PDF Tools

This plugin provides various Appian Cloud compatible Smart Services and functions for managing PDF files. Download the plugin from AppMarket on Appian Community.

Configuring the Merge PDF Smart Service

Merge multiple PDF files into a new consolidated PDF. Documents are merged in the order they're provided.

Data Tab

Input	Data Type	Required	Multiple	Description
Create New Document	Boolean	Yes	No	Whether to create a new document, or update an existing one.
New Document Name	Text	No	No	If creating a new document, use this as the name.
New Document Desc	Text	No	No	If creating a new document, use this as the description.
Save In Folder	Folder	No	No	If creating a new document, save into this folder.
Existing Document	Document	No	No	If not creating a new document, overwrite this one.
Documents	Document	Yes	Yes	The PDFs to merge together.
Pause On Error	Boolean	Yes	No	When set to false the process flow will continue when an error occurs.
Optimize Document	Boolean	Yes	No	When set to true the document will be scanned and duplicate objects (ex: embedded fonts) will be removed.

Output	Data Type	Multiple	Description

Output	Data Type	Multiple	Description
New Document Created	Document	No	A PDF of the merged Documents.
Error Occurred	Boolean	No	Returns true if an error has occurred. Save this value to a process variable to enable exception processing on the subsequent activity in the process flow.
Error Text	Text	No	Lists the text of the error message if one occurred.

Configuring the Extract PDF Pages Smart Service

Extract a number of pages from an existing PDF into a new PDF.

Data Tab

Input	Data Type	Required	Multiple	Description
Create New Document	Boolean	Yes	No	Whether to create a new document, or update an existing one.
New Document Name	Text	No	No	If creating a new document, use this as the name.
New Document Desc	Text	No	No	If creating a new document, use this as the description.
Save In Folder	Folder	No	No	If creating a new document, save into this folder.
Existing Document	Document	No	No	If not creating a new document, overwrite this one.
Document	Document	Yes	No	The PDF to extract pages from.
Start Page	Integer	Yes	No	The first page to extract. Must be equal or greater than 1.
End Page	Integer	Yes	No	The last page to extract. Must be equal or greater than Start Page.
Pause On Error	Boolean	Yes	No	When set to false the process flow will continue when an error occurs.

Output	Data Type	Multiple	Description
New Document Created	Document	No	A PDF of the extracted pages.
Error Occurred	Boolean	No	Returns true if an error has occurred. Save this value to a process variable to enable exception processing on the subsequent activity in the process flow.
Error Text	Text	No	Lists the text of the error message if one occurred.

Configuring the Fill PDF Smart Service

Populate a PDF form with the given values. Use the `getPdfFields` function to retrieve the existing PDF field names and values. PDF may be flattened to stop further edits being made to the form.

Data Tab

Input	Data Type	Required	Multiple	Description
Create New Document	Boolean	Yes	No	Whether to create a new document, or update an existing one.
New Document Name	Text	No	No	If creating a new document, use this as the name.
New Document Desc	Text	No	No	If creating a new document, use this as the description.
Save In Folder	Folder	No	No	If creating a new document, save into this folder.
Existing Document	Document	No	No	If not creating a new document, overwrite this one.
Document	Document	Yes	No	The PDF that contains the form to be filled.
Pdf Fields	PdfField	Yes	Yes	The fields and values to populate.
Flatten	Boolean	Yes	No	Whether to make the generated PDF uneditable or not.
Pause On Error	Boolean	Yes	No	When set to false the process flow will continue when an error occurs.

Output	Data Type	Multiple	Description
New Document Created	Document	No	A PDF with the form fields populated.
Error Occurred	Boolean	No	Returns true if an error has occurred. Save this value to a process variable to enable exception processing on the subsequent activity in the process flow.
Error Text	Text	No	Lists the text of the error message if one occurred.

Configuring the Convert PDF To Image Smart Service

This Smart Service allows a PDF document be converted to an image of type JPG or PNG and specified resolution. JPG is the default format if no format is provided. 300dpi will be used as the default resolution if no resolution is provided. The service will create an array of new images in the specified folder.

Data Tab

Input	Data Type	Required	Multiple	Description
DPI Resolution	Integer	No	No	Resolution value in DPI. Default 300.
Image Format	Text	Yes	No	Format of the output images. Only JPG, PNG, MP_TIFF & TIFF values are supported. Default JPG. MP_TIFF will create a single multi-page TIFF document.
Folder	Folder	No	No	Save into this folder.
Document	Document	No	No	PDF document to be converted.

Output	Data Type	Multiple	Description
Images Created	Document	Yes	An array of document types referencing the images created.

Configuring the Compress PDF Smart Service

Compressed the JPG images in a PDF to reduce the total file size.

Data Tab

Input	Data Type	Required	Multiple	Description
Create New Document	Boolean	Yes	No	Whether to create a new document, or update an existing one.
New Document Name	Text	No	No	If creating a new document, use this as the name.
New Document Desc	Text	No	No	If creating a new document, use this as the description.
Save In Folder	Folder	No	No	If creating a new document, save into this folder.
Existing Document	Document	No	No	If not creating a new document, overwrite this one.
Document	Document	Yes	No	The PDF to compress.
Quality	Integer	Yes	No	The quality in percent to use (0 - 100). Defaults to 50.
Pause On Error	Boolean	Yes	No	When set to false the process flow will continue when an error occurs.

Output	Data Type	Multiple	Description
New Document Created	Document	No	The compressed PDF.
Error Occurred	Boolean	No	Returns true if an error has occurred. Save this value to a process variable to enable exception processing on the subsequent activity in the process flow.
Error Text	Text	No	Lists the text of the error message if one occurred.

Configuring the Create PDF Smart Service

This Smart Service allows text to be added to a PDF with control over the position and style. An existing PDF can be updated or a new PDF can be created from scratch.

Data Tab

Input	Data Type	Required	Multiple	Description
Create New Document	Boolean	Yes	No	Whether to create a new document, or update an existing one.

Input	Data Type	Required	Multiple	Description
New Document Name	Text	No	No	If creating a new document, use this as the name.
New Document Desc	Text	No	No	If creating a new document, use this as the description.
Save In Folder	Folder	No	No	If creating a new document, save into this folder.
Existing Document	Document	No	No	If not creating a new document, overwrite this one.
Document	Document	No	No	The PDF to populate with content. This is optional.
Contents	PdfContent	Yes	Yes	Whether to make the generated PDF uneditable or not.
Pause On Error	Boolean	Yes	No	When set to false the process flow will continue when an error occurs.

Output	Data Type	Multiple	Description
New Document Created	Document	No	A PDF with the form fields populated.
Error Occurred	Boolean	No	Returns true if an error has occurred. Save this value to a process variable to enable exception processing on the subsequent activity in the process flow.
Error Text	Text	No	Lists the text of the error message if one occurred.

Configuring the Extract PDF Attachments Service

Extract attachments embedded within a PDF and save them in a folder.

Data Tab

Input	Data Type	Required	Multiple	Description
Document	Document	No	No	The document to extract attachments from
Save In Folder	Folder	No	No	The folder to save attachments in

Output	Data Type	Multiple	Description
Error Occurred	Boolean	No	Returns true if an error has occurred. Save this value to a process variable to enable exception processing on the subsequent activity in the process flow.
Error Text	Text	No	Lists the text of the error message
Attachments	Document	Yes	Array of documents that were embedded in the input document as attachments.
Contexts	Text	Yes	Parallel array to Attachments. Each value describes the PDF field that the corresponding attachment was pulled from

Configure the Encrypt PDF Service

This smart service encrypts an existing PDF with a password.

Data Tab

Input	Data Type	Required	Multiple	Description
Document to Encrypt	Document	Yes	No	The document to encrypt.
Destination Folder	Folder	Yes	No	The folder in which the password encrypted document will be saved..
Encrypted Document File Name	Text	Yes	No	The filename of the password encrypted document
Owner Password	Text	No	No	The password to open the document as owner, with all permissions enabled.
User Password	Text	Yes	No	The password to open the document as user with limited permissions.
Read Only	Boolean	No	No	If true, then the PDF will be read only.
Allow Printing	Boolean	No	No	If true, printing of the PDF will be allowed.

Output	Data Type	Multiple	Description
Encrypted Document	Document	No	The password encrypted PDF document
Error Message	Text	No	The error message if any encountered
Is Success	Boolean	Yes	If true the encryption is successful.

getPdfMetadata

Returns the PDF metadata attributes. (e.g page count, title, author, security, encryption, etc)

Syntax

```
getPdfMetadata(document)
```

- document: (Document) The PDF document to extract fields from.

getPdfBookmarks

Returns a list of all bookmarks associated with a PDF document

Syntax

```
getPdfBookmarks(document)
```

- document: (Document) The PDF document to extract bookmarks from.

getPdfText

Returns the text contained in the PDF.

Syntax

```
getPdfText(document)
```

- document: (Document) The PDF document to extract the text from.
- startPage: (Integer) The first page to start extracting from.
- endPage: (Integer) The last page to extract from.

getPdfFields

Returns an array of PdfField from the PDF.

Syntax

```
getPdfFields(document)
```

- document: (Document) The PDF document to extract fields from.

getPdfSignatures

Returns an array of PdfSignatureField from the PDF.

Syntax

```
getPdfSignatures(document)
```

- document: (Document) The PDF document to extract fields from.

Data Types

PdfField

Attribute	Data Type	Multiple	Description
name	Text	No	The name of the PDF form field.
value	Text	No	The PDF form field value.

PdfBookmark

Attribute	Data Type	Multiple	Description
name	Text	No	The name of the PDF bookmark.

PdfContent

Attribute	Data Type	Multiple	Description
page	Integer	No	The page in the PDF to update. Multiple PdfContent can reference the same page. Use 0 to create a new page (default). Use -1 to apply this content to all pages.
xPercent	Decimal	No	The percentage to start displaying content from the left side of the document in percent. Defaults to 5.
yPercent	Decimal	No	The percentage to start displaying content from the bottom of the document in percent. Defaults to 95 (e.g 5% from the top).
angle	Decimal	No	The angle of the content (e.g 0 - 360)
sections	PdfSection	Yes	The sections to display one after the other.

PdfSignature

Attribute	Data Type	Multiple	Description

Attribute	Data Type	Multiple	Description
name	Text	No	The name of the person who signed.
date	Date&Time	No	The time when it was signed.
reason	Text	No	The reason why it was signed.

PdfSignatureField

Attribute	Data Type	Multiple	Description
name	Text	No	The name of the PDF form field.
value	PdfSignature	No	The PdfSignature object

PdfSection

Attribute	Data Type	Multiple	Description
style	PdfStyle	No	The default style to apply to components.
components	Any Type	Yes	Any number of PdfText and PdfImage

PdfStyle

Attribute	Data Type	Multiple	Description
font	Text	No	The font name. Defaults to Helvetica.
fontSize	Decimal	No	The font size. Defaults to 12.
color	PdfColor	No	The text color in RGB. Defaults to black.

PdfColor

Attribute	Data Type	Multiple	Description
r	Integer	No	The amount of Red. Defaults to 0, max 255.
g	Integer	No	The amount of Green. Defaults to 0, max 255.
b	Integer	No	The amount of Blue. Defaults to 0, max 255.

PdfText

Attribute	Data Type	Multiple	Description
style	PdfStyle	No	The style for this component.
value	Text	No	The text to output to the PDF. Supported auto-replace keywords: $pageNum, $numPages

PdfImage

Attribute	Data Type	Multiple	Description
document	Document	No	The image document. Supported types: png, jpg, bmp and gif
quality	Integer	No	The quality in percent to store the image as.
scale	Decimal	No	The size of the image. If blank, the image will automatically be scaled down to fit onto the page. 1.0 = normal size, 2.0 = twice the size, 0.5 = half the size, etc

Packaging for Testing

Run the following command from the root folder to build everything or from a specific module folder to only build this module:

- `mvn clean package` . Use the extra argument `-DskipTests=true` to skip the JUnit tests
- Commit, squash and push all changes to your branch
- Merge the Pull Request to `master`

Packaging for Release

Perform the following steps from the root of the project:

- checkout `master` after all code has been reviewed and merged
- `mvn clean`
- `mvn release:clean`
- `mvn release:prepare -DautoVersionSubmodules=true -DpushChanges=false` . This command asks for a release version number and the number of the next snapshot. It also creates a tag to use in Github to upload the release. Optionally, use the parameter `-Darguments="-DskipTests"` to skip the tests when releasing. ONLY do so if all tests have been run successfully previously.
  - For the release version number, format it as `#.#.#` .
  - For the tag, format it as `v#.#.#` .
  - For the next snapshot, format like the release version number. This should be a single increment from the release version number entered earlier.
- Push generated commits to master `git push`
  - If you are on a detached head, create a new branch, push it to master, and create a release PR
- Push the auto generated tag to master `git push --tags`
- Create a new release on Github:
  - Navigate to the `Releases` section of the repo
  - Click "Draft a new release"
  - Populate "Tag version" with the new tag
  - Set the "Release title" as the tag version (v#.#.#)
  - In the section to "Describe this release", add the release notes (Enhancements & Bug Fixes)

    - Upload the ZIP artifact built by Maven in the section "Attach binaries by dropping..."
- On the Shared Components on Forum:
    - Got to "Related Actions" > "Update a Shared Component"
    - In the section "Description (Long..." add release notes under "Latest Release Notes"
    - Select "Yes" for "Update Component File or Attachements?" and click "Next"
    - Update "Component" with the new version