Extract semi structured data activity

Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>May 21 2019</td>
<td>Antoine Chaussin</td>
<td>First release</td>
</tr>
<tr>
<td>1.1</td>
<td>August 26 2019</td>
<td>Antoine Chaussin</td>
<td>API Key property added. Improved Build Taxonomy</td>
</tr>
</tbody>
</table>

Overview

The Extract semi-structured data activity allows RPA developers to easily take advantage of UiPath’s machine learning models for semi-structured documents processing. The models currently can currently analyze invoices and receipts, providing various information (total price, vendor address, list of items, etc.). By simply specifying the image to process and the address of the model to use, the activity returns the model’s response already formatted for parsing or verification using UiPath’s Validation Station. *The activity is compatible with IntelligentOCR Activities version 2.0.0, previous versions are not supported.*

Activity properties
**Inputs**

- **Image path** (string): the path of the image you want to send to the machine learning model. The activity supports png, jpeg, bmp, tiff and pdf formats. **For pdf, please note that the document can only be 3 pages long.**

**Options**

- **Document Type** (string): Used in scenarios where multiple document types are used in the same Studio project. This field can be populated with a DocumentTypeID (which can be retrieved from Studio’s [Taxonomy Manager](#)) which will indicate to the activity what the expected structure of the document is. This is needed to generate the [Validation Station](#) objects. If left blank, the activity will look in the current project’s taxonomy for a default document type. If it cannot find one, the RawText, DocumentModel and AutomaticExtractionResults outputs will all be null.

**Outputs**

- **ModelResult** (ModelResult): a simple representation of the model’s response. The type of this output is a custom object which will be described below.

- **AutomaticExtractionResults** (ExtractionResult): Extracted data formatted to be compatible with UiPath’s [IntelligentOCR Activities](#)

- **RawText** (string): The full text extracted from the input document. Needed for [Validation Station](#)
• **DocumentModel** (Document): A Document Object Model representation of the input document. Needed for Validation Station

### Server

• **URL** (string): the URL of the model you are targeting. It must include the http:// or https:// header (e.g. https://invoices.uipath.com)

• **APIKey** (string): Authentication key provided by UiPath.

### Custom classes

**ModelResult**

A ModelResult is the top level representation of a response from a UiPath machine learning model. It has the following fields:

• **OCRPages** (Array of OCRPage): This field holds the raw representation of a document after it has been analyzed by an OCR engine. The information is split by pages (OCRPage), lines (OCRLine) and words (OCRWords). Each object holds the start index and length of its corresponding section (e.g. OCRPage.Length returns the number of characters in OCRPage). OCRWords have a Description and BoundingPoly field. Description holds the actual word after it has been processed by the OCR engine. This is relevant as some words can be slightly modified by the engine (e.g. dates get formatted). BoundingPoly holds a list of 4 vertices (xy coordinates) and a Description which represent the as-is word in the document and its location.
• **Fields** (Array of ModelField): All single value fields can be accessed from here. The ModelField object is described below.

• **TableFields** (Array of TableField): All multi value fields can be accessed from here. The TableField object is described below.

• **APIVersion** (string): the version of the UiPath model that has been used to produce this result.

ModelResult objects can be serialized to an indented json string by calling ModelResult.ToString.

---

**ModelField**

A ModelField represents a single value field in a document (e.g. the total amount in a receipt or the vendor address in an invoice). It has the following fields:

• **Name** (string): The field name as defined in the taxonomy. You can use the Taxonomy Manager to check and edit the value that will be set here.

• **Value** (string): The information in the document that corresponds to this field. Can be null if the field is not found.

• **Confidence** (string): If set, will always be a string containing a number between 0 and 1, 1 representing a 100% confidence in the result.

• **Boxes** (List of boxes): A cross reference to the OCR data held by the ModelResult. Holds the raw text and location of each word that composes the Value field.

• **Page** (int): The index of the page where you can find this field.
TableField
A TableField represents a multi value field in the model (e.g. the list of items bought in a receipt). It has the following fields:

- **Name** (string): The field name as defined in the taxonomy. You can use the Taxonomy Manager to check and edit the value that will be set here.

- **Items** (Array of TableItem): Each element of the array represents one value in this multi-value field. Following on the list of items bought in a receipt example, a receipt for apples, bananas and oranges will have three Items: one for apples, one for bananas and one for oranges. Each item will be a TableItem which wraps an array of ModelField (Fields). So in our example, the apples item will have multiple fields (description, price, quantity), so will the bananas and oranges ones.

Setup

*If you’re not going to use Validation Station or the IntelligentOCR Activities you can skip this part.*

In order to use various IntelligentOCR Activities, a document structure needs to be provided. We call this structure the *Document Type*. It contains the fields that are expected to appear in a document (e.g. in a receipt, we expect to see a total amount, a tax amount, a date, etc.). Studio projects that use IntelligentOCR need to build a *Taxonomy*, which is basically a list of *Document Types*. *Document Types* can be created using the Taxonomy Manager in Studio.
The activity can also build the *Document Type* for a specific endpoint automatically. After dragging the activity in your workflow, you can click the Build Taxonomy button to automatically generate a new *Document Type* or update an existing one. The After clicking the button, you will be prompted for the following information:

- **Group**: A *Document Type* in the taxonomy model belongs to a category which belongs to a group. This field defines the group of the *Document Type*
- **Category**: The category of the *Document Type*
- **Type**: The name of the *Document Type*. Documents are identified by their Group, Category and Type. If this triplet does not already exist in the *Taxonomy* a new *Document Type* will be created. If it does, the dialog box will display a warning and won’t let you validate. If you need to erase a document type, you can do it from the *Taxonomy Manager*
- **URL**: The URL of the endpoint you want to target, including the http:// or https:// header (e.g. https://invoices.uipath.com)

Once you’ve clicked the OK button, the new *Document Type* will appear in the activity’s drop down menu. It will also appear in the *Taxonomy Manager* where you will be able to edit field names and remove fields that are not needed in your validation.