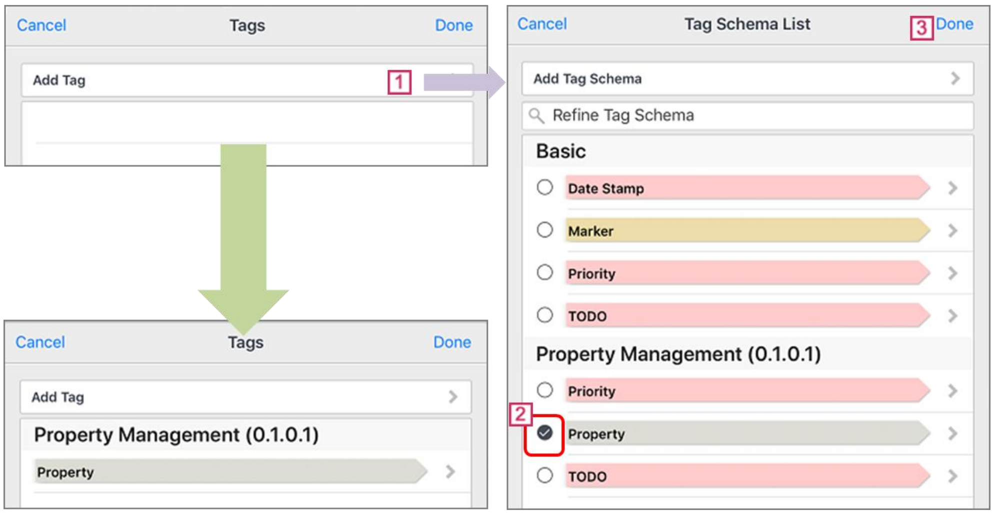


Figure 11: Specifying a Tag Schema from the list

[2] Linking Each Form Control to a Tag

Holding down on a Form Control brings the context menu. Go to [Form] > [Link to Tag]. You have the list of available tag properties such as Id, Name, Type, or Location. Choose an appropriate tag property.

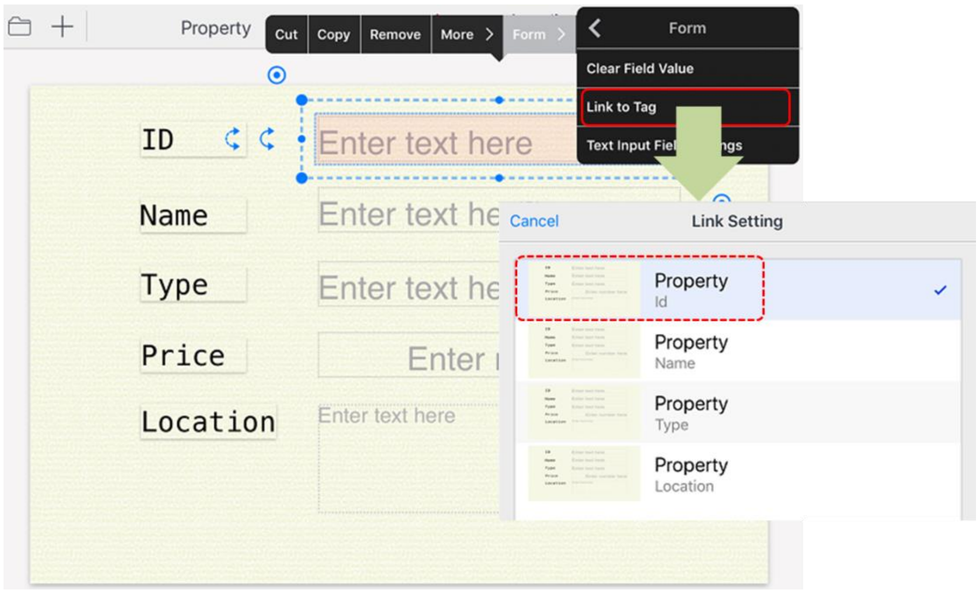


Figure 12: Linking a Form Control to a Tag

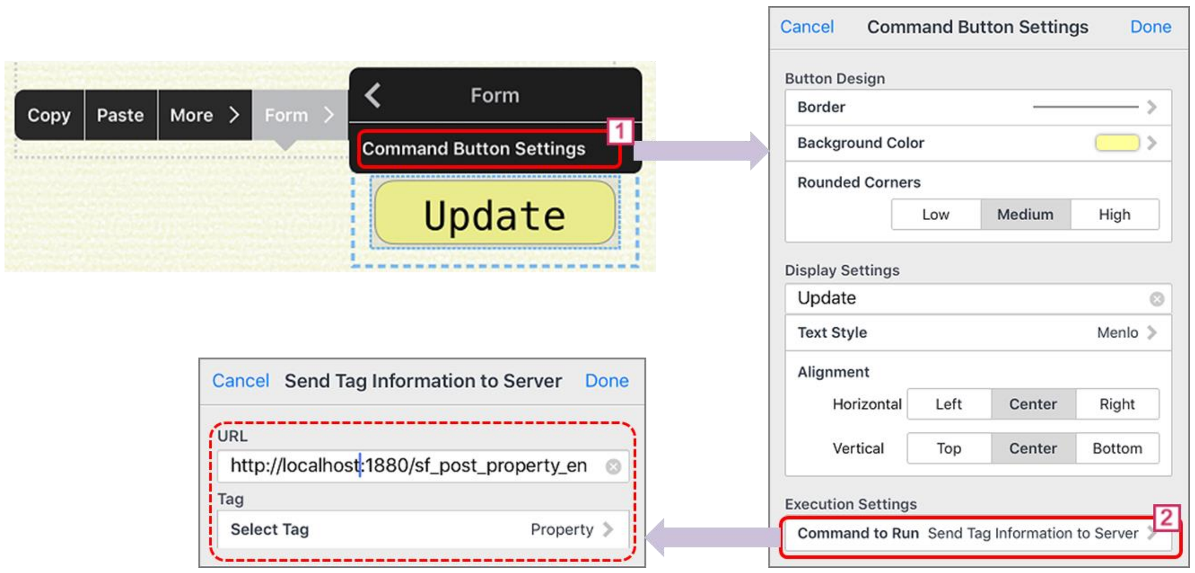
Repeat the same operation for each form control on the page.

MetaMoJi

4.5 Setting a POST Method to a Command Button

The command button “Update” in the Figure 4 is used to send input field values from the form controls to the Salesforce server.

When you hold down on the button, you have the context menu. Navigating [Form] > [Command Button Settings] from the context menu, you will get the Command Button Settings dialog.



Cancel Command Button Settings Done

Button Design
- Border
- Background Color
- Rounded Corners: Low / Medium / High

Display Settings
- Update
- Text Style: Menlo
- Alignment
  - Horizontal: Left / Center / Right
  - Vertical: Top / Center / Bottom

Execution Settings
- Command to Run: Send Tag Information to Server

Cancel Send Tag Information to Server Done

URL
http://localhost:1880/sf_post_property_en

Tag
Select Tag: Property

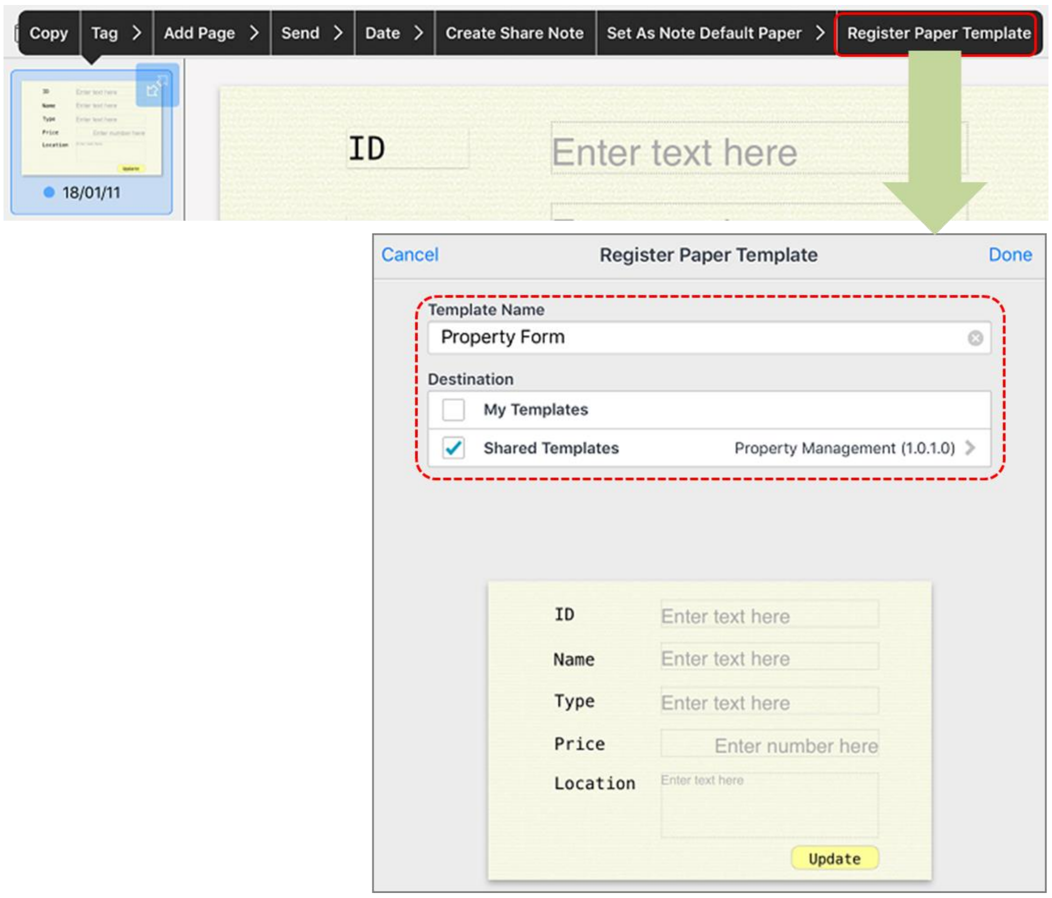
Figure 13: Setting URL to a Command Button

On the Command Button Settings dialog, tap [Command to Run] and choose [Send Tag Information to Server]. You will have the Send Tag Information to Server dialog on which you need to specify the following info:

Parameter	Description
URL	URL for an HTTP POST method. Specify what you implemented in the section 6.4.
Tag	Name of a tag schema. Specify what you created in the section 4.34.3.

4.6 Registering a Form as a Paper Template

To retrieve Salesforce data into the GEMBA Note app, you will be asked which paper template is used to generate pages. To register your created form as a paper template, on the Pages List hold down on a target page. Then select [Register Paper Template] from the context menu. You have the Register Paper Template dialog below:



Template Name

Property Form

Destination

My Templates

✓ Shared Templates — Property Management (1.0.1.0)

ID	Enter text here
Name	Enter text here
Type	Enter text here
Price	Enter number here
Location	Enter text here

Figure 14: Registering a Paper Template

Enter a template name (e.g. Property Form) and choose a package name. The form will be registered in the package as a paper template.

To confirm whether the template is registered, go to ＋ > [Add Page] > [Add using Paper Style] > [Shared Templates] > [*Your package name*].

4.7 Defining a Search Condition

GEMBA Note embeds the search function based on tag schemata. Based on this mechanism external data such as Salesforce can be imported.

A search condition file is an XML file that contains the following structure:

```
<?xml  version=”1.0”  encoding=”UTF-8”?>
<searchSetting
xmlns=”http://xmlns.metamoi.com/noteanytime/aggregation/searchSetting/1.0”>
   <settingId>
      <name>...</name>
   </settingId>
   <extractSettings>
      <connector>…</connector>
   </extractSettings>
   <sql>…</sql>
   <resultClass tagId="Property"/>
   <searchParameters>
      <searchParameter>…</searchParameter>
   </searchParameters>
</searchSetting>
```

Element Name	Description
`searchSetting`	The root element
`settingId`	Defines the search setting ID
`extractSettings`	Defines how to retrieve data
`sql`	SQL statement
`resultClass`	Tag schema to be mapped from the search result
`searchParameters`	Defines parameters to be used for the search condition

MetaMoJi

[1] Search Setting ID

The <settingID> element includes <name> elements that represent names of the search condition in different languages with the "xml:lang" attribute.

Sample:

```
<settingId id="REST_SFProperty">
        <name xml:lang="en">[Salesforce] Property Search</name>
        <name xml:lang="ja">[Salesforce] 物件検索</name>
</settingId>
```

[2] Extraction Settings

The <extractSettings> element includes a <connector> element that shows how to get data from a target REST http. The <connector> element contains a <url> element and a <method> element. The extractTableName attribute of the <extraSettings> element is used in the <sql> element.

Sample:

```
<extractSettings>
   <connector type="REST" extractTableName="property_list">
      <url>http://localhost:1880/sf_get_property_en</url>
      <method>GET</method>
   </connector>
</extractSettings>
```

The <url> content will be implemented in 6.1 section.

[3] SQL

The <sql> element declares an SQL statement that retrieves a search result.

Sample:

```
<sql>SELECT id, Name, Type, Price, Location
   FROM property_list WHERE Name LIKE '%$PNAME%' ORDER BY id ASC
</sql>
```

MetaMoJi

[4] Result Class

The <resultClass> element declares which tag schema is applied to for mapping data from the search result.

Sample:

```
<resultClass tagId="Property"/>
```

[5] Search Parameters

The <searchParameters> element contains parameters for the search condition. <searcParameter> elements represent parameters to be used in other places. For example, “PNAME” in the following sample is used for the WHERE statement in the <sql> element above. You can specify an initial value for each parameter.

Sample:

```
<searchParameters>
   <searchParameter type="String">
      <parameterId id="PNAME">
          <name xml:lang="en">Property Name</name>
          <name xml:lang="ja">物件名</name>
      </parameterId>
      <initialValue>Apt</initialValue>
   </searchParameter>
</searchParameters>
```

Please save the XML file in iTunes, your WebDAV server, or any external storage apps such as OneDrive, DropBox, Google Drive, etc.

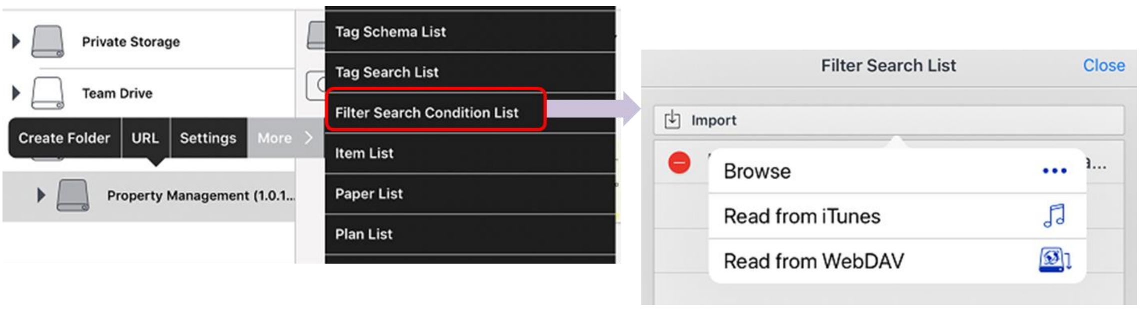
Sample Search Condition (REST_Property_Settings.mmjconf):

```
<?xml version="1.0" encoding="UTF-8"?>
<searchSetting xmlns="http://xmlns.metamoji.com/noteanytime/searchSetting/1.0">
        <settingId id="REST_SFProperty">
                <name xml:lang="en">[Salesforce] Property Search</name>
                <name xml:lang="ja">[Salesforce] 物件検索</name>
        </settingId>
        <extractSettings>
                <connector type="REST" extractTableName="property_list">

        <url>http://mps101a-nodered-beta.metamoji.com:1880/sf_get_property_en</url>
                        <method>GET</method>
                </connector>
        </extractSettings>
        <sql>SELECT id,Name,Type,Price,Location
                FROM property_list WHERE Name LIKE '%$PNAME%' ORDER BY id ASC
        </sql>
        <resultClass tagId="Property"/>
        <searchParameters>
                <searchParameter type="String">
                        <parameterId id="PNAME">
                                <name xml:lang="en">Property Name</name>
                                <name xml:lang="ja">物件名</name>
                        </parameterId>
                        <initialValue>Apt</initialValue>
                </searchParameter>
        </searchParameters>
</searchSetting>
```

Download → https://www.dropbox.com/s/u4ajctbwzujnhkv/REST_Property_Setting.agss?dl=0

4.8 Importing a Search Condition File

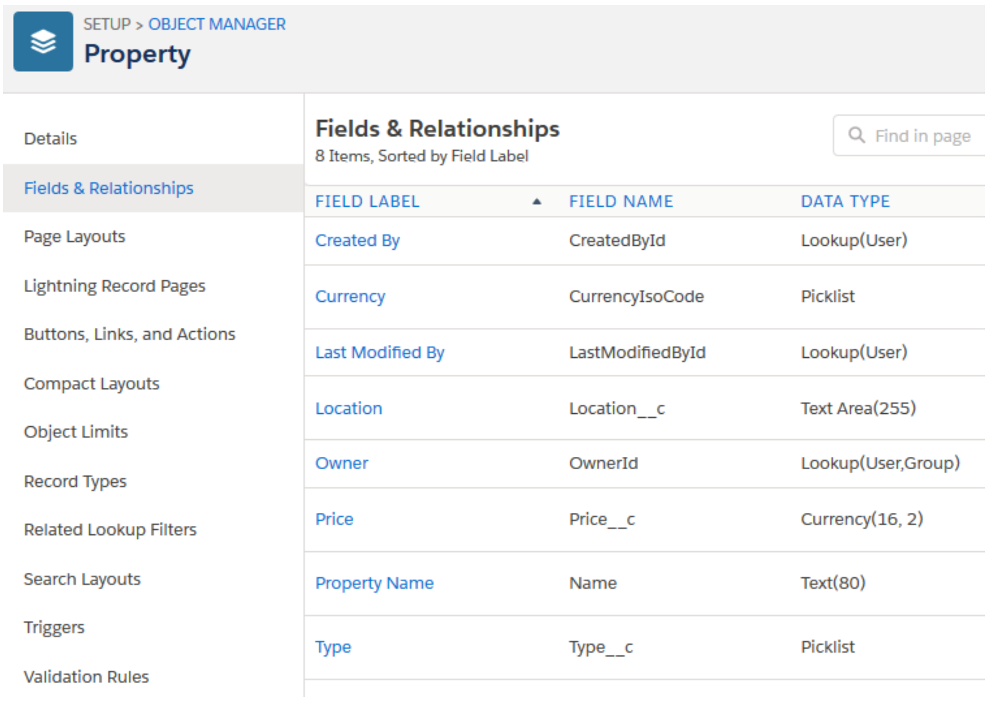
You need to import the XML file created in the previous section into a Development Team Drive. To do that, hold down on the package you have created in the 4.1 section.



Select [More] > [Filter Search Condition List] from the context menu. Tapping [Import] on the “Filter Search List” dialog brings a pop-up menu including “Browse”, “Read from iTunes”, and “Read from WebDAV”. Select one of the commands to navigate to the target place where you saved the XML file. When you tap the XML file, the search condition will be imported to your package.

5 Customization in Salesforce

This chapter describes how Salesforce components are customized.
To describe that, a custom object “Property” is used.
https://trailhead.salesforce.com/en/modules/data_modeling/units/objects_intro



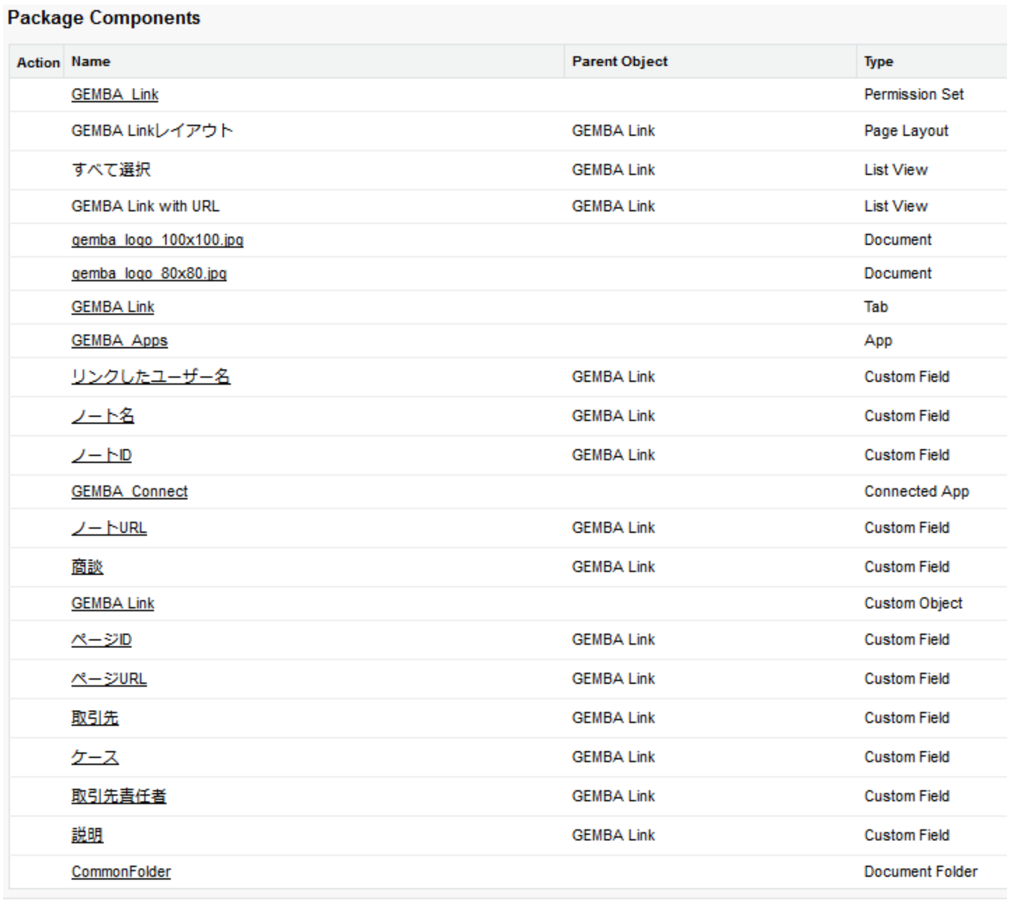
Fields & Relationships
8 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE
Created By	CreatedById	Lookup(User)
Currency	CurrencyIsoCode	Picklist
Last Modified By	LastModifiedById	Lookup(User)
Location	Location__c	Text Area(255)
Owner	OwnerId	Lookup(User,Group)
Price	Price__c	Currency(16, 2)
Property Name	Name	Text(80)
Type	Type__c	Picklist

Figure 15: Custom Object “Property” in Salesforce

5.1 Installing GEMBA Note package from AppExchange

The GEMBA Note package has been released from the AppExchange. Please install the app from the store. The installed components (*) are as follows:


Package Components

Action	Name	Parent Object	Type
	GEMBA_Link		Permission Set
	GEMBA Linkレイアウト	GEMBA Link	Page Layout
	すべて選択	GEMBA Link	List View
	GEMBA Link with URL	GEMBA Link	List View
	gemba_logo_100x100.jpg		Document
	gemba_logo_80x80.jpg		Document
	GEMBA Link		Tab
	GEMBA_Apps		App
	リンクしたユーザー名	GEMBA Link	Custom Field
	ノート名	GEMBA Link	Custom Field
	ノートID	GEMBA Link	Custom Field
	GEMBA_Connect		Connected App
	ノートURL	GEMBA Link	Custom Field
	商談	GEMBA Link	Custom Field
	GEMBA Link		Custom Object
	ページID	GEMBA Link	Custom Field
	ページURL	GEMBA Link	Custom Field
	取引先	GEMBA Link	Custom Field
	ケース	GEMBA Link	Custom Field
	取引先責任者	GEMBA Link	Custom Field
	説明	GEMBA Link	Custom Field
	CommonFolder		Document Folder

(*) You can see this list from [Setup] > [Installed Packages] > [GEMBA] > [View Components].

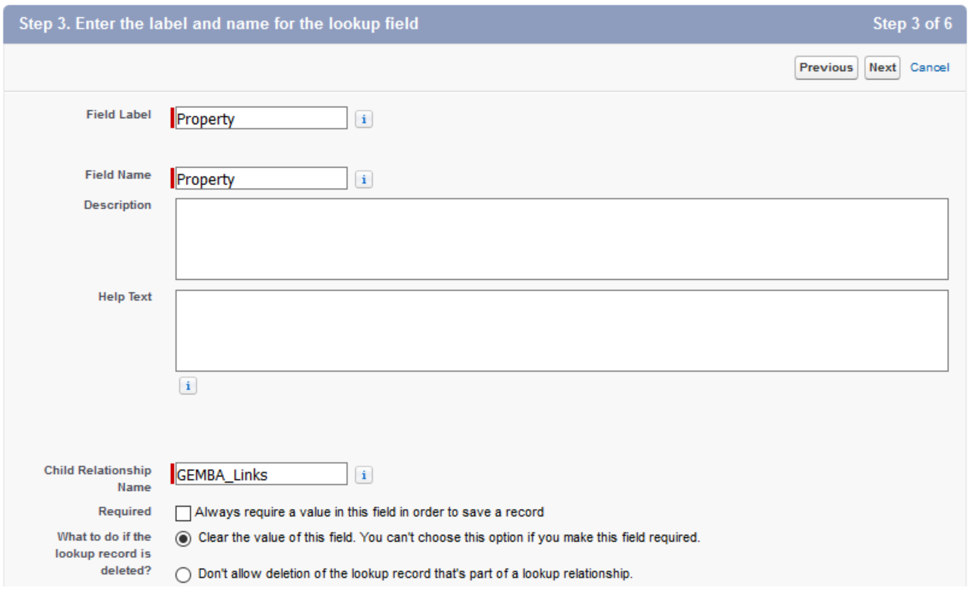
5.2 Creating a Relationship from GEMBA Link

This operation is required when using objects other than standard objects such as Account, Case, Contact, and Opportunity.

The following steps show how to add a reference field for a custom object “Property.”

1. Tap ⚙ > [Setup] on the Salesforce screen.
2. Open [Object Manage] from the tap.
3. Select [GEMBA Link] from the list of available objects.
4. Click [Fields & Relationships] on the side bar.
5. Tap [New] and choose [Lookup Relationship] as the data type.
6. Tap [Next] at the top or bottom of the screen.
7. Select [Property] to which the Property object is related and click [Next].
8. Specify “Property” for the Field Label and Field Name and “GEMBA_Links” as

the Child Relationship Name, then click [Next].

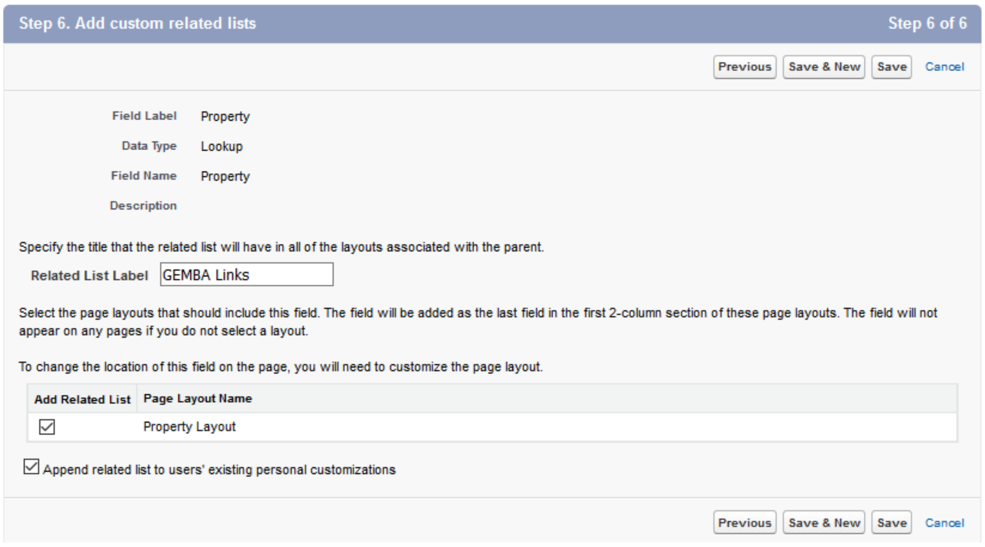


Step 3. Enter the label and name for the lookup field — Step 3 of 6

Previous | Next | Cancel

Field Label	Property
Field Name	Property
Description	
Help Text	
Child Relationship Name	GEMBA_Links
Required	☐ Always require a value in this field in order to save a record
What to do if the lookup record is deleted?	● Clear the value of this field. You can't choose this option if you make this field required.
	○ Don't allow deletion of the lookup record that's part of a lookup relationship.

9. Click [Next] (Step 4. Establish field-level security for reference field) and [Next] (Step 5. Add reference field to Page Layouts)
10. Confirm whether the Related List Label is set to “GEMBA Links”, click [Save].



Step 6. Add custom related lists — Step 6 of 6

Previous | Save & New | Save | Cancel

Field Label	Property
Data Type	Lookup
Field Name	Property
Description	

Specify the title that the related list will have in all of the layouts associated with the parent.

Related List Label: GEMBA Links

Select the page layouts that should include this field. The field will be added as the last field in the first 2-column section of these page layouts. The field will not appear on any pages if you do not select a layout.

To change the location of this field on the page, you will need to customize the page layout.

Add Related List	Page Layout Name
✓	Property Layout

✓ Append related list to users' existing personal customizations

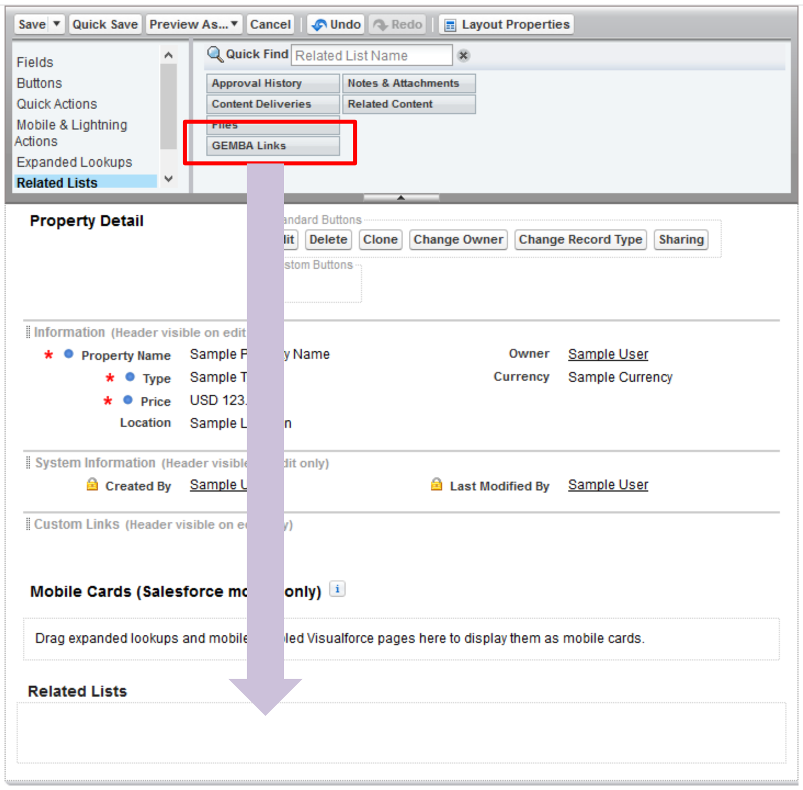
Previous | Save & New | Save | Cancel

The Property field (the name is *Property*__c) is added to the object.

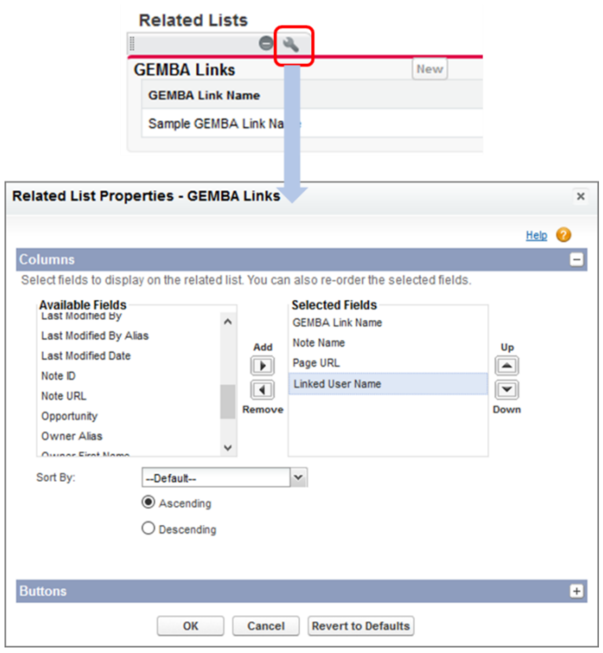
5.3 Setting Up Layout of the Target Object

The GEMBA Note for Salesforce package includes a custom object named GEMBA Link. The object can refer to any other standard and custom objects in Salesforce.
For example, if you need to have the link function for a custom object "Property", the following steps show how to modify a page layout.

1. Tap > [Setup] on the Salesforce screen.
2. Open [Object Manage] from the tap.
3. Go to [Property] > [Page Layouts] > [Property Layout].
4. Select [Related Lists] on the sub-window at the top of the layout screen.
   You can find [GEMBA Links].
5. Drag [GEMBA Links] and drop it to the Related Lists area (see the figure below).



6. Tap the wrench icon to open the Related List Properties of GEMBA Links.


Related Lists

GEMBA Links

GEMBA Link Name

Sample GEMBA Link Na

Related List Properties - GEMBA Links

Columns

Select fields to display on the related list. You can also re-order the selected fields.

Available Fields: Last Modified By, Last Modified By Alias, Last Modified Date, Note ID, Note URL, Opportunity, Owner Alias, Owner First Name

Selected Fields: GEMBA Link Name, Note Name, Page URL, Linked User Name

Sort By: --Default--

[✓ Ascending] Descending

7. Move “Note Name”, “Page URL”, and “Linked User Name” from the Available Fields to the Selected Fields and tap the OK button.
8. Tap [Save] to store the property layout.

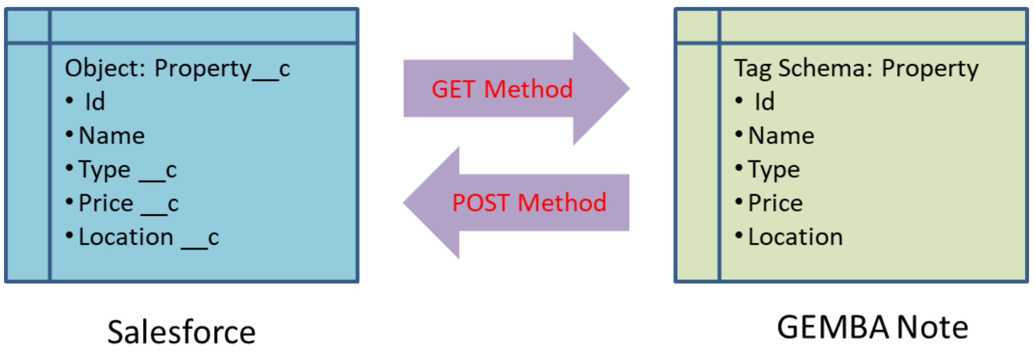
6 Developing HTTP Methods for a RESTful Service

GEMBA Note for Salesforce works with a web service using REST as described in the section 2.3.

We assume that specific web services will be designed and implemented by system integrators to access data as required for individual installations. At MetaMoJi we have developed the service utilizing Node-RED (https://nodered.org/). In this chapter, we show you two samples based on Node-RED.

6.1 Why is a RESTful Service necessary?

As you implemented the Property tag schema in GEMBA Note and the Property custom object in Salesforce, the schemata are different in both systems.



The RESTful service is used to map different item names and their values from Salesforce to GEMBA Note (GET methods) and vice versa (POST methods).

6.2 What is Node-RED?

Node-RED is a programming tool for wiring together hardware devices, APIs and online services. It provides a browser-based editor that makes it easy to wire together flows using a variety of nodes in the palette that can be deployed to its runtime.

The programming in Node-RED is to drag and drop nodes, set information for each node, write some Java Script code, and connect nodes together with wires.

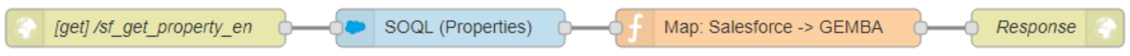
Here are basic nodes that are used for the two samples:

Node	Node Icon	Description
HTTP IN	http	Creates an HTTP end-point for creating web services.
HTTP OUT (response)	http response	Sends responses back to requests received from an HTTP Input node.
Function	function	A JavaScript function block to run against the messages being received by the node.
SOQL	soql	Executes an SOQL query. Library: node-red-contrib-salesforce (*)
DML	dml	Executes an insert, update, upsert or delete DML statement. Library: node-red-contrib-salesforce (*)

(*1) https://flows.nodered.org/node/node-red-contrib-salesforce

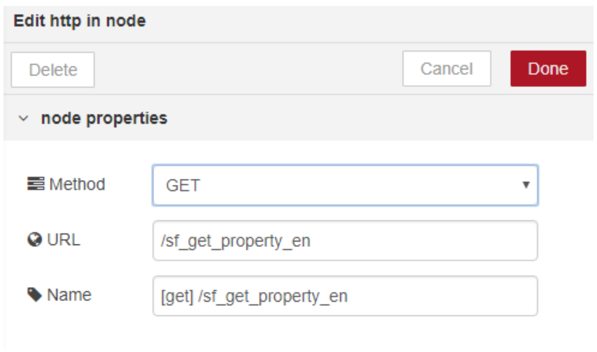
6.3 Implementing a GET Method

To get the Property objects (Property__c) from Salesforce, an HTTP IN node, an SOQL node, a Function node, and an HTTP OUT node are wired as follows:



[1] [get] /sf_get_property_en (HTTP IN)

The HTTP end point is specified as */sf_get_property_en*.

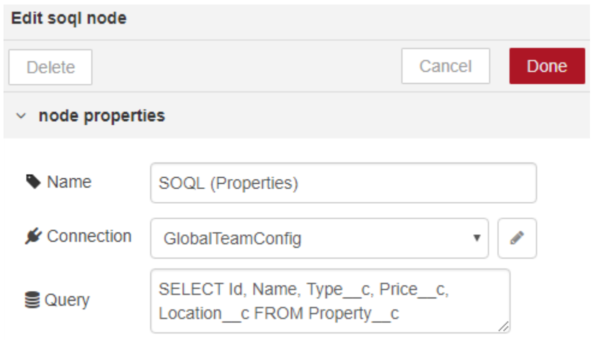


Method: GET
URL: /sf_get_property_en
Name: [get] /sf_get_property_en

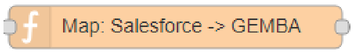
MetaMoJi

[2] (SOQL)

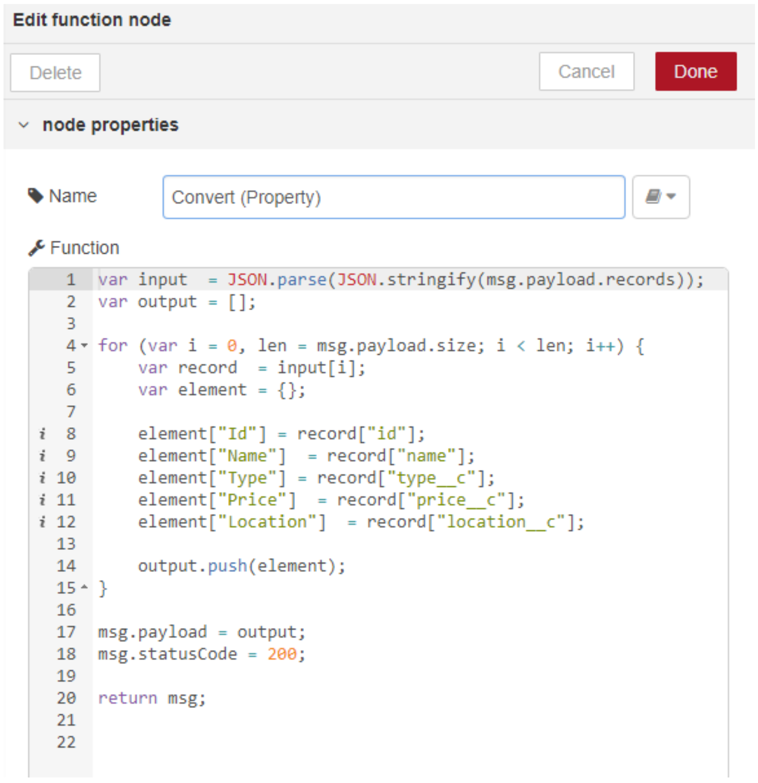
An SOQL statement
"SELECT Id, Name, Type__c, Price__c, Location__c FROM Property__c"
is specified to the Query field.



Name: SOQL (Properties)
Connection: GlobalTeamConfig
Query: SELECT Id, Name, Type__c, Price__c, Location__c FROM Property__c

[3]  (Function)

This function alters a JSON structure from Salesforce to the corresponding JSON structure for GEMBA Note.



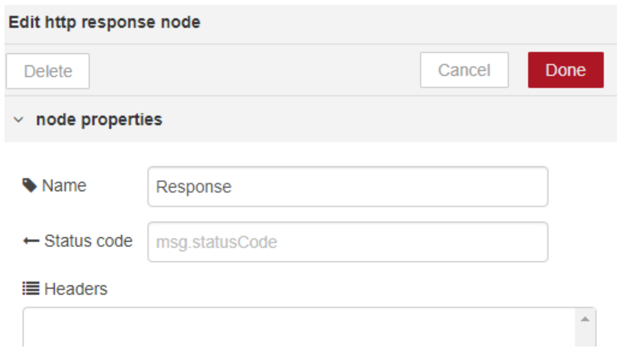
Name: Convert (Property)

Function

```
var input  = JSON.parse(JSON.stringify(msg.payload.records));
var output = [];

for (var i = 0, len = msg.payload.size; i < len; i++) {
    var record  = input[i];
    var element = {};

    element["Id"] = record["id"];
    element["Name"]  = record["name"];
    element["Type"] = record["type__c"];
    element["Price"]  = record["price__c"];
    element["Location"]  = record["location__c"];

    output.push(element);
}

msg.payload = output;
msg.statusCode = 200;

return msg;
```

[4] (HTTP OUT)

Only the name of the node is specified.



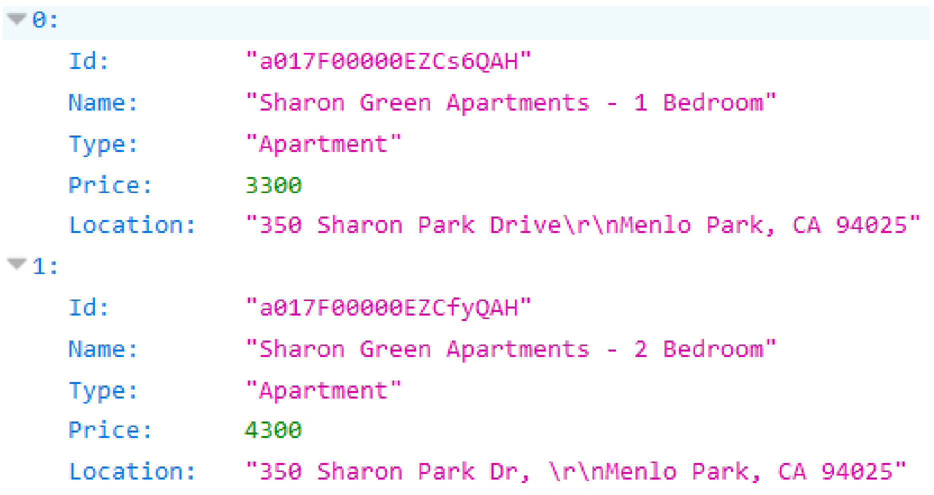
Edit http response node

Name: Response

Status code: msg.statusCode

Headers

A sample URL http://mps101a-nodered-beta.metamoji.com:1880/sf_get_property_en returns a JSON structure on a web browser as follows:

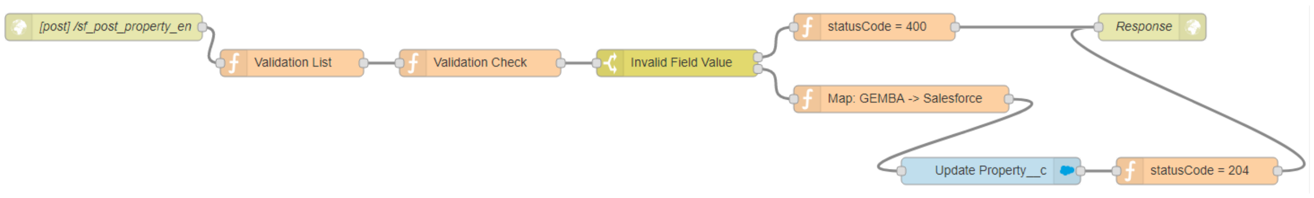


```
▼0:
    Id:          "a017F00000EZCs6QAH"
    Name:        "Sharon Green Apartments - 1 Bedroom"
    Type:        "Apartment"
    Price:       3300
    Location:    "350 Sharon Park Drive\r\nMenlo Park, CA 94025"
▼1:
    Id:          "a017F00000EZCfyQAH"
    Name:        "Sharon Green Apartments - 2 Bedroom"
    Type:        "Apartment"
    Price:       4300
    Location:    "350 Sharon Park Dr, \r\nMenlo Park, CA 94025"
```

MetaMoJi

6.4 Implementing a POST Method

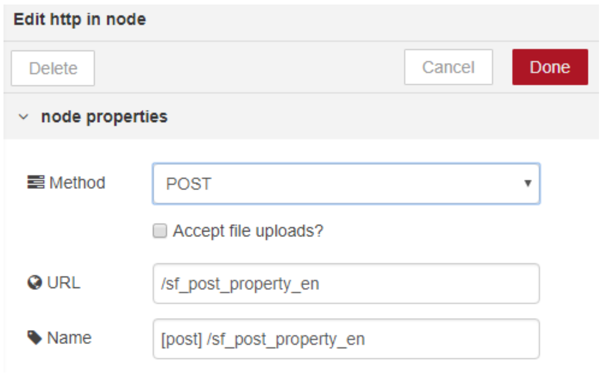
This method is embedded to a Command Button, a form control of the GEMBA Note appl. It is used to put values from GEMBA Note to Salesforce. The flow figure is as follows:



[1]

(HTTP IN)

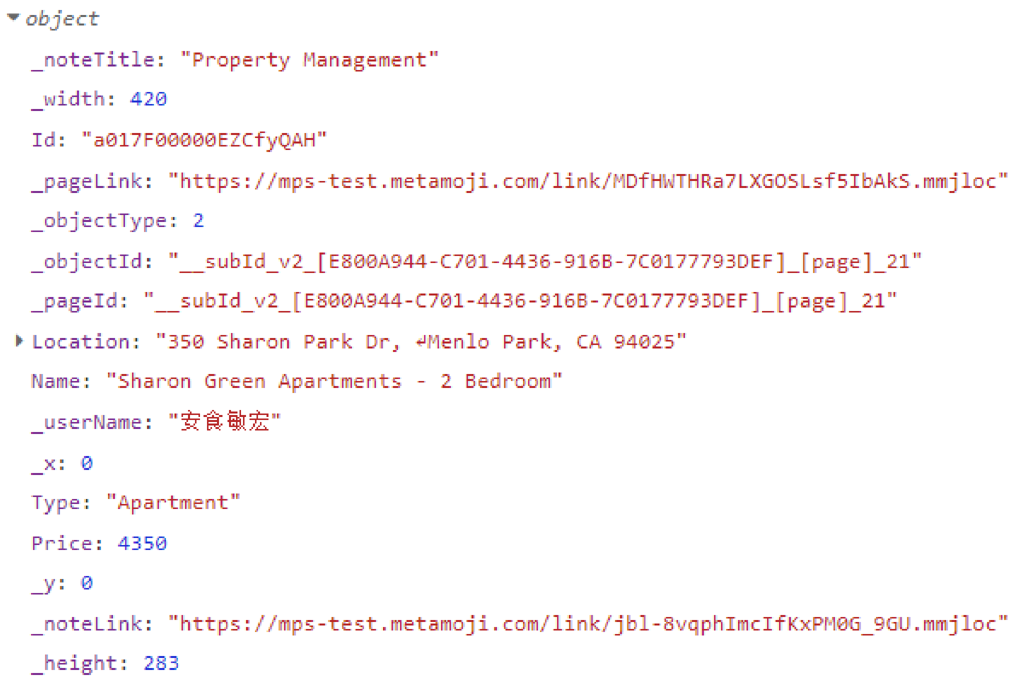
The HTTP end point is specified as /sf_post_property_en.



Method: POST
Accept file uploads?
URL: /sf_post_property_en
Name: [post] /sf_post_property_en

This node obtains a JSON structure from GEMBA Note. The following figure shows a sample structure of the JSON data:

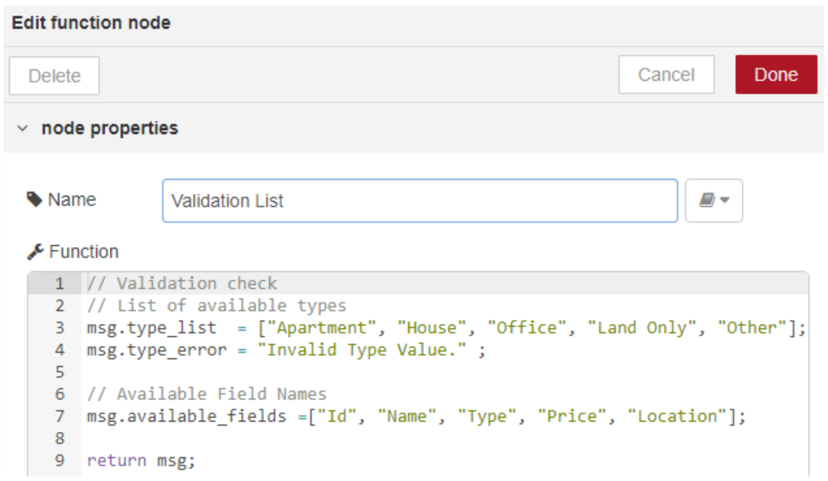
MetaMoJi



```
▾object
   _noteTitle: "Property Management"
   _width: 420
   Id: "a017F00000EZCfyQAH"
   _pageLink: "https://mps-test.metamoji.com/link/MDfHWTHRa7LXGOSLsf5IbAkS.mmjloc"
   _objectType: 2
   _objectId: "__subId_v2_[E800A944-C701-4436-916B-7C0177793DEF]_[page]_21"
   _pageId: "__subId_v2_[E800A944-C701-4436-916B-7C0177793DEF]_[page]_21"
 ▸ Location: "350 Sharon Park Dr, ↵Menlo Park, CA 94025"
   Name: "Sharon Green Apartments - 2 Bedroom"
   _userName: "安食敏宏"
   _x: 0
   Type: "Apartment"
   Price: 4350
   _y: 0
   _noteLink: "https://mps-test.metamoji.com/link/jbl-8vqphImcIfKxPM0G_9GU.mmjloc"
   _height: 283
```

[2] Validation List (Function)

The node includes the list of available type names, an error message, and available field names that are used in the next node.



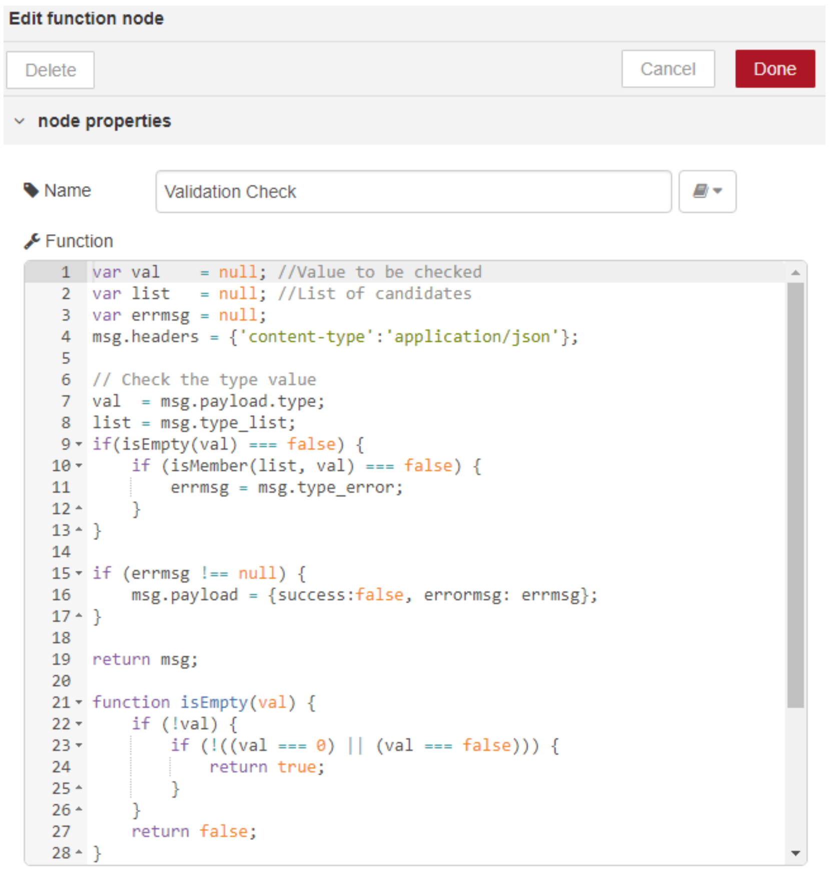
Name: Validation List

Function

```
// Validation check
// List of available types
msg.type_list  = ["Apartment", "House", "Office", "Land Only", "Other"];
msg.type_error = "Invalid Type Value." ;

// Available Field Names
msg.available_fields =["Id", "Name", "Type", "Price", "Location"];

return msg;
```

[3] Validation Check (Function)

The value of the Type field is checked whether it is valid with referring to msg.type_list.



Name: Validation Check

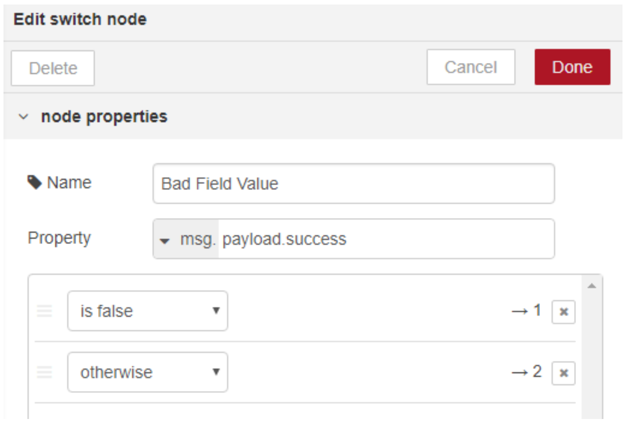
Function

```
var val    = null; //Value to be checked
var list   = null; //List of candidates
var errmsg = null;
msg.headers = {'content-type':'application/json'};

// Check the type value
val  = msg.payload.type;
list = msg.type_list;
if(isEmpty(val) === false) {
    if (isMember(list, val) === false) {
        errmsg = msg.type_error;
    }
}

if (errmsg !== null) {
    msg.payload = {success:false, errormsg: errmsg};
}

return msg;

function isEmpty(val) {
    if (!val) {
        if (!((val === 0) || (val === false))) {
            return true;
        }
    }
    return false;
}
```

MetaMoJi

[4] (Switch)

If any invalid values exist, go to the node 1. Else go to the node 2.



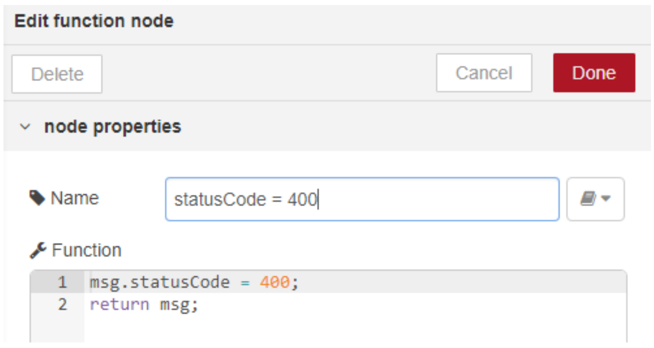
Name: Bad Field Value

Property: msg. payload.success

is false	→ 1
otherwise	→ 2

[5] (Function)

The statusCode for the HTTP out is set to 400.



Name: statusCode = 400

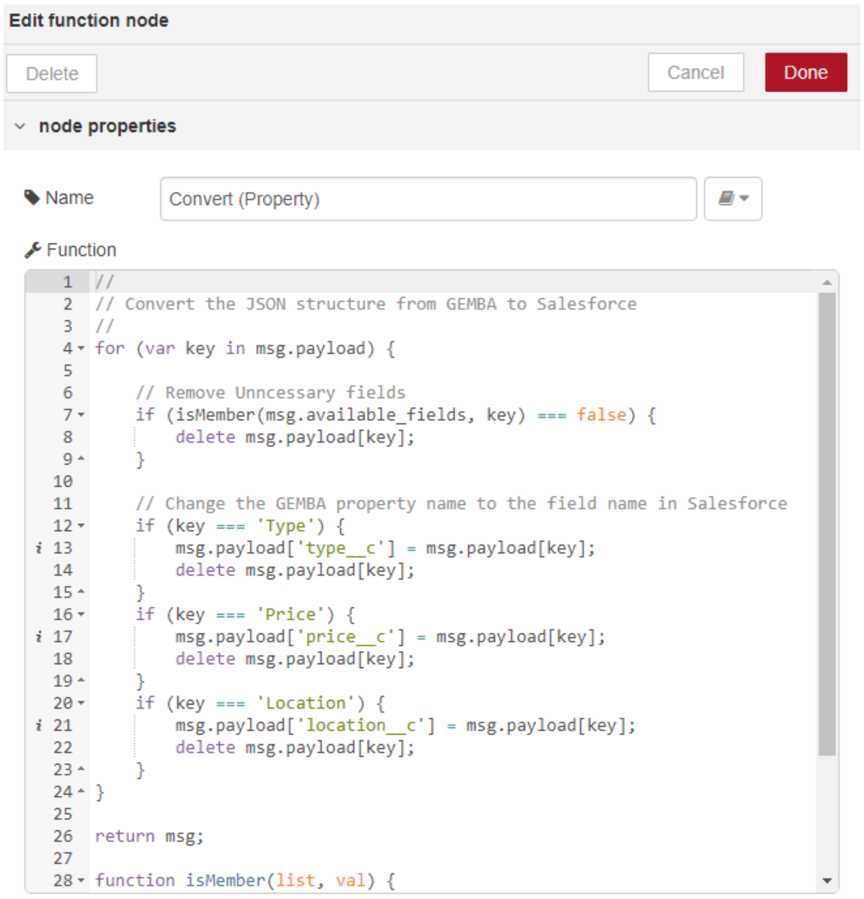
Function

```
msg.statusCode = 400;
return msg;
```

MetaMoJi

[6] Map: GEMBA -> Salesforce (Function)

Put values from GEMBA Note to corresponding fields in Salesforce.



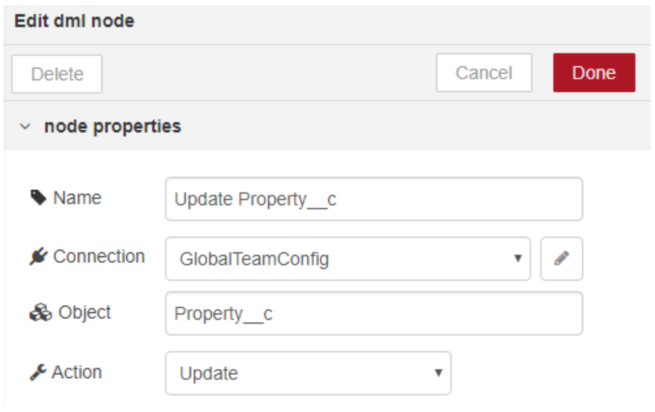
Name: Convert (Property)

Function

```
//
// Convert the JSON structure from GEMBA to Salesforce
//
for (var key in msg.payload) {

    // Remove Unncessary fields
    if (isMember(msg.available_fields, key) === false) {
        delete msg.payload[key];
    }

    // Change the GEMBA property name to the field name in Salesforce
    if (key === 'Type') {
        msg.payload['type__c'] = msg.payload[key];
        delete msg.payload[key];
    }
    if (key === 'Price') {
        msg.payload['price__c'] = msg.payload[key];
        delete msg.payload[key];
    }
    if (key === 'Location') {
        msg.payload['location__c'] = msg.payload[key];
        delete msg.payload[key];
    }
}

return msg;

function isMember(list, val) {
```

[7] Update Property__c (DML)

Update the values of the Property__c instance in Salesforce.



Name: Update Property__c

Connection: GlobalTeamConfig

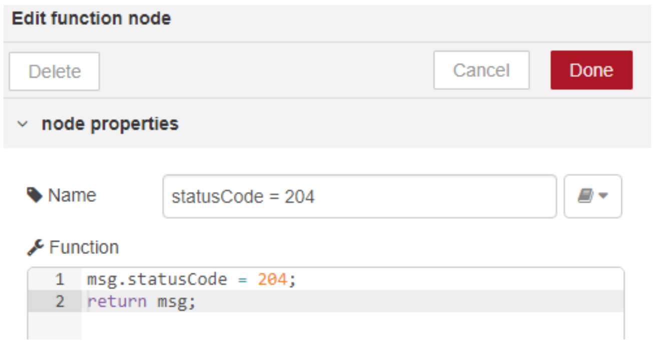
Object: Property__c

Action: Update

[8]

(Function)

The statusCode for the HTTP out is set to 204.



Name: statusCode = 204

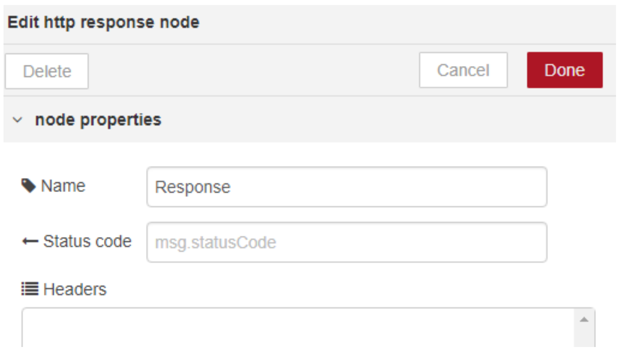
Function

```
msg.statusCode = 204;
return msg;
```

[9]

(HTTP OUT)

Only the name of the note is specified.



Name: Response

Status code: msg.statusCode

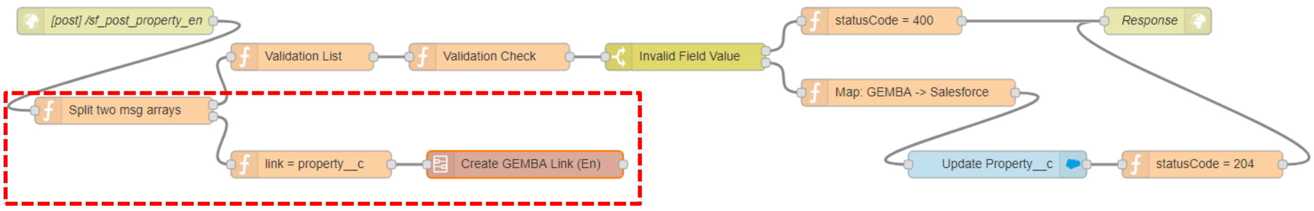
Headers

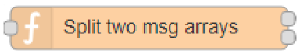
If the object is successfully updated, the node will return a JSON structure as follows:

```
{ success: true, object: "property__c", action: "update" }
```

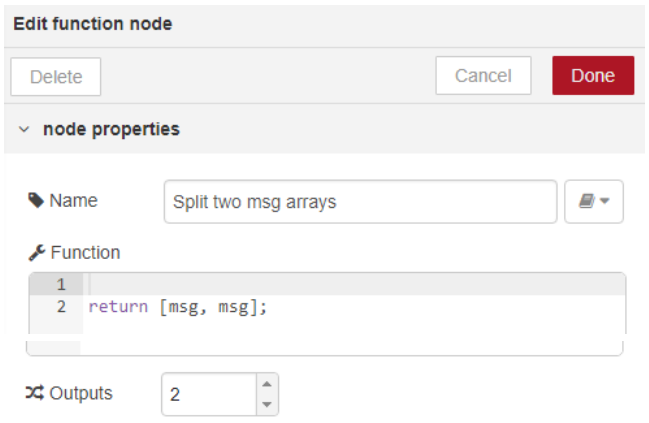
6.5 Generating a GEMBA Link

The flow described in the previous section does not include the function to generate a GEMBA Link object in Salesforce.



[1]  (Function)

The msg structure is forwarded to the next two nodes so that the structure cannot be modified.


Edit function node

Delete | Cancel | Done

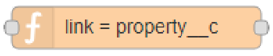
node properties

Name: Split two msg arrays

Function

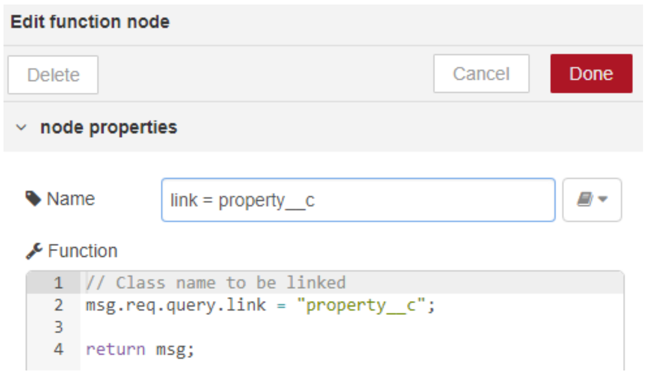
```

return [msg, msg];
```

Outputs: 2

[2]  (Function)

The msg.req.query.link element is set to “property__c”, a target Salesforce class name. It is used to connect the Property__c instance and the GEMBA_Link instance in Salesforce.

MetaMoJi



Name: link = property__c

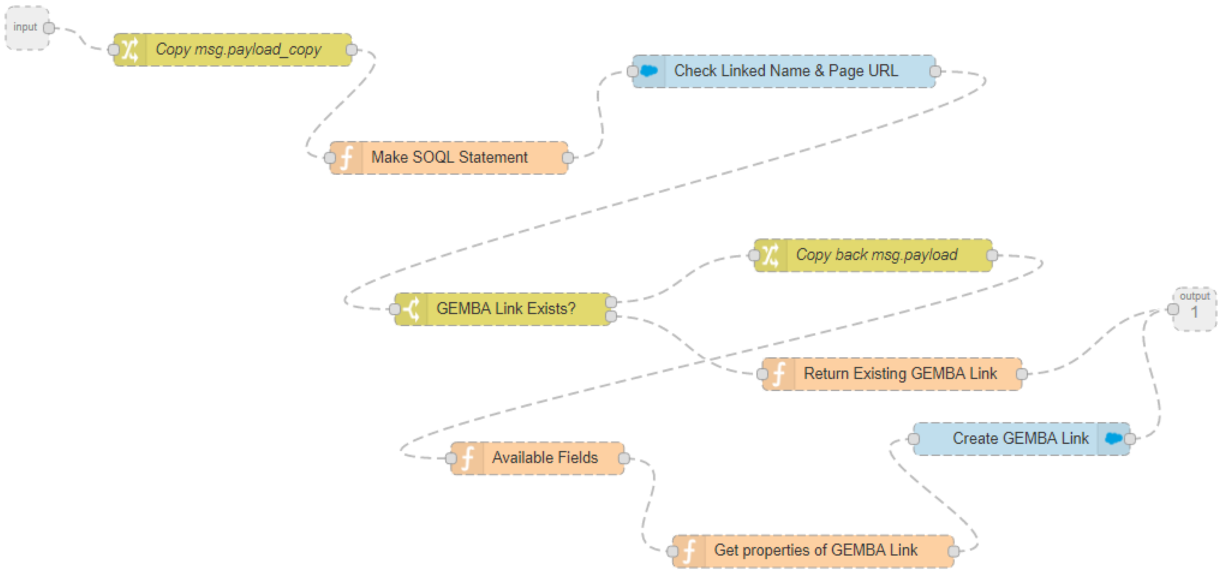
Function

```
// Class name to be linked
msg.req.query.link = "property__c";

return msg;
```

[3]

(Sub flow)

The node expands to the following flow:



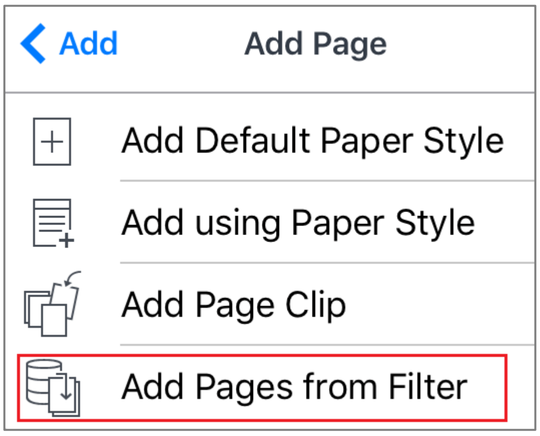
This sub-flow generates a GEMBA_Link instance in Salesforce.

7 Testing

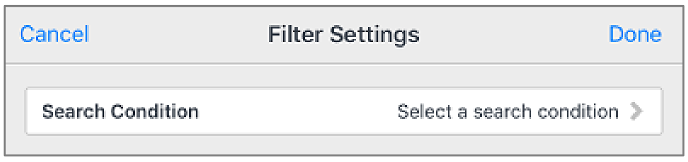
After you finish customizing components for GEMBA Note and Salesforce, you need to confirm whether your customization will work as you expected.

7.1 Importing Data from Salesforce

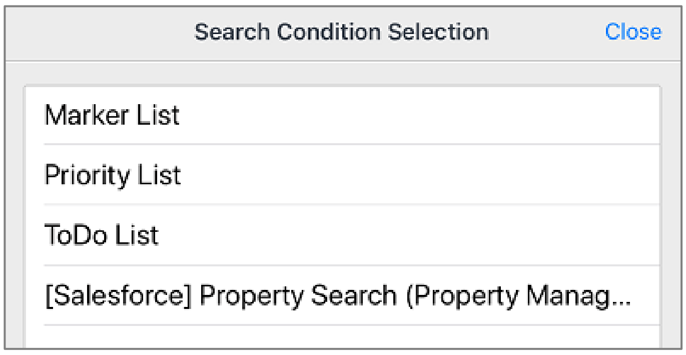
From the ＋ icon on top bar menu, you can add objects (pages, items, spreadsheet, shapes, etc.) into your GEMBA Note.
Importing data from Salesforce is implemented in [Add Page] > [Add Pages from Filter]:



1. Tap [Add Pages from Filter].
   You have the Filter Settings dialog:



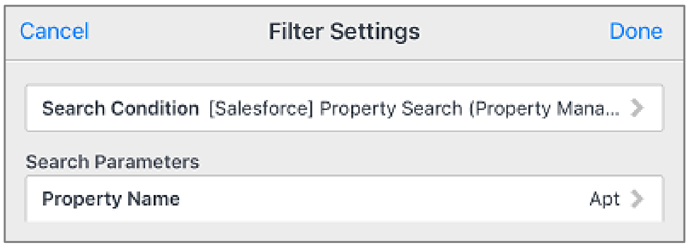
2. Tap [Search Condition] on the dialog.
   The Search Condition Selection dialog appears:


Search Condition Selection
Close

Marker List
Priority List
ToDo List
[Salesforce] Property Search (Property Manag...

Your created search condition names appears in the list.

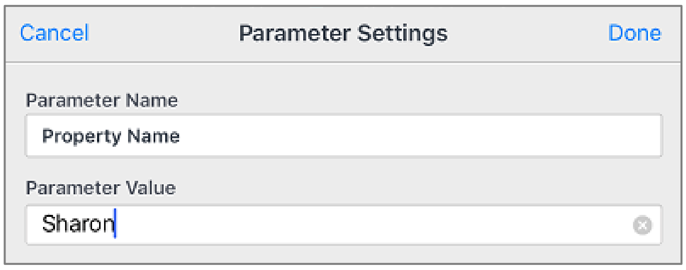
3. Choose [Salesforce] Property Search. You can see [Property Name] which is designed for parameter settings in the XML file.


Cancel Filter Settings Done

Search Condition [Salesforce] Property Search (Property Mana...

Search Parameters
Property Name Apt

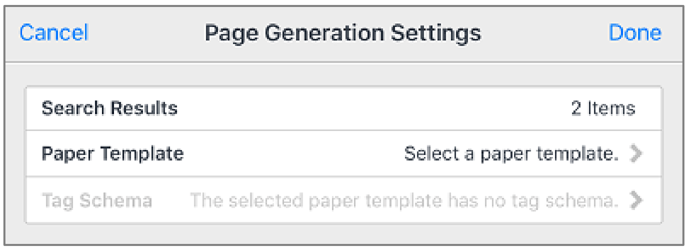
4. When you tap [Property Name], you are asked to specify the parameter value as follows:


Cancel Parameter Settings Done

Parameter Name
Property Name

Parameter Value
Sharon

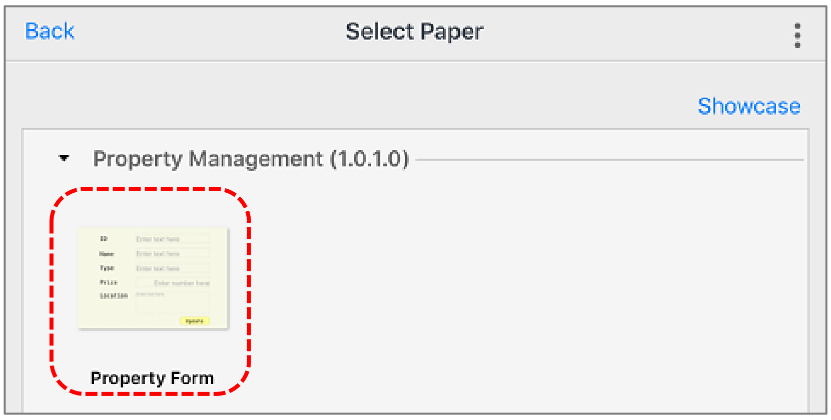
5. Tap the Done button on the Filter Settings dialog (see step 3). Now open the Page Generation Settings dialog.


Cancel Page Generation Settings Done

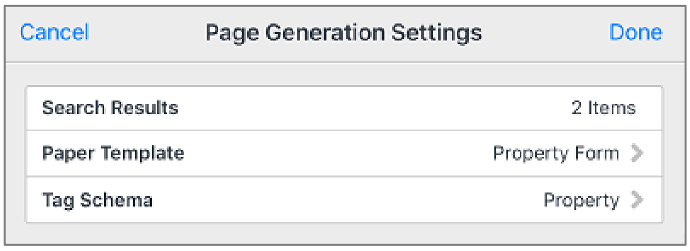
Search Results	2 Items
Paper Template	Select a paper template.
Tag Schema	The selected paper template has no tag schema.

[Search Results] shows the number of target Salesforce instances found with the search condition specified in the previous steps.

6. Tap [Paper Template] on the Page Addition Settings dialog.
   The Select Paper dialog allows you to choose a target paper template.

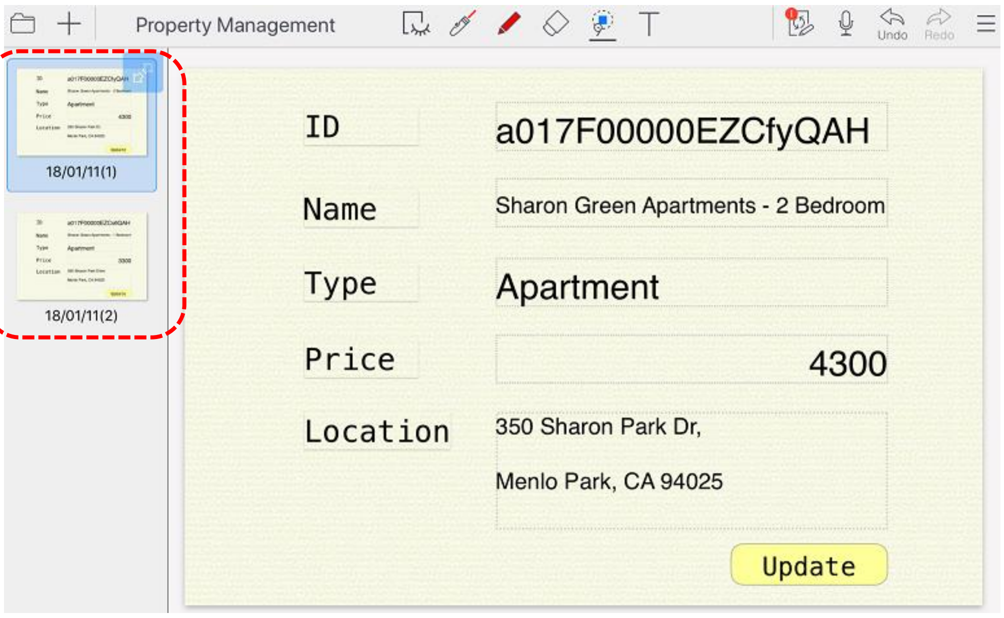


7. Select "Property Form" which has been created and registered.



Search Results	2 Items
Paper Template	Property Form
Tag Schema	Property

8. Tap the Done button on the Page Generation Settings dialog.
   Pages based on the selected paper template ("Property Form") will be generated as follows:



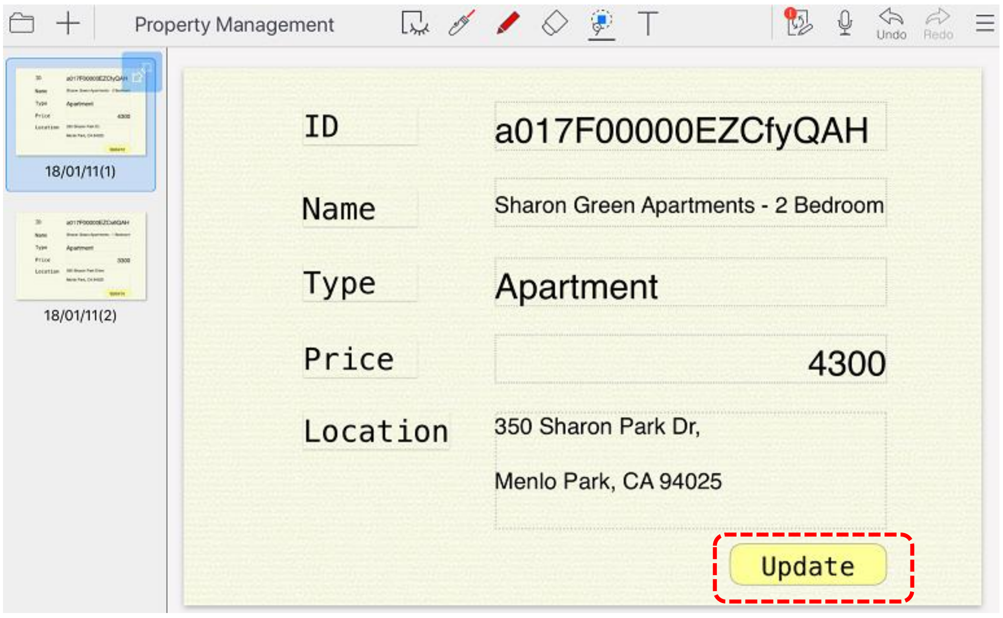
ID	a017F00000EZCfyQAH
Name	Sharon Green Apartments - 2 Bedroom
Type	Apartment
Price	4300
Location	350 Sharon Park Dr, Menlo Park, CA 94025

On the generated pages, you can draw, add text, web unit, photos, video, and so on.

MetaMoJi

7.2 Updating Salesforce Data from GEMBA Note

You can send data back to Salesforce from GEMBA Note.
Tap the Update button on a target generated page.



ID	a017F00000EZCfyQAH
Name	Sharon Green Apartments - 2 Bedroom
Type	Apartment
Price	4300
Location	350 Sharon Park Dr, Menlo Park, CA 94025

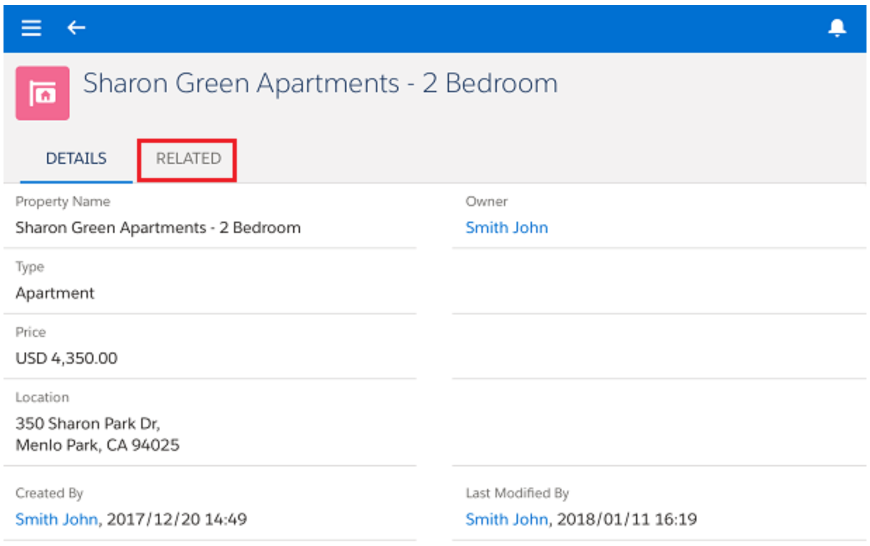
All field values are not updated. Each REST URL is designed for which field values are updated. On the Case Sheet template, you can update Status, Priority, Type, Reason, Origin, and Description. The specifications depend on what you implemented the HTTP POST method.

MetaMoJi

7.3 Opening a GEMBA Note Page from Salesforce

Using the Safari app on iOS, you can open a target GEMBA page from a Salesforce object.

1. Log in to a Salesforce account using the Safari app.
2. Navigate to a Salesforce object updated from the GEMBA Note app.

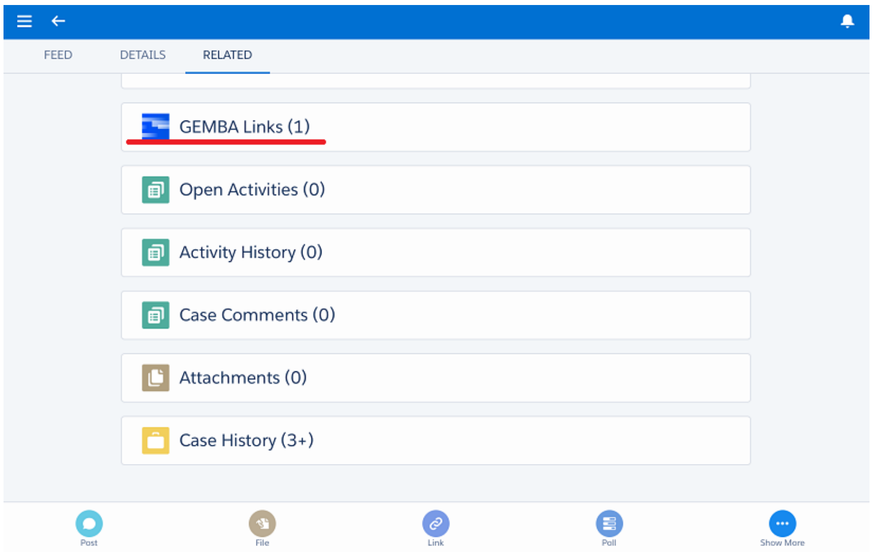


Sharon Green Apartments - 2 Bedroom

DETAILS RELATED

Property Name Sharon Green Apartments - 2 Bedroom	Owner Smith John
Type Apartment	
Price USD 4,350.00	
Location 350 Sharon Park Dr, Menlo Park, CA 94025	
Created By Smith John, 2017/12/20 14:49	Last Modified By Smith John, 2018/01/11 16:19

3. Tap the RELATED tab and scroll the window so that you can see [GEMBA Links].



FEED DETAILS RELATED

GEMBA Links (1)

Open Activities (0)

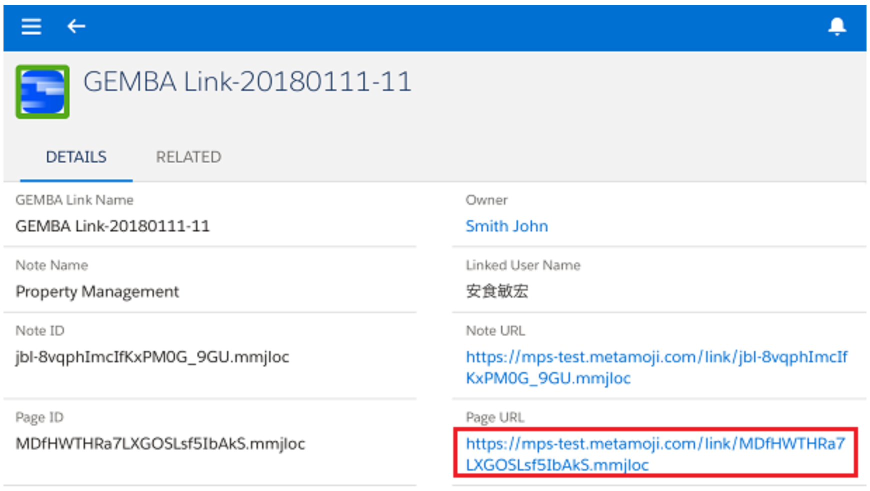
Activity History (0)

Case Comments (0)

Attachments (0)

Case History (3+)

4. Tap a target GEMBA Link from a list of GEMBA Link instances to open it.

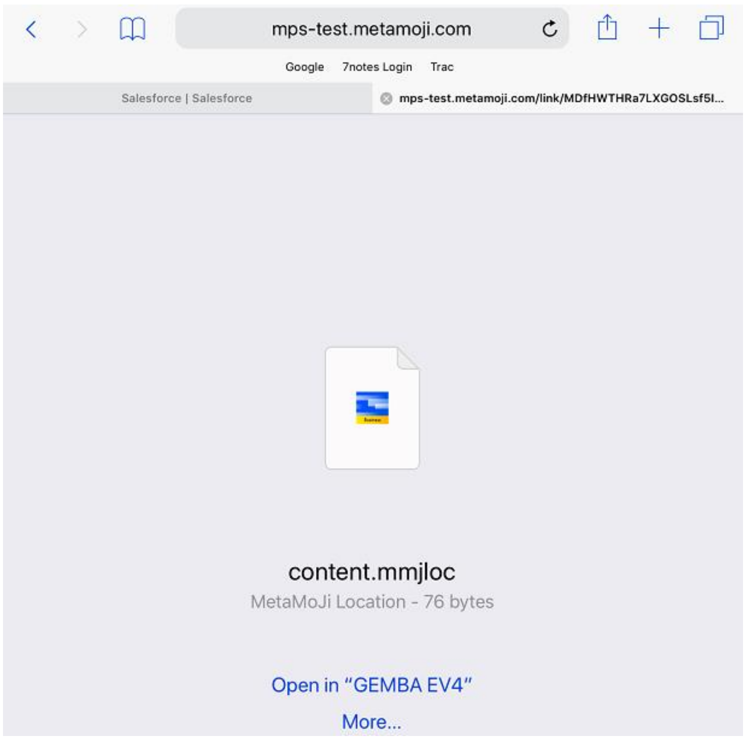


GEMBA Link-20180111-11

DETAILS RELATED

GEMBA Link Name GEMBA Link-20180111-11	Owner Smith John
Note Name Property Management	Linked User Name 安食敏宏
Note ID jbl-8vqphImcIfKxPM0G_9GU.mmjloc	Note URL https://mps-test.metamoji.com/link/jbl-8vqphImcIfKxPM0G_9GU.mmjloc
Page ID MDfHWTHRa7LXGOSLsf5IbAkS.mmjloc	Page URL https://mps-test.metamoji.com/link/MDfHWTHRa7LXGOSLsf5IbAkS.mmjloc

5. Tap the Page URL on the page:



mps-test.metamoji.com

Google 7notes Login Trac

Salesforce | Salesforce mps-test.metamoji.com/link/MDfHWTHRa7LXGOSLsf5I...

content.mmjloc

MetaMoJi Location - 76 bytes

6. On the Safari screen, tap [Open in GEMBA Note 4].
If you don't see the target app, tap [More…] to select the correct application name. The target GEMBA note will open at the correct page (*)
(*) If you delete the GEMBA note or page this cannot be opened.

MetaMoJi

Update History

Date	Description
January 11, 2018	First draft