iParse

iParse.io Introduction **2**
iParse.io Security & Your Data **2**
iParse.io Support **2**
Setup Salesforce Email Service **3**
To setup the Salesforce Email Service 3
Setup Salesforce Named Credentials 4
Assign Permission Sets **5**
To Assign a Permission Set 5
Create iParse Document Parser **6**
Step 1: To create a new iParse sample template: 6
Document Parser Custom Options 7
Step 2: Use Tag Designer 9
Step 3: Add Routing tags 9
Step 4: Add Form Tags 11
Other Parsing Options **12**
Use Capture Table data 12
To Capture Signatures 13
To Capture A Selected Area 13
Create Query Tags 15
Transforming Parsed Data **16**
Using Transformation Designer 16
Viewing Processed Documents 20
Process Status Overview 20
Document Pages (Related Object) 20
iParse Ai Engine For Invoices **21**
Introduction 21
Setup: 21
CSV Parsing **25**
Introduction 25
Setup: 25
Processing CSV Documents 28
Compliance **29**
Introduction 29
Setup Field Mappings
Setup: 29

Add Target Object 31
Create Field Mappings 31
Add Compliance Check Component to the layout 31
Create a Flow for Compliance Check by Upload 31

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. Whether that involves updating or inserting data into standard or custom objects within Salesforce, iParse is a tool that will greatly increase your organisation's productivity by removing manual data entry.

iParse.io Security & Your Data

iParse is a product built for Salesforce with Microsoft Azure foundation. Any data extracted from your files using iParse is not stored by us whatsoever. View Data Flow Diagram

iParse.io Support

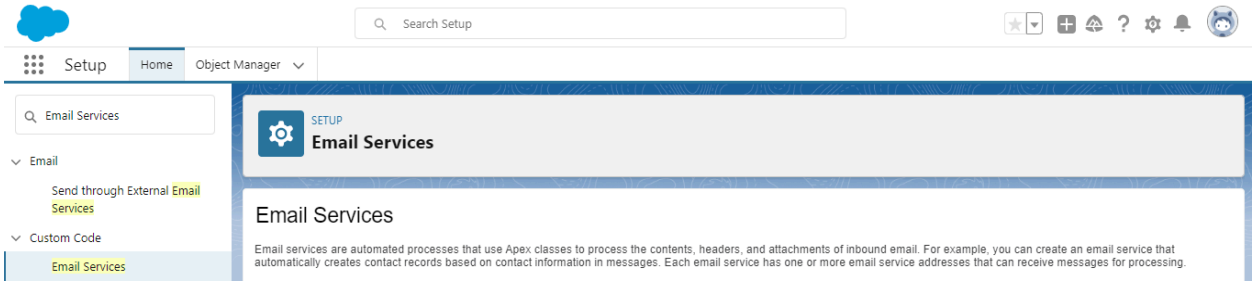
Our dedicated support team is here to assist you with any support needs that you have. If you have any questions please email support@iparse.io.

Setup Salesforce Email Service

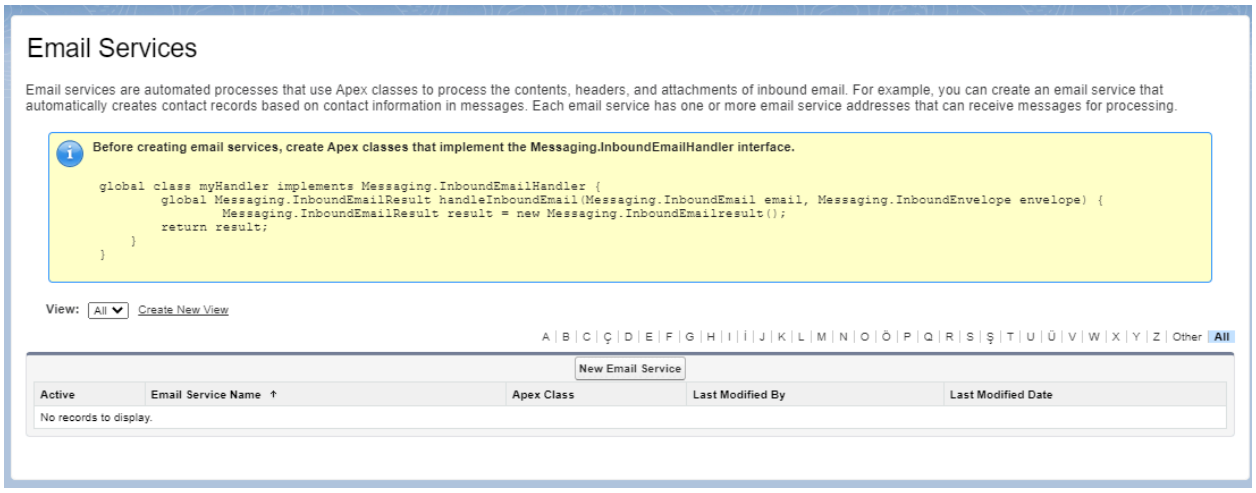
The iParse email service allows you to simply forward your documents to your email service address and iparse.io will start to extract the data from your documents based on the routing and configuration rules you have set up… more on those rules later.

To setup the Salesforce Email Service

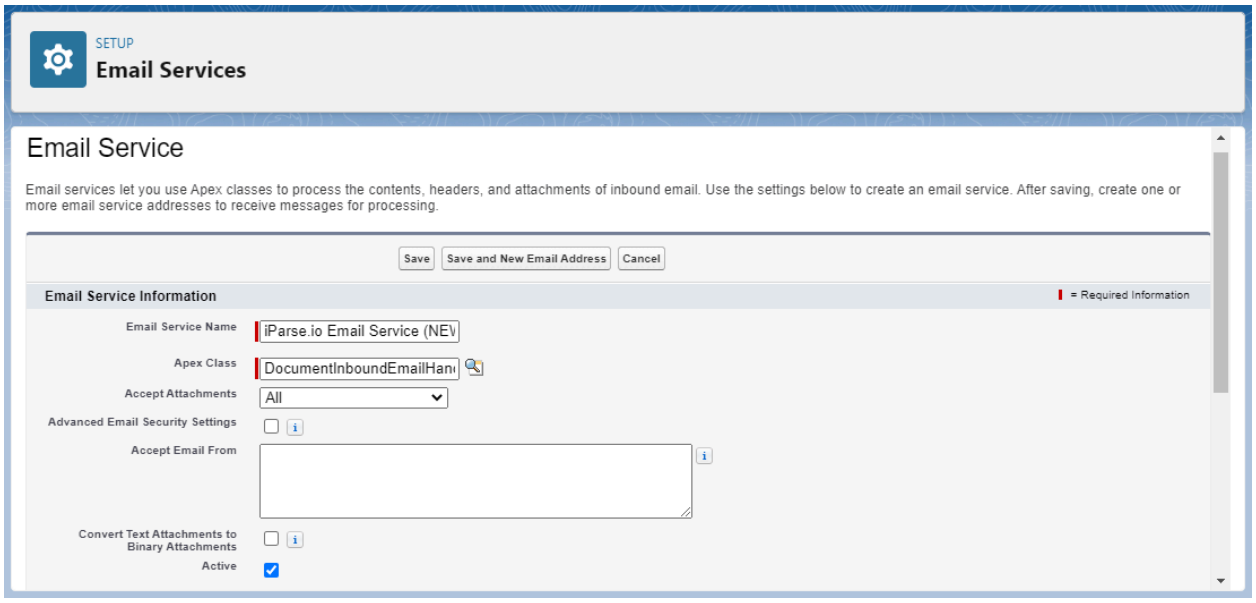
1. Click **‘Setup’**
2. Search for **‘Email Services’**



3. Click the **‘New Email Service’** button

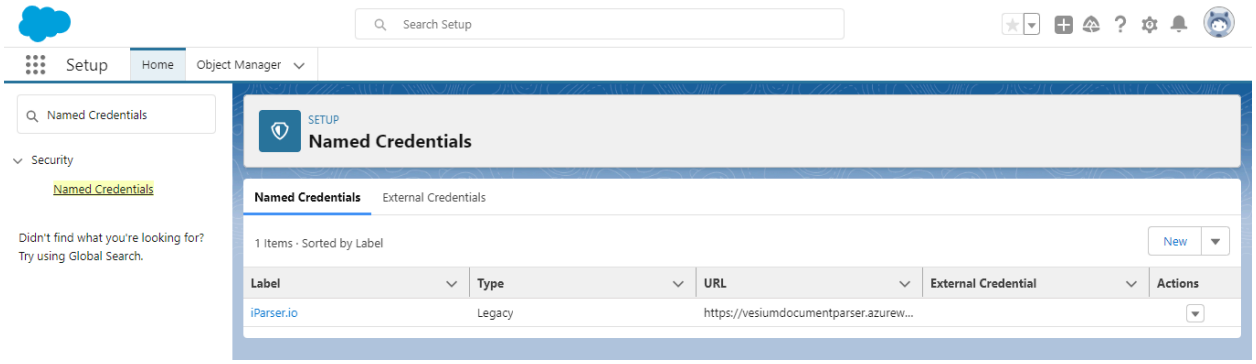


4. Add Service Name
5. Select Apex Class : **DocumentInboundEmailHandler**
6. Select Accept Attachments : **All**
7. Make **Active**
8. Click **Save**

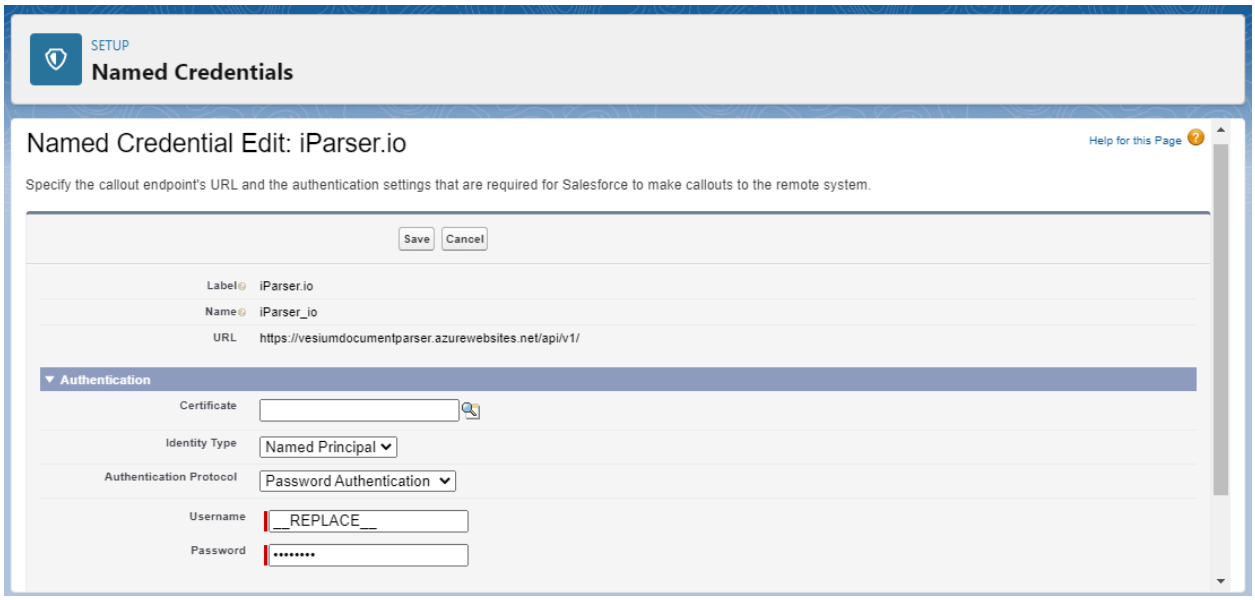


Setup Salesforce Named Credentials

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. (iParse team will provide these credentials).

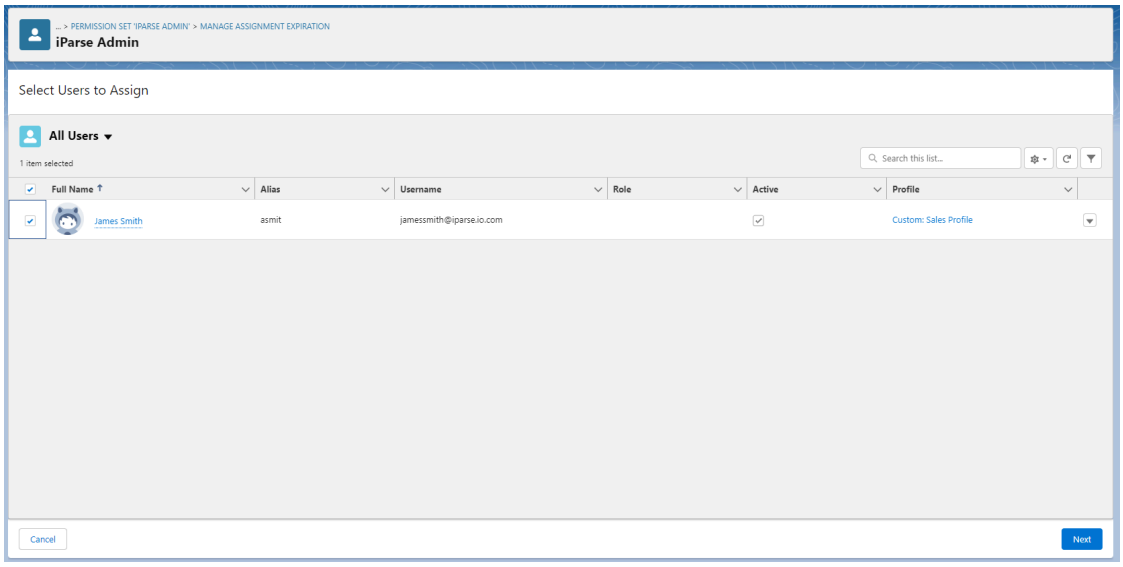


Assign Permission Sets

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'**

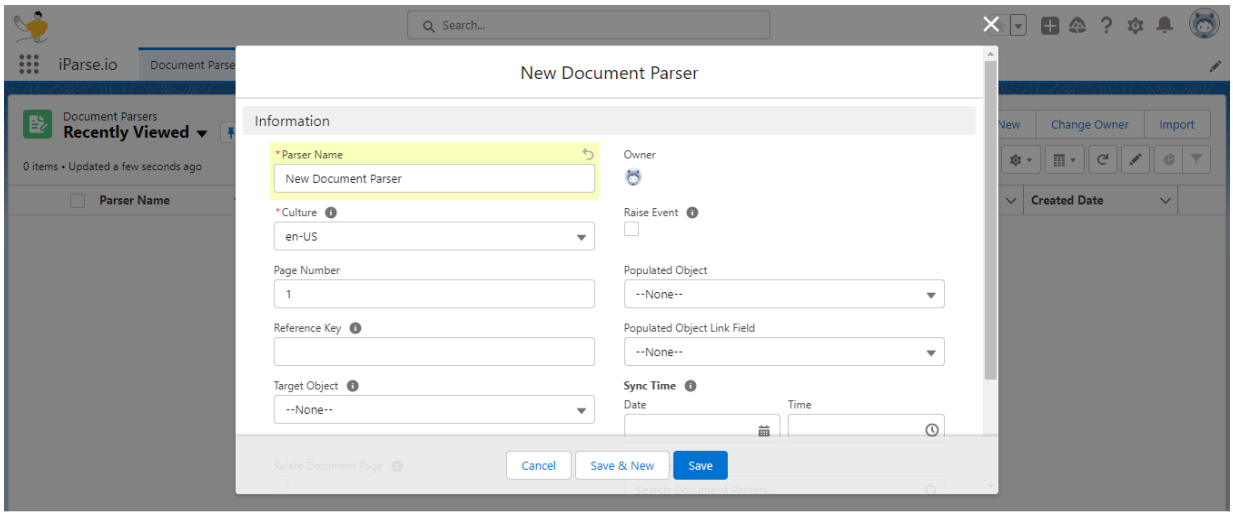


7. Click **‘Assign’** and click **‘Done’**

Create iParse Document Parser

Step 1: To create a new iParse sample template:

1. In the Document Parsers object, click **‘New’**
    a. Add ‘Parser Name’
    b. Select your required custom options (See Document Parser Custom Options Below)
    c. Click **‘Save’**



Document Parser Custom Options

The below options are useful depending on the file you are configuring.

Field	Description	Scenario / Example
Relate Document Page	Assigns the relative page (PDF) to the processed document page. This is helpful if you have more than a single page in a PDF but you want to route the different pages down different Document Parse routes. Single page related to each document page.	You receive 2+ page files from your customers. • Page 1 is the agreement. • Page 2 is an order breakdown. You set up a Document Parser record for each page (treating them individually). This box is checked on both Document Parser records. When the file is received, the parsers extract the data and the relevant page from the PDF is stored on the processed document (I.e. the PDF is split from a 2 page file to individual files for page 1 and 2).

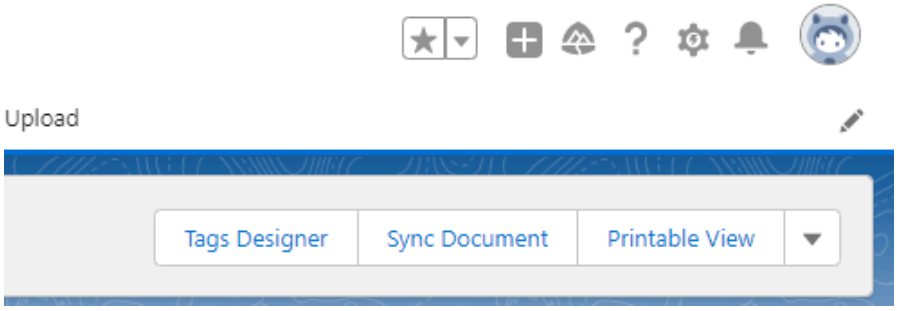
Use The Same Parser For Following Pages	iParse analyzes each page individually & the routing tags of the page informs iParse what fields it should focus on extracting. However, at times if you have a document that spans multiple pages, perhaps the routing tags only appear on the first page (This is typical for tables on invoices that continue onto multiple pages). In this scenario, by enabling this feature, you are telling iParse to continue to grab the same information even if the routing tags are not detected.	You have an invoice template with a table that details the line items. Your document parser has its routing tags configured (E.g. the supplier name and account number), however this information is not on other pages. Table capture configuration grabs all the data from your chosen columns. **The Issue:** As the routing tags are not picked up beyond page 1, we need to tell iParse to apply the same tags other pages; **The Solution:** By checking the 'Use The Same Parser For Following Pages', iParse will use the same table
Relate All Pages Up To Page (for Files)	Relate all pages up to the page when the continuous form feature is used.	In a 10-page document, if your parser definition matches page 3, the pdf file contains pages 3-10 on the results. *Output - Document Page Files, the PDF would show pages 3-10.
Retrieve Headerless Tables	Option for tables with no headers where the table is not captured by the designer.	Use this feature If you have a table-like structure without headers and not grabbed by the designer as table.
Next Page	Allows for documents where no routing tags are available on a page that is different.	Your file is 3 pages. ● Page 1+ forms an invoice. ● Next Page is an Agreement. **The Issue:** You want to extract data from the Agreement page, but there are no feasible routing tags. **The solution:** Create a new document parser

		record. Select the page that has no routing tags by using the page selector.  Select your Form Tags, Table Capture & Queries. Save the Document Parser Record. Next, go to the original Document Parser you created for Page 1+ and use the 'Next Page' lookup field to populate the Agreement page you just created.

Step 2: Use Tag Designer

Tag Designer allows you to select what information you want to grab each time from your file.

1. Click the **'Tags Designer'** button



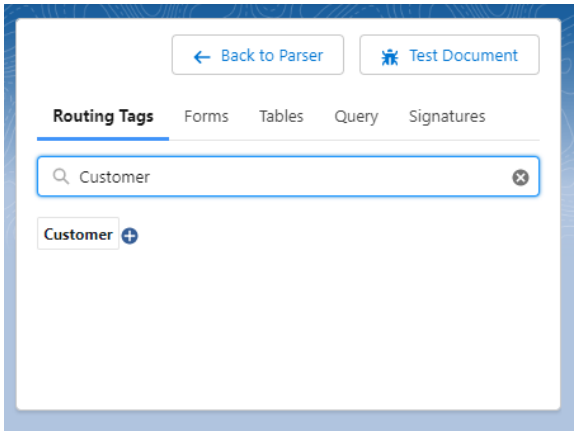
   d. Upload file (PDF, JPEG, PNG)

Sync Document: Sync the document to the iParse Engine if you have made changes to an existing document you have created. (E.g. If you add additional routing or form tags to a Document Parser record).

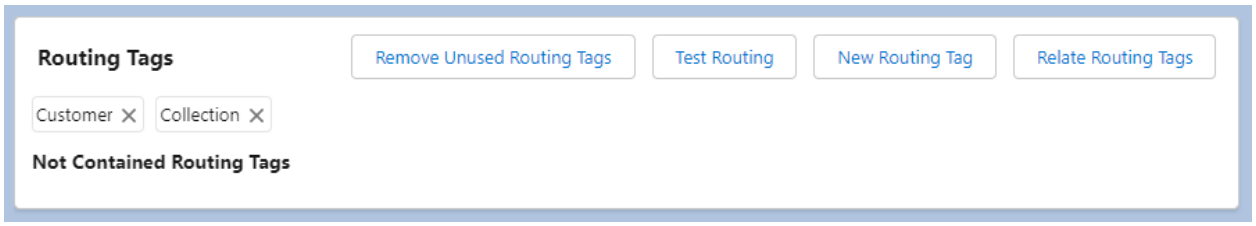
Step 3: Add Routing tags

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.

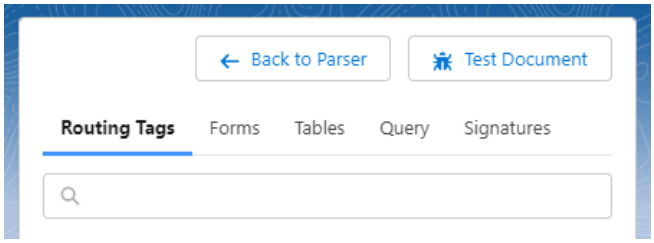
1. In the right column, click the **'Routing Tags'** tab
2. Select which text to use as routing tags via "+" sign next to the text (You can select more than one)



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



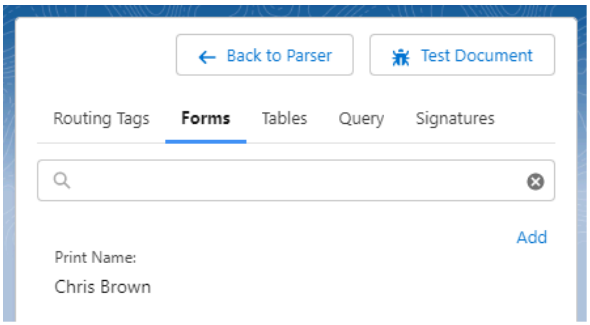
3. Click the **'Test Document'** button to confirm the routing tags trigger as expected.



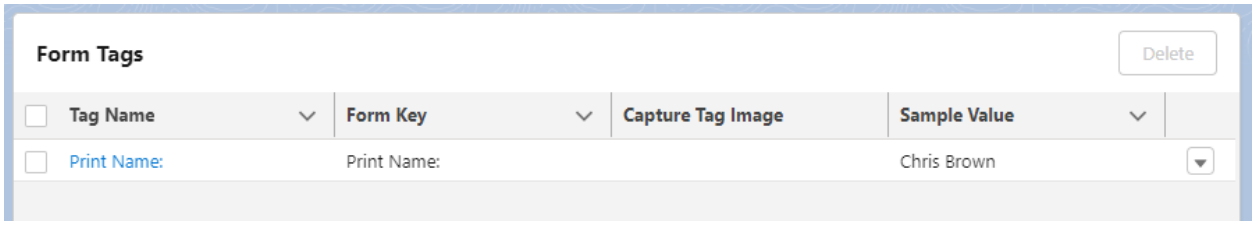
Step 4: Add Form Tags

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



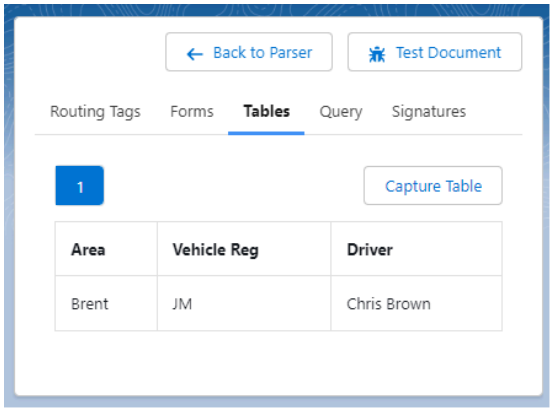
3. Click the **‘Test Document’** button to confirm the form tags parse the data as expected.

Other Parsing Options

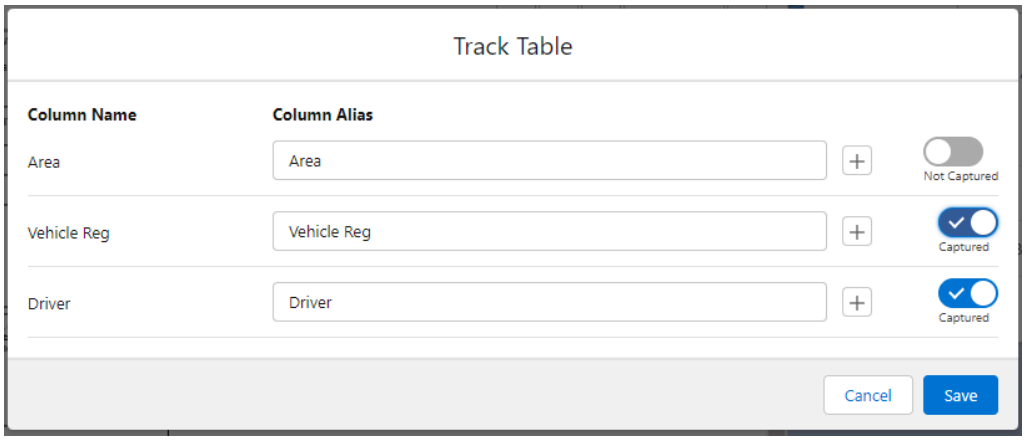
Use Capture Table data

If your document has data stored in a table, you can capture this data within Tag Designer 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.

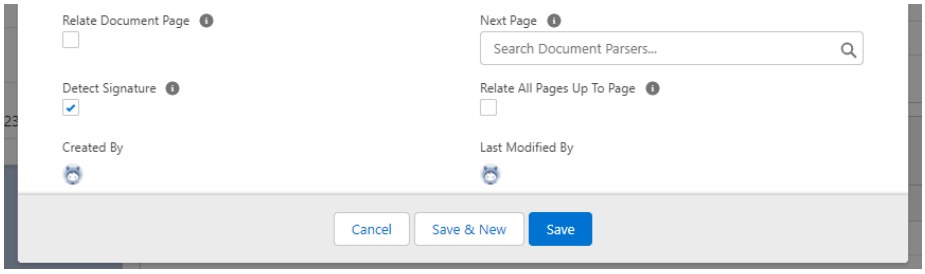


3. Click the **‘Test Document’** button to confirm the table data parses the data as expected.

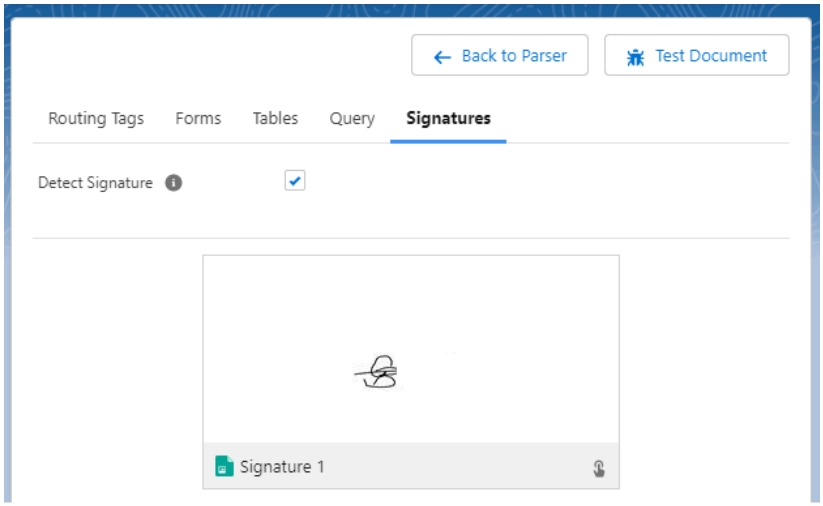
To Capture Signatures

You can detect and capture signatures on documents automatically.

1. Check **'Detect Signature'** on your Document Parser record



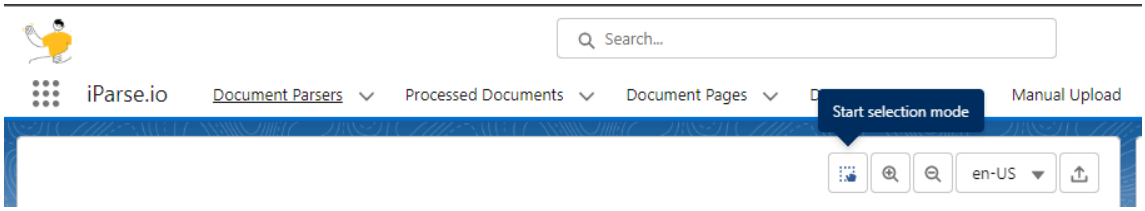
2. Upload your document using **Tag Designer**
3. In the right column, click the '**Signatures'** tab to display captured signatures



To Capture A Selected Area

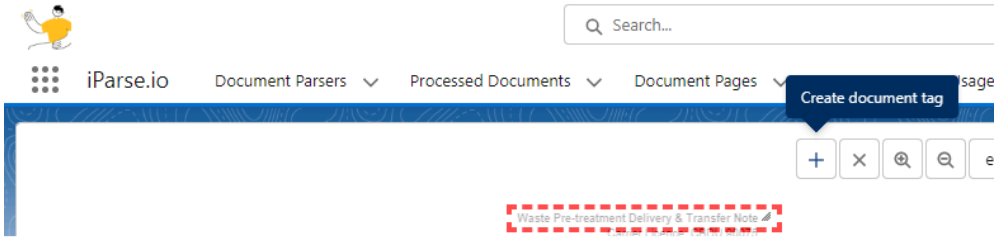
If your want to grab a data from a very specific area of a document that may not be available using other options;

1. Within Tag Manager, Click **'Start Selection'** mode 

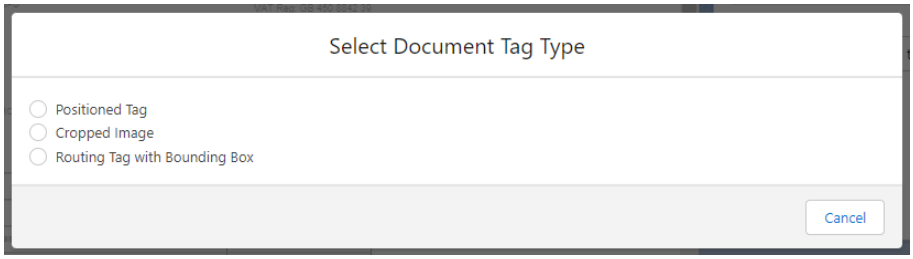


a. Position the selection tool over the area you want

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



c. Choose either 'Position Tag' or 'Cropped Image' (Note: Position Tags will extract the readable data, whereas Cropped Image, will save the selected area as an 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.

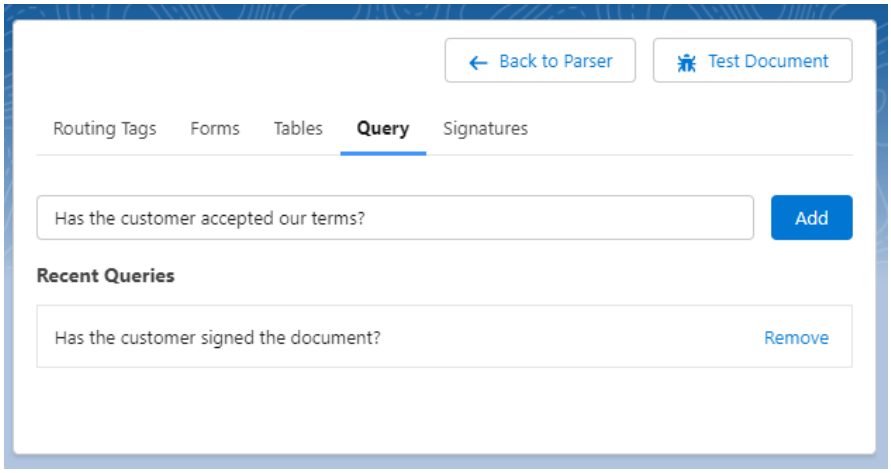
Create Query Tags

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'**

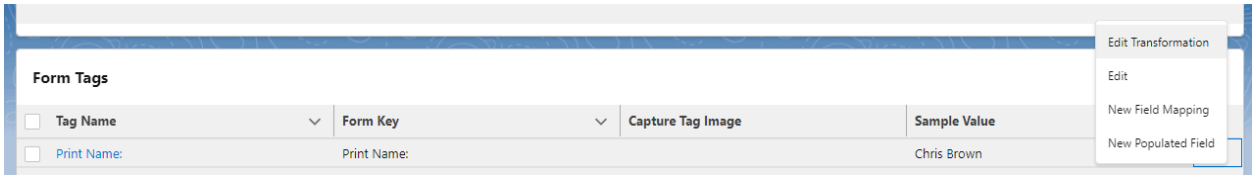


Transforming Parsed Data

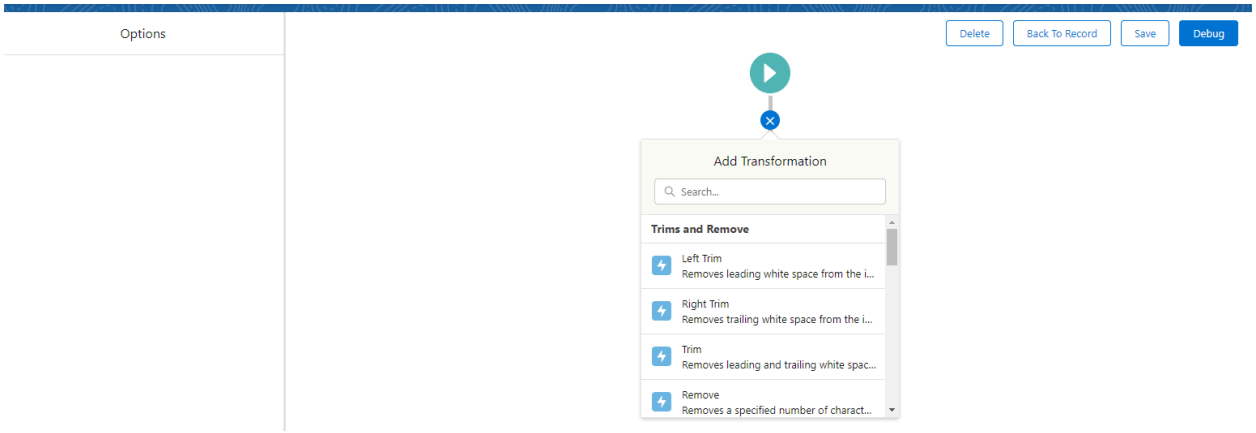
The values parsed from your Form Tags may need some adjusting - After all, data can sometimes be inconsistent. This is where the Transformation feature comes into play.

Using Transformation Designer

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

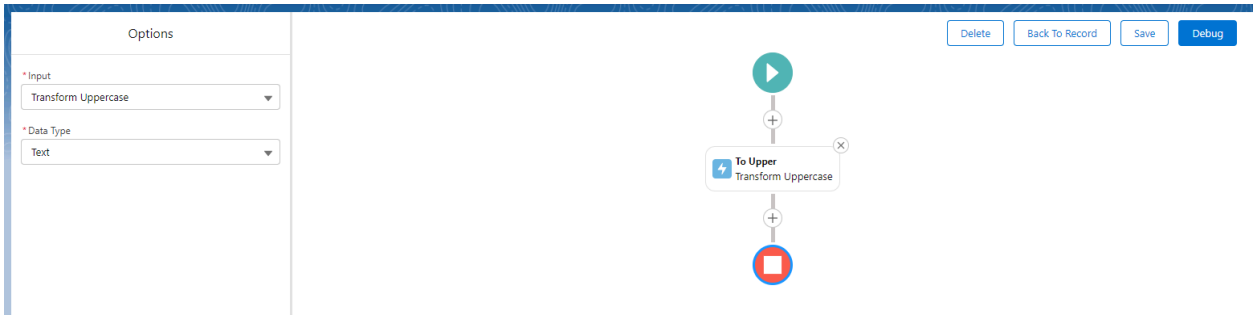


   a. Transformation designer will open
2. Click the green play button to begin
   a. View the input
3. Click the plus icon on the flow designer
   a. Select the relevant transformation options to output a different format for this tag.

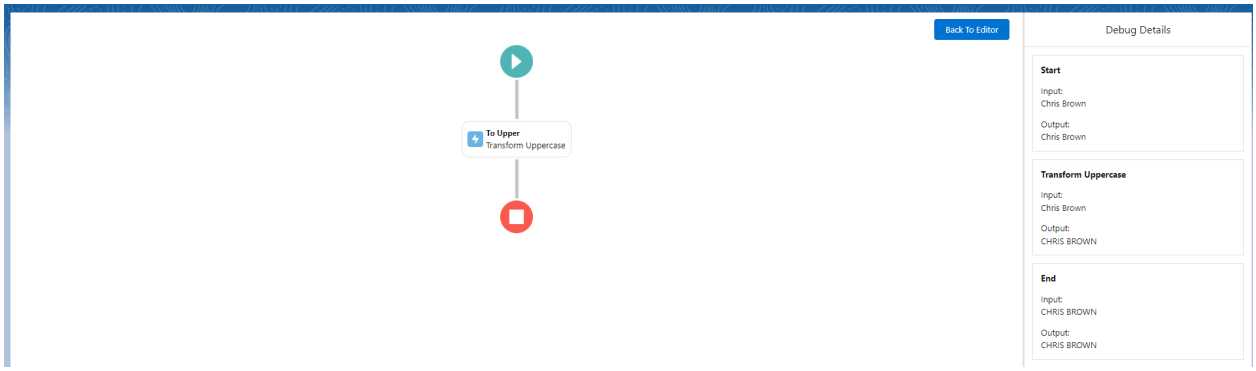


   b. Click on the transformation
      i. Give your transformation a relevant label name
      ii. Select Node Input (Alternatively choose a static value)
      iii. Select the Input
      iv. Apply transformation you want to make

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`

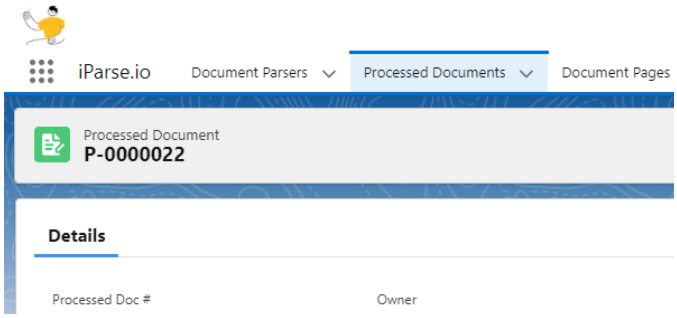
		}\s?[A-Z]{3}) Output Format : {0} Output: YK22 VGE
Substring	Retrieves a substring from start to end	Input: BMW YK22 VGE Vehicle Start: 0 Length: 3 Output: BMW
Get Part After	Retrieves a substring after occurrence of string part	Input: BMW YK22 VGE Vehicle String Part : VGE Output: Vehicle
Get Part Before	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 by manually uploading them. 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

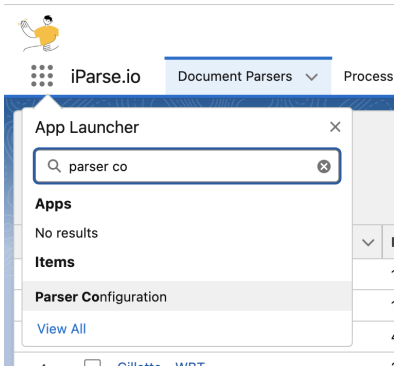
iParse Ai Engine For Invoices

Introduction

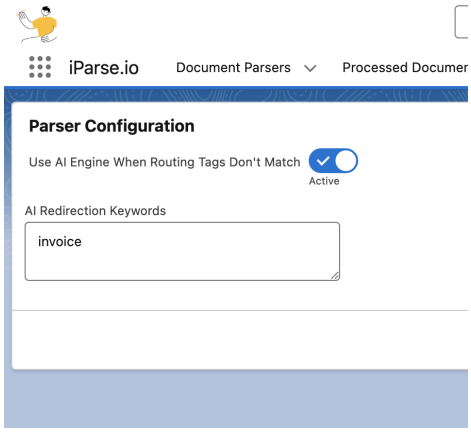
If you receive a high number of Invoices from various suppliers, setting each one up using routing tags may not be a feasible or ideal solution.With iParse.io, you can use our in-built Invoice Ai engine which will automatically extract data.

Setup:

1. In App Launcher go to Parser Configuration



2. Enable ‘Use AI Engine When Routing Tags Don’t Match’
3. Enter Redirection Keywords separated by a comma (These are the keywords iParse extracts. E.g. If no routing tags are matched AND iParse picks up your keyword, it will parse the invoice document using the Ai Engine for Invoices.)



Every new line is OR option

Every value in the line separated with semicolon (;) is AND

The operation is Case Insensitive

4. When you send a file that is processed via the AI Engine, iParse will relate the extracted data to some predefined tags that are often captured across the majority of invoices. These tags include;

- Invoice Receipt Date — `INVOICE_RECEIPT_DATE`
- Invoice Receipt ID — `INVOICE_RECEIPT_ID`
- Invoice Tax Payer ID — `TAX_PAYER_ID`
- Customer Number — `CUSTOMER_NUMBER`
- Account Number — `ACCOUNT_NUMBER`
- Vendor Name — `VENDOR_NAME`
- Receiver Name — `RECEIVER_NAME`
- Vendor Address — `VENDOR_ADDRESS`
- Receiver Address — `RECEIVER_ADDRESS`
- Order Date — `ORDER_DATE`
- Due Date — `DUE_DATE`

- Delivery Date — `DELIVERY_DATE`
- PO Number — `PO_NUMBER`
- Payment Terms — `PAYMENT_TERMS`
- Total — `TOTAL`
- Amount Due — `AMOUNT_DUE`
- Amount Paid — `AMOUNT_PAID`
- Subtotal — `SUBTOTAL`
- Tax — `TAX`
- Service Charge — `SERVICE_CHARGE`
- Gratuity — `GRATUITY`
- Prior Balance — `PRIOR_BALANCE`
- Discount — `DISCOUNT`
- Shipping and Handling Charge — `SHIPPING_HANDLING_CHARGE`
- Vendor ABN Number — `VENDOR_ABN_NUMBER`
- Vendor GST Number — `VENDOR_GST_NUMBER`
- Vendor PAN Number — `VENDOR_PAN_NUMBER`
- Vendor VAT Number — `VENDOR_VAT_NUMBER`
- Receiver ABN Number — `RECEIVER_ABN_NUMBER`
- Receiver GST Number — `RECEIVER_GST_NUMBER`
- Receiver PAN Number — `RECEIVER_PAN_NUMBER`
- Receiver VAT Number — `RECEIVER_VAT_NUMBER`
- Vendor Phone — `VENDOR_PHONE`
- Receiver Phone — `RECEIVER_PHONE`
- Vendor URL — `VENDOR_URL`
- Line Item/Item Description — `ITEM`
- Line Item/Quantity — `QUANTITY`
- Line Item/Total Price — `PRICE`
- Line Item/Unit Price — `UNIT_PRICE`
- Line Item/ProductCode — `PRODUCT_CODE`

- Address (Bill To, Ship To, Remit To, Supplier) — `ADDRESS`
- Name (Bill To, Ship To, Remit To, Supplier) — `NAME`
- Core Address (Vendor, Receiver, Bill To, Ship To, Remit To, Supplier) — `ADDRESS_BLOCK`
- Street Address (Vendor, Receiver, Bill To, Ship To, Remit To, Supplier) — `STREET`
- City (Vendor, Receiver, Bill To, Ship To, Remit To, Supplier) — `CITY`
- State (Vendor, Receiver, Bill To, Ship To, Remit To, Supplier) — `STATE`
- Country (Vendor, Receiver, Bill To, Ship To, Remit To, Supplier) — `COUNTRY`
- ZIP Code (Vendor, Receiver, Bill To, Ship To, Remit To, Supplier) — `ZIP_CODE`

You can then use these tags in your automations (i.e. Apex or Flows) to Insert or update records in your org.

CSV Parsing

Introduction

CSV Parsers allow you to forward CSV's you receive on a regular basis to the iParse Email Service (or Manually Upload), which allows you to extract and process your data automatically.

Setup:

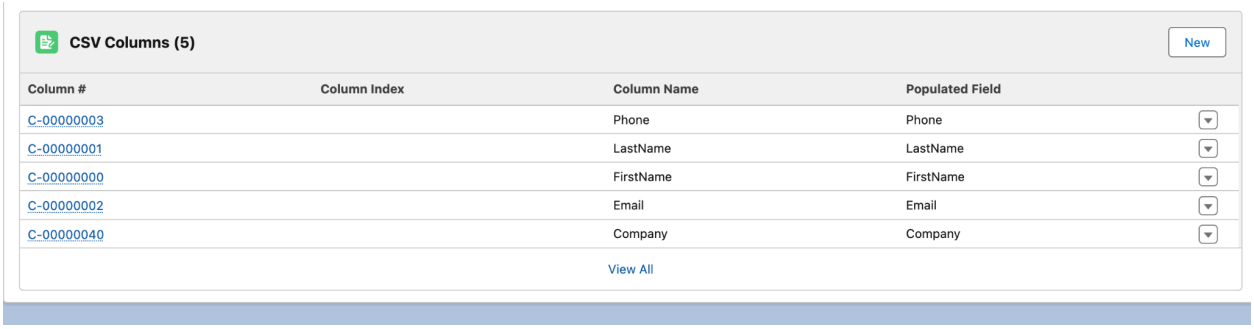
1. Click 'CSV Parsers'
2. Create 'New' CSV Parser
3. Add 'CSV Parser Name'
4. Select how you wish to identify this csv (Similar to routing tags)

    a. File Name Pattern (Contains word/phrase)
    b. Sender Name Pattern (Contains word/phrase)
    c. Sender Address Pattern (Contains word/phrase)
    d. Subject Pattern (Contains word/phrase)

5. Select 'Match Column' Pattern, if you want to respect the columns index.
6. Select an Object that you wish to populate the CSV extracted data to (Note: You must add the object name as a picklist value on the Populate_Object__c field to select it.
7. Parse By either Delimiter, Position or Tab (Delimiter is the default option)
8. Set Delimiter as a comma ','

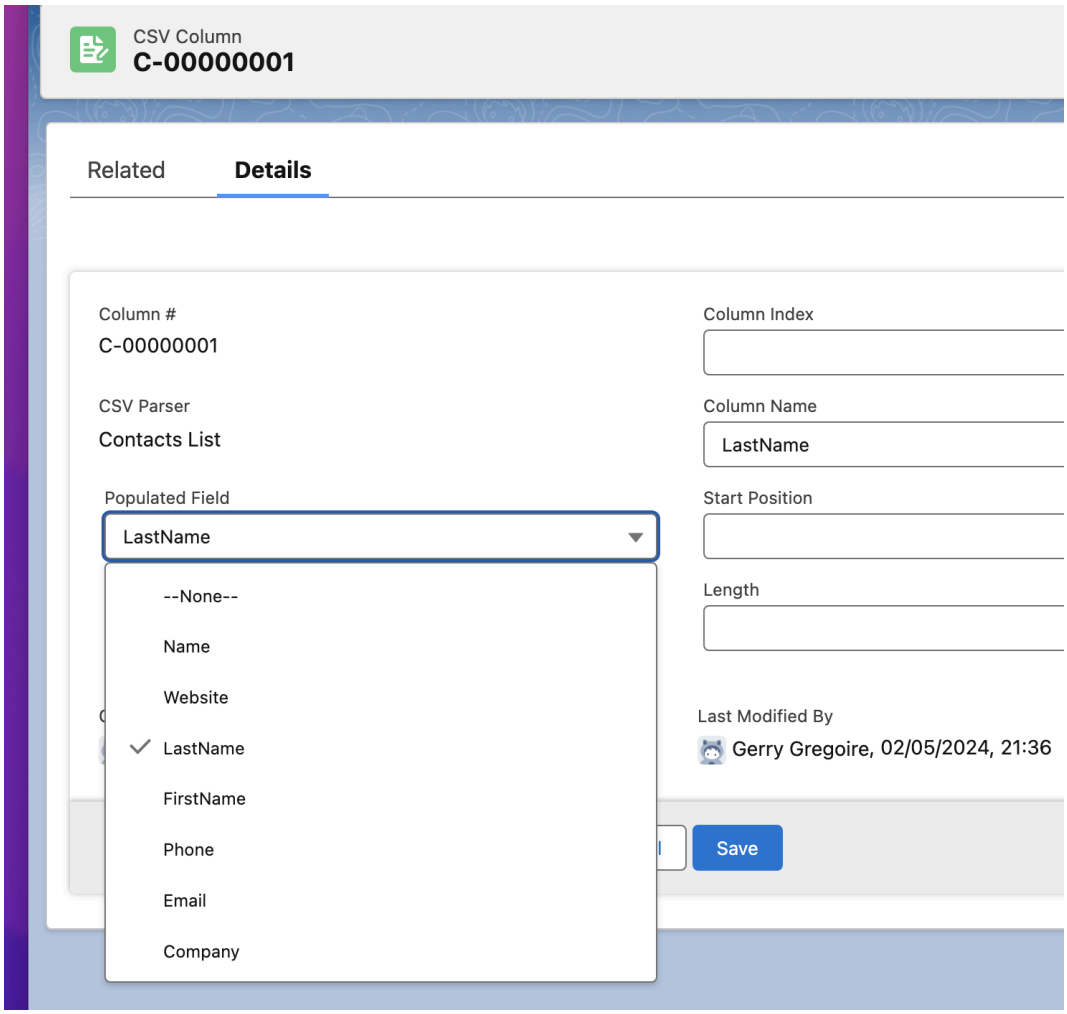
9. Assign Columns by either Name, Index or Position (We recommend Name)
10. Save

11. Next, you can create the CSV Columns that you wish to extract



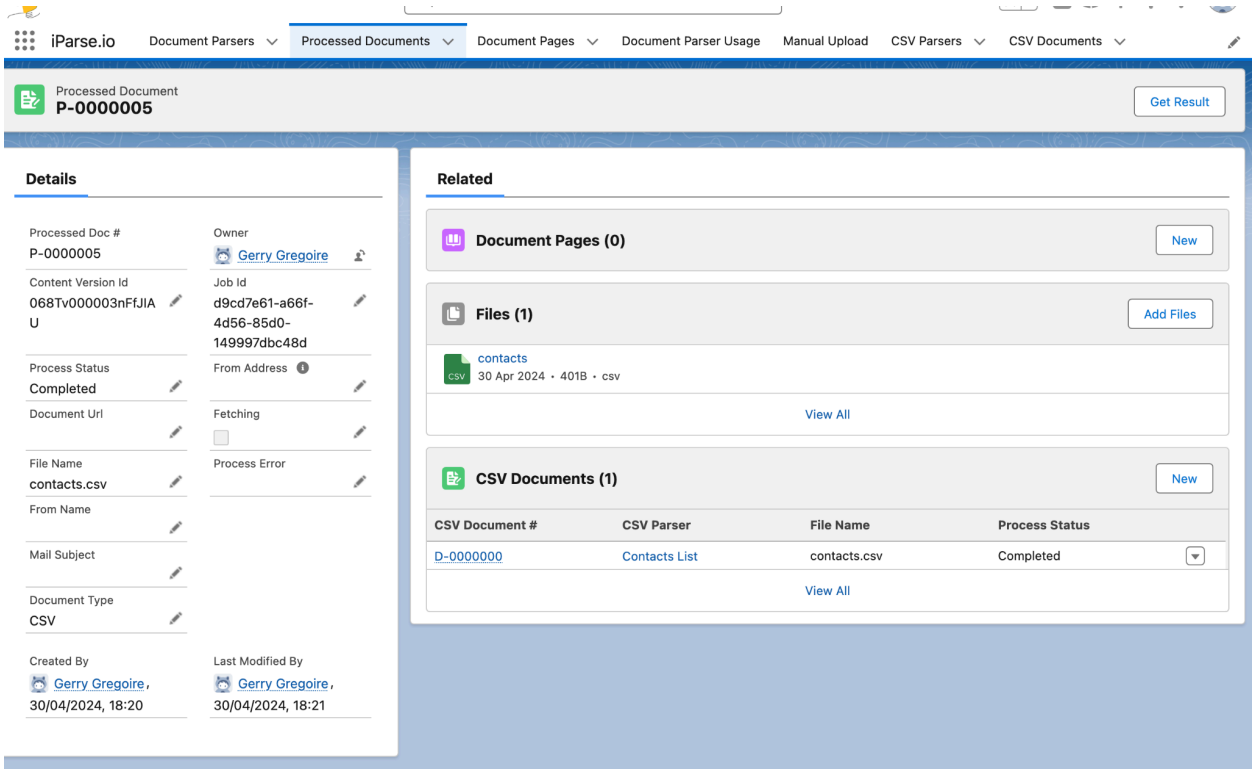
12. You can use ‘Trim Value’ to ensure any unwanted white space is removed.
13. If you are inserting the data into an Object, select a field you want the data to be inserted to (Note: You must add the API names of the object fields to the Populated_Field__c).

E.g. Below LastName is added which is the API name of the field on the Lead object.



Processing CSV Documents

1. You can email your CSV parser to the Email Service address you have set up. (Or manually upload by clicking into the ‘Manual Upload’ tab)
2. A Processed Document will be created (Note; This process run Asynchronously and so to trigger the process you can click on the the ‘Get Result’ button, otherwise it can take up to 15 min to process)
3. A CSV Document will be created (This is where the CSV data has been extracted to)



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..

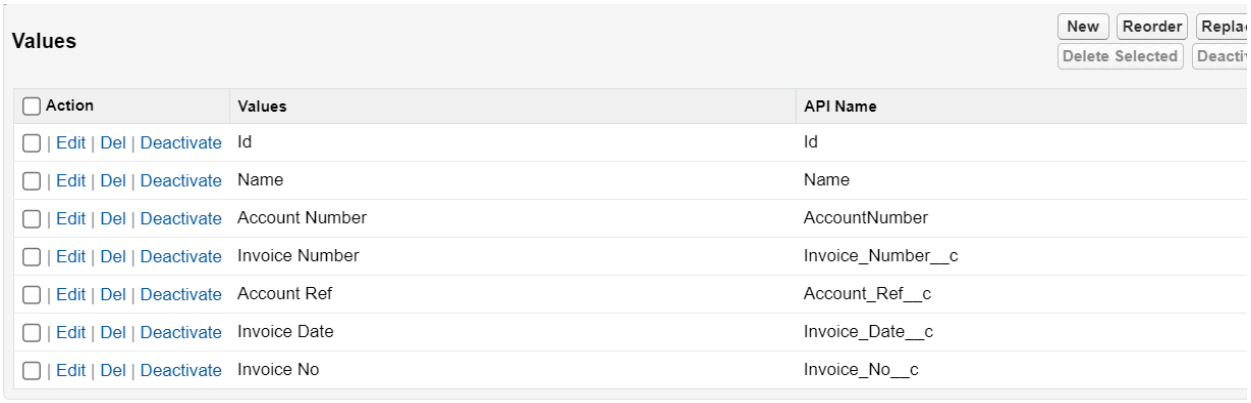
Setup Field Mappings

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.

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”).

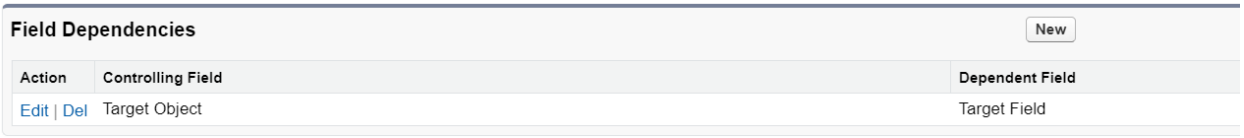


Values

New | Reorder | Repla
Delete Selected | Deacti

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

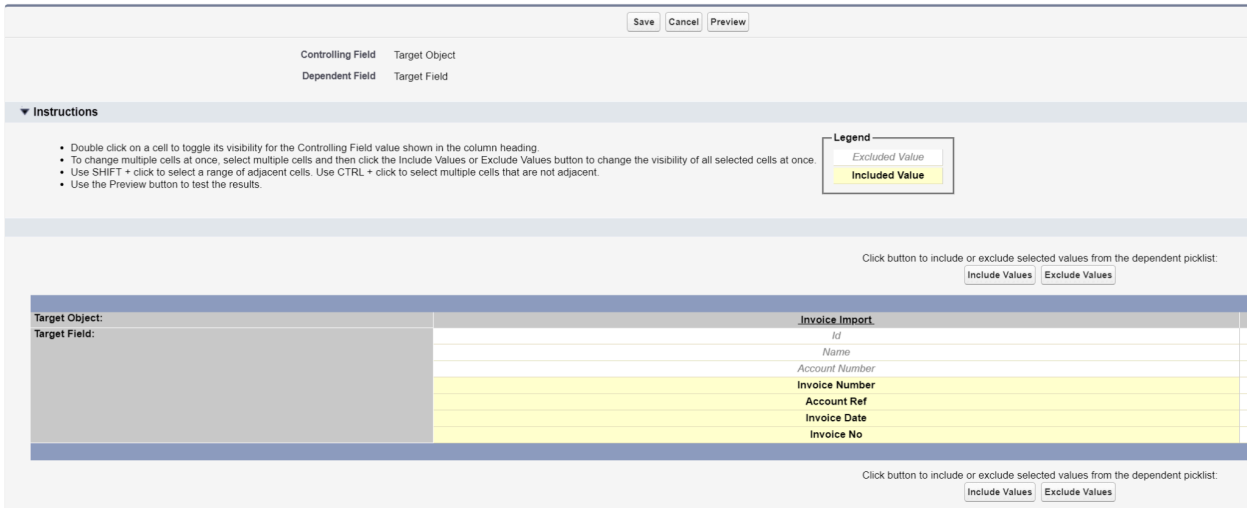


Field Dependencies

New

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

Edit Field Dependency



Save | Cancel | Preview

Controlling Field: Target Object
Dependent Field: Target Field

Instructions

- Double click on a cell to toggle its visibility for the Controlling Field value shown in the column heading.
- To change multiple cells at once, select multiple cells and then click the Include Values or Exclude Values button to change the visibility of all selected cells at once.
- Use SHIFT + click to select a range of adjacent cells. Use CTRL + click to select multiple cells that are not adjacent.
- Use the Preview button to test the results.

Legend: Excluded Value | Included Value

Click button to include or exclude selected values from the dependent picklist:
Include Values | Exclude Values

Target Object:	Invoice Import
Target Field:	Id
	Name
	Account Number
	Invoice Number
	Account Ref
	Invoice Date
	Invoice No

Click button to include or exclude selected values from the dependent picklist:
Include Values | Exclude Values

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.