

OpenText Hightail

API Development Guide 21.1

Contents

1	Introduction.....	2
2	Getting Started.....	2
3	Sending Files	2
	3.1 Overview.....	2
	3.2 Procedure	3
4	Moving to production.....	14
	About OpenText.....	14

1 Introduction

The Hightail API makes it easy to develop custom tools or direct integrations to automate key tasks and make your team more productive. The API is available to Hightail Teams and Business users.

2 Getting Started

To access the Hightail API, first set up a Hightail Business Trial account and obtain an API token by following these steps:

1. Sign up for a temporary 30-day Hightail Business Trial account on test environment.
2. Generate an API token.
3. Refer to the '[Sending Files](#)' workflow guide for the next steps.

You have now set up a Hightail test environment. This is a temporary Hightail Business account of which you are the administrator. Your account runs on our test server and will expire after 30 days. If you need to extend the duration, please contact Hightail customer service.

3 Sending Files

3.1 Overview

This set of API calls invokes Hightail's 'File Send' function. It provides the same 'File Send' capabilities as those which you will find in the Hightail Web application. The general flow is as follows:

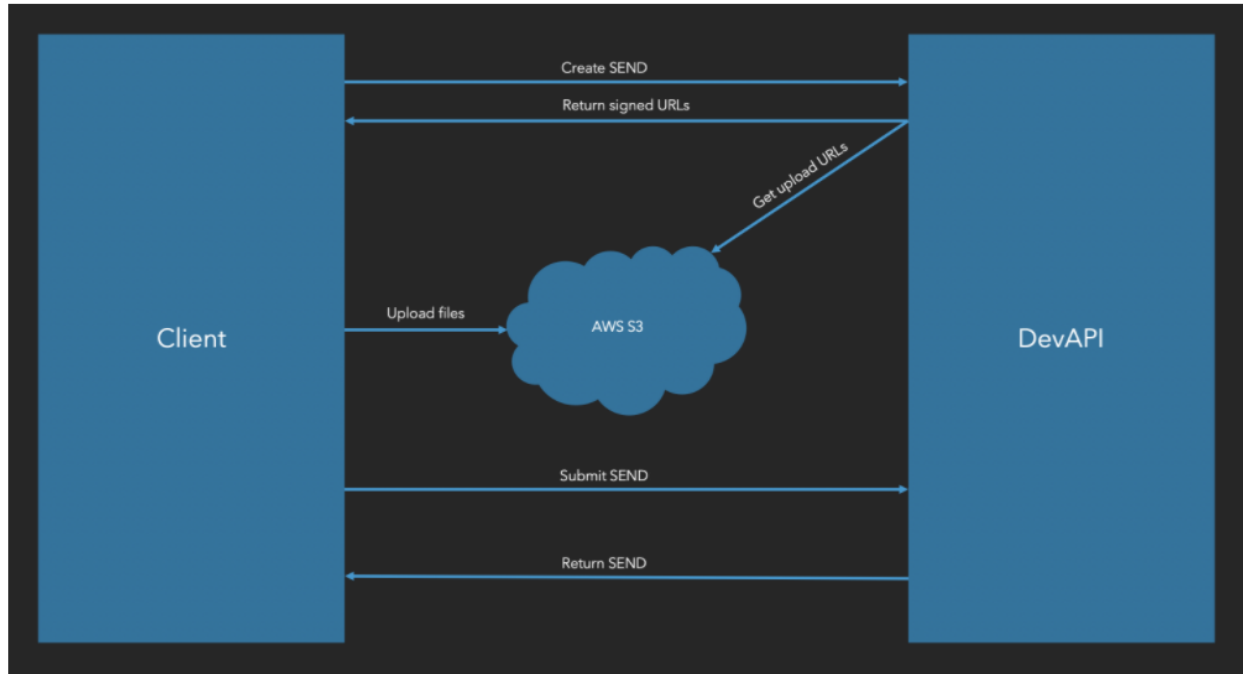
1. Create a Send on Hightail, which will act as the repository for the files you wish to send. Hightail will return the following:

- Space ID of the new Send Repository.
- URLs to upload files to

These URLs are currently set to expire after 24 hours, but this duration is subject to change.

4. Upload your files directly to S3 with URLs returned in the previous step.
5. 'Send' the files. You may set the following parameters for this call:

- Subject
- Message
- Recipient email address(es)
- Access permissions



3.2 Procedure

1. Create Send Repository:

The following is an example of a request:

```

curl 'https://api.stage.htspaces.com/devapi/v1/send/create' \
-H 'Accept: application/json, text/plain, */*' \
-H 'content-type: application/json;charset=UTF-8' \
-H 'Authorization: 2f145795e4a1e8a8db8efba8a2806f176698eef35b4c0e32d0fafeddd566dfdd_d' \
-d '{
  "email": "sub1@yopmail.com",
  "chunkSize": 10485760,
  "fullUrl": true,
  "files": [
    {
      "filename": "AmazonChime.dmg",
      "fileSize": 17569972
    },
    {
      "filename": "1.md",
    }
  ]
}'
    
```

```

        "fileSize": 1
      }
    ]
  }
}
```

Parameters:

Parameter	Type	Comments
Authorization Header		API token
email	string	Sender's email address. Sender must be a registered Hightail user and a member of the organization linked to the API token.
chunksize	long	<p>If this value is less than 5MB (5,242,880), this value will default to 5MB. If the file is very large, Hightail will adjust the chunkSize value to ensure that chunk counts remain under 10,000.</p> <p>Keep in mind that this value may change from your input, so be sure to use the chunkSize provided in response.</p>
fullURL	boolean	<p>Controls whether the response to this call should return full URLs or packed URLs in response.</p> <p>Our recommendation is to use fullURLs for small files and packedURLs for large files to reduce the response payload size. When uploading a mix of small and large files, we recommend packedURLs.</p>
Files		This is a list of files to be uploaded for sending. Each item in this list will contain a filename and file size. This list can contain one or more files.
filename	string	
filesize	long	

The following is an example of a response for fullURL mode (that is, when fullURL is set to true):

```

{
  "spaceId": "sp-8ad94976-4816-482e-a763-510e44a0a533",
}
```

```

    "fullUrls": [
      {
        "chunkSize": 10485760,
        "urls": [
          "https://newsendit-dev.s3.us-west-2.amazonaws.com/61758b7e-2add-4b60-9d9c-5d748b6987a0?uploadId=fbPsdYJiqVIjnzdBkmuE8t.4QdJNbPzQCa8pEc._gkEwlOsxMC3PRlbcboueE64RKovJ5bYTXE6QV6MnDUKFGxKzvfCOKEHJHblzEq1ewzMLfRl8sNBKYzgxrkdUQ9PV&partNumber=1&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20201113T205756Z&X-Amz-SignedHeaders=host&X-Amz-Expires=604799&X-Amz-Credential=AKIAJ7FUEDMYSZEGDPA%2F20201113%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-Signature=4eb687f30840ff44b67ec52792ca5568b17d8121bc0c230c871647fba2730194",
          "https://newsendit-dev.s3.us-west-2.amazonaws.com/61758b7e-2add-4b60-9d9c-5d748b6987a0?uploadId=fbPsdYJiqVIjnzdBkmuE8t.4QdJNbPzQCa8pEc._gkEwlOsxMC3PRlbcboueE64RKovJ5bYTXE6QV6MnDUKFGxKzvfCOKEHJHblzEq1ewzMLfRl8sNBKYzgxrkdUQ9PV&partNumber=2&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20201113T205756Z&X-Amz-SignedHeaders=host&X-Amz-Expires=604800&X-Amz-Credential=AKIAJ7FUEDMYSZEGDPA%2F20201113%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-Signature=f536f6445355fe2f6842bff00f96abec755040c16e2cbf4c229891ea0ac6b38"
        ]
      },
      {
        "chunkSize": 10485760,
        "urls": [
          "https://newsendit-dev.s3.us-west-2.amazonaws.com/e7730ea9-b26b-4328-99ff-29c9937439d8?uploadId=4v481Ayx_Jv0500Q4iO9y_uAijpVOM9c53GH5vohDLKhoX5g_gTHVvkobatzbyG9IhZiD4rTKE5IrhxTJvme1BjIYJVqT8kfKSGsqGdyzk3qJm3PZq28POtHNYjT08XZ&partNumber=1&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20201113T205756Z&X-Amz-SignedHeaders=host&X-Amz-Expires=604800&X-Amz-Credential=AKIAJ7FUEDMYSZEGDPA%2F20201113%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-Signature=34e3509a3529637078b71c08ad0edf52a8f1da18b036709870c87fdadcdeb89e"
        ]
      }
    ]
  }
}

```

Note:

- **spaceId (string):** Keep this on hand. You will need it for the submit call.

- fullUrls: List of chunk size and upload URLs for each file. Each item is associated with one file.
- chunkSize (long): Make sure to use this value to generate chunks for each file.
- urls: These signed full URLs can be used to upload files. Make sure to apply the URLs and chunks in the correct order (using the first URL for the first chunk, the second URL for the second chunk, and so on).

The following is an example of a response for packedURL mode (fullURL set to false):

```
{
  "spaceId": "sp-8bda763d-d51f-4aca-ba2b-ad4ccb7411da",
  "packedUrls": [
    {
      "urlPattern": "https://newsendit-dev.s3.us-west-2.amazonaws.com/
84f1de5a-5a54-4e8f-b629-
13a7746d4f13?uploadId=wwnC6zr9G7PORPvD5gBcfzyyCci4V0j_
RQcP6HnweJk8bAngmLpOCrHzYCQbuo9ie6_6sixcWG5Z6NfNVSCWs.
hRfsGKs8DUSKUWjxTqRuYZa8XTWA7uJwpfeGf.AEQw&partNumber={{partNumber}}
&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20201114T020932Z&X-Amz-
SignedHeaders=host&X-Amz-Expires={{expires}}&X-Amz-Credential=
AKIAJ7FUEDMYSZEGDPHA%2F20201114%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-
Signature={{signature}}",
      "chunkSize": 10485760,
      "items": [
        {
          "partNumber": 1,
          "expires": 604800,
          "signature": " fea3f74b8fe6f4199f2646d89f8360bb946d6f9081c3a779
0f37ba7871567c2d"
        },
        {
          "partNumber": 2,
          "expires": 604800,
          "signature":
"944bf20c0belee069c5517ff66992299d9e7db57171fcc81a88b
f1f1bd9f0c4c"
        }
      ]
    },
    {
      "urlPattern": "https://newsendit-dev.s3.us-west-
2.amazonaws.com/73850d9e-
6771-4214-86a5-fbdd3ba13ddb?uploadId=Fme5NqvrMnJMSTrSOdoT4Ps5X032IGX
Kyj1EbrfqPE69.8Gz_CE6GaGNq4DGxsHXKt9MKevg3sdB_
rF3pWneUBcelm0Xna0lUisF5BWlBeFYnCbfdOT9zr56ySx4sdSh
&partNumber={{partNumber}}&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
```

```
Date=20201114T020932Z&X-Amz-SignedHeaders=host&X-Amz-Expires=
{{expires}}&X-Amz-Credential=AKIAJ7FUEDMYSZEGDPA%2F20201114%2
Fus-west-2%2Fs3%2Faws4_request&X-Amz-Signature=
{{signature}}",
  "chunkSize": 10485760,
  "items": [
    {
      "partNumber": 1,
      "expires": 604800,
      "signature":
"30410b0b718f689e0435e9b8bdf4a89d09cfd285adaf344b66c
c9dfbed0ce780"
    }
  ]
}
```

Note:

- `spaceId` (string): Keep this on hand. You will need it for the submit call.
- `packedUrls`: Similar to the `fullUrls` list above, this list contains the items needed to construct full URLs for each file. Each item is associated with one file.
 - `urlPattern`: The pattern for constructing full URLs (one pattern per file), Replace `{{partNumber}}`, `{{expires}}` and `{{signature}}` with the values in the items to construct the full URL.
 - `chunkSize` (long): Make sure to use this value to generate chunks for each file.
 - `items`: This is a list which contains all values needed to generate full URLs for each file. Apply these values to the patterns in `urlPattern` to construct the full URLs.
 - `partNumber` (int): sequence number of each chunk
 - `expires` (int): expiration time of the URL (in seconds)
 - `signature`: signature of the signed URL

2. Prepare the chunks for upload:

The following is an example of a command on Mac OS:

```
split -b 10485760 AmazonChime.dmg AmazonChime.
```

Note:

- Use the `chunkSize` value from the create Send Repository response. The sample command will create chunk files with extensions `.aa`, `.ab`, ...

- Skip this step for single chunk files where the file size is less than the chunkSize value. These files will only have one URL.

3. Upload chunks to S3:

The following is a Curl example that uploads chunks (in practice, your application can upload chunks as streams instead of split files):

```
curl -i -X PUT 'https://newsendit-dev.s3.us-west-2.amazonaws.com/de535c54-
adec-11ea-9c4f-e20c93bf03cf?uploadId=SPstilHuT6uuJ0lyXDBnkyZh9v.
K85SX5tdwP502H9bobVF7F8Gf.33kJenqeFS4P6TCKJnJ3KaA3Cyl_EpLT2so6JaqSga..QSFm
.VooWxT45OpnyuNIUw_Ul_Hlc34&partNumber=1&X-Amz-Algorithm=AWS4-HMAC-
SHA256&X-Amz-Date=20201106T185449Z&X-Amz-SignedHeaders=host&X-Amz-
Expires=604800&X-Amz-Credential=AKIAJ7FUEDMYSZEGDPA%2F20201106%2Fus-west-
2%2Fs3%2Faws4_request&X-Amz-
Signature=882f5cf11e4aeaa388f7ae5493d8c5e61fddb2ec88a82
f4cefde3ff44a344022' \
--upload-file AmazonChime.aa
```

```
curl -i -X PUT 'https://newsendit-dev.s3.us-west-2.amazonaws.com/de535c54-
adec-11ea-9c4f-e20c93bf03cf?uploadId=SPstilHuT6uuJ0lyXDBnkyZh9v.
K85SX5tdwP502H9bobVF7F8Gf.33kJenqeFS4P6TCKJnJ3KaA3Cyl_EpLT2so6JaqSga..QSFm
.VooWxT45OpnyuNIUw_Ul_Hlc34&partNumber=2&X-Amz-Algorithm=AWS4-HMAC-
SHA256&X-Amz-Date=20201106T185449Z&X-Amz-SignedHeaders=host&X-Amz-
Expires=604799&X-Amz-Credential=AKIAJ7FUEDMYSZEGDPA%2F20201106%2Fus-west-
2%2Fs3%2Faws4_request&X-Amz-
Signature=d2f0a1b33d31296632ed9719a841baa56b19b08d3a1c
c5702cbf9320950b9d11' \
--upload-file AmazonChime.ab
```

Note: Make sure to use the return URLs and chunks in the correct order. Chunks can be uploaded in parallel, as long as all chunks use the correct URLs.

The following is an example of a response from chunk upload:

```
HTTP/1.1 100 Continue
```

```
HTTP/1.1 200 OK
```

```
x-amz-id-2:
```

```
7OmLQJpps1AQhKLZjQtt6qdoXFqvP0GwLDY6KX1//iQyxj8q22eF4RMOarORVVHRWUqEWNNU5xw
w=x-amz-request-id: C310F66E781DEC35
```

```
Date: Fri, 06 Nov 2020 19:24:54 GMT
```

```
ETag: "09c47d8dd9c8d788f01f3e0089ea76f3"
```



```
x-amz-server-side-encryption: AES256
Content-Length: 0
Server: AmazonS3
```

4. Submit Send Repository:

The following is a sample request:

```
curl 'https://api.stage.htspaces.com/devapi/v1/send/submit' \
  -H 'Accept: application/json, text/plain, */*' \
  -H 'content-type: application/json; charset=UTF-8' \
  -H 'Authorization:
2f145795e4a1e8a8db8efba8a2806f176698eef35b4c0e32d0fafa
ddd566dfdd_d' \
  -d '{
    "spaceId": "sp-859ec598-d81a-4ea6-a98f-cb4770fc9695",
    "subject": "Subject of the Send Repository",
    "message": "Message of the Send Repository",
    "emails": [
      "usr666@yapmail.com",
      "usr666@mailinator.com"
    ]
  }'
```

Note:

- spaceId (string): Returned value from the create Send Repository response.
- subject (string): Subject of the Send Repository (optional)
- message (string): Message of the Send Repository (optional)
- emails: List of email address(es) of intended recipient(s). Needs to contain at least one value and can contain multiple values, up to the user's daily limit (which varies based on account type).

The following is an example of a response:

```
{
  "id": "sp-859ec598-d81a-4ea6-a98f-cb4770fc9695",
  "status": "ACTIVE",
  "url": "UvbDZsVaEf",
  "downloadReceipt": false,
  "creationTime": "1605318332763",
  "updateTime": "1605318332763",
  "activeAt": "1605318332763",
  "type": "SEND",
```

```
"purged": false
}
```

5. Send Options:

When submitting Send Repositories, the following send options are available (except where limited by product entitlement or organization policy):

- Access Code
- Expiration
- Verify Recipient Identity
- Allow Recipient Comments
- Prevent File Downloads
- Get A Download Receipt

1. Access Code

This option allows senders to set a passcode for the sent repository. All recipients must enter the correct passcode before accessing the send.

This is an example of a request:

```
curl 'https://api.stage.htspaces.com/devapi/v1/send/submit' \
-H 'Accept: application/json, text/plain, */*' \
-H 'content-type: application/json; charset=UTF-8' \
-H 'Authorization: 2f145795e4a1e8a8db8efba8a2806f176698eef35b4c0e32d0fafeddd566dfdd_d' \
-d '{
  "spaceId": "sp-859ec598-d81a-4ea6-a98f-cb4770fc9695",
  "subject": "Subject of the Send Repository",
  "message": "Message of the Send Repository",
  "accessCode": "Pa$$w0rd",
  "emails": [
    "usr666@yapmail.com",
    "usr666@mailinator.com"
  ]
}'
```

Note:

- The accessCode must contain at least 1 character.
- If your organization's policy requires an accessCode and your send request does not have one, the send request will be rejected.

- If your organization's policy requires an accessCode of a certain complexity and your send request contains an accessCode which fails to meet the requirements, the send request will be rejected.

Example: Your organization's policy may require accessCode values to include at least 1 number, 1 uppercase letter, 1 lowercase letter, and 1 special character while being at least 8 characters total in length.

2. Expiration

This option is used to control the expiration time of a sent repository. Recipients will no longer be able to access the send after the set expiration time.

The following is an example of a request:

```
curl 'https://api.stage.htspaces.com/devapi/v1/send/submit' \
  -H 'Accept: application/json, text/plain, */*' \
  -H 'content-type: application/json; charset=UTF-8' \
  -H 'Authorization: 2f145795e4a1e8a8db8efba8a2806f176698eef35b4c0e32
d0fafeddd566dfdd_d' \
  -d '{
    "spaceId": "sp-859ec598-d81a-4ea6-a98f-cb4770fc9695",
    "subject": "Subject of the Send Repository",
    "message": "Message of the Send Repository",
    "expiresAt": "2020-12-06T03:02:13.552Z",
    "emails": [
      "usr666@yapmail.com",
      "usr666@mailinator.com"
    ]
  }'
```

Note:

- expiresAt can be long or string
 - long: Milliseconds from 1970-01-01T00:00:00Z
For example: 1607223733552
 - String: DateTime string in ISODateTimeFormat
For example: "2020-12-06T03:02:13.552Z", "2020-12-06T03:02:13.552-08:00"

3. VRI (Verify Recipient Identity)

This is a boolean value which can be set to **true** or **false**. (The default value will be **false**.) When set to **true**, only user logged into email accounts listed as recipients of the sent repository can view the space.

The following is an example of a request:

```
curl 'https://api.stage.htspaces.com/devapi/v1/send/submit' \
  -H 'Accept: application/json, text/plain, */*' \
  -H 'content-type: application/json; charset=UTF-8' \
  -H 'Authorization: 2f145795e4a1e8a8db8efba8a2806f176698eef35b4c0e32d0fafeddd566dfdd_d' \
  -d '{
    "spaceId": "sp-859ec598-d81a-4ea6-a98f-cb4770fc9695",
    "subject": "Subject of the Send Repository",
    "message": "Message of the Send Repository",
    "verifyRecipient": true,
    "emails": [
      "usr666@yapmail.com",
      "usr666@mailinator.com"
    ]
  }'
```

Note:

- **verifyRecipient** can be overridden by your organization's policies. If the "Verify Recipient Required" policy has been set to **YES**, then all spaces sent by members from that organization will have **verifyRecipient** set to **true**, regardless the input in the send request.

4. Allow Recipient Comments

This is a boolean value which can be set to **true** or **false**. (The default value will be **false**.) When set to **true**, recipients will be able to comment on the sent space.

This is an example of a request:

```
curl 'https://api.stage.htspaces.com/devapi/v1/send/submit' \
  -H 'Accept: application/json, text/plain, */*' \
  -H 'content-type: application/json; charset=UTF-8' \
  -H 'Authorization: 2f145795e4a1e8a8db8efba8a2806f176698eef35b4c0e32d0fafeddd566dfdd_d' \
  -d '{
    "spaceId": "sp-859ec598-d81a-4ea6-a98f-cb4770fc9695",
    "subject": "Subject of the Send Repository",
    "message": "Message of the Send Repository",
    "allowComment": true,
    "emails": [
      "usr666@yapmail.com",
    ]
  }'
```

```
    "usr666@mailinator.com"
  ]
}'
```

5. Prevent File Downloads

This is a boolean value which can be set to **true** or **false**. (The default value will be **false**.) When set to **true**, recipients will not be able to download files from the sent space.

This is an example of a request:

```
curl 'https://api.stage.htspaces.com/devapi/v1/send/submit' \
-H 'Accept: application/json, text/plain, */*' \
-H 'content-type: application/json; charset=UTF-8' \
-H 'Authorization: 2f145795e4a1e8a8db8efba8a2806f176698eef35b4c0e32d0fafeddd566dfdd_d' \
-d '{
  "spaceId": "sp-859ec598-d81a-4ea6-a98f-cb4770fc9695",
  "subject": "Subject of the Send Repository",
  "message": "Message of the Send Repository",
  "preventDownload": true,
  "emails": [
    "usr666@yapmail.com",
    "usr666@mailinator.com"
  ]
}'
```

6. Get a Download Receipt

This is a boolean value which can be set to **true** or **false**. (The default value will be **false**.) When set to **true**, the sender will receive an email notification when recipients download files from the sent space.

This is an example of a request:

```
curl 'https://api.stage.htspaces.com/devapi/v1/send/submit' \
-H 'Accept: application/json, text/plain, */*' \
-H 'content-type: application/json; charset=UTF-8' \
-H 'Authorization: 2f145795e4a1e8a8db8efba8a2806f176698eef35b4c0e32d0fafeddd566dfdd_d' \
-d '{
  "spaceId": "sp-859ec598-d81a-4ea6-a98f-cb4770fc9695",
  "subject": "Subject of the Send Repository",
  "message": "Message of the Send Repository",
  "downloadReceiptRequested": true,
}
```

```
"emails": [  
  "usr666@yapmail.com",  
  "usr666@mailinator.com"  
]  
'
```

4 Moving to production

When you are ready to use the Hightail API within your organization's production environment, do the following:

1. Request a production API token from your organization's Hightail administrator.
2. Update your calls with the production API token received in the previous step.
3. Update the host in the API calls to point to your production system.

About OpenText

OpenText enables the digital world, creating a better way for organizations to work with information, on-premises or in the cloud. For more information about OpenText (NASDAQ/TSX: OTEX), visit opentext.com.

Connect with us:

[OpenText CEO Mark Barrenechea's blog](#)

[Twitter](#) | [LinkedIn](#)