AFAX PLATFORM API (beta v 1.3)

THE FAX SERVICE YOU'VE BEEN WAITING FOR

API

From your system

Use our Application Programming Interface (API) to securely send and receive your document faxes from applications, systems or platforms.

Perpetual beta v1.30

EMPOWER YOUR BUSINESS

STARTING WITH AFAX™ API

The AFAX API is an Application Programming Interface that provides developers the functionality to integrate our fax capabilities into their own applications. The integration itself is done via a web services using HTTPS operations.

Each method has mandatory required data elements and option pieces of information, for instant, at least you need to have your "**api key**" and "**account number**" in order to perform a basic call. (requests/inquiries)

By default, all responses are sent back from our server as a **JSON** (JavaScript Object Notation) encoded string.

SEND FAX

This method queues a Fax for delivery to its destination.

Field Name	Data Type	Required?	Description
sid_key	string	Yes	API Key. Retrieve / Reset that using AFAX Portal
haccno	string	Yes¥	AFAX Account Number
FaxDoc	Array	Yes	Related Fax document information.
clientReferenceCode	string	No	Any client reference values. Max 128 bytes
fromFaxNumber	string	Yes	Your Fax Number. 10 digits north of American format
fromEmailAddress	string	Yes	Your email address, valid sender email in your account
toMultipleRecipients	string	Yes	Value YES / NO. If there is more than one recipient
toFaxNumbers	string	Yes	Recipients(s) fax number including country code.
typeOfCoverpage	string	Yes	Digits between 0 to 9
fromFaxNumber_OnCoverPage	string	No	A Fax number in defined / selected coverage
fromName_OnCoverPage	string	No	Sender Name in defined coverage
toName_OnCoverPage	string	No	Recipient Name defined coverage
toOrgName_OnCoverPage	string	No	Recipient Company Name (if applicable)
faxSubject	string	No	Subject of Fax in defined coverage
faxMessage	string	No	Body of your Fax in defined coverage
qualityOfFax	string	No	Default is “fine”
clientCallBackURL	string	No	A URL for getting result in call back (aka post the result of fax status back to caller)
sendFaxAtDateTime	string	No	If you wish to send a fax in future, you can identify the
faxDocumentsURLs	Array string	Yes**	It’s a array of strings. Please see examples
faxDocumentsContent	string	Yes***	If used, must be in Base64 Encoding String.
encryption	string	Yes	Must be set to YES

¥’When you using this API in **production**, this value should be provided in HASH. The following logic should apply while invoking this method. The “**haccno**” value claculated as:
UpperCase(**MD5Hash** ("account-no" + UTC Date Time Format "YYMMDDhhnn"))
YYMMDDhhnn (24-hour clock): Two digits Year, Month, Day, two digits hour of two digits minutes.
Example:

Account No: 101587
Current Request Date Time (in **UTC timezone**): 2019-01-21 10:21:05 AM
str_value: "101587" + "1901221021" is equal to: "1015871901221021”
haccno: uppercase(MD5Hash(str_value)), equal to: **UpperCase**(**MD5Hash**("1015871901221021”))
haccno:= “*4C744559D42AB5E669AA4CEF4CD7FCBB*” is the value should pass for haccno

**’ For sending fax, system required to have an actual fax document or the reference in form of https URL to the fax documents.

***’ if not provided, system will decide to use “FaxDocumentsURLs”

The result value of the call is the **AFX REFID**, this result could be used in other methods; for instant when you sending a fax. Developer may decide to store this vlaue in the client system (i.e. in database or file) socould be refrenced to that when needed within other methods.

The following value is the result of a sample call for sending two fax documents.


```
{
  "value": "AFX43C86FA90CE44E9DBDB0AE086F924D43"
}
```

API Model

```
{
  sid_key*          string
  haccno*           string
  FaxDoc*           FaxService.TFaxDocuments {
                      clientReferenceCode       string
                      fromFaxNumber             string
                      fromEmailAddress          string
                      toMultipleRecipients      string
                      toFaxNumbers              string
                      typeOfCoverpage           string
                      coverPageURL              string
                      fromFaxNumber_OnCoverPage string
                      fromName_OnCoverPage      string
                      toName_OnCoverPage        string
                      toOrgName_OnCoverPage     string
                      faxSubject                string
                      faxMessage                string
                      qualityOfFax              string
                      clientCallBackURL         string
                      sendFaxAtDateTime         string
                      faxDocumentsURLs          [string]
                      faxDocumentsContent       string($base64)
                    }
  encryption*       string
}
```

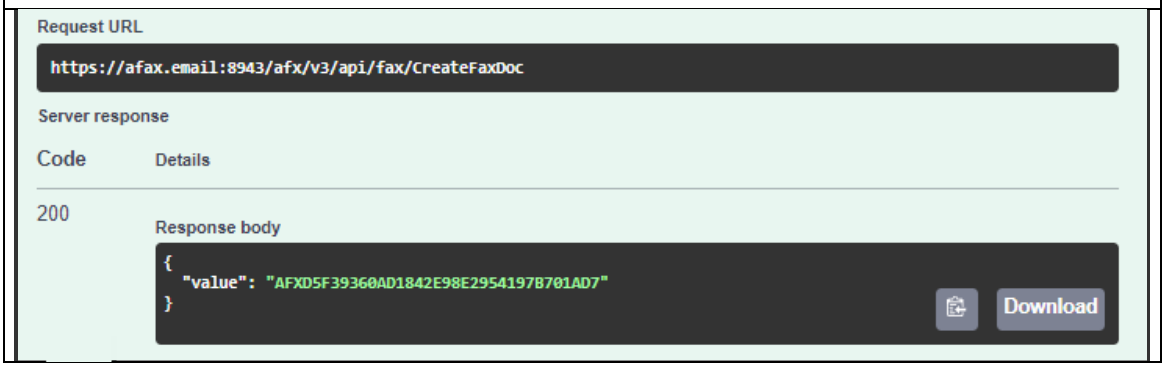
Sample JSON Request


```
{
  "sid_key": "ITpi9e890ae3fccwb3e815sqb8db03fdL",
  "haccno": "101587",
  "FaxDoc": {
    "clientReferenceCode": "API TEST",
    "fromFaxNumber": "1-613-555-4958",
    "fromEmailAddress": "demo@afax.ca",
    "toMultipleRecipients": "NO",
    "toFaxNumbers": "1-416-226-1602",
    "typeOfCoverpage": "0",
    "coverPageURL": "",
    "fromFaxNumber_OnCoverPage": "",
    "fromName_OnCoverPage": "",
    "toName_OnCoverPage": "",
    "toOrgName_OnCoverPage": "",
    "faxSubject": "",
    "faxMessage": "",
    "qualityOfFax": "fine",
    "clientCallBackURL": null,
    "sendFaxAtDateTime": null,
    "faxDocumentsURLs": ["https://qfax.ca/misc/afax.test.page.pdf"],
    "faxDocumentsContent": ""
  },
  "encryption": "YES",
  "aplatform": "afax"
}
```

Sample cURL Request

```
curl -X 'POST' \
  'https://afax.email:8943/afx/v3/api/fax/CreateFaxDoc' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "sid_key": " ITpi9e890ae3fccwb3e815sqb8db03fdL ",
  "haccno": "101587",
  "FaxDoc": {
    "clientReferenceCode": "API TEST",
    "fromFaxNumber": "1-613-555-4958",
    "fromEmailAddress": "demo@afax.ca",
    "toMultipleRecipients": "NO",
    "toFaxNumbers": "1-416-226-1602",
    "typeOfCoverpage": "0",
    "coverPageURL": "",
    "fromFaxNumber_OnCoverPage": "",
    "fromName_OnCoverPage": "",
    "toName_OnCoverPage": "",
    "toOrgName_OnCoverPage": "",
    "faxSubject": "",
    "faxMessage": "",
    "qualityOfFax": "fine",
    "clientCallBackURL": null,
    "sendFaxAtDateTime": null,
    "faxDocumentsURLs": ["https://qfax.ca/misc/afax.test.page.pdf"],
    "faxDocumentsContent": ""
  },
  "encryption": "YES",
  "aplatform": "afax"
}
```

Result Value



RECEIVE FAX

This method helps to getting the fax document using reference id.

Input values:

Field Name	Data Type	Required?	Description
sid_key	string	Yes	API Key. Retrieve / Reset that using AFAX Portal
haccno	string	Yes¥	AFAX Account Number
rfid	string	Yes	Reference AFX ID*
encryption	string	No	If set to Yes, system will send the encrypted data
embdoc	string	No	Default to “Embedded” **

¥’ When you using this API in **production**, this value should be provided in HASH. The following logic should apply while invoking this method. The “**haccno**” value claculated as:
UpperCase(**MD5Hash** ("account-no" + UTC Date Time Format "YYMMDDhhnn"))

Please read the details provided in SEND FAX

*' The result value of the call is the **AFX REFID** which could be use in other methods. For instant when you send a fax, you sould store tis ref-if in your side (i.e. in database for file) so you can refrence to that when try to check the status of fax.
The following value is the result of a sample API call for sending a fax document.


```
{
  "value": " AFXD5F39360AD1842E98E2954197B701AD7 "
}
```


**If "embdoc" filed set to "Embedded", the content of the Fax Document is embedded into JSON response. Please remember the binary data is encoded using Base64 algorithms in order to pass through http.
You have to implement a code for decoding the string to binary data. Please refer to this site for testing: https://base64.guru/converter/decode/pdf

if for any reason the size of fax document more than the size which could be carry using HTTP protocol, our system may switch and send you the URL for stored data in our secure server. In this case, the URL presented in "**fax_documents_urls**" filed.

Example:"fax_documents_urls":"**https://afax.io/md/stg/fx0ff39b941b329773a51a7ad940f5d7c4.pdf**"

Sample cURL Request

Curl

```
curl -X GET "https://www.afax.email:8743/afx/api/fax/GetFaxDocByRFid?sid_key=IVHBiwELisoYwF2E3DNEVei5KoCAX3st&haccno=103270&rfid=AFXCB2F471BC7844103AFEB747AAEF3F656&encryption=Yes&embdoc=Embedded" -H  "accept: application/json"
```

Request URL

```
https://www.afax.email:8743/afx/api/fax/GetFaxDocByRFid?sid_key=IVHBiwELisoYwF2E3DNEVei5KoCAX3st&haccno=103270&rfid=AFXCB2F471BC7844103AFEB747AAEF3F656&encryption=Yes&embdoc=Embedded
```

Sample HTTP Response

Server response

Code	Details
200	Response body

JSON Response


```
{
  "$id": 1,
  "faxid": 5678,
  "refid": "AFXCB2F471BC7844103AFEB747AAEF3F656",
  "status": "Sent",
  "extended_status": "Completed | OK",
  "from_faxnumber": "6135552021",
  "to_faxnumber": "12898427115",
  "from_emailaddress": "demo@afax.ca",
  "to_emailaddress": "",
  "fax_protocol": "T38",
  "fax_quality": "14400, 8031/7700",
  "fax_doc_size": 17,
  "farend_station_id": "AFAX",
  "timezone_diff": 0,
  "fax_ttl": 0,
  "fax_queued_datetime": "2020-11-30T16:32",
  "fax_start_datetime": "2020-11-30T16:32:27",
  "fax_end_datetime": "2020-11-30T16:32:53",
  "fax_duration": 26,
  "numberof_pages": 2,
  "fax_documents_urls": "",
  "fax_doc_content":
"JVBERi0xLjcKCjQgMCBvYmoKPDwKL0JpdHNQZXJDb21wb251bnQgOAovQ29sb3JTcGFjZSAvRGV2aWNlR3JheQovRmlsdGVyIC9GbGF0ZURlY29kZQovSGVpZ2h0IDIzMzkKL0xlbmd0aCAxMDYxMgovU3VidHlwZSAvSW1hZ2UKL1R5cGUgL1hPYmplY3QKL1dpZHRoIDE3MjgKPj4Kc3RyZWFtCnic7N2NctuqFgbQ/f4vzelpE5ufjWzHlizitWburQQIJDrDF2Q3pwAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Download

JSON Response Details

Response data elements:


```
{
  "faxid": 5678,
  "refid": "AFXCB2F471BC7844103AFEB747AAEF3F656",
  "status": "Sent",
  "extended_status": "Completed | OK",
  "from_faxnumber": "6135552021",
  "to_faxnumber": "12898427115",
  "from_emailaddress": "demo@afax.ca",
  "to_emailaddress": "",
  "fax_protocol": "T38",
  "fax_quality": "14400, 8031/7700",
  "fax_doc_size": 17,
  "farend_station_id": "AFAX",
  "timezone_diff": 0,
  "fax_ttl": 0,
  "fax_queued_datetime": "2020-11-30T16:32",
  "fax_start_datetime": "2020-11-30T16:32:27",
  "fax_end_datetime": "2020-11-30T16:32:53",
  "fax_duration": 26,
  "numberof_pages": 2,
  "fax_documents_urls": "",
  "fax_doc_content": "JVBERi0xLjc.... base64",
  "api_version": "v1.3 - (c) AFAX Platform."
}
```


Field Name	Type	Size	Description
faxid	Integer	8 bytes	Unique ID for each fax record, could be use internally within your development and coding.
rfid	string	36 chars	Unique AFX Reference id. (aka **AFX REFID**)
status	string	16 chars	The status of Fax – like: Queued, Sending, Sent, Receiving, Received, Processing, Processed, Deleted, Failed, Known.
extended_status	string	64 chars	Contained more information for each fax status.
from_faxnumber	string	32 chars	Fax number for sender of Fax
to_faxnumber	string	32 chars	Fax number of the recipient of the Fax
from_emailaddress	string	128 chars	Sender's email address
to_emailaddress	string	128 chars	Recipient email address
fax_protocol	string	32 chars	Fax communication protocol. Useful for partner
fax_quality	string	32 chars	Fax quality
fax_doc_size	Integer	4 bytes	Size of fax document in bytes
farend_station_id	string	128 chars	Far end fax station ID or Name
timezone_diff	number	4.2	Associated TZ based on account profile based on EST. For example, if account is in Vancouver (BC), this value would be (-3.00)
fax_ttl	integer	4 bytes	TBD
fax_queued_datetime	datetime	4 bytes	Date time when fax is queued in system. The data format is "Unix Format", like:2020-11-30T16:32:53
fax_start_datetime	datetime	4 bytes	Date time when processing fax is started
fax_end_datetime	datetime	4 bytes	Date time when processing fax is ended
fax_duration	Integer	4 bytes	Duration of fax communication.
numberof_pages	Integer	4 bytes	Number of fax pages
fax_documents_urls	string	4 bytes	A URL for fax document. This URL is provided by system if no Embedded selected as a format of receiving the fax document
fax_doc_content	string	4 bytes	A Base64 Encoding string, content of the fax document is Embedded selected as a format of receiving the fax
api_version	string	32 chars	AFAX API version

HOUSE KEEPING

Please consider following details when you are planning to test our API.

- Send us your public IP address to open a firewall to our SIT environment
- Arrange not to sensendding more than 5 ~ 10 requests per second
- Our API will assess the size of fax documents (for both sent and received faxes) and if the size of document is bigger than the size that could be transferred within Hypertext in JSON via HTTP protocol, system will switch the “embedded” request to “url”.
- If you have any questions or need any addition information please contact us via email at support@afax.ca
- Please note for a security measurement, we only allow our support team to deal with authorized person in account and/or a developer(s) which have induced to us.

Response and Error Codes

Whenever you make a request that fails for some reason, an error is returned also in the JSON format. The errors include an error code and description, which you can find in detail below.

Code	Type	Details
200	OK	Everything worked as expected.
400	Bad request	Bad request, missing attribute in payload
401	Unauthorized	The request was unacceptable. Typically due to the API key needing to be added or corrected.
404	Data not found	
444	data format/type	Unexpected data format/type!
500	Internal Server Error	The request could not be completed due to an error on the server side.
503	Service Unavailable	The server was unavailable.
521	Data encryption issue	Data encryption issue

Some useful tips

Most developers have their own resources and toolsets for building and testing their API. Our developers are using Postman Platform, Swagger, and SoapUI resources for their testing.

If you want to learn more about BASE64 Format and why is used, please search the in the internet. This site Base64 Encoding may help you to start.

AFAX Platform Methods

The list of avilable methods that could be used by developer to extend their applications functionalties.

Please contcat our support team if you have any questions, or need any additional information.