iParse

iParse.io Introduction

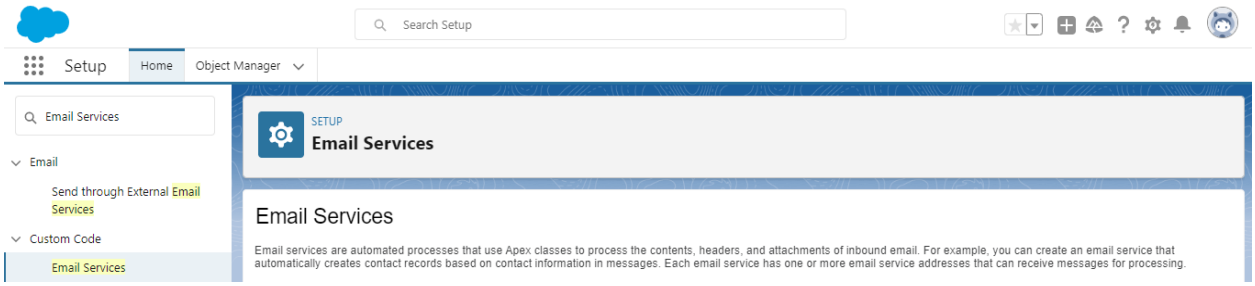
iParse is a tool designed to save you time extracting information from your documents. Once extracted, you can choose what to do with the data.

Setup Salesforce Email Service

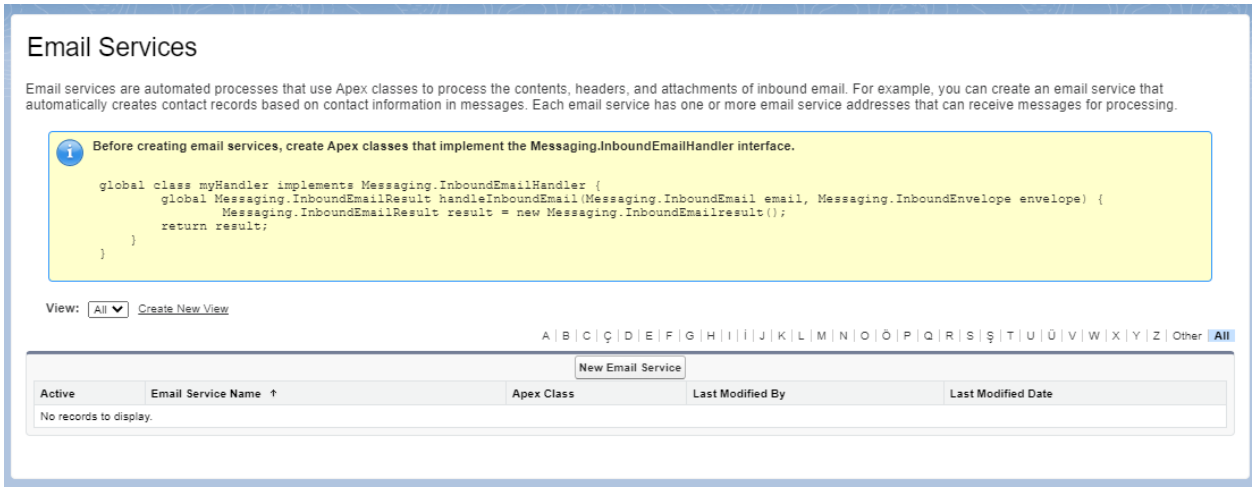
When you have set up your email service and sample document parsing templates, you can simply forward your documents to your email service address and iparse.io will do the rest.

To setup the Salesforce Email Service

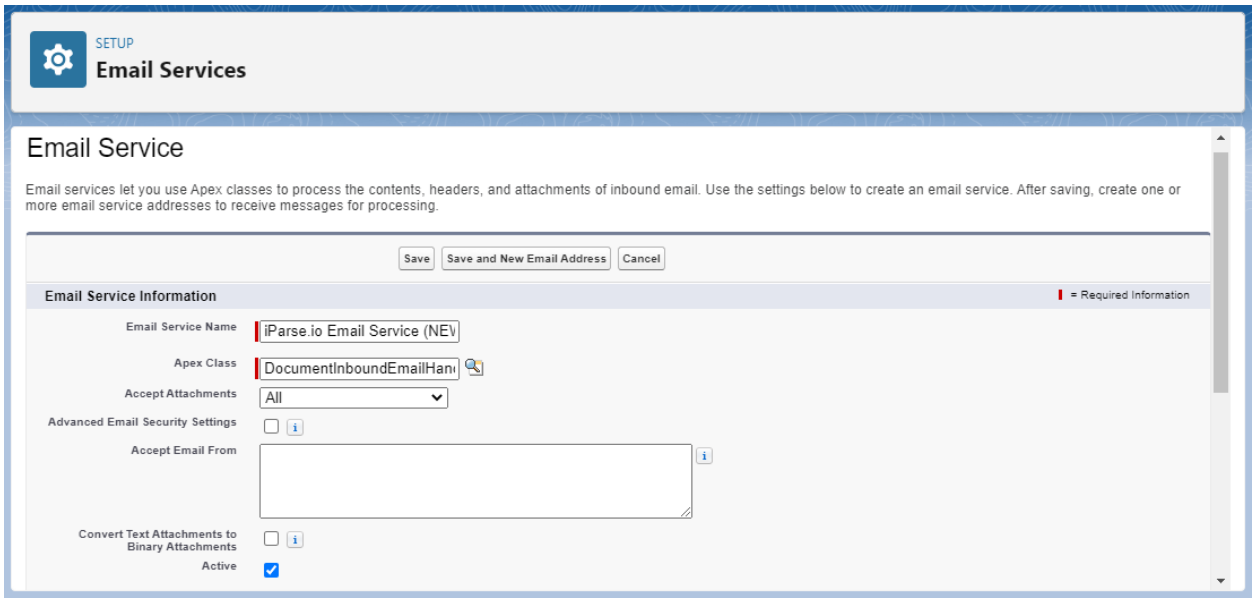
1. Click **'Setup'**
2. Search for **'Email Services'**



3. Click the **'New Email Service'** button

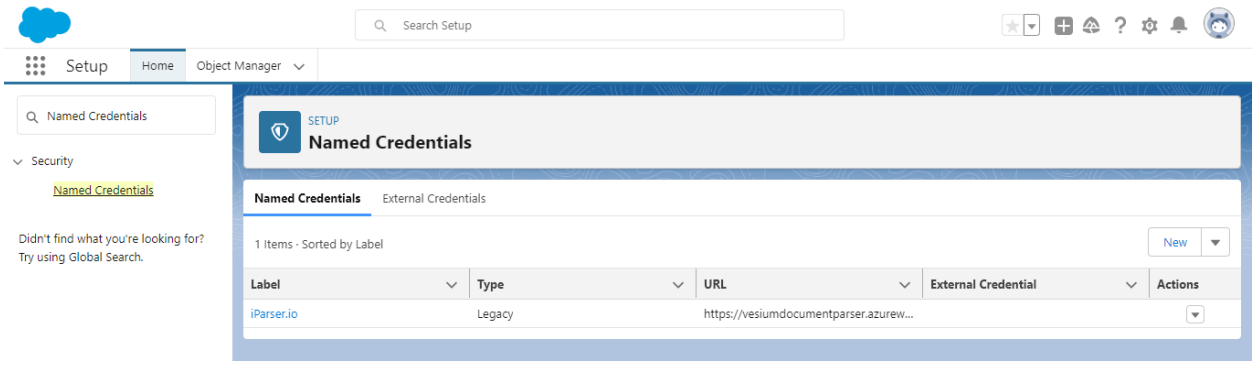


4. Add Service Name
5. Select Apex Class : **DocumentInboundEmailHandler**
6. Select Accept Attachments : **All**
7. Make **Active**
8. Click **Save**

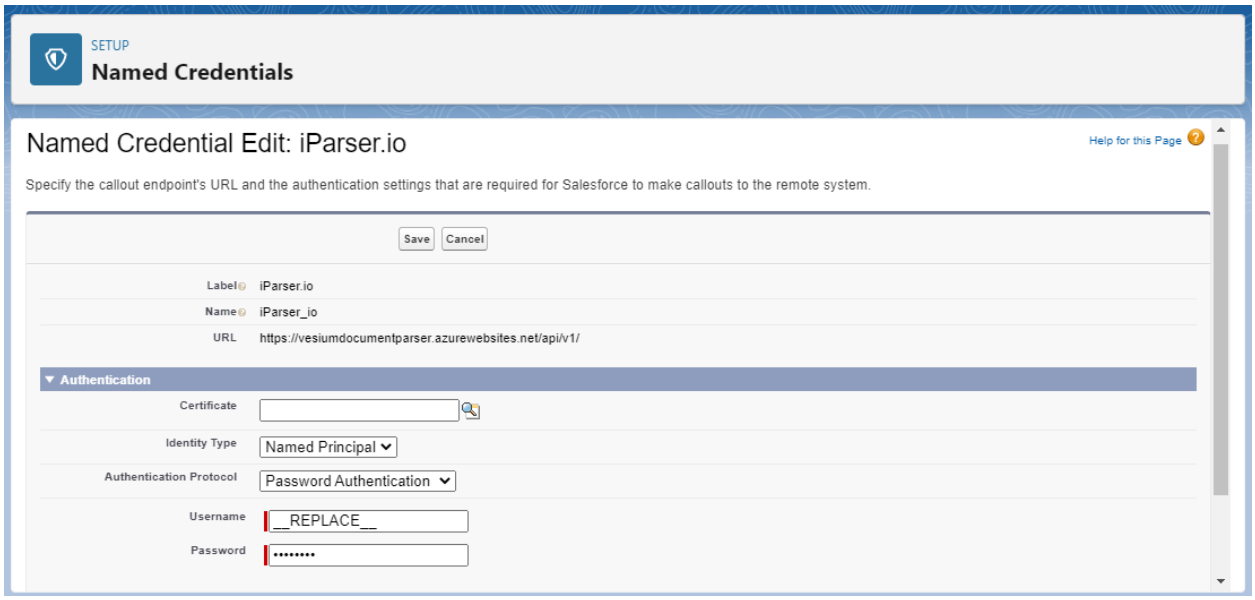


Setup Salesforce Named Credential

1. Click **'Setup'**
2. Search for **'Named Credentials'**



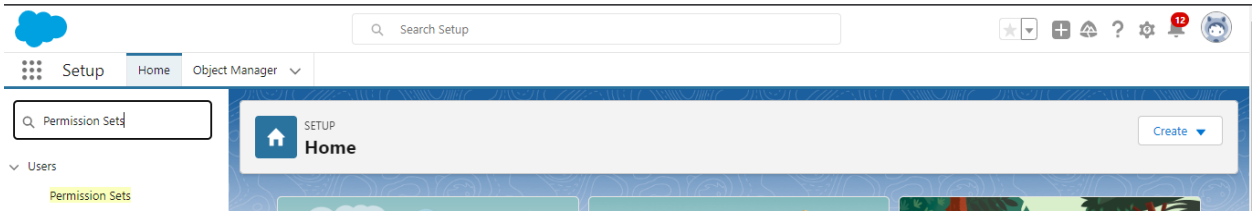
3. Edit the already created **‘iParser.io’** credential to add your **Username** and **Password** and hit save.



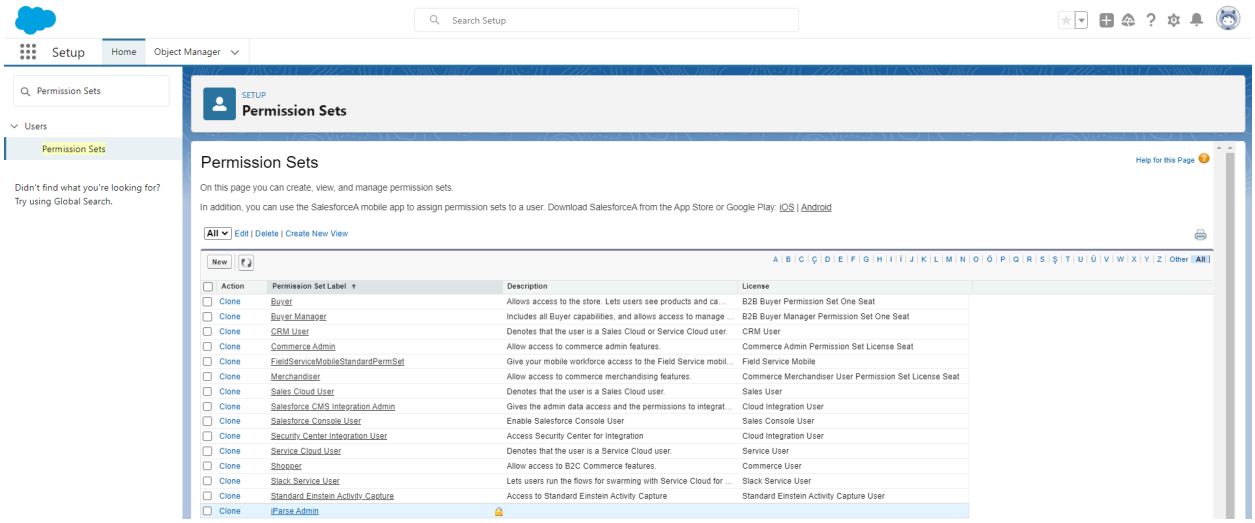
Assign Permission Sets

To Assign a Permission Set

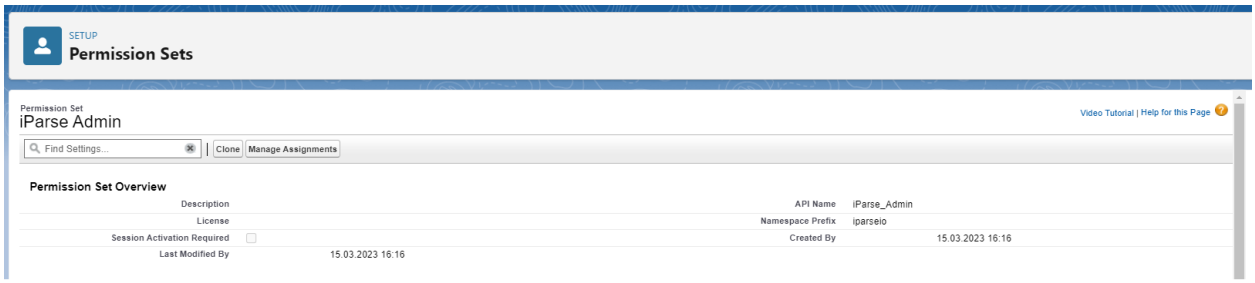
1. Click **‘Setup’**
2. Search **‘Permission Sets’** in the Quick Find search bar and Click **‘Permission Sets’**



3. Click **‘iParse Admin’** under Permission Set Label

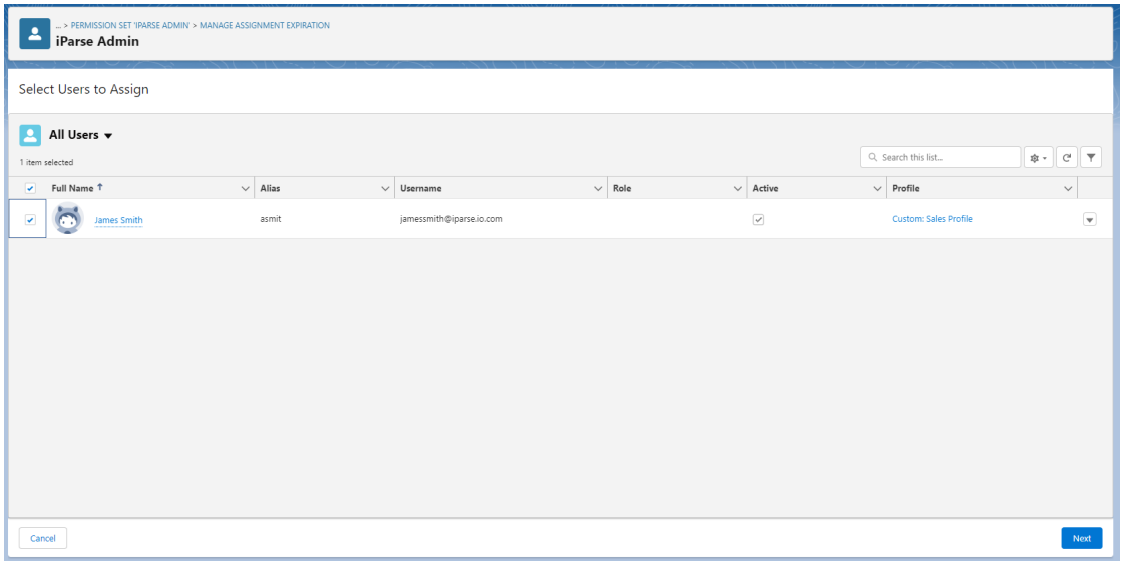


4. Click **‘Manage Assignments’**



5. Click **‘Add Assignment’**

6. Select user(s) and click **‘Next’**

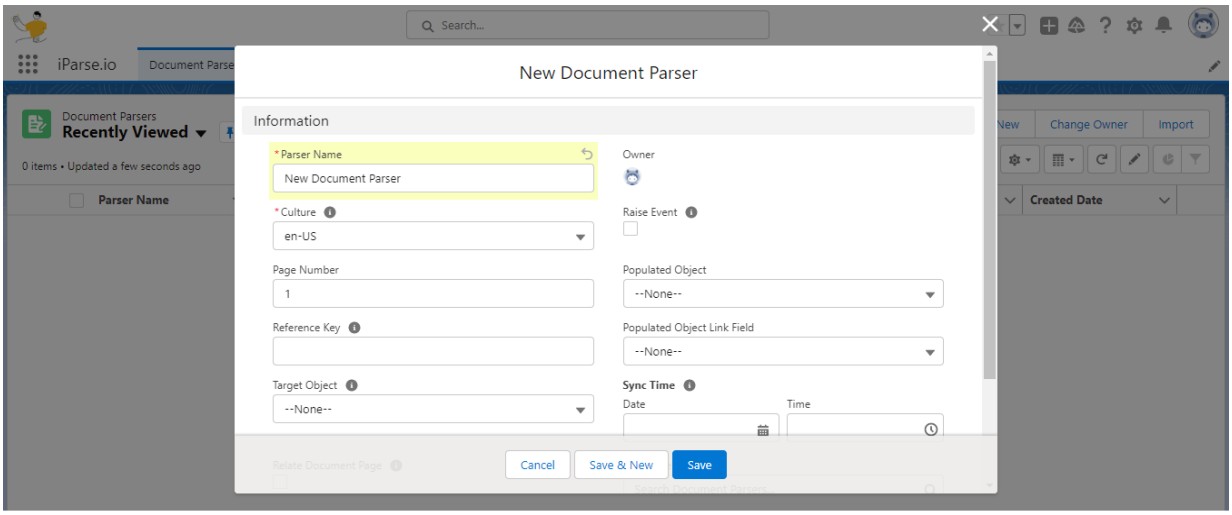


7. Click **‘Assign’** and click **‘Done’**

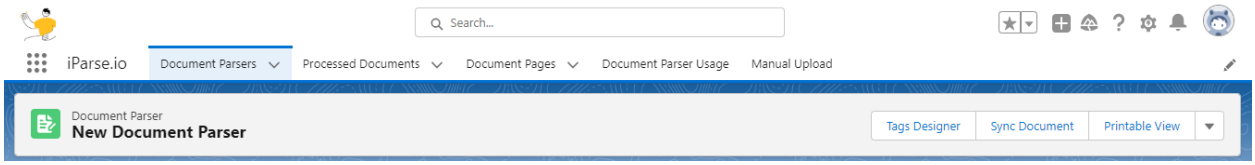
Create your first iParse document sample

Step 1: To create a new iParse sample template:

1. In the Document Parsers object, click **‘New’**
    a. Add ‘Parser Name’
    b. Click **‘Save’**



2. Click the **‘Tags Designer’** button



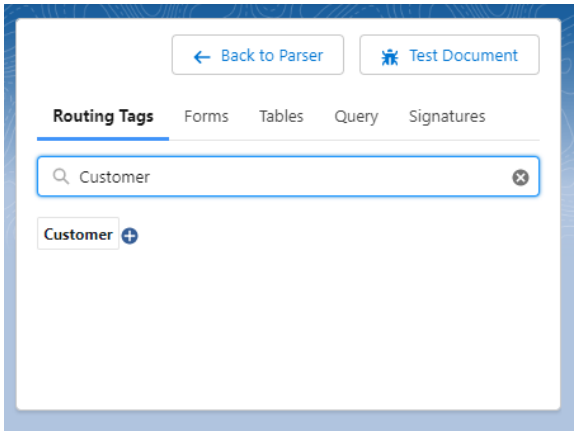
   a. Upload file (PDF, JPEG, PNG)

Your document will be parsed and a preview of the parsed data will be displayed.

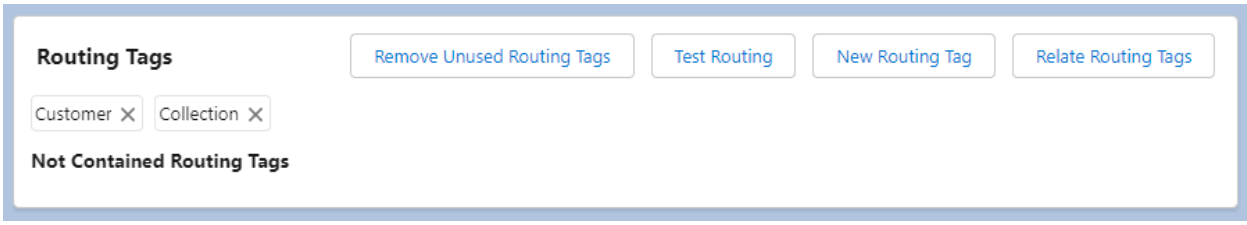
Step 2: To add Routing tags

Routing tags allow you to differentiate your document and lets the iParse engine know what document we will be processing. For example, a supplier name can be used to differentiate documents (which may have different layouts) from various suppliers.

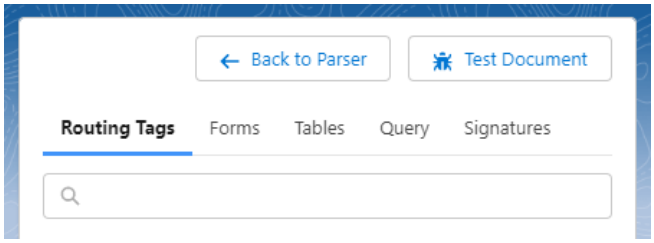
1. In the right column, click the **‘Routing Tags’** tab
2. Select which text to use as routing tags via “+” sign next to the text (You can select more than one)



   a. Your selected routing tags will be displayed under the **‘Routing Tags’** related list



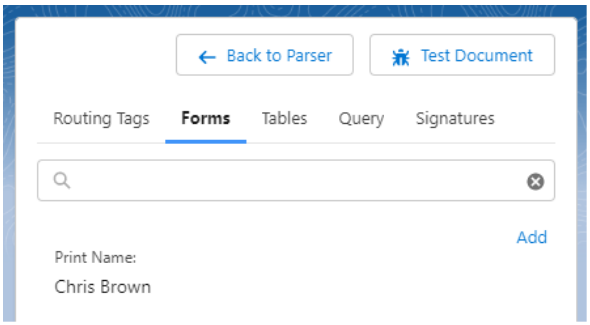
3. Click the **‘Test Document’** button to confirm the routing tags trigger as expected.



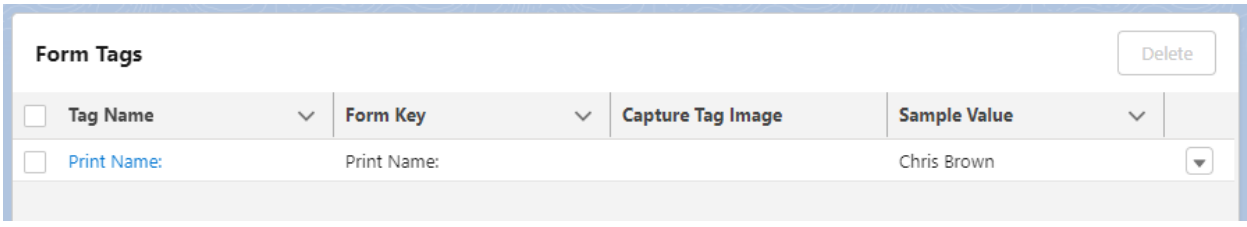
Step 3: To add Form Tags

Form tags specify what data we are going to extract from the document each time it's processed.

1. In the right column, click the **‘Forms’** tab
    a. You will be able to preview what iParse has extracted automatically.



2. Click **‘Add’** against the form tags you wish to parse each time
    a. Your selected form tags will display under the **‘Form Tags’** related list
    b. Edit the ‘Tag Name’ to something meaningful if required



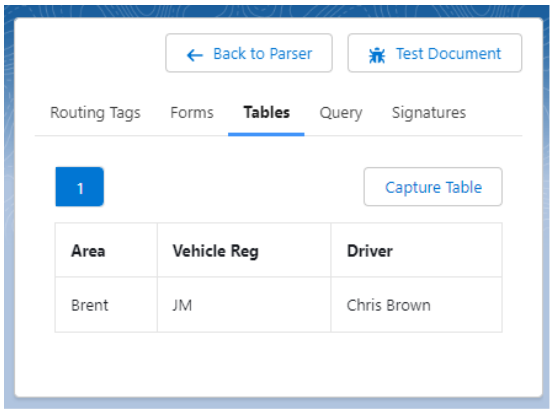
3. Click the **‘Test Document’** button to confirm the form tags parse the data as expected.

Other Parsing Options

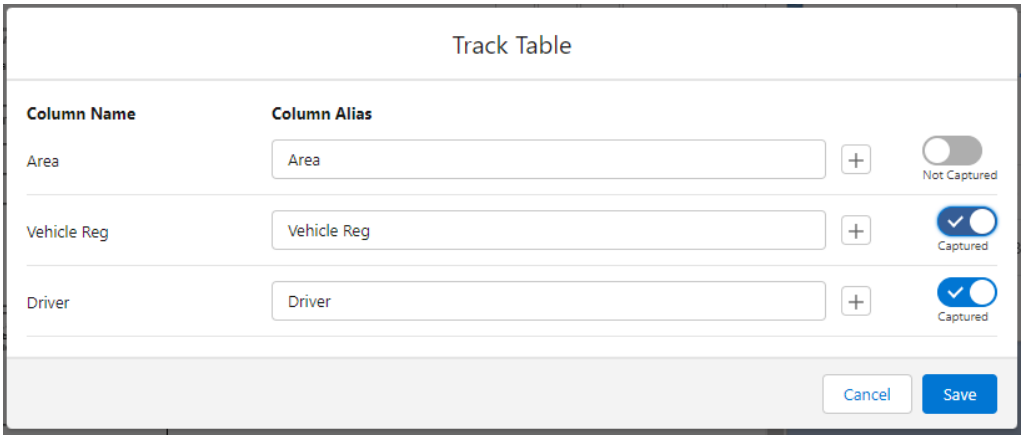
To Capture Table data

If your document has data stored in a table, you can capture this data using the 'Tables' feature.

1. In the right column, click the **'Tables'** tab



2. Click the 'Capture Table' button
    a. Click the toggle buttons against the columns you wish to capture the data from.

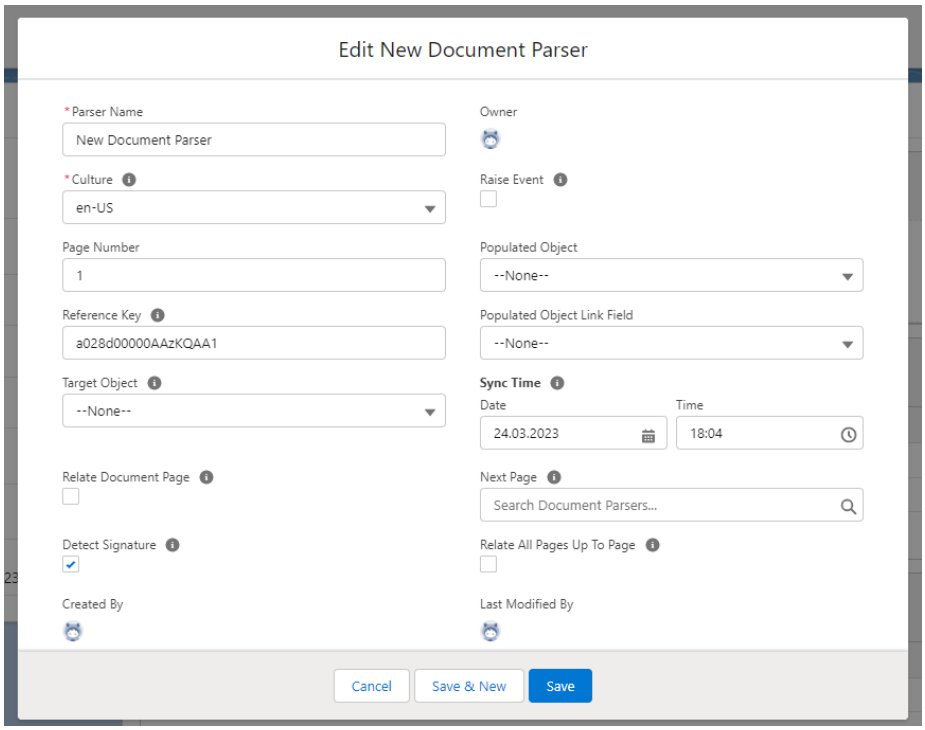


3. Click the **'Test Document'** button to confirm the table data parses the data as expected.

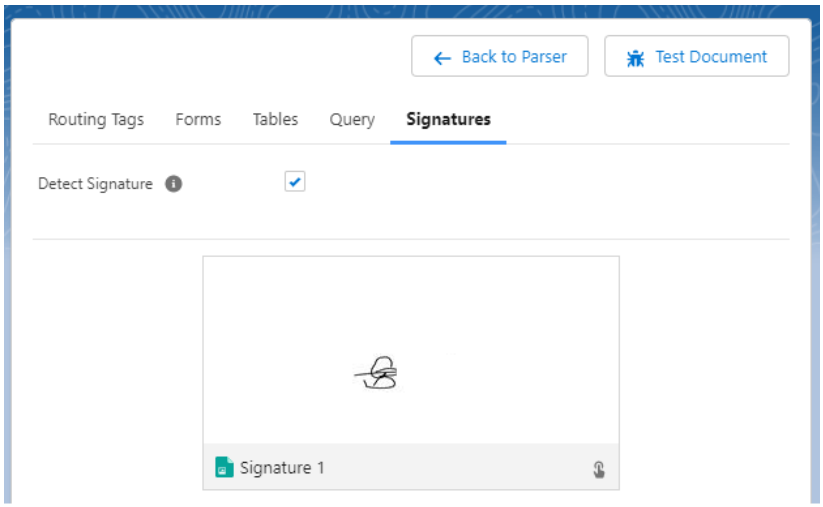
To Capture Signatures

You can detect and capture signatures on documents automatically.

1. Check **‘Detect Signature’** on your Document Parser record



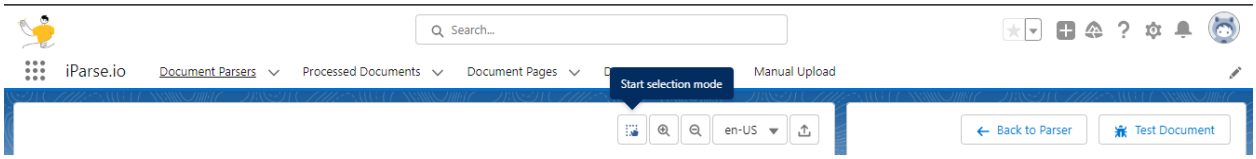
2. Upload your document using **Tag Designer**
3. In the right column, click the ‘**Signatures’** tab to display captured signatures



To capture a Selected Area

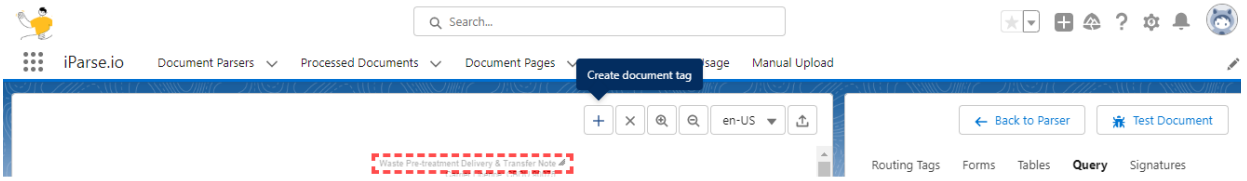
If your want to grab a data from a very specific area of a document that may not be available using other options;

1. Click **'Start Selection'** mode 

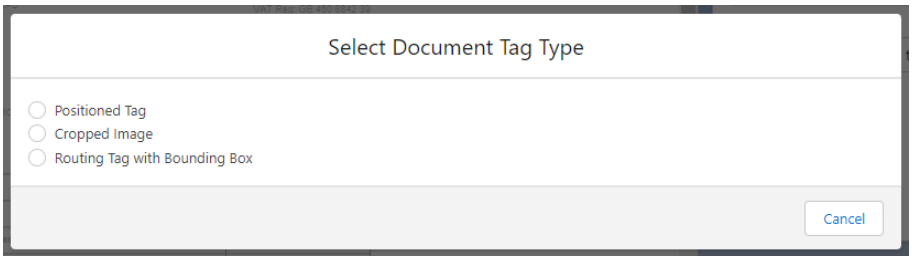


   a. Position the selection tool over the area you want

   b. Click on **'Create Document Tag'** 



   c. Choose either 'Position Tag' or 'Cropped Image'



   d. Give your Tag a suitable name
   e. Click **'Save'**

2. Click the **'Test Document'** button to confirm your 'Position Tag' or 'Cropped Image' is parsed as expected.

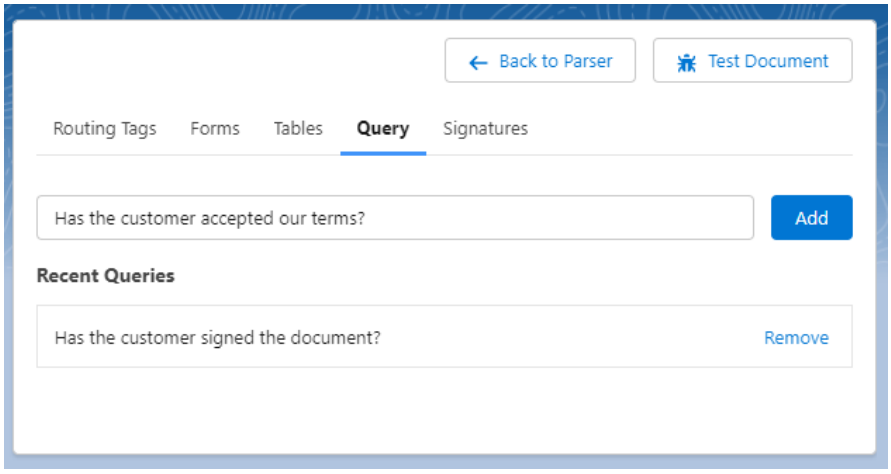
Create Query Tags

Query tags allow you to ask simple questions about your documents. For example;

- Has the customer signed the document?
- Has the customer accepted our terms?

iParse.io uses AI to answer these questions to the best of its ability.

1. Click on the **'Query'** tab
2. Enter your question to be answered and click **'Add'**

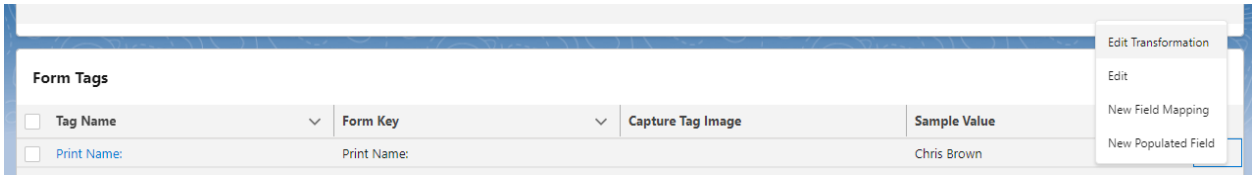


Transforming Parsed Data

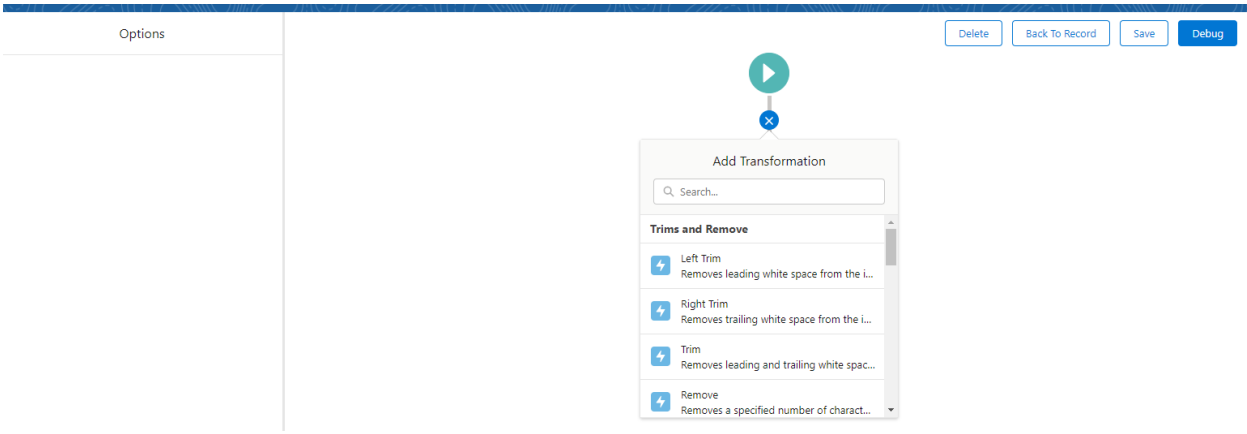
The values parsed from your Form Tags may need some adjusting - After all, data can sometimes be inconsistent. This is where the Transformation feature comes into play.

Using Transformation Designer

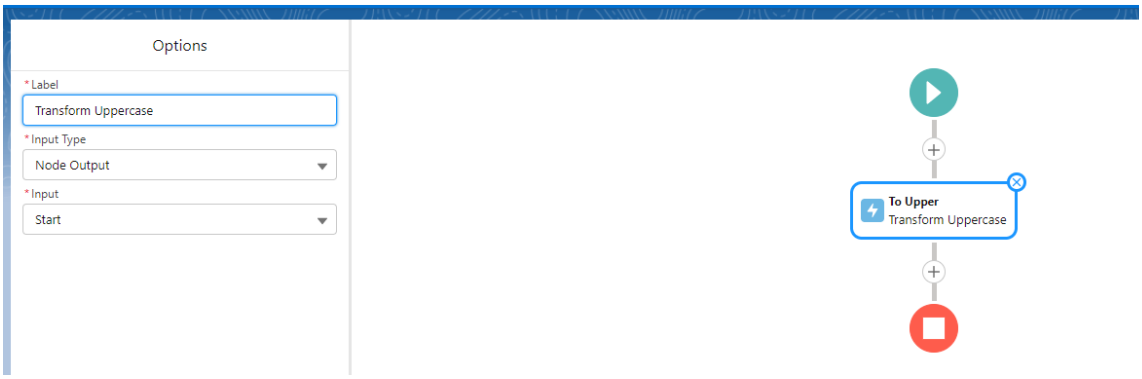
1. Click ‘Edit Transformation’ against your chosen Form Tag



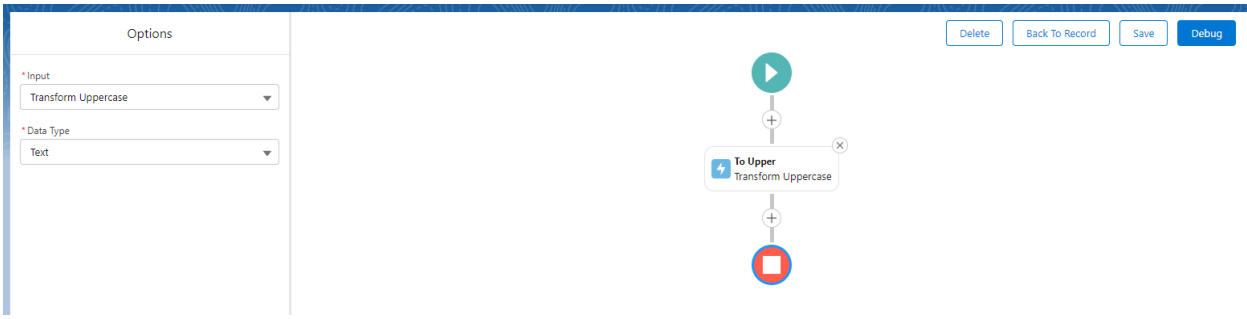
   a. Transformation designer will open
2. Click the green play button to begin
   a. View the input
3. Click the plus icon  on the flow designer
   a. Select the relevant transformation options to output a different format for this tag.



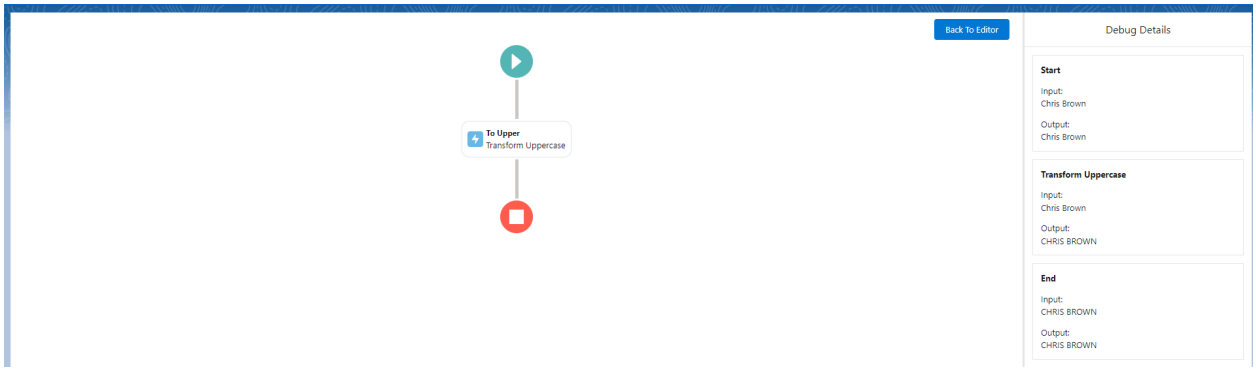
   b. Click on the transformation
      i. Give your transformation a relevant label name
      ii. Select Node Input (Alternatively choose a static value)
      iii. Select the Input
      iv. Apply transformation you want to make



4. Click the Stop flow icon 
    a. Select the **Input** from the dropdown



5. Click **‘Debug’**
6. View the **Debug Details**



7. Click **‘Back to Editor’** and ‘**Save**’

Transformation Options

Below outlines some of the transformation options along with some examples on how they can be used.

Transformation	Description	Example
Trims & Remove		
Left Trim	Removes leading white space from the input	Input: `iParse-000251631` Output: `iParse-000251631`
Right Trim	Removes leading white space from the input	Input: `iParse-000251631` Output: `iParse-000251631`

Trim	Removes leading & Trailing white space from the input	As above
Remove	Removes a specified number of characters from a specified index position in a string	Input: `iParse-000251631` Output: `000251631`
Remove Part	Removes a specified string occurrences from a string	Input: `iParse-000251631` Output: `i-000251631`

Remove Line Breaks	Removes All Line Breaks	Input:
		`Line1`
		`Line2`
		`Line3`
		Output:
		`Line1 Line2 Line3`

Change Case		
To Lower	Coverts all characters in the string to lower case	Input: `iParse-000251631` Output: `iparse-000251631`
To Upper	Coverts all characters in the string to Upper case	Input: `iParse-000251631` Output: `IPARSE-000251631`
Create New String occurrence		
Format	Build a formatted string from a set of objects	Input: `50689` Format: `Job-{0}` Output: `Job-50689`
Concat	Builds strings from two or more strings	Input: 1-`JobNumber:` 2-`50689` Output: `JobNumber:50689`

Regex Capture	Retrieve all REGEX pattern matches in an input string	Input:
		`BMW YK22 VGE Vehicle Ready`
		Regex :
		`\s?([A-Z]{1,2}[0-9]{1,2}\s?[A-Z]{3})`
		Output Format : `{0}`
		Output: `YK22 VGE`

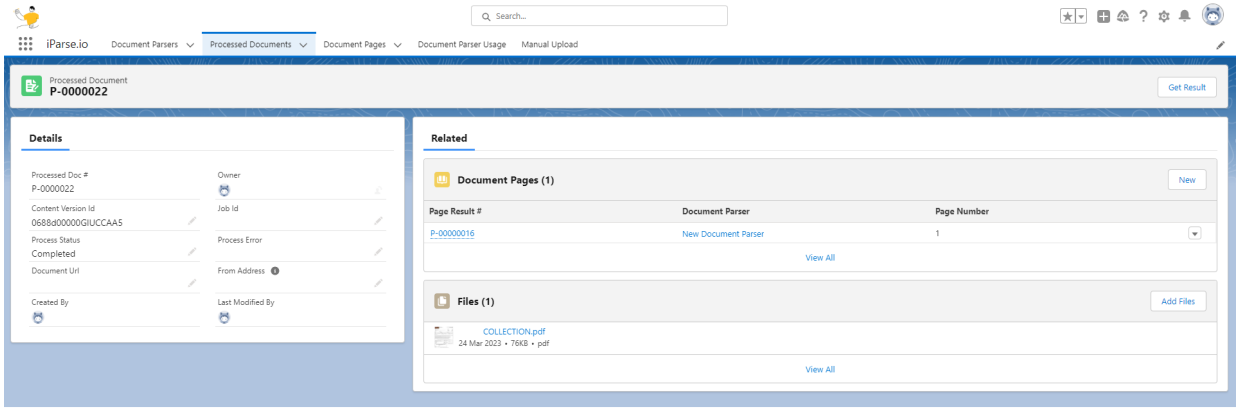
Substring	Retrieves a substring from start to end	Input: `BMW YK22 VGE Vehicle` Start: `0` Length: `3` Output: `BMW`
Get Part After	Retrieves a substring after occurrence of string part	Input: `BMW YK22 VGE Vehicle` String Part : `VGE` Output: `Vehicle`
Get Part Before	Retrieves a substring before of string part	Input: `BMW YK22 VGE Vehicle` String Part : `VGE` Output: `BMW YK22`

Replace	Replace each matching character/substring in the string	Input: `BMW YK22 VGE Vehicle` Old Value: `22` New Value: `33` Output: `BMW YK33 VGE Vehicle`
Pad Strings		
PAD Left	Creates a new string by concatenating enough leading pad characters to an original string to achieve a specified total length	Input: `282189` Total Character: `8` Padding Char: `0` Output: `00282189`
PAD Right	Creates a new string by concatenating enough leading pad characters to an original string to achieve a specified total length	Input: `282189` Total Character: `8` Padding Char: `0` Output: `28218900`

Processed Documents

Once you have your Document Parser Templates set up, you can begin to process documents automatically via the email service or manually. Each time a document is processed, a **Processed Document** record is created.

Viewing Processed Documents

1. Go to the **'Processed Documents'** tab
2. 

   Open the record

Process Status Overview

Status	Description
Initialized	Document is sent to the server for processing.
Queued	When the document has pages more than 8 it is queued for batch, the results arrive with a batch process.
Error	The document can not be processed.
Complete	Document parsed successfully.

Document Pages (Related Object)

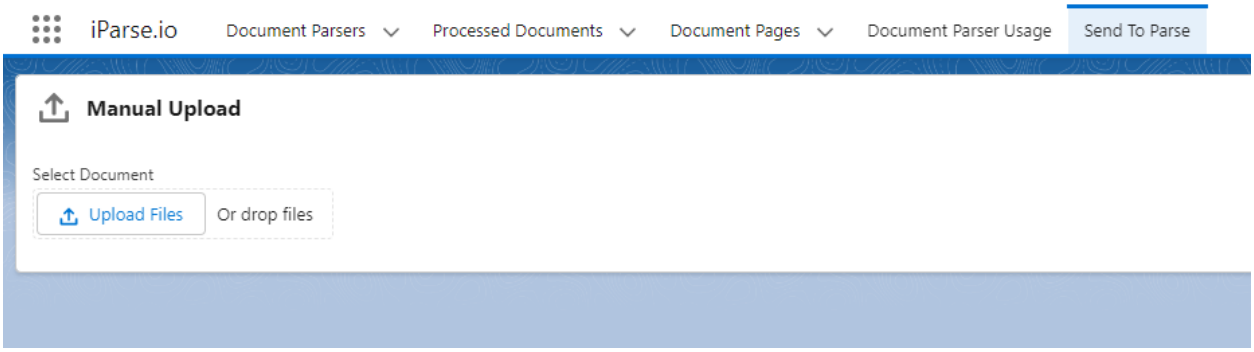
Each page of your processed document is handled individually. If your document has more than one page that successfully parsed, each page will be visible within the **Document Pages** related list.

Click into each document page record to preview what has been parsed from that page. Within each Document page record, you will be able to view the following;

- Related Routing Tags
- Related Page Form Tags & Parsed Values
- Related Pages Query Tags & Answers
- Related Position Tag
- Related Cropped Images
- Related Table Cell Parsed Values

Manual Upload

Users also have the ability to upload the document manually by using the Send to Parse feature.



Compliance

Introduction

When setting up your document parser records, you can also map your documents **FormTag** fields to Salesforce fields. This feature allows you to **compare the parsed data** against actual data held on a record. If any anomalies exist, the **compliance check** component will display the file.

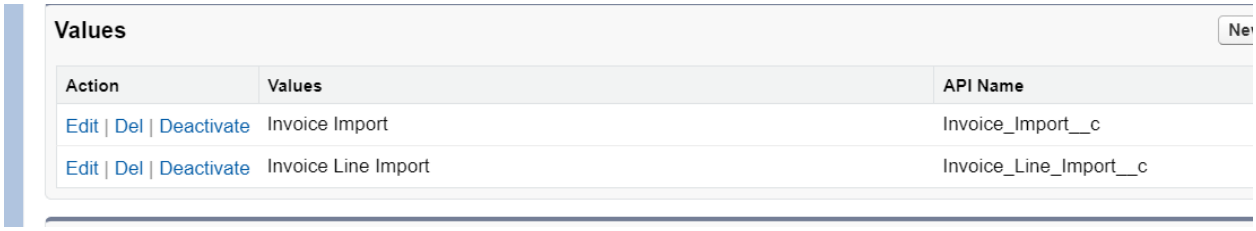
This component is useful for customers who want to ensure the data they have stored on the Salesforce matches with parsed documents..

Setup Field Mappings

Setup:

The objects and the fields used in compliance must be defined as a part of metadata.

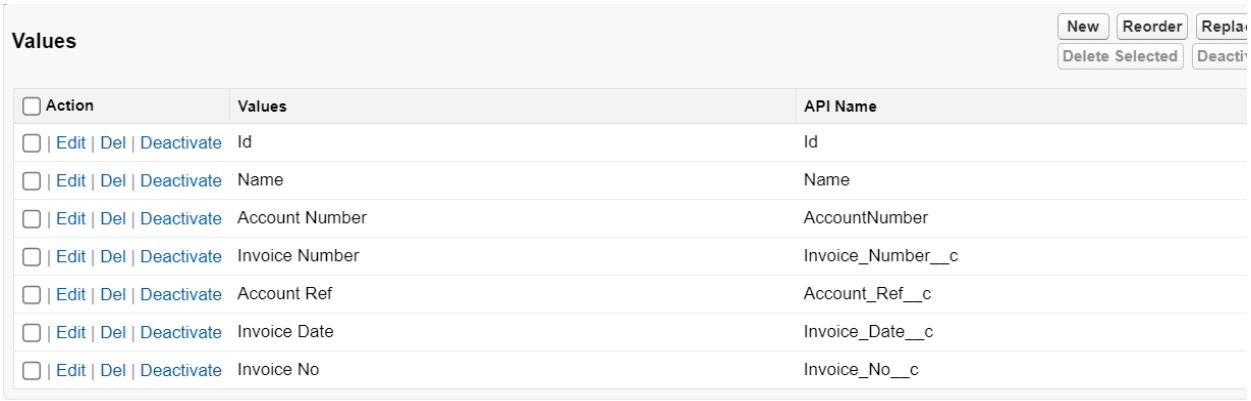
The objects defined in; “Target Objects” Picklist Value set with name and API name.



Values

Action	Values	API Name
Edit | Del | Deactivate	Invoice Import	Invoice_Import__c
Edit | Del | Deactivate	Invoice Line Import	Invoice_Line_Import__c

The Fields defined in; “iparseio__FieldMapping__c” object on "Target Field” which has a dependency to “Target Object” field on the same object (Global Picklist Value Set “Target Objects”).

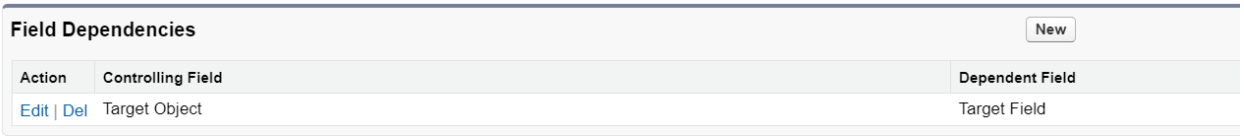


Values

New | Reorder | Delete Selected

Action	Values	API Name
Edit | Del | Deactivate	Id	Id
Edit | Del | Deactivate	Name	Name
Edit | Del | Deactivate	Account Number	AccountNumber
Edit | Del | Deactivate	Invoice Number	Invoice_Number__c
Edit | Del | Deactivate	Account Ref	Account_Ref__c
Edit | Del | Deactivate	Invoice Date	Invoice_Date__c
Edit | Del | Deactivate	Invoice No	Invoice_No__c

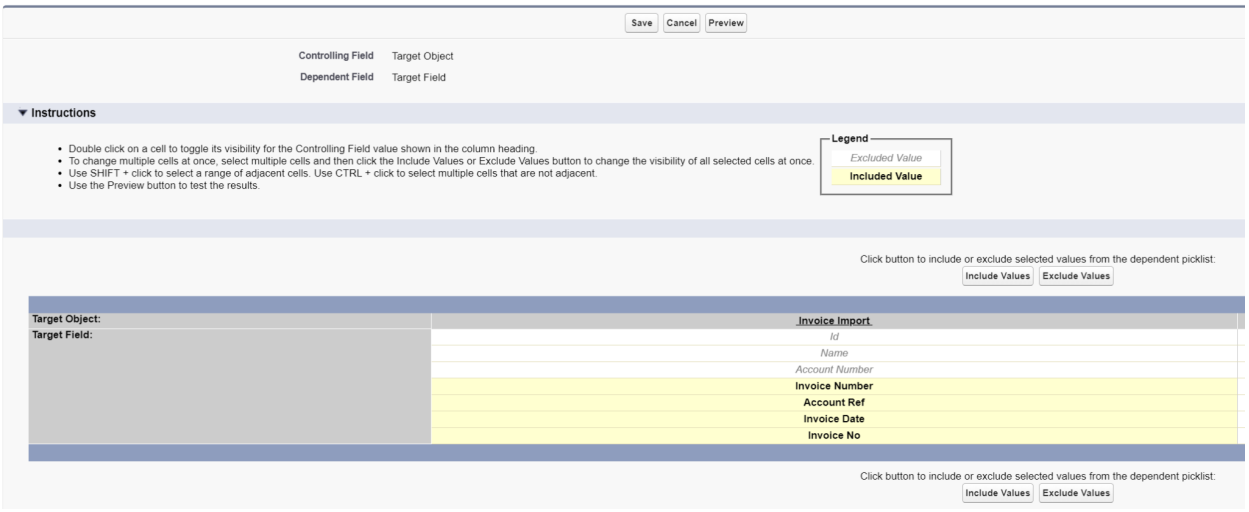
Set the field dependency values according to the Object - Fields



Field Dependencies New

Action	Controlling Field	Dependent Field
Edit | Del	Target Object	Target Field

Edit Field Dependency



Add Target Object

1. Open Document Parser Record
2. Select **'Target Object'** where compliance check is needed
3. Save Document Parser Record

Create Field Mappings

1. Open Document Parser Record
2. Click **'New'** on **Field Mapping** related list
    a. Select a Document Tag
    b. Select your desired operator
    c. Select the same **Target Object**
    d. Select your desired **Target Field**
    e. Optionally, you can insert a value in the **Constant Value** field (iParse will compare against this constant value)

Add Compliance Check Component to the layout

1. Go to the Target Object
2. Click setup > *'Edit Page'*
3. Add the **'Record Processed Documents'** component to the layout
4. In your flow process for the Document Page set the Document Page -> Related Record id to target object id.

Create a Flow for Compliance Check by Upload

Use a flow to upload your document to the record. (Please watch our extra guidance on setting up flows)

1. Within your flow, Add the **'Uploaded Document Compliance'** component
2. Add the following **key** parameters

Field	Value
`API Name`	Select unique value
`Content Document ID`	`con`
`contentDocumentIds`	`{!ContentDocIds}`
`distributionUrl`	`{!documentURL}`
`Editable`	If you want the user to be able to edit Fields select `{!$GlobalConstant.True}`
`relatedRecordId`	`{!recordId}`

When you upload a file via your flow, the ‘Upload Document Compliance’ preview component will display your Mapped Fields against the mapped Salesforce data we are checking against.

Example:

The compliance component will display the Fields we mapped. If the Operator is highlighted in Red, this means a discrepancy occurred; If Green, the data comparison passed the validation.