

sign pro PDF API Guide

Contents

sign pro PDF API Guide	1
Downloads.....	1
API Modes.....	3
Starting sign pro PDF API	3
Submit	4
Submit IE.....	4
Command Line	5
Demonstration	5
Text Tag Mode	5
API Mode	8
SharePoint	11
OneDrive	12
Troubleshooting.....	12
Error starting API	12
Invalid JSON	13
Error message displayed by sign pro PDF	13

Downloads

The zip file API-Demo-V4.zip contains samples as follows:

Unzipped file: C:\API-Demo	Description
bin\	Contains test app

scripts\API Command Demo	Contains API command tests in .txt files with supporting PDF files
scripts\Text Tag Demo	Contains text tag command tests in .txt files with supporting PDF files

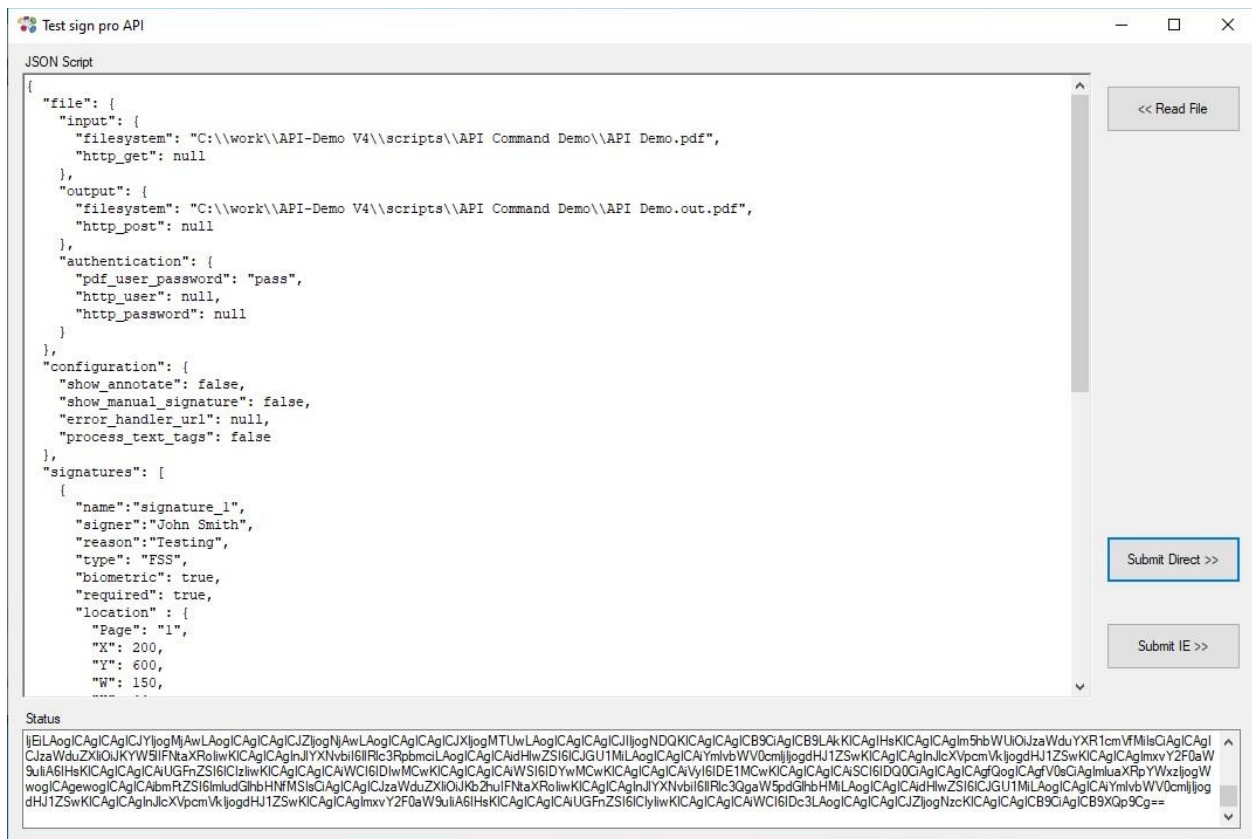
API Modes

There are two modes of API operation:

- Text Tag Mode: in this mode sign pro PDF instructions are inserted in text tags within the PDF
- API Mode: in this mode all of the sign pro PDF instructions are included in the command-line

Starting sign pro PDF API

To simplify testing the demo application **TestSignProAPI.exe** is included.



Run the application then press **Read File**

Browse to the scripts folder and select one of the JSON script .txt files provided in either:

scripts\\API Command Demo

or

scripts\\TextTag Demo

The script files are configured to demonstrate the API with the PDF files included. Once the file has been read into the edit window the contents can be modified to trial alternative API functions.

Preprocessor:

Note that the original file contents text string ‘(\$currentdir)’ is converted to the current folder name to simplify script file management. This means that the script folders can be relocated and continue to function correctly.

Submit

Select Submit to validate and encode the JSON structure entered in the edit window. If successful the sign pro PDF application is started with the command-line parameters:

```
C:\Program Files (x86)\Wacom sign pro PDF\Sign Pro PDF.exe -api signpro:<base64-json>
```

To support signpro v3, Ctrl-Submit starts the signpro application as:

```
C:\Program Files (x86)\Wacom sign pro PDF\ WacomGSS.signpro.exe -api signpro:<base64-json>
```

The operation can be started manually by entering the command in a DOS command prompt. Either copy the encoded string from the status window or encode the JSON structure using an online utility such as <https://www.base64encode.org/>

Submit IE

Select Submit IE to validate and encode the JSON structure entered in the edit window. If successful the sign pro PDF application is started by supplying Internet Explorer with the signpro URL:

```
IExplore.exe signpro:<base-64-json>
```

The operation can be started manually by entering the URL in a browser:

```
signpro:<base-64-json>
```

Browser limitation

Please be aware that the length of the URL command is limited. Internet Explorer is limited to 2083 characters, see:

<https://support.microsoft.com/en-us/help/208427/maximum-url-length-is-2-083-characters-ininternet-explorer>

Other browsers have similar limitations.

Command Line

Once the JSON structure has been saved it can be passed in the command line for direct processing, e.g.:

```
C:\Program Files (x86)\Wacom sign pro PDF\Sign Pro PDF.exe -apiFile C:\dev\json-file.txt
```

For successful operation file parameters will be saved with the full pathname, e.g.:

```
"input": {  
  "filesystem": "C:\\docs\\mydoc.pdf",  
  "http_get": null  
},
```

In addition, the sign pro PDF installer creates the file association for the extension `.signpro`. A JSON file saved with the `.signpro` extension can then be double-clicked to launch the application with direct processing.

Demonstration

Text Tag Mode

The file “**API Demo Text Tags.pdf**” demonstrates the use of text tags.

The Tags in the file are as follows:

Create signature area on Page 1 & 3:

```
{[[[Type=Signature,Page=1:3,Height=44,Width=150,X=200,Y=600,Name=TestSignature,Signer=John Smith,Reason=Testing]]]}
```

Create Initials area on Page2:

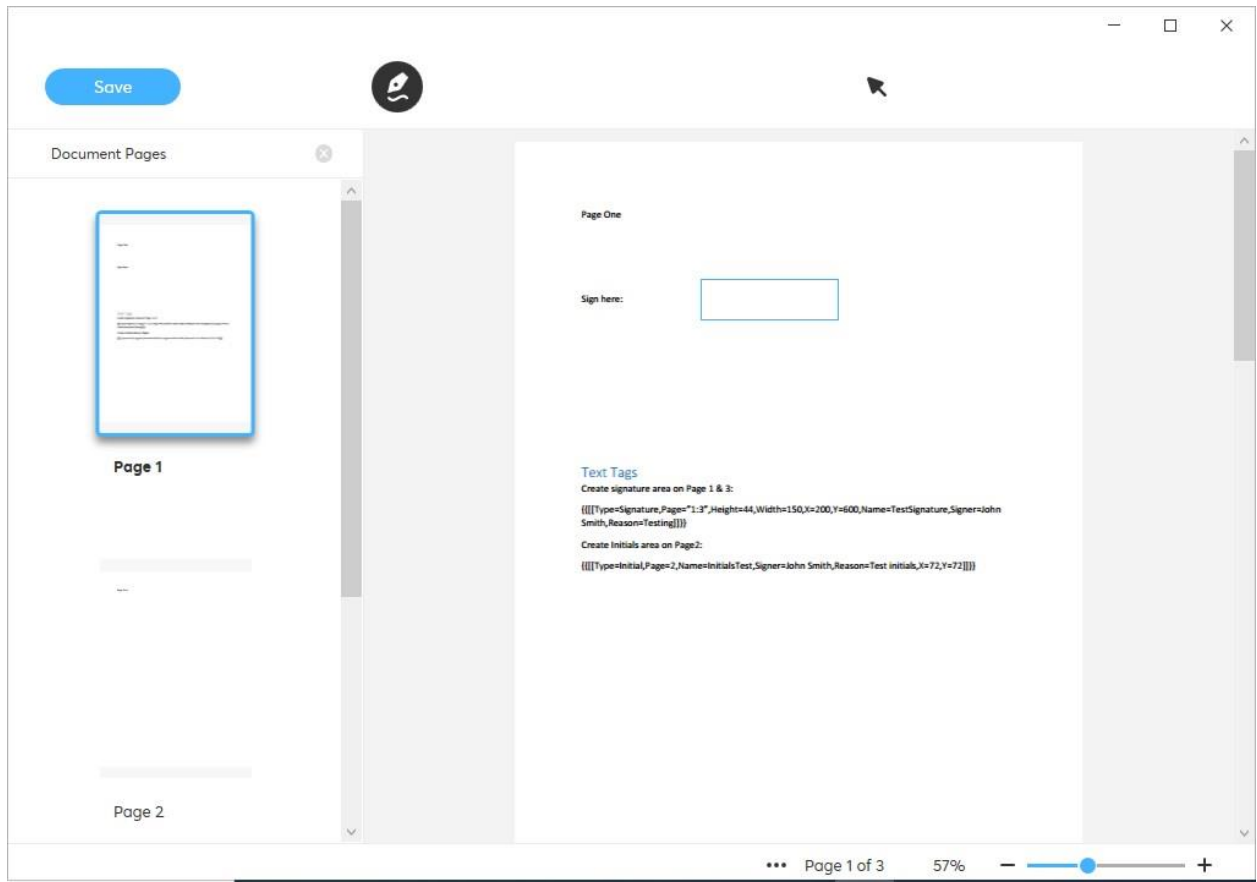
```
{[[[Type=Initial,Page=2,Name=InitialsTest,Signer=John Smith,Reason=Test initials,X=72,Y=72]]]}
```

To start the demonstration select the 'demo text tags.txt' script then press Submit:

```
{  
  "file": {  
    "input": {  
      "filesystem": "C:\\API-Demo\\scripts\\TextTag Demo\\API Demo - Text  
Tags.pdf",
```

```
    "http_get": null
  },
  "output": {
    "filesystem": "C:\\API-Demo\\scripts\\TextTag Demo\\API-Demo - Text
Tags.out.pdf",
    "http_post": null
  },
  "authentication": {
    "pdf_user_password": null,
    "http_user": null,
    "http_password": null
  }
},
"configuration": {
  "show_annotate": false,
  "show_manual_signature": false,
  "error_handler_url": null,
  "process_text_tags": true
},
"initials": null
}
```

The command starts sign pro PDF and creates the signing areas defined:



In this demo sign pro PDF has been started without manual signature placement or annotations enabled. By modifying the JSON structure these can be enabled in subsequent tests, e.g.:

```
"configuration": {  
  "show_annotate": true,  
  "show_manual_signature": true,  
  "error_handler_url": null,  
  "process_text_tags": true  
},
```

Similarly the JSON structure can be modified to test other functions, such as document password protection.

API Mode

The file “**API Demo.pdf**” can be used to demonstrate API mode where all commands are passed in the JSON structure.

Select the script file 'demo api.txt':

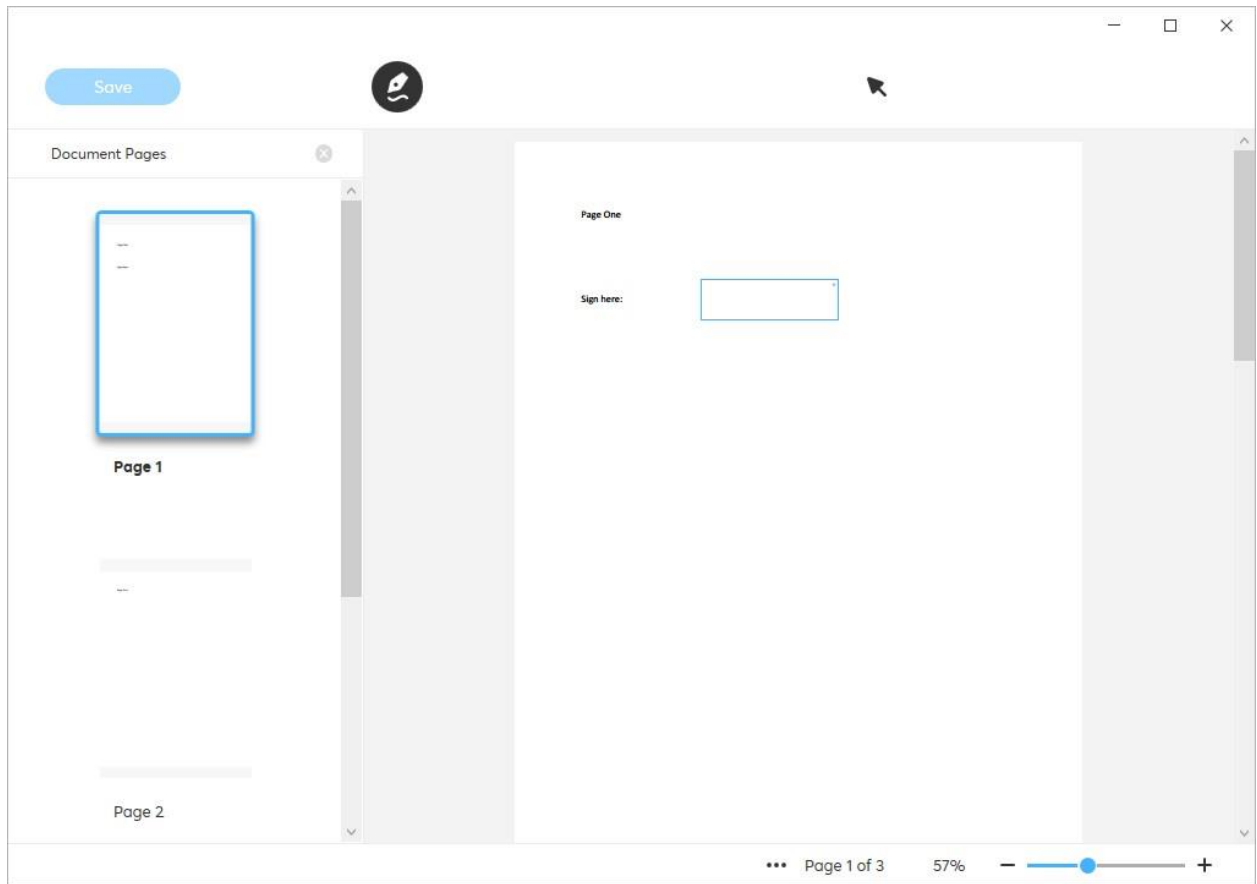
```
{
  "file": {
    "input": {
      "filesystem": "C:\\API-Demo\\scripts\\API Command Demo\\API Demo.pdf",
      "http_get": null
    },
    "output": {
      "filesystem": "C:\\API-Demo\\scripts\\API Command Demo\\API
Demo.out.pdf",
      "http_post": null
    },
    "authentication": {
      "pdf_user_password": "pass",
      "http_user": null,
      "http_password": null
    }
  },
  "configuration": {
    "show_annotate": false,
    "show_manual_signature": false,
    "error_handler_url": null,
    "process_text_tags": false
  },
  "signatures": [
    {
      "name": "signature_1",
      "signer": "John Smith",
      "reason": "Testing",
      "type": "FSS",
      "biometric": true,
      "required": true,
      "location": {
        "Page": "1",
        "X": 200,
        "Y": 600,
        "W": 150,
        "H": 44
      }
    },
    {
      "name": "signature_2",
      "signer": "Jane Smith",
      "reason": "Testing",
      "type": "FSS",
```



```
"biometric": true,  
"required": true,
```

```
    "location" : {
      "Page": "3",
      "X": 200,
      "Y": 600,
      "W": 150,
      "H": 44
    }
  }],
  "initials": [
    {
      "name": "intials_1",
      "signer": "John Smith",
      "reason": "Test initials",
      "type": "FSS",
      "biometric": true,
      "required": true,
      "location" : {
        "Page": "2",
        "X": 77,
        "Y": 77
      }
    }
  ]
}
```

The command starts sign pro PDF and creates the signing areas similar to the Text Tags demo. In this case signatures are marked as 'Required' and must be completed before the document can be saved:



SharePoint

When using SharePoint, functions such as registering an application or getting an access token need to be done via API calls.

For OneDrive, input parameters look something like:

```
"file": {
  "input": {
    "ms_graph": {
      "document_id": "012...",
      "user_email": "abc..."
    },
  },
  "output": {
    "ms_graph": {
```

```

        "document_id": "012...",
        "user_email": "abc..."
    },
},
"authentication": {
    "ms_graph_access_token": "xyz...",
},
},
},

```

OneDrive

OneDrive's functions, like SharePoint, must once again be accessed by calling the app. Unlike SharePoint, a drive id is utilised instead of a user's email.

```

"file": {
    "input": {
        "ms_graph": {
            "document_id": "012...",
            "drive_id": "abc..."
        },
    },
    "output": {
        "ms_graph": {
            "document_id": "012...",
            "drive_id": "abc..."
        },
    },
    "authentication": {
        "ms_graph_access_token": "xyz...",
    },
},
},

```

In both cases, the app must be properly programmed to know or obtain the document_id, ms_graph_access_token and either drive_id or user_email.

Troubleshooting

Error starting API

To use the API functionality the sign pro PDF license must be set to Evaluation or Enterprise to include "SIGNPRO_API_ACCESS". If this is not enabled an error message will be displayed: 'API access prohibited'.

Invalid JSON

Before attempting to start sign pro PDF the test validates the JSON structure. An error message will be displayed in the Status window, for example a missing a comma:

```
Validating JSON...  
After parsing a value an unexpected character was encountered: ".  
Path 'file', line 17, position 2. Invalid  
Json
```

The structure must be corrected before attempting to submit.

Error message displayed by sign pro PDF

If the API commands cannot be processed an error message will be displayed. Check the content of the JSON structure and/or the text tags, particularly file paths.

The JSON structure can be validated online, for example here:

<https://jsonformatter.curiousconcept.com/>
