

Version: Quebec

API Reference - Server Side Scoped

ServiceNow provides JavaScript APIs for use within scripts running on the ServiceNow platform to deliver common functionality. This reference lists available classes and methods along with parameters, descriptions, and examples to make extending the ServiceNow platform easier.

Please note: The APIs below are intended for scoped applications and may behave differently in the global scope.

PDFGenerationAPI API

Provides support for PDF conversion and handling PDF fields.

This API is part of the ServiceNow PDF Generation Utilities plugin (com.snc.apppdfgenerator) and is provided within the `sn_pdfgeneratorutils` namespace. The plugin is activated by default.

These methods can also be used for documents created by non-catalog items. The methods in this class enable the following tasks:

- Dynamically generate a PDF from an HTML string and attach it to a record
- Fill fields in a PDF
- Sign a PDF
- Unflattened, flattened, or partially flattened
- Retrieving PDF field data

Related APIs:

- [PdfMergeSignRequestor](#) ([dev.do#!/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PdfMergeSignRequestorBothAPI](#))
- [SVGToPDFConversionAPI](#) ([dev.do#!/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/SVGToPDFConversionAPIBothAPI](#))

PDFGenerationAPI()

Instantiates a new PDFGenerationAPI object.

Example

The following shows how to create a PDFGenerationAPI object.

```
var v = new sn_pdfgeneratorutils.PDFGenerationAPI;
```

convertToPDF(String html, String targetTable, String targetTableSysId, String pdfName)

Converts an HTML string to a PDF document.

To generate a PDF with header and footer information, such as page numbers, use [convertToPDFWithHeaderFooter\(\)](#) ([dev.do#!/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-convertToPDFWithHdrFtr_S_S_S_S_O](#)).

Parameters

Name	Type	Description
html	String	HTML to convert to a PDF document.
targetTable	String	Name of the table on which to attach the converted PDF.
targetTableSysId	String	Sys_id of the record on which to attach the converted PDF.
pdfName	String	Name to give the PDF. Default: Sys_id of the PDF in the Attachments [sys_attachment] table.

Returns

Type	Description
Object	<p>Object containing sys_id of the PDF attachment if conversion is successful, error message otherwise.</p> <pre>{ "attachment_id": "String", "message": "String", "request_id": "String", "status": "String" }</pre>
<Object>.attachment_id	<p>If HTML conversion is successful, sys_id of the converted and attached PDF. The file is listed in the Attachments [sys_attachment] table.</p> <p>Data type: String</p>
<Object>.message	<p>Message confirming success or error.</p> <p>Possible values:</p> <ul style="list-style-type: none"> Conversion failed. – No PDF created. Make sure the values provided are accurate. Conversion is successful. – The HTML successfully converted to PDF. Exception while reading Source document contents. PDF header not found. – Input attachment provided is not a valid PDF. Provide the correct attachment sys_id. Given target record [<tableName> - <targetTableSysId>] does not exist. – Target table sys_id is not in the table provided. Make sure you include the correct table name for the record. No Form associated with pdf to fill. attachmentSysId: <sys_id> No editable fields exist with specified names. Please check and try again. field names: <field names> Request cannot proceed as the attachment with sys_id [{0}] did not pass security scan – The PDF did not pass the antivirus scan. Request cannot proceed as the attachment with sys_id [{0}] is pending security scan – The PDF requires an antivirus scan. Request completed successfully – Operation is successful. Undefined – Sys_id provided does not exist or is not a PDF attachment. <p>Data type: String</p>
<Object>.request_id	<p>Sys_id of the change producer request record.</p> <p>Data type: String</p>
<Object>.status	<p>Status indicating whether the operation is successful.</p> <p>Valid values:</p> <ul style="list-style-type: none"> success - Operation was successful. failure – Operation was not successful. The message provides details. <p>Data type: String</p>

Example

The following shows how to convert HTML to a PDF and attach it to a record in the Incident [incident] table.

```
var v = new sn_pdfgeneratorutils.PDFGenerationAPI;

// (Option) get HTML from the description field of an incident record
var gr = new GlideRecord("incident");
var html;

if (gr.get("<tableSysId>")) {
  html = gr.description.toString();
}

var result = v.convertToPDF(html, "incident", "<target_sys_id>", "myPDF");
gs.info(JSON.stringify(result));
```

```
{"attachment_id":"<sys_id>","message":"Conversion is successful","request_id":"<change_sys_id>","status":"success"}
```

convertToPDFWithHeaderFooter(String html, String targetTable, String targetTableSysId, String pdfName, Object headerFooterInfo)

Converts an HTML string into a PDF with header and footer content.

Use this method to generate PDFs with page settings:

- Header and footer information
- Margin sizes
- Orientation

- Enumeration
- Page size

Parameters

Name	Type	Description
html	String	HTML to convert to a PDF document.
targetTable	String	Name of the table on which to attach the converted PDF.
targetTableSysId	String	Sys_id of the record on which to attach the converted PDF.
pdfName	String	Name to give the PDF. Default: Sys_id of the PDF in the Attachments [sys_attachment] table.
headerFooterInfo	Object	Defines PDF header and footer details. <pre>{ "FooterImageAlignment": "String", "FooterImageAttachmentId": "String", "FooterImageHeight": "String", "FooterText": "String", "FooterTextAlignment": "String", "GeneratePageNumber": "String", "HeaderImageAlignment": "String", "HeaderImageAttachmentId": "String", "HeaderImageHeight": "String", "LeftOrRightMargin": "String", "PageOrientation": "String", "PageSize": "String", "TopOrBottomMargin": "String" }</pre>
headerFooterInfo.FooterImageAlignment	String	Sets the image position in the footer. Valid values: <ul style="list-style-type: none">BOTTOM_CENTER: Position the image in the bottom center of the footer.BOTTOM_LEFT: Position the image in the bottom left area of the footer.BOTTOM_RIGHT: Position the image in the bottom right area of the footer.TOP_CENTER: Position the image in the top center of the footer.TOP_LEFT: Position the image in the top left area of the footer.TOP_RIGHT: Position the image in the top right area of the footer.
headerFooterInfo.FooterImageAttachmentId	String	Sys_id of the footer image in the Attachments [sys_attachment] table. To determine if the file type is supported in your instance, Navigate to System Properties > Security and check if it's listed in List of file extensions (comma-separated) that can be attached field.
headerFooterInfo.FooterImageHeight	String	Height of footer image. Default: 50 points
headerFooterInfo.FooterText	String	Footer text to place at the bottom of each PDF page.
headerFooterInfo.FooterTextAlignment	String	Sets the text position in the footer. Make sure this value does not match or conflict with the area provided in headerFooterInfo.FooterImageAlignment. Valid values: <ul style="list-style-type: none">BOTTOM_CENTER: Position the text in the bottom center of the footer.BOTTOM_LEFT: Position the text in the bottom left area of the footer.BOTTOM_RIGHT: Position the text in the bottom right area of the footer.TOP_CENTER: Position the text in the top center of the footer.TOP_LEFT: Position the text in the top left area of the footer.TOP_RIGHT: Position the text in the top right area of the footer.
headerFooterInfo.GeneratePageNumber	String	Flag that indicates whether to generate a PDF page number. Valid values: <ul style="list-style-type: none">true: Generate page numbers.false: Do not generate page numbers. Default: true
headerFooterInfo.HeaderImageAlignment	String	Sets the image position in the header. Valid values: <ul style="list-style-type: none">center: Position the image in the center of the header.left: Position the image on the left side of the header.right: Position the image on the right side of the header.
headerFooterInfo.HeaderImageAttachmentId	String	Sys_id of the header image in the Attachments [sys_attachment] table. To determine if the file type is supported in your instance, Navigate to System Properties > Security and check if it's listed in List of file extensions (comma-separated) that can be attached field.
headerFooterInfo.HeaderImageHeight	String	Height of the header image. Default: 50 points

headerFooterInfo.LeftOrRightMargin	String	Size of the left and right margins. If positioned in the left or right side of the page, header/footer details are placed within in this area. Default: 36 points
headerFooterInfo.PageOrientation	String	Page orientation. Valid values: <ul style="list-style-type: none">• PORTRAIT• LANDSCAPE Default: Portrait
headerFooterInfo.PageSize	String	Document page size. Valid values: <ul style="list-style-type: none">• A4 – 595 × 842 points• LETTER – 612 × 792 points• LEDGER – 792 x 1224 points
headerFooterInfo.TopOrBottomMargin	String	Size of the top and bottom margins. Header and footer details are placed within in this area. Default: 72 points

Returns

Type	Description
Object	<p>Object containing sys_id of the PDF attachment if conversion is successful, error message otherwise.</p> <pre>{ "attachment_id": "String", "message": "String", "request_id": "String", "status": "String" }</pre>
<Object>.attachment_id	<p>If HTML conversion is successful, sys_id of the converted and attached PDF. The file is listed in the Attachments [sys_attachment] table.</p> <p>Data type: String</p>
<Object>.message	<p>Message confirming success or error. Possible values:</p> <ul style="list-style-type: none">• Conversion failed. – No PDF created. Make sure the values provided are accurate.• Conversion is successful. – The HTML successfully converted to PDF.• Footer Image alignment and text alignment cannot be in the same region with same alignment: <footerImageAlignment value> – Make sure that headerFooterInfo.FooterImageAlignment and headerFooterInfo.FooterTextAlignment values are not in the same area.• Exception while reading Source document contents. PDF header not found. – Input attachment provided is not a valid PDF. Provide the correct attachment sys_id.• Given target record [<tableName> - <targetTableSysId>] does not exist. – Target table sys_id is not in the table provided. Make sure you include the correct table name for the record.• Invalid footer image alignment: <invalid_option> is provided. – Provide a valid option in the headerFooterInfo.FooterImageAlignment property.• Invalid footer text alignment: " + <invalid_option> + " is provided. – Provide a valid option in the headerFooterInfo.footerTextAlignment property.• No Form associated with pdf to fill. attachmentSysId: <sys_id>• No editable fields exist with specified names. Please check and try again. field names: <field names>• Request cannot proceed as the attachment with sys_id [{0}] did not pass security scan – The PDF did not pass the antivirus scan.• Request cannot proceed as the attachment with sys_id [{0}] is pending security scan – The PDF requires an antivirus scan.• Request completed successfully – Operation is successful.• Unable to get the footer image. sysId: + <value provided> – Make sure the sys_id provided for headerFooterInfo.footerImageId is accurate.• Unable to get the header image. sysId: + <value provided> – Make sure the sys_id provided for headerFooterInfo.headerImageId is accurate.• Undefined – Sys_id provided does not exist or is not a PDF attachment. <p>Data type: String</p>
<Object>.request_id	<p>Sys_id of the change producer request record.</p> <p>Data type: String</p>
<Object>.status	<p>Status indicating whether the operation is successful. Valid values:</p> <ul style="list-style-type: none">• success - Operation was successful.• failure – Operation was not successful. The message provides details. <p>Data type: String</p>

Example

The following shows how to convert HTML to a PDF named "myPDF" and add the PDF as an attachment to a record in the Incident [incident] table. The PDF contains header and footer provided via attachment.

```
var v = new sn_pdfgeneratorutils.PDFGenerationAPI;

// (Option) get HTML from the description field of an incident record
var gr = new GlideRecord("incident");
var html;

if (gr.get("<tableSysId>")) {
    html = gr.description.toString();
}

var hfInfo = new Object();
hfInfo["HeaderImageAttachmentId"] = "<hdrImgAttSysId>";
hfInfo["HeaderImageAlignment"] = "left";
hfInfo["FooterImageAttachmentId"] = "<ftrImgAttSysId>";
hfInfo["FooterImageAlignment"] = "TOP_CENTER";
hfInfo["FooterText"] = "Sample Footer Message";
hfInfo["PageSize"] = "A4";
hfInfo["GeneratePageNumber"] = "false";
hfInfo["TopOrBottomMargin"] = "36";
hfInfo["LeftOrRightMargin"] = "24";

var result = v.convertToPDFWithHeaderFooter(html, "incident", "<targetTbl_sys_id>", "myPDF", hfInfo);
gs.info(JSON.stringify(result));

{"attachment_id":"<sys_id>","message":"Conversion is successful.","request_id":"<change_sys_id>","status":"success"}
```

`fillDocumentFields(Object fieldsMap, String sysId, String tableName, String tableSysId, String pdfName)`

Fills fields in an editable PDF and attaches it to the provided record.

Use the following methods to determine if the PDF is fillable and get field information:

- [isDocumentFillable\(\)](#) (dev.do#/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-isDocumentFillable_S)
- [getDocumentFields\(\)](#) (dev.do#/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-getDocumentFields_S)
- [getDocumentFieldsType\(\)](#) (dev.do#/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-getDocumentFieldsType_S)

PDFGenerationAPI provides additional fill methods with different options:

- [fillDocumentFieldsAndFlatten\(\)](#) (dev.do#/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-fillDocFieldsFlatten_O_S_S_S_S_O_O) – Fills fields in an editable PDF, flattens the data fields, and attaches it to the provided record.
- [fillFieldsAndMergeSignature\(\)](#) (dev.do#/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-fillFieldsMergeSig_O_S_S_S_S_O_O_O) – Fills fields in an editable PDF, adds signature image, flattens the data fields, and attaches it to the provided record.
- [getFilledDocumentWithSignatureAsBase64\(\)](#) (dev.do#/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-getFilDocWithSigAsBase64_O_S_O_O_O) – Fills fields in an editable PDF, creates an image, and converts it to a Base64-encoded PDF.

Parameters

Name	Type	Description
fieldsMap	Object	Optional. Key value map by PDF field name and value to fill. Use the <code>getDocumentFields()</code> method to get the list of available fields.
sysId	String	Sys_id of a PDF in the Attachments [sys_attachment] table.
tableName	String	Name of the table containing the record to which the PDF is attached. You can find this value in the same row as the attachment listed in the Attachments [sys_attachment] table.
tableSysId	String	Sys_id of the record to which the PDF is attached. You can find this value in the same row as the attachment listed in the Attachments [sys_attachment] table.
pdfName	String	Name to give the PDF. Default: Sys_id of the PDF in the Attachments [sys_attachment] table.

Returns

Type	Description
	Object containing sys_id of the updated PDF attachment if successful, error message otherwise.
Object	<pre>{ "attachment_id": "String", "message": "String", "status": "String" }</pre>
<Object>.attachment_id	<p>If the operation is successful, sys_id of the filled PDF. The file is listed in the Attachments [sys_attachment] table.</p> <p>Data type: String</p>
<Object>.message	<p>Message confirming success or error.</p> <p>Valid values:</p> <ul style="list-style-type: none">Exception while reading Source document contents. PDF header not found. – Input attachment provided is not a valid PDF. Provide the correct attachment sys_id.Given target record [<tableName> - <targetTableSysId>] does not exist. – Target table sys_id is not in the table provided. Make sure you include the correct table name for the record.No Form associated with pdf to fill. attachmentSysId: <sys_id>No editable fields exist with specified names. Please check and try again. field names: <field names>Request cannot proceed as the attachment with sys_id [{0}] did not pass security scan – The PDF did not pass the antivirus scan.Request cannot proceed as the attachment with sys_id [{0}] is pending security scan – The PDF requires an antivirus scan.Request completed successfully – Operation is successful.Undefined – Sys_id provided does not exist or is not a PDF attachment. <p>Data type: String</p>
<Object>.status	<p>Status indicating whether the operation is successful.</p> <p>Valid values:</p> <ul style="list-style-type: none">success - Operation was successful.failure – Operation was not successful. The message provides details. <p>Data type: String</p>

Example

The following shows how to fill fields in an editable PDF.

```
var fieldMap = new Object();
fieldMap["Address"] = "Address value here";
fieldMap["State"] = "State value here";

var v = new sn_pdfgeneratorutils.PDFGenerationAPI;
var result = v.fillDocumentFields(fieldMap, "<attachmentSysId>", "<tableName>", "<tableSysId>", "pdfName");
gs.info(JSON.stringify(result));
```

```
{"attachment_id":"<sys_id>","message":"Request completed successfully.", "status":"success"}
```

`fillDocumentFieldsAndFlatten(Object fieldsMap, String sysId, String tableName, String tableSysId, String pdfName, Object flatten)`

Fills fields in an editable PDF, flattens the data fields, and attaches it to the provided record.

Use the following methods to determine if the PDF is fillable and get field information:

- [isDocumentFillable\(\)](#) (dev.do#/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-isDocumentFillable_S)
- [getDocumentFields\(\)](#) (dev.do#/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-getDocumentFields_S)
- [getDocumentFieldsType\(\)](#) (dev.do#/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-getDocumentFieldsType_S)

PDFGenerationAPI provides additional fill methods with different options:

- [fillDocumentFields\(\)](#) (dev.do#/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-fillDocumentFields_O_S_S_S_S_S) – Fills fields in an editable PDF and attaches it to the provided record.
- [fillFieldsAndMergeSignature\(\)](#) (dev.do#/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-fillFieldsMergeSig_O_S_S_S_S_O_O) – Fills fields in an editable PDF, adds signature image, flattens the data fields, and

attaches it to the provided record.

- `getFilledDocumentWithSignatureAsBase64()` (dev.do#/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-getFilDocWithSigAsBase64_O_S_O_O) – Fills fields in an editable PDF, creates an image, and converts it to a Base64-encoded PDF.

Parameters

Name	Type	Description
fieldsMap	Object	Optional. Key value map by PDF field name and value to fill. Use the <code>getDocumentFields()</code> method to get the list of available fields.
sysId	String	Sys_id of a PDF in the Attachments [sys_attachment] table.
tableName	String	Name of the table containing the record to which the PDF is attached. You can find this value in the same row as the attachment listed in the Attachments [sys_attachment] table.
tableSysId	String	Sys_id of the record to which the PDF is attached. You can find this value in the same row as the attachment listed in the Attachments [sys_attachment] table.
pdfName	String	Name to give the PDF. Default: Sys_id of the PDF in the Attachments [sys_attachment] table.
flatten	Object	Optional. Flattening fields enable locking the fields so that other users cannot change the information. Specify the key as "FlattenType" and provide a flattening option as a string. Valid values: <ul style="list-style-type: none">• <code>do_not_flatten</code> - Do not flatten any fields.• <code>partially_flatten</code> - Flatten only the fields which are modified.• <code>fully_flatten</code> - Flattens all the fields. Default: <code>fully_flatten</code> <pre>{ "FlattenType": "String"}</pre>

Returns

Type	Description
Object	Object containing sys_id of the updated PDF attachment if successful, error message otherwise. <pre>{ "attachment_id": "String", "message": "String", "status": "String"}</pre>
<Object>.attachment_id	If the operation is successful, sys_id of the filled PDF. The file is listed in the Attachments [sys_attachment] table. Data type: String
<Object>.message	Message confirming success or error. Valid values: <ul style="list-style-type: none">• Exception while reading Source document contents. PDF header not found. – Input attachment provided is not a valid PDF. Provide the correct attachment sys_id.• Given target record [<tableName> - <targetTableSysId>] does not exist. – Target table sys_id is not in the table provided. Make sure you include the correct table name for the record.• No Form associated with pdf to fill. attachmentSysId: <sys_id>• No editable fields exist with specified names. Please check and try again. field names: <field names>• Request cannot proceed as the attachment with sys_id [{0}] did not pass security scan – The PDF did not pass the antivirus scan.• Request cannot proceed as the attachment with sys_id [{0}] is pending security scan – The PDF requires an antivirus scan.• Request completed successfully – Operation is successful.• Undefined – Sys_id provided does not exist or is not a PDF attachment. Data type: String
<Object>.status	Status indicating whether the operation is successful. Valid values: <ul style="list-style-type: none">• <code>success</code> - Operation was successful.• <code>failure</code> – Operation was not successful. The message provides details. Data type: String

Example

The following shows how to fill fields and flatten an editable PDF.

```
var fieldMap = new Object();
fieldMap["Last Name First Name Middle Initial"] = "Tuter Abel E.";
fieldMap["Date of Birth"] = "08101952";
fieldMap["US SSN"] = "111-22-9999";
fieldMap["Address"] = "PO Box 344";
fieldMap["City"] = "Jerome";
fieldMap["State"] = "AZ";
fieldMap["Zip"] = "86331";

var flatten = new Object();
flatten["FlattenType"] = "partially_flatten";

var v = new sn_pdfgeneratorutils.PDFGenerationAPI;
var result = v.fillDocumentFieldsAndFlatten(fieldMap, "<attachmentSysId>", "<tableName>", "<tableSysId>", "pdfName", flatten);
gs.info(JSON.stringify(result));
```

```
"attachment_id":"<sys_id>","message":"Request completed successfully.,"status":"success"
```

`fillFieldsAndMergeSignature(Object fieldsMap, String sysId, String tableName, String tableSysId, String pdfName, PdfMergeSignRequestor requestor, Object flatten)`

Fills fields in an editable PDF, adds signature image, flattens the data fields, and attaches it to the provided record.

Use the following methods to determine if the PDF is fillable and get field information:

- `isDocumentFillable()` ([dev.do#!/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-isDocumentFillable_S](#))
- `getDocumentFields()` ([dev.do#!/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-getDocumentFields_S](#))
- `getDocumentFieldsType()` ([dev.do#!/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-getDocumentFieldsType_S](#))

PDFGenerationAPI provides additional fill methods with different options:

- `fillDocumentFields()` ([dev.do#!/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-fillDocumentFields_O_S_S_S_S](#)) – Fills fields in an editable PDF and attaches it to the provided record.
- `fillDocumentFieldsAndFlatten()` ([dev.do#!/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-fillDocFieldsFlatten_O_S_S_S_S_O](#)) – Fills fields in an editable PDF, flattens the data fields, and attaches it to the provided record.
- `getFilledDocumentWithSignatureAsBase64()` ([dev.do#!/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-getFilDocWithSigAsBase64_O_S_O_Q](#)) – Fills fields in an editable PDF, creates an image, and converts it to a Base64-encoded PDF.

Parameters

Name	Type	Description
fieldsMap	Object	Optional. Key value map by PDF field name and value to fill. Use the getDocumentFields() method to get the list of available fields.
sysId	String	Sys_id of a PDF in the Attachments [sys_attachment] table.
tableName	String	Name of the table containing the record to which the PDF is attached. You can find this value in the same row as the attachment listed in the Attachments [sys_attachment] table.
tableSysId	String	Sys_id of the record to which the PDF is attached. You can find this value in the same row as the attachment listed in the Attachments [sys_attachment] table.
pdfName	String	Name to give the PDF. Default: Sys_id of the PDF in the Attachments [sys_attachment] table.
requestor	PdfMergeSignRequestor (dev.do#/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PdfMergeSignRequestorBothAPI)	Signature input returned from pdfMergeSignRequestor.
flatten	Object	<p>Optional. Flattening fields enable locking the fields so that other users cannot change the information. Specify the key as "FlattenType" and provide a flattening option as a string. Valid values:</p> <ul style="list-style-type: none"> • donot_flatten - Do not flatten any fields. • partially_flatten - Flatten only the fields which are modified. • fully_flatten - Flattens all the fields. <p>Default: fully_flatten</p> <pre>{ "FlattenType": "String" }</pre>

Returns

Type	Description
Object	<p>Object containing sys_id of the updated PDF attachment if successful, error message otherwise.</p> <pre>{ "attachment_id": "String", "message": "String", "status": "String" }</pre>
<Object>.attachment_id	<p>If the operation is successful, sys_id of the filled PDF. The file is listed in the Attachments [sys_attachment] table.</p> <p>Data type: String</p>
<Object>.message	<p>Message confirming success or error.</p> <p>Valid values:</p> <ul style="list-style-type: none"> Exception while reading Source document contents. PDF header not found. – Input attachment provided is not a valid PDF. Provide the correct attachment sys_id. Given target record [<tableName> - <targetTableSysId>] does not exist. – Target table sys_id is not in the table provided. Make sure you include the correct table name for the record. No Form associated with pdf to fill. attachmentSysId: <sys_id> No editable fields exist with specified names. Please check and try again. field names: <field names> Request cannot proceed as the attachment with sys_id [{0}] did not pass security scan – The PDF did not pass the antivirus scan. Request cannot proceed as the attachment with sys_id [{0}] is pending security scan – The PDF requires an antivirus scan. Request completed successfully – Operation is successful. Undefined – Sys_id provided does not exist or is not a PDF attachment. <p>Data type: String</p>
<Object>.status	<p>Status indicating whether the operation is successful.</p> <p>Valid values:</p> <ul style="list-style-type: none"> success - Operation was successful. failure – Operation was not successful. The message provides details. <p>Data type: String</p>

Example

The following shows how to fill fields with signature with default settings to completely flatten the fields.

```
var fieldMap = new Object();
fieldMap["Address_Salutation"] = "Address value here";

var paramMap = new Object();
paramMap["FlattenType"] = "partially_flatten";

var requestor = new sn_pdfgeneratorutils.PdfMergeSignRequestor;
requestor.createRequest("<attachmentSysId>", "incident", "<tableSysId>", "filledPdf");
requestor.addSignatureMapping(6, 40, 50, 188, 44, "<signatureSysId>");

var v = new sn_pdfgeneratorutils.PDFGenerationAPI;
var result = v.fillFieldsAndMergeSignature(fieldMap, "<attachmentSysId>", "incident", "<tableSysId>", requestor, "filledPdf", paramMap);
gs.info(JSON.stringify(result));
```

```
{"attachment_id":"5440d993dbed3010d66be1191396194e","message":"Request completed successfully.", "status":"success"}
```

getDocumentFields(String sysId)

Gets a list of editable fields in a PDF document. Enables listing editable PDF fields without manually opening the file to check.

Parameters

Name	Type	Description
sysId	String	Sys_id of a PDF in the Attachments [sys_attachment] table.

Returns

Type	Description
Object	<p>Object containing ID of the signed PDF, error message otherwise.</p> <pre>{ "attachment_id": "String", "message": "String", "status": "String" }</pre>
<Object>.fields	<p>If the request is successful, list containing the name of each field in the PDF.</p> <p>Data type: Array of strings</p> <pre>"fields": ["field_name"]</pre>
<Object>.message	<p>Message confirming success or error.</p> <p>Possible values:</p> <ul style="list-style-type: none">Exception while reading Source document contents. PDF header not found. – Input attachment provided is not a valid PDF. Provide the correct attachment sys_id.Request cannot proceed as the attachment with sys_id {{0}} did not pass security scan – The PDF did not pass the antivirus scan.Request cannot proceed as the attachment with sys_id {{0}} is pending security scan – The PDF requires an antivirus scan.Request completed successfully – Operation is successful.Undefined – Sys_id provided does not exist or is not a PDF attachment. <p>Data type: String</p>
<Object>.status	<p>Status indicating whether the operation is successful.</p> <p>Valid values:</p> <ul style="list-style-type: none">success - Operation was successful.failure – Operation was not successful. The message provides details. <p>Data type: String</p>

Example

The following shows how to retrieve fields in a PDF attachment.

```
var v = new sn_pdfgeneratorutils.PDFGenerationAPI;
var result = v.getDocumentFields("attachmentSysId");
gs.info(JSON.stringify(result));
```

```
{"message":"Request completed successfully.", "fields":["NP_formFillable","reset","print","1SSN","Signature.1","5sigDate","Check Box21"], "status":"success"}
```

getDocumentFieldsType(String sysId)

Gets the field type of set of editable fields from a PDF document.

Parameters

Name	Type	Description
sysId	String	Sys_id of a PDF in the Attachments [sys_attachment] table.

Returns

Type	Description
Object	<p>Object containing each PDF field type if successful, error message otherwise.</p> <pre>{ "fields_type": (Object), "message": "String", "status": "String" }</pre>
<Object>.fields_type	<p>Object listing each field in the specified PDF if successful, error message otherwise.</p> <p>Data type: Object</p> <pre>"fields_type": { "<field type>": (Object), }</pre>
<Object>.fields_type.<field>	<p>Object containing page number of each field. The name represents the field label, for example, "SSN", or an automated label representing the type.</p> <p>Data type: Object</p> <pre>"<field>": { "fieldsDetails": [Array], // Check boxes, radio buttons, choice boxes only "pageNumber": "String", "type": "String" }</pre>
<Object>.fields_type.<field>.fieldsDetails	<p>List of objects containing field name and corresponding value of each option for choice field types. Applicable types:</p> <ul style="list-style-type: none">• Check box• Choice box• Combo box• Multi select choice box <p>Data type: Array</p> <pre>"fieldsDetails": ["fieldName": "String", "value": "String"]</pre>
<Object>.fields_type.<field>.fieldsDetails.fieldName	<p>Name of a choice field.</p> <p>Data type: String</p>
<Object>.fields_type.<field>.fieldsDetails.value	<p>Value of a choice field.</p> <p>Data type: String</p>
<Object>.fields_type.<field>.pageNumber	<p>PDF page number corresponding to this field.</p> <p>Data type: String</p>
<Object>.fields_type.<field>.type	<p>PDF field type.</p> <p>Possible values:</p> <ul style="list-style-type: none">• check_box• choice_box• combo_box• multi_select_choice_box• push_button• radio_button• signature• text <p>Data type: String</p>

<p><Object>.message</p>	<p>Message confirming success or error. Valid values:</p> <ul style="list-style-type: none"> Exception while reading Source document contents. PDF header not found. – Input attachment provided is not a valid PDF. Provide the correct attachment sys_id. Request cannot proceed as the attachment with sys_id {{0}} did not pass security scan – The PDF did not pass the antivirus scan. Request cannot proceed as the attachment with sys_id {{0}} is pending security scan – The PDF requires an antivirus scan. Request completed successfully – Operation is successful. Undefined – Sys_id provided does not exist or is not a PDF attachment. <p>Data type: String</p>
<p><Object>.status</p>	<p>Status indicating whether the operation is successful. Valid values:</p> <ul style="list-style-type: none"> success - Operation was successful. failure – Operation was not successful. The message provides details. <p>Data type: String</p>

Example

The following shows how to retrieve field types in a PDF attachment. Results include manual returns for readability and are truncated for brevity.

```
var v = new sn_pdfgeneratorutils.PDFGenerationAPI;
var result = v.getDocumentFieldsType("<attachmentSysId>");
gs.info(JSON.stringify(result));
```

```
{
  "fields_type": {
    "1ADDLINE2.25": {
      "pageNumber": 2,
      "type": "text"
    },
    "1ADDLINE2.24": {
      "pageNumber": 2,
      "type": "text"
    },
    "1ADDLINE2.23": {
      "pageNumber": 2,
      "type": "text"
    },
    "1ADDLINE2.22": {
      "pageNumber": 2,
      "type": "text"
    },
    "1ADDLINE2.11": {
      "pageNumber": 2,
      "type": "text"
    },
    "Check Box1": {
      "fieldsDetails": {
        "fieldName": "Yes"
      },
      "pageNumber": 2,
      "type": "check_box"
    },
    "4consentDate.6": {
      "pageNumber": 4,
      "type": "text"
    },
    "4consentDate.7": {
      "pageNumber": 4,
      "type": "text"
    },
    "3SSN.9": {
      "pageNumber": 3,
      "type": "text"
    },
    "3SSN.8": {
      "pageNumber": 3,
      "type": "text"
    },
    "3SSN.7": {
      "pageNumber": 3,
      "type": "text"
    },
    "pageNumber": 2,
    "type": "check_box",
    "Check Box8": {
      "fieldsDetails": {
        "fieldName": "Off",
        "fieldName": "yes"
      },
      "pageNumber": 2,
      "type": "check_box"
    },
    "4planAdminDate.8": {
      "pageNumber": 4,
      "type": "text"
    },
    "4planAdminDate.7": {
      "pageNumber": 4,
      "type": "text"
    },
    "1FirstName_ID.7": {
      "pageNumber": 2,
      "type": "text"
    },
    "Check Box9": {
      "fieldsDetails": {
        "fieldName": "Yes"
      },
      "pageNumber": 3,
      "type": "check_box"
    },
    "1LN.1": {
      "pageNumber": 2,
      "type": "text"
    },
    "1LN.2": {
      "pageNumber": 2,
      "type": "text"
    },
    "Check Box11": {
      "fieldsDetails": {
        "fieldName": "Yes"
      },
      "pageNumber": 3,
      "type": "check_box"
    },
    "1LN.9": {
      "pageNumber": 2,
      "type": "text"
    },
    "Check Box17": {
      "fieldsDetails": {
        "fieldName": "Yes"
      },
      "pageNumber": 3,
      "type": "check_box"
    },
    "Check Box16": {
      "fieldsDetails": {
        "fieldName": "Yes"
      },
      "pageNumber": 3,
      "type": "check_box"
    },
    "1LN.7": {
      "pageNumber": 2,
      "type": "text"
    },
    "Check Box19": {
      "fieldsDetails": {
        "fieldName": "Yes"
      },
      "pageNumber": 2,
      "type": "text"
    },
    "1LN.8": {
      "pageNumber": 2,
      "type": "text"
    },
    "Check Box18": {
      "fieldsDetails": {
        "fieldName": "Yes"
      },
      "pageNumber": 2,
      "type": "text"
    },
    "print": {
      "pageNumber": 2,
      "type": "push_button",
      "4planAdministrator.1": {
        "pageNumber": 4,
        "type": "text"
      },
      "1TaxID.9": {
        "pageNumber": 2,
        "type": "text"
      },
      "4SSN.1": {
        "pageNumber": 3,
        "type": "text"
      },
      "4SSN.2": {
        "pageNumber": 3,
        "type": "text"
      },
      "Signature.1": {
        "pageNumber": 4,
        "type": "text"
      },
      "1ZIP.2": {
        "pageNumber": 2,
        "type": "text"
      },
      "1ZIP.3": {
        "pageNumber": 2,
        "type": "text"
      }
    },
    "message": "Request completed successfully.",
    "status": "success"
  }
}
```

getFilledDocumentWithSignatureAsBase64(Object fieldsMap, String sysId, PdfMergeSignRequestor requestor, Object flatten)

Fills fields in an editable PDF, creates an image, and converts it to a Base64-encoded PDF.

Base64 encoding enables you to output a PDF as a string within a text document, such as HTML or JSON, without damaging the binary character syntax.

Use the following methods to determine if the PDF is fillable and get field information:

- [isDocumentFillable\(\)](#) ([dev.do#!/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-isDocumentFillable_S](#))
- [getDocumentFields\(\)](#) ([dev.do#!/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-getDocumentFields_S](#))
- [getDocumentFieldsType\(\)](#) ([dev.do#!/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-getDocumentFieldsType_S](#))

PDFGenerationAPI provides additional fill methods with different options:

- [fillDocumentFields\(\)](#) ([dev.do#!/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-fillDocumentFields_O_S_S_S_S_S](#)) – Fills fields in an editable PDF and attaches it to the provided record.
- [fillDocumentFieldsAndFlatten\(\)](#) ([dev.do#!/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-fillDocFieldsFlatten_O_S_S_S_S_S_O](#)) – Fills fields in an editable PDF, flattens the data fields, and attaches it to the provided record.
- [fillFieldsAndMergeSignature\(\)](#) ([dev.do#!/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PDFGenerationAPIBothAPI#P-fillFieldsMergeSig_O_S_S_S_S_S_O_O](#)) – Fills fields in an editable PDF, adds signature image, flattens the data fields, and attaches it to the provided record.

Parameters

Name	Type	Description
fieldsMap	Object	Optional. Key value map by PDF field name and value to fill. Use the getDocumentFields() method to get the list of available fields.
sysId	String	Sys_id of a PDF in the Attachments [sys_attachment] table.
requestor	PdfMergeSignRequestor (dev.do#/reference/api/quebec/server/sn_pdfgeneratorutils-namespace/PdfMergeSignRequestorBothAPI)	Signature input returned from pdfMergeSignRequestor.
flatten	Object	<p>Optional. Flattening fields enable locking the fields so that other users cannot change the information. Specify the key as "FlattenType" and provide a flattening option as a string. Valid values:</p> <ul style="list-style-type: none"> • donot_flatten - Do not flatten any fields. • partially_flatten - Flatten only the fields which are modified. • fully_flatten - Flattens all the fields. <p>Default: fully_flatten</p> <pre>{ "FlattenType": "String" }</pre>

Returns

Type	Description
String	If successful, PDF converted to Base64 format is added to the Attachments table [sys_attachment]. Contents reflect the PDF attachment provided with fields and signature filled. The fields are not editable unless an alternative flattening option was provided with the flatten parameter.
<Object>.message	<p>Message confirming success or error. Valid values:</p> <ul style="list-style-type: none"> • Exception while reading Source document contents. PDF header not found. – Input attachment provided is not a valid PDF. Provide the correct attachment sys_id. • Given target record [<tableName> - <targetTableSysId>] does not exist. – Target table sys_id is not in the table provided. Make sure you include the correct table name for the record. • No Form associated with pdf to fill. attachmentSysId: <sys_id> • No editable fields exist with specified names. Please check and try again. field names: <field names> • Request cannot proceed as the attachment with sys_id [{0}] did not pass security scan – The PDF did not pass the antivirus scan. • Request cannot proceed as the attachment with sys_id [{0}] is pending security scan – The PDF requires an antivirus scan. • Request completed successfully – Operation is successful. • Undefined – Sys_id provided does not exist or is not a PDF attachment. <p>Data type: String</p>
<Object>.status	<p>Status indicating whether the operation is successful. Valid values:</p> <ul style="list-style-type: none"> • success - Operation was successful. • failure – Operation was not successful. The message provides details. <p>Data type: String</p>

Example

The following shows how to load two fields in a PDF attachment, flatten the fields, and convert the PDF to Base64 format.

```
var mymap = new Object();
mymap["City"] = "City value here";
mymap["State"] = "XX";

// create a requestor
var requestor = new sn_pdfgeneratorutils.PdfMergeSignRequestor;
requestor.createRequest("<sys_id>", "tableName", "<tableSysId>", "pdfName");
requestor.addSignatureMapping(6, 40, 50, 188, 44, "<signImgSysId>");
var processedRequestObj = requestor.processRequest();

var v = new sn_pdfgeneratorutils.PDFGenerationAPI;

var result = v.getFilledDocumentWithSignatureAsBase64(mymap, "<attachmentSysId>", processedRequestObj);
gs.info (JSON.stringify(result));
```

getPdfPageSizes(String sysId)

Gets the page size of a PDF document.

Parameters

Name	Type	Description
sysId	String	Sys_id of a PDF in the Attachments [sys_attachment] table.

Returns

Type	Description
Object	Object containing the size of each page if successful, error message otherwise. <div> <pre>{ "pages_size": {Object}, "message": "String", "status": "String" }</pre> </div>
<Object>.pages_size	<p>If the operation is successful, width and height of each PDF page in points. The page number is returned as a string and the measurement values are returned as number data types.</p> <p>Data type: Object</p> <div> <pre>"pages_size": {"<page number>":["<width>,<height>"]}</pre> </div>
<Object>.message	<p>Message confirming success or error. Possible values:</p> <ul style="list-style-type: none"> Request cannot proceed as the attachment with sys_id [{0}] did not pass security scan – The PDF did not pass the antivirus scan. Request cannot proceed as the attachment with sys_id [{0}] is pending security scan – The PDF requires an antivirus scan. Request completed successfully – Operation is successful. Undefined – Sys_id provided does not exist or is not a PDF attachment. <p>Data type: String</p>
<Object>.status	<p>Status indicating whether the operation is successful. Valid values:</p> <ul style="list-style-type: none"> success - Operation was successful. failure – Operation was not successful. The message provides details. <p>Data type: String</p>

Example

The following shows how to display the width and height of each page in a PDF attachment.

```
var v = new sn_pdfgeneratorutils.PDFGenerationAPI;
var result = v.getPdfPageSizes("<attachmentSysId>");
gs.info(JSON.stringify(result));
```

```
{"pages_size":{"1":[612,792],"2":[612,792],"3":[612,792],"4":[612,792],"5":[612,792]},"message":"Request completed successfully.","status":"success"}
```


isDocumentFillable(String sysId)

Checks if the PDF document contains editable fields.

Parameters

Name	Type	Description
sysId	String	Sys_id of a PDF in the Attachments [sys_attachment] table.

Returns

Type	Description
Object	<p>Object containing the size of each page if successful, error message otherwise.</p> <pre>{ "document_editable": "String", "message": "String", "status": "String" }</pre>
<Object>.document_editable	<p>If the operation is successful, flag indicating whether the document is editable. Valid values:</p> <ul style="list-style-type: none">• true: PDF document has editable fields.• false: PDF document does not have editable fields. <p>Data type: Boolean value provided as a string</p>
<Object>.message	<p>Message confirming success or error. Valid values:</p> <ul style="list-style-type: none">• Exception while reading Source document contents. PDF header not found. – Input attachment provided is not a valid PDF. Provide the correct attachment sys_id.• Request cannot proceed as the attachment with sys_id [{0}] did not pass security scan – The PDF did not pass the antivirus scan.• Request cannot proceed as the attachment with sys_id [{0}] is pending security scan – The PDF requires an antivirus scan.• Request completed successfully – Operation is successful.• Undefined – Sys_id provided does not exist or is not a PDF attachment. <p>Data type: String</p>
<Object>.status	<p>Status indicating whether the operation is successful. Valid values:</p> <ul style="list-style-type: none">• success - Operation was successful.• failure – Operation was not successful. The message provides details. <p>Data type: String</p>

Example

The following shows how to determine if PDF document fields are editable.

```
var v = new sn_pdfgeneratorutils.PDFGenerationAPI;
var result = v.isDocumentFillable("<attachmentSysId>");
gs.info(JSON.stringify(result));
```

```
{"message":"Request completed successfully.","document_editable":"true","status":"success"}
```