

# Cloud Elements

## ***Documents Hub Provisioning Guide***

## Table of Contents

[Introduction](#)

[Set up your Application with the Endpoint](#)

[Sign up for the Cloud Elements Service](#)

[Provision the Documents Hub Elements via the Cloud Elements Provisioning API](#)

[Multiple Application Users](#)

[Create Instance Template](#)

[HTTP Headers](#)

[Input JSON](#)

[Successful Response JSON](#)

[Example](#)

[Add Account User](#)

[HTTP Headers](#)

[Input JSON](#)

[Successful Response JSON](#)

[Example](#)

[Get Endpoint OAuth URL](#)

[HTTP Headers](#)

[Input Query Parameters](#)

[Successful Response JSON](#)

[Example](#)

[Handle Callback from the Endpoint](#)

[Create Instance for Account User](#)

[HTTP Headers](#)

[Input JSON](#)

[Successful Response JSON](#)

[Example](#)

[Multiple Application Users belonging to Multiple Customers \(Organizations\) within your Organization](#)

[Add Sub-account](#)

[Add Account User](#)

[Use The Documents Hub In Your Application](#)

[Ping](#)

[Upload File](#)

[Upload Files](#)

[Retrieve a File](#)

[Retrieve Access URLs](#)

[Retrieve Download Link](#)

[Delete a File or Folder](#)

[Search](#)

[Search Tags](#)

[List](#)

[Details](#)

[Move a File or Folder](#)

[Copy a File or Folder](#)

[Rename a File or Folder](#)

[Create a Folder](#)

[Add Tags](#)

[Remove Tags](#)

[Usage](#)

[Appendix](#)

[Create a new Dropbox platform application](#)

[Create a new Box platform application](#)

[Create a new Google Drive platform application](#)

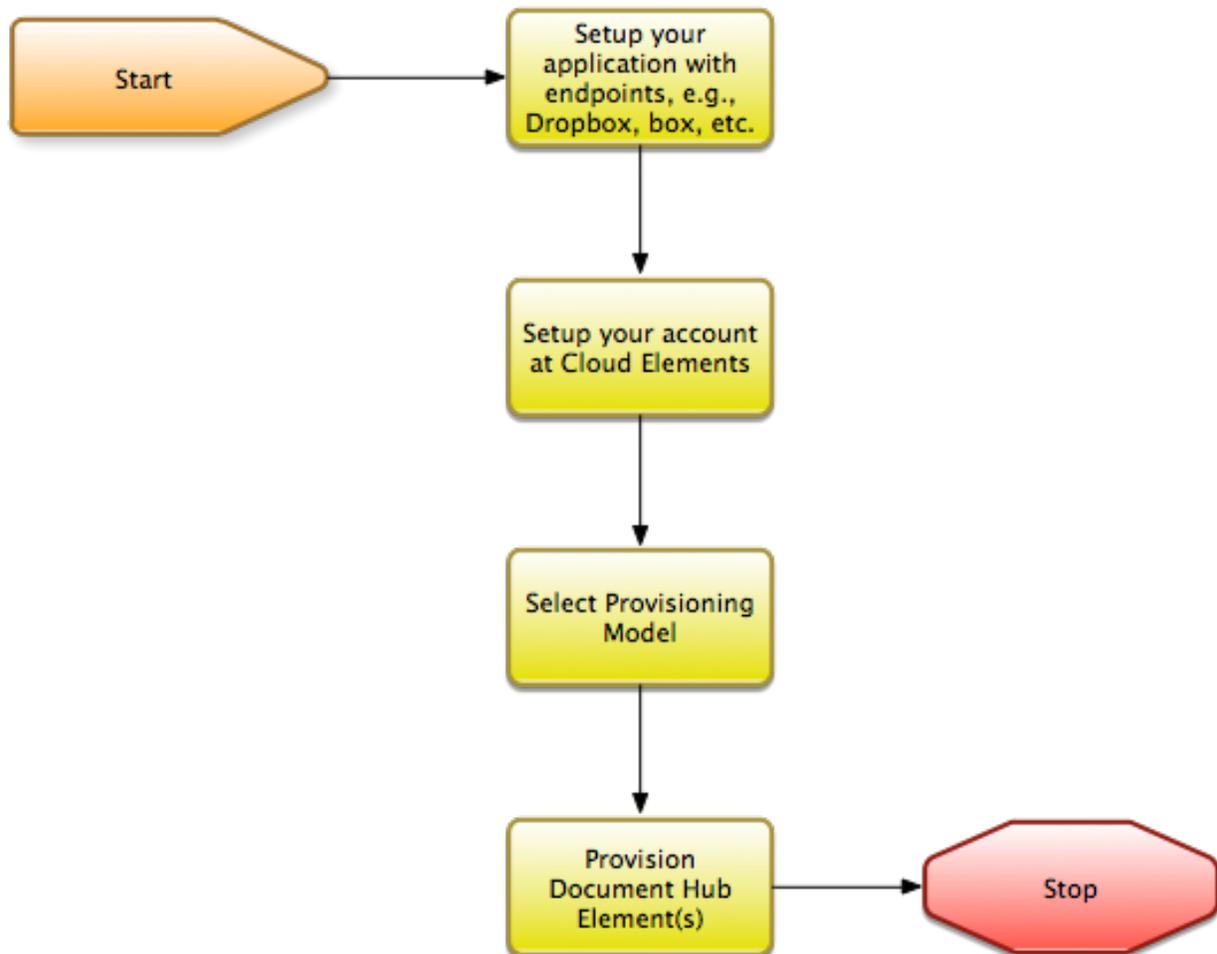
[Create a new OneDrive platform application](#)

[Create a new SharePoint platform application](#)

## Introduction

The Documents Hub provides a uniform API to allow applications to use various endpoints such as Box, Dropbox, Google Drive, OneDrive and SharePoint.

The following illustration shows the overall flow of the Documents Hub Element provisioning process.



In order to use one or more of the Documents Hub Elements in your application, via the uniform API, you have to first follow three steps:

1. Set up your application with the Documents Hub Element endpoint, e.g., Box, Dropbox, etc.
2. Sign up for the Cloud Elements Services.

3. Provision the Documents Hub Element.

The following sections will guide you through this process.

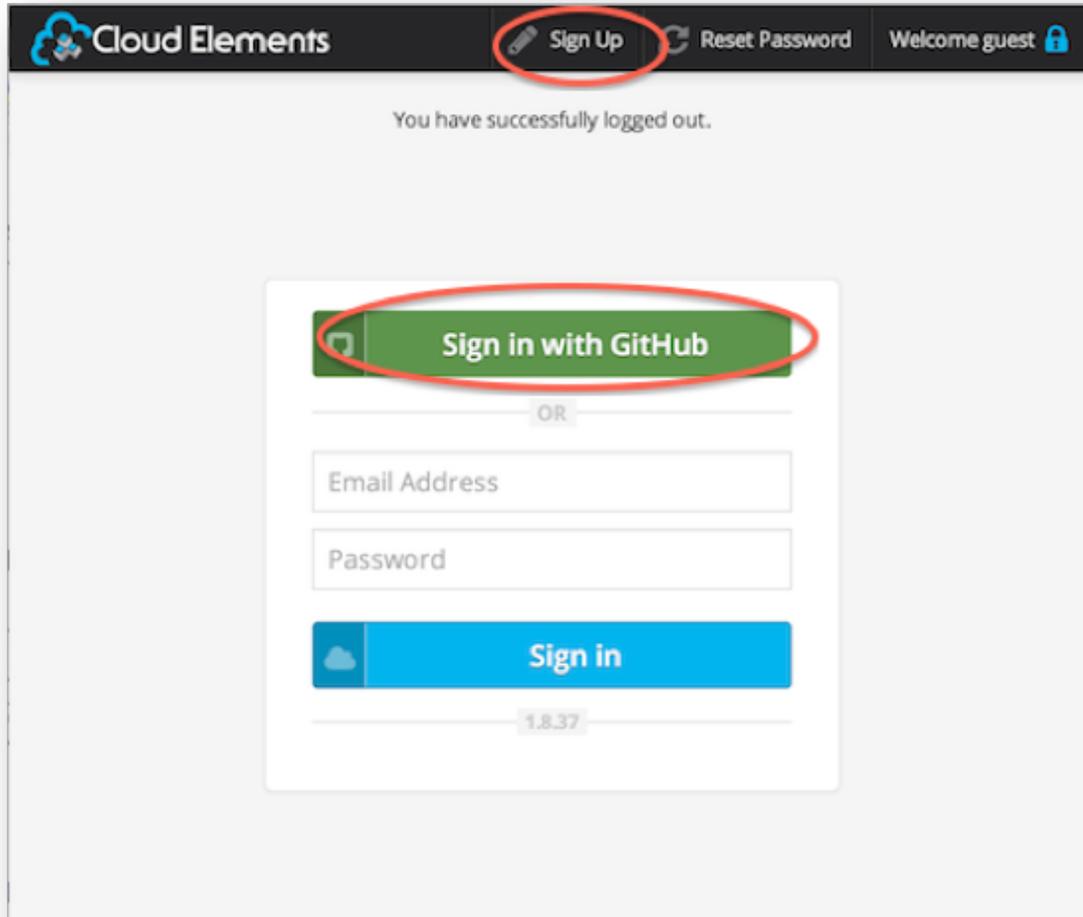
OAuth is an open standard for authorization. Cloud Elements uses OAuth to interact with elements in the Document Hub during the provisioning process. See [this document](#) for more information on OAuth.

### **Set up your Application with the Endpoint**

The Appendix contains instructions to set up your application with the endpoint for each supported Documents Hub endpoint ([Dropbox](#), [Box](#), [Google Drive](#), [OneDrive](#) and [SharePoint](#)).

After following one of these guides, the next step in the process is to sign up for the Cloud Elements service. The remainder of this guide assumes the use of Dropbox.

## Sign up for the Cloud Elements Service



To sign up for the Cloud Elements service, using a web browser, go to <https://console.cloud-elements.com>. To ensure security, you will need to sign up using a web browser, versus using the API, so you can keep your organization and user secrets (assigned during the sign up process) secure.

You can sign up for the Cloud Elements service using your GitHub account, or create a new account with Cloud Elements using the "Sign Up" link shown above.

If you choose to not use GitHub to sign up, you will then be required to validate your new account, and will then be required to reset your password.

After completing this process, choose the user Profile Settings as shown below to reset and retrieve the Organization and User secret.



At the bottom of the next screen, you will see your user and organization secrets, which you need to reset, copy and make a note of. These secrets will be used in the Element provisioning process.

Organization Secret	672aa88bb4e3235091de77900e3e299b	 Reset
User Secret	21fe6ec3374507e5415d5bd03f0e8415	 Reset

You're now ready to start provisioning the Documents Hub Elements via the Cloud Elements Provisioning APIs.

## Provision the Documents Hub Elements via the Cloud Elements Provisioning API

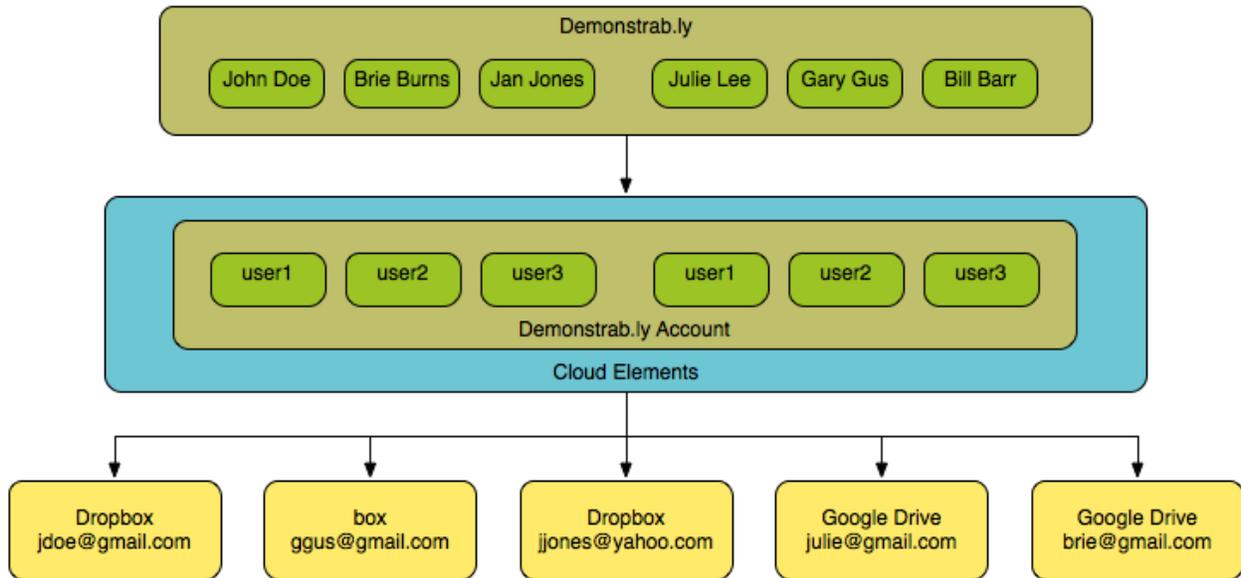
Before we start provisioning Elements, we need to understand how your application will use the Documents Hub Elements. There are three provisioning models for all Elements:

1. Single application user for your organization.
2. Multiple application users for your organization.
3. Multiple application users belonging to multiple customers (organizations) within your organization.

### Multiple Application Users

This model allows multiple application users of demonstrab.ly to provision Elements and in turn, use endpoint services via the Elements.

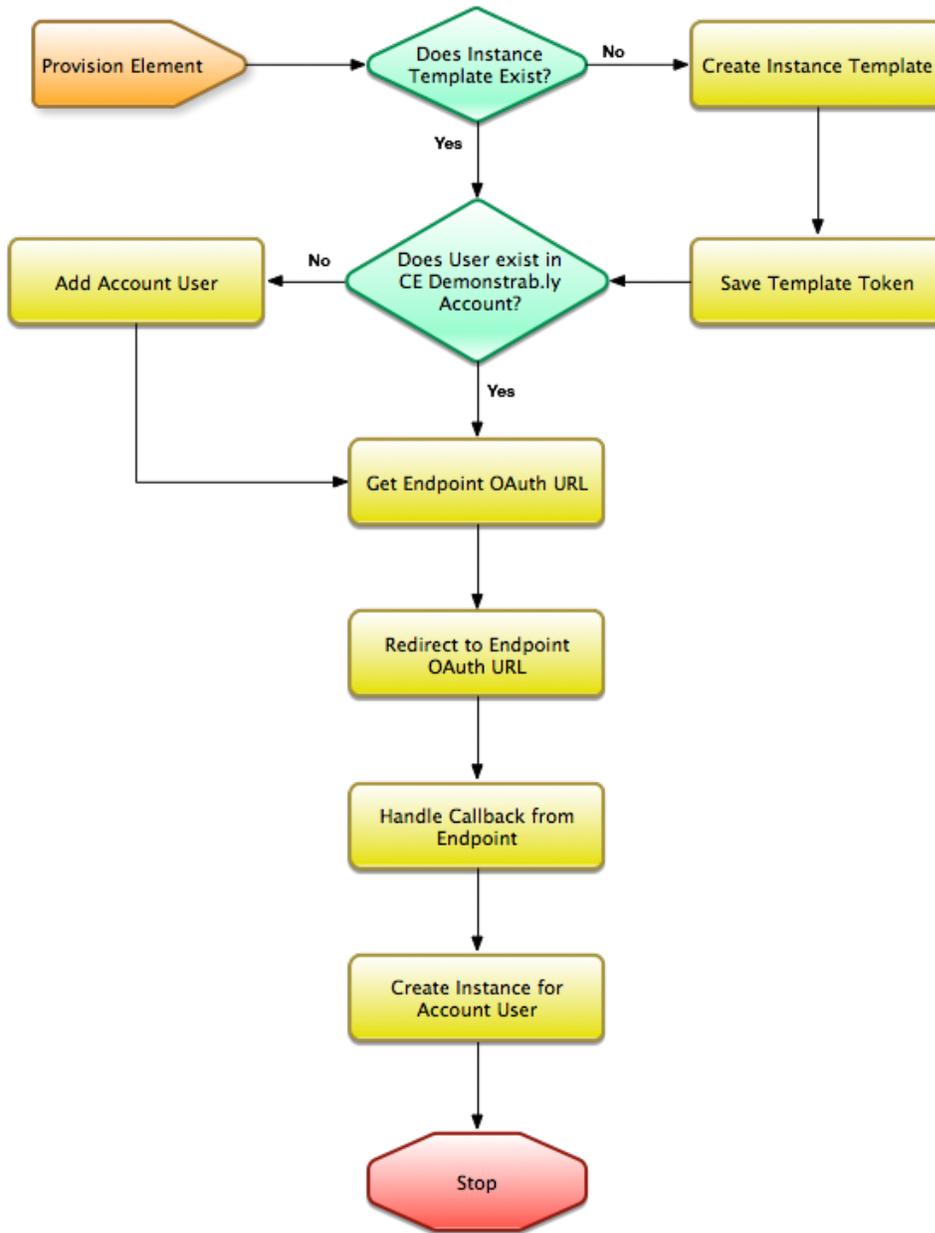
The following illustration shows how your application, demonstrab.ly, will access the various endpoints via the Documents Hub, via multiple users of the application.



Your application, [demonstrab.ly](#), will create a user in the Cloud Elements [demonstrab.ly](#) account, associated with each application user that needs to use the Documents Hub Elements. The [demonstrab.ly](#) account is simply a wrapper for all the users, and is implicitly created upon signup.

**Note:** The [demonstrab.ly](#) application is not required to pass any user information other than a user ID to Cloud Elements. The user ID is encrypted at rest.

The multiple users provisioning flow is illustrated by the following diagram.



The following sections describe the API to use for each step in the above flowchart.

### Create Instance Template

The API for this step is `/provisioning/createInstanceTemplate` and is documented in the API Documentation, which can be found at [https://api.cloud-elements.com/elements/apidocs/#!/provisioning/\\_post\\_9](https://api.cloud-elements.com/elements/apidocs/#!/provisioning/_post_9).

An Element token is returned upon successful execution of this API. This token needs to be retained by the application for use as the "ParentElement" header value for subsequent requests.

A sample request illustrating the `/provisioning/createInstanceTemplate` API is shown below.

## HTTP Headers

```
Authorization : User <INSERT_USER_SECRET>, Organization <INSERT_ORGANIZATION_SECRET>
```

## Input JSON

```
{
  account: {
    key: "<INSERT_ACCOUNT_KEY>" (email address used to sign up)
  },
  instance: {
    name: "<INSERT_INSTANCE_NAME>",
    element: {
      key: "dropbox"
    },
    properties: [
      {
        key: "oauth.api.key",
        propertyValue : "<INSERT_OAUTH_API_KEY>"
      },
      {
        key: "oauth.api.secret",
        propertyValue : "<INSERT_OAUTH_API_SECRET>"
      },
      {
        key: "oauth.callback.url",
        propertyValue : "<INSERT_OAUTH_CALLBACK_URL>"
      }
    ]
  }
}
```

## Successful Response JSON

```
{
  dataReturned: true,
  success: true,
  accountElementConfig: {
    id: <YOUR_ID>,
    token: "<YOUR_TOKEN>",
    name: "<YOUR_INSTANCE_NAME>",
    isAccountProvision: true,
    isValid: true,
    resoldProvision: false
  }
}
```

## Example

### Request

```
curl -X POST \
  -H 'Authorization: \
    User 39fceb6cdfd3363d7c5d3ad1a6ae2b4c, \
    Organization 6d0c70e362c3ecc7fc8bab5d3d6a30e3' \
  -H 'Content-Type: application/json' \
  -d @create-instance-template.json \
  'https://console.cloud-elements.com/elements/api-
  v1/provisioning/createInstanceTemplate'
```

**create-instance-template.json**

```
{
  "account": {
    "key": "joseph.pulaski@oldschool.com"
  },
  "instance": {
    "name": "documents-demo-provisioning",
    "element": {
      "key": "dropbox"
    }
  },
  "properties": [
    {
      "key": "oauth.api.key",
      "propertyValue": "fake_dropbox_api_key"
    },
    {
      "key": "oauth.api.secret",
      "propertyValue": "fake_dropbox_api_secret"
    },
    {
      "key": "oauth.callback.url",
      "propertyValue": "http://fake.oauth.callback/url"
    }
  ]
}
```

### Response

```
{
  "dataReturned": true,
  "success": true,
  "accountElementConfig": {
    "id": 16846,
    "token": "f0a2621b48c0f0aa56fc6c42f76dc22d",
    "name": "documents-demo-provisioning",
    "isAccountProvision": true,
    "isValid": true,
    "resoldProvision": false
  }
}
```

## Add Account User

The API for this step is `/provisioning/addAccountUser` and is documented in the API Documentation, which can be found at [https://api.cloud-elements.com/elements/apidocs/#!/provisioning/\\_post\\_7](https://api.cloud-elements.com/elements/apidocs/#!/provisioning/_post_7).

This step is not required if your application is always going to have just one user, i.e., the user that signed up for the Cloud Elements service.

However, most applications support more than just one user, so if that is the case with your application, this step is necessary so that you can ensure that each Element instance is associated with the user that owns the endpoint account. Segregating the instance by users also allows you to get API usage metrics by user.

A sample request illustrating the `/provisioning/addAccountUser` API is shown below.

## HTTP Headers

```
Authorization : User <INSERT_USER_SECRET>, Organization <INSERT_ORGANIZATION_SECRET>
```

## Input JSON

```
{
  account: {
    key: "<INSERT_ACCOUNT_KEY>"
  },
  user: {
    lastName: "<INSERT_LAST_NAME>",
    firstName: "<INSERT_FIRST_NAME>",
    password: "<INSERT_PASSWORD>",
    phone: "<INSERT_PHONE_NUMBER>",
    emailAddress: "<INSERT_EMAIL_ADDRESS>",
    username: "<INSERT_USERNAME>"
  }
}
```

## Successful Response JSON

```
{
  id: <YOUR_ID>,
  lastName: "<YOUR_LAST_NAME>",
  dataReturned: true,
  phone: "<YOUR_PHONE_NUMBER>",
  accountId: <YOUR_ACCOUNT_ID>,
  expired: false,
  credentialsExpired: false,
  active: true,
  emailAddress: "<YOUR_EMAIL_ADDRESS>",
  locked: false,
  fullName: "<YOUR_FULL_NAME>",
  firstName: "<YOUR_FIRST_NAME>",
  success: true
}
```

## Example

### Request

```
curl -X POST \
  -H 'Authorization: \
    User 39fceb6cdfd3363d7c5d3ad1a6ae2b4c, \
    Organization 6d0c70e362c3ecc7fc8bab5d3d6a30e3' \
  -H 'Content-Type: application/json' \
  -d @add-account-user.json \
  'https://console.cloud-elements.com/elements/api-v1/provisioning/addAccountUser'
```

### add-account-user.json

```
{
  "account": {
    "key": "joseph.pulaski@oldschool.com"
  },
  "user": {
    "lastName": "ricard",
    "firstName": "frank",
    "password": "password1",
    "phone": "2122122222",
    "emailAddress": "frank.ricard@oldschool.com",
    "username": "frank.ricard@oldschool.com"
  }
}
```

### Response

```
{
  "accountId": 2,
  "active": true,
  "credentialsExpired": false,
  "dataReturned": true,
  "emailAddress": "frank.ricard@oldschool.com",
  "expired": false,
  "firstName": "frank",
  "fullName": "frank ricard",
  "id": 16,
  "lastName": "ricard",
  "locked": false,
  "phone": "2122122222",
  "success": true,
  "username": "frank.ricard@oldschool.com"
}
```

## Get Endpoint OAuth URL

The API for this step is `/provisioning/getOAuthUrl` and is documented in the API Documentation, which can be found at [https://api.cloud-elements.com/elements/apidocs/#!/provisioning/\\_get\\_18](https://api.cloud-elements.com/elements/apidocs/#!/provisioning/_get_18).

**Note:** Provide the instance template token obtained from the [Create Instance Template](#) step for the "ParentElement" header value, in addition to the user secret and the organization secret.

The result of this API invocation is an OAuth redirect URL from the endpoint. Your application should now redirect to this URL, which in turn will present the OAuth authentication and authorization page to the user.

A sample request illustrating the `/provisioning/getOAuthUrl` API is shown below.

## HTTP Headers

```
Authorization : User <INSERT_USER_SECRET>, Organization <INSERT_ORGANIZATION_SECRET>,
ParentElement <INSERT_PARENT_ELEMENT_TOKEN>
```

## Input Query Parameters

```
?elementKey=dropbox
  &apiKey=<INSERT_API_KEY>
  &apiSecret=<INSERT_API_SECRET>
  &callbackUrl=<INSERT_CALLBACK_URL>
  &state : "any custom value to pass in"
```

## Successful Response JSON

```
{
  dataReturned: true,
  value:
  "https://www.dropbox.com/1/oauth2/authorize?response_type=code&client_id=kurrvca7gldc
  lvq&redirect_uri=https%3A%2F%2Fconsole.cloud-
  elements.com%2Felements%2Fdocumentshub%2Fhubcallback.html&state=sadfsad",
  success: true
}
```

## Example

### Request

```
curl -X GET \
  -H 'Authorization: \
    User 39fceb6cdfd3363d7c5d3ad1a6ae2b4c, \
    Organization 6d0c70e362c3ecc7fc8bab5d3d6a30e3 \
    ParentElement f0a2621b48c0f0aa56fc6c42f76dc22d' \
  -H 'Content-Type: application/json' \
  'https://console.cloud-
  elements.com/elements/api-
  v1/provisioning/getOAuthUrl?elementKey=dropbox&apiKey=fake_dropbox_api_key&apiSecret
  =fake_dropbox_api_secret&callbackUrl=http://fake.oauth.callback/url&state=joseph-
  blue-palaski'
```

### Response

```
{
  "dataReturned": true,
  "success": true,
  "value":
  "https://www.dropbox.com/1/oauth2/authorize?response_type=code&client_id=zulws95ofcvh
  el5&redirect_uri=http%3A%2F%2Ffake.oauth.callback%2Furl&state=joseph-blue-pulaski"
}
```

## Handle Callback from the Endpoint

Upon successful authentication and authorization by the user, the endpoint will redirect to the callback URL you provided when you setup your application with the endpoint, in our example, <https://www.demonstrab.ly/authz>.

The endpoint will also provide two query string parameters: "state" and "code". The value for the "state" parameter will be the name of the endpoint, e.g., "dropbox" in our example, and the value for the "code" parameter is the code required by Cloud Elements to retrieve the OAuth access and refresh tokens from the endpoint.

If the user denies authentication and/or authorization, there will be a query string parameter called "error" instead of the "code" parameter. In this case, your application can handle the error gracefully.

## Create Instance for Account User

Upon successful authentication and authorization by the user, and when the OAuth token access code is successfully retrieved from the endpoint callback, invoke the `/provisioning/createInstanceForAccountUser` API, which can be found at [https://api.cloud-elements.com/elements/apidocs/#!/provisioning/post\\_15](https://api.cloud-elements.com/elements/apidocs/#!/provisioning/post_15).

The result of this API will be an Element token, which needs to be retained by your application as the authentication key associated with this user for the specified endpoint. Each endpoint will result in a separate Element token for the same user.

This API ends the provisioning flow, and now your application can start using the Documents Hub APIs using the token returned from this API.

A sample request illustrating the `/provisioning/createInstanceForAccountUser` API is shown below. See the [Create Instance Template](#) section for details about the `ParentElement` header value.

### HTTP Headers

```
Authorization : User <INSERT_USER_SECRET>, Organization <INSERT_ORGANIZATION_SECRET>,
ParentElement <INSERT_PARENT_ELEMENT_TOKEN>
```

## Input JSON

```
{
  account: {
    key: "<INSERT_ACCOUNT_KEY>"
  },
  instance: {
    tags: [
      {
        name: "<INSERT_OPTIONAL_TAG_NAME>"
      }
    ],
    element: {
      key: "dropbox"
    },
    name: "<INSERT_INSTANCE_NAME>",
    properties: [
      {
        key: "document.tagging",
        propertyValue: "<INSERT_TRUE_OR_FALSE>"
      },
      {
        key: "oauth.user.id",
        propertyValue: "<INSERT_OAUTH_USER_ID>"
      },
      {
        key: "oauth.user.token",
        propertyValue: "<INSERT_OAUTH_USER_TOKEN>"
      }
    ]
  },
  user: {
    emailAddress: "<INSERT_EMAIL_ADDRESS>"
  }
}
```

## Successful Response JSON

```
{
  dataReturned: true,
  success: true
}
```

## Example

### Request

```
curl -X POST \
  -H 'Authorization: \
    User 39fceb6cdfd3363d7c5d3ad1a6ae2b4c, \
    Organization 6d0c70e362c3ecc7fc8bab5d3d6a30e3, \
    ParentElement f0a2621b48c0f0aa56fc6c42f76dc22d' \
  -H 'Content-Type: application/json' \
  -d @create-instance-for-account-user.json \
  'https://console.cloud-elements.com/elements/api-
  v1/provisioning/createInstanceForAccountUser'
```

#### create-instance-for-account-user.json

```
{
  "account": {
    "key": "joseph.pulaski@oldschool.com"
  },
  "code": "3e0fmj5xTsAAAAAAAAAAEK2bnmLtpvfsXd3CwOXG650A",
  "instance": {
    "tags": [
      {
        "name": "sandbox"
      }
    ],
    "element": {
      "key": "dropbox"
    },
    "name": "frank-ricard-dropbox",
    "properties": [
      {
        "key": "document.tagging",
        "propertyValue": "false"
      },
      {
        "key": "oauth.callback.url",
        "propertyValue": "http://fake.oauth.callback/url"
      },
      {
        "key": "oauth.api.key",
        "propertyValue": "fake_dropbox_api_key"
      },
      {
        "key": "oauth.api.secret",
        "propertyValue": "fake_dropbox_api_secret"
      }
    ]
  },
  "user": {
    "emailAddress": "frank.ricard@oldschool.com"
  }
}
```

#### Response

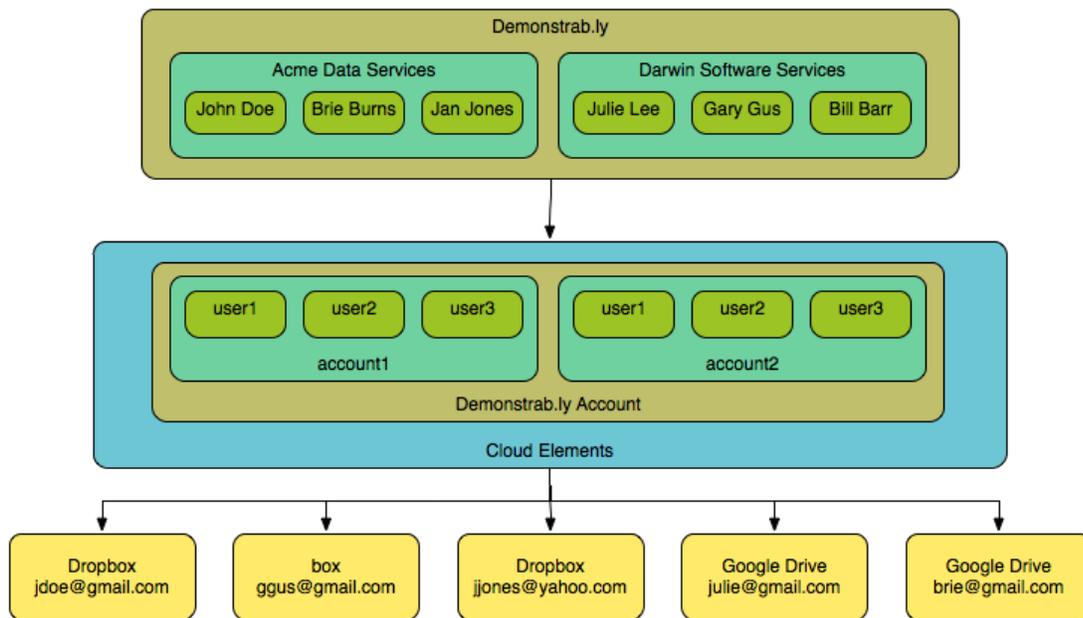
```

{
  "dataReturned": true,
  "name": "frank-ricard-dropbox",
  "success": true,
  "token": "852e8bad78b4eab8cea6e912cfad6b4b"
}

```

## Multiple Application Users belonging to Multiple Customers (Organizations) within your Organization

This model allows multiple application users of customers (organizations) within demonstrab.ly to provision Elements and in turn, use endpoint services via the Elements.

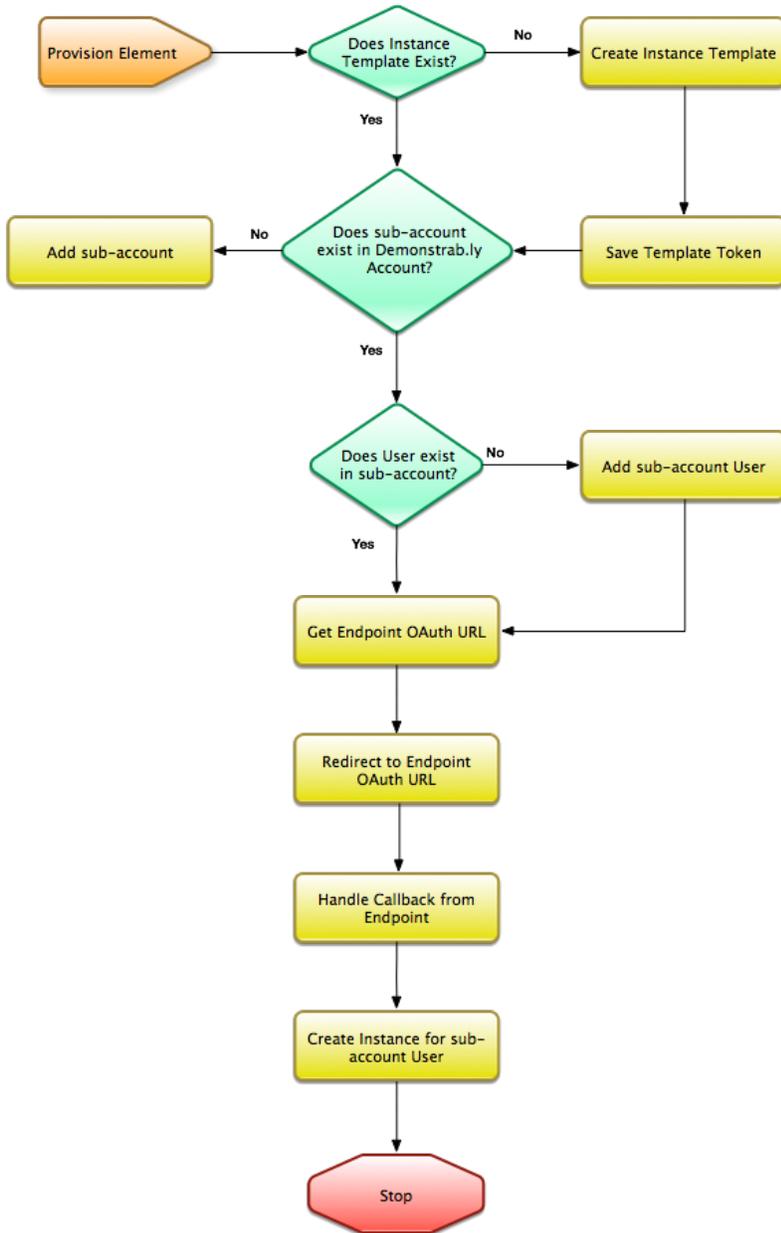


The following illustration shows how your application, demonstrab.ly, will access the various endpoints via the Documents Hub, via multiple users of the application, within different organizations that are customers of demonstrab.ly.

Your application, demonstrab.ly, will create a sub-account for each of your customer (organization) within the Cloud Elements demonstrab.ly account. Each of your customer's user will be associated with a user in the customer's sub-account in Cloud Elements.

Direct users of demonstrab.ly can also be supported in this provisioning model.

The multiple users within demonstrab.ly customers (organizations) provisioning flow are illustrated by the following diagram.



**Note:** The *demonstrab.ly* application is not required to pass any user information other than a user ID to Cloud Elements. The user ID is encrypted at rest.

The primary difference between the multiple user flow and the multiple customer (organization) user flow is that in the first case, all the *demonstrab.ly* users are directly within the *demonstrab.ly* account in the Cloud Elements system, while in the second case, the users are within sub-accounts in the *demonstrab.ly* account.

The following sections describe the API to use for each step in the above flowchart.

The APIs for this provisioning model are almost the same as the previous provisioning model. The exceptions are the following APIs.

### **Add Sub-account**

The API for this step is `/provisioning/createAccount` and is documented in the API Documentation, which can be found at [https://api.cloud-elements.com/elements/apidocs/#!/provisioning/\\_post\\_2](https://api.cloud-elements.com/elements/apidocs/#!/provisioning/_post_2).

This API will create a sub-account for your customer's organization, under the `demonstrab.ly` account, and further, will allow you to create your customer's users under this sub-account.

### **Add Account User**

The API for this step is `/provisioning/addAccountUser` and is documented in the API Documentation, which can be found at [https://api.cloud-elements.com/elements/apidocs/#!/provisioning/\\_post\\_7](https://api.cloud-elements.com/elements/apidocs/#!/provisioning/_post_7).

The difference in the invocation of this API for this provisioning model versus the previous model is that you will pass in the sub-account, i.e., your customer's account, as the parent account instead of the `demonstrab.ly` account.

All the other APIs are the same between the two provisioning models. The only difference is that for the previous provisioning model, the Account key is the `demonstrab.ly` account key, while for this provisioning model, the Account key passed in is the sub-account key, or the key for customers of `demonstrab.ly`.

### **Use The Documents Hub In Your Application**

Now that you have finished provisioning your Dropbox Element, you are ready to use the Document APIs to manipulate files and folders in your Dropbox account. Below are examples for each of the Documents Hub APIs.

**Note:** Any of the tag APIs require that you chose to enable tagging when you provisioned your Documents Hub Element.

## Ping

```
curl -X GET \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  'https://console.cloud-elements.com/elements/api-v1/document/ping'
```

```
{
  "dataReturned": true,
  "element": {
    "value": "dropbox"
  },
  "success": true
}
```

## Upload File

```
curl -X POST \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  -F file=@test.txt \
  'https://console.cloud-elements.com/elements/api-v1/document/uploadFile?path=/&description=Test%20file&tags=ow|now|brown|cow&overwrite=false'
```

```
{
  "dataReturned": true,
  "fileDescription": "Test file",
  "fileName": "test.txt",
  "fileTags": [
    "ow",
    "now",
    "brown",
    "cow"
  ],
  "fileType": "text/plain",
  "hash": "-1726965043",
  "isDirectory": false,
  "modifiedDate": "2014-04-17T13:50:06 UTC",
  "path": "/test.txt",
  "revision": "8550209611e",
  "size": 30,
  "success": true
}
```

## Upload Files

```

curl -X POST \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  -F file=@test.txt \
  -F file=@test1.txt \
  'https://console.cloud-elements.com/elements/api-v1/document/uploadFiles?path='
{
  "dataReturned": true,
  "success": true,
  "recordCount": 2,
  "records": [
    {
      "fileName": "test.txt",
      "fileType": "text/plain",
      "hash": "11745080",
      "isDirectory": false,
      "modifiedDate": "2014-04-16T23:08:18 UTC",
      "originalFileSize": 29,
      "path": "/test.txt",
      "revision": "8460209611e",
      "size": 29
    },
    {
      "fileName": "test1.txt",
      "fileType": "text/plain",
      "hash": "1587229379",
      "isDirectory": false,
      "modifiedDate": "2014-04-16T23:08:19 UTC",
      "originalFileSize": 34,
      "path": "/test1.txt",
      "revision": "8470209611e",
      "size": 34
    }
  ]
}

```

## Retrieve a File

```

curl -X GET \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  'https://console.cloud-elements.com/elements/api-
v1/document/get?path=/Running/marathon_smart_coach_plan.pdf'

```

<FILE DATA>

## Retrieve Access URLs

```

curl -X GET \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  'https://console.cloud-elements.com/elements/api-
v1/document/getAccessUrls?path=/Running/marathon_smart_coach_plan.pdf'

```

```
{
  "dataReturned": true,
  "links": {
    "download":
    "https://dl.dropboxusercontent.com/1/view/73gc1x9942owrcb/Running/marathon_smart_coach_plan.pdf",
    "expires": "Wed, 16 Apr 2014 23:12:12 +0000",
    "view":
    "https://www.dropbox.com/s/wkhfs8ggygfh2e/marathon_smart_coach_plan.pdf"
  },
  "success": true
}
```

## Retrieve Download Link

```
curl -X GET \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  'https://console.cloud-elements.com/elements/api-v1/document/getDownloadLink?path=/Running/marathon_smart_coach_plan.pdf'
```

```
{
  "dataReturned": true,
  "success": true,
  "value": "https://console.cloud-elements.com/elements/api-v1/dropbox/download?id=6987f3fb0d6330c47961f83c127796a4"
}
```

## Delete a File or Folder

```
curl -X DELETE \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  'https://console.cloud-elements.com/elements/api-v1/document/delete?path=/test.txt&emptyTrash=false'
```

```
{
  "dataReturned": true,
  "success": true,
  "value": true
}
```

## Search

```
curl -X GET \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  'https://console.cloud-elements.com/elements/api-v1/document/search?path=/Running&text=*&startTime=01-01-2012%2011:11:11&endTime=01-01-2015%2011:11:11&pageNumber=1&pageSize=100'
```

```

{
  "dataReturned": true,
  "pageCount": 1,
  "results": {
    "recordCount": 2,
    "records": [
      {
        "fileName": "alderfer_closure_10_16_13.pdf",
        "hash": "1915502852",
        "isDirectory": false,
        "modifiedDate": "2014-01-17T16:39:38 UTC",
        "path": "/Running/alderfer_closure_10_16_13.pdf",
        "revision": "6a50209611e",
        "size": 278993
      },
      {
        "fileName": "marathon_smart_coach_plan.pdf",
        "hash": "-1374199044",
        "isDirectory": false,
        "modifiedDate": "2014-01-17T16:39:38 UTC",
        "path": "/Running/marathon_smart_coach_plan.pdf",
        "revision": "6a40209611e",
        "size": 139276
      }
    ]
  },
  "rowStartNumber": 1,
  "success": true,
  "totalCount": 2
}

```

## Search Tags

```

curl -X GET \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  'https://console.cloud-elements.com/elements/api-
v1/document/searchTags?path=/&tags=brown|cow&pageNumber=1&pageSize=100'

```

```

{
  "dataReturned": true,
  "pageCount": 0,
  "results": {
    "recordCount": 1,
    "records": [
      {
        "fileDescription": "Test file",
        "fileName": "test.txt",
        "fileTags": [
          "ow",
          "now",
          "brown",
          "cow"
        ],
        "hash": "-688952649",
        "id": 313,
        "isDirectory": false,
        "modifiedDate": "2014-04-17T13:56:07 UTC",
        "path": "/test.txt",
        "revision": "8570209611e",
        "size": 30
      }
    ]
  },
  "rowStartNumber": 0,
  "success": true,
  "totalCount": 0
}

```

## List

```

curl -X GET \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  'https://console.cloud-elements.com/elements/api-
v1/document/list?path=/&fetchTags=true&pageNumber=1&pageSize=100'

```

```
{
  "dataReturned": true,
  "pageCount": 1,
  "results": {
    "recordCount": 2,
    "records": [
      {
        "fileName": "Running",
        "fileType": "dir",
        "isDirectory": true,
        "path": "/Running"
      },
      {
        "fileDescription": "Test file",
        "fileName": "test.txt",
        "fileTags": [
          "ow",
          "now",
          "brown",
          "cow"
        ],
        "hash": "-688952649",
        "id": 313,
        "isDirectory": false,
        "modifiedDate": "2014-04-17T13:56:07 UTC",
        "path": "/test.txt",
        "revision": "8570209611e",
        "size": 30
      }
    ]
  },
  "rowStartNumber": 1,
  "success": true,
  "totalCount": 2
}
```

## Details

```
curl -X GET \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  'https://console.cloud-elements.com/elements/api-
  v1/document/details?path=/Development'
```

```
{
  "dataReturned": true,
  "fileName": "Development",
  "fileType": "dir",
  "isDirectory": true,
  "path": "/Development",
  "success": true
}
```

## Move a File or Folder

```
curl -X PUT \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  -H 'Content-Type: application/json' \
  -d @move.json \
  'https://console.cloud-elements.com/elements/api-v1/document/move'
```

**move.json**

```
{
  "dest": {
    "deepCopy": "true",
    "path": "/test_update.txt"
  },
  "src": {
    "path": "/test.txt"
  }
}
```

```
{
  "dataReturned": true,
  "fileName": "test_update.txt",
  "hash": "-1902794092",
  "isDirectory": false,
  "modifiedDate": "2014-04-16T23:13:13 UTC",
  "path": "/test_update.txt",
  "revision": "8500209611e",
  "size": 34,
  "success": true
}
```

## Copy a File or Folder

```
curl -X POST \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  -H 'Content-Type: application/json' \
  -d @copy.json \
  'https://console.cloud-elements.com/elements/api-v1/document/copy'
```

**copy.json**

```
{
  "dest": {
    "deepCopy": "false",
    "path": "/test_copy.txt"
  },
  "src": {
    "path": "/test_update.txt"
  }
}
```

```
{
  "dataReturned": true,
  "fileName": "test.txt",
  "hash": "556942229",
  "isDirectory": false,
  "modifiedDate": "2014-04-16T23:13:24 UTC",
  "path": "/test.txt",
  "revision": "8510209611e",
  "size": 34,
  "success": true
}
```

## Rename a File or Folder

```
curl -X PUT \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  -H 'Content-Type: application/json' \
  -d @rename.json \
  'https://console.cloud-elements.com/elements/api-v1/document/rename'
```

**rename.json**

```
{
  "file": {
    "path": "/test_update.txt"
  },
  "newName": "test_rename.txt"
}
```

```
{
  "dataReturned": true,
  "fileName": "test_rename.txt",
  "hash": "11872810",
  "isDirectory": false,
  "modifiedDate": "2014-04-16T23:13:34 UTC",
  "path": "/test_rename.txt",
  "revision": "8530209611e",
  "size": 34,
  "success": true
}
```

## Create a Folder

```
curl -X POST \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  -H 'Content-Type: application/json' \
  -d @create-folder.json \
  'https://console.cloud-elements.com/elements/api-v1/document/createFolder'
```

**create-folder.json**

```
{
  "folder": {
    "deepCopy": "false",
    "path": "/test_folder"
  },
  "description": "This is a test folder created with Cloud Elements"
}

{
  "createdDate": "2014-04-16T23:13:44 UTC",
  "dataReturned": true,
  "fileDescription": "This is a test folder created with Cloud Elements",
  "fileName": "test_folder",
  "fileType": "dir",
  "isDirectory": true,
  "path": "/test_folder",
  "success": true
}
```

## Add Tags

```
curl -X PUT \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  -H 'Content-Type: application/json' \
  -d @add-tags.json \
  'https://console.cloud-elements.com/elements/api-v1/document/addTags'
```

### add-tags.json

```
{
  "tags": "ow|now|brown|cow",
  "file": {
    "path": "/test.txt"
  }
}

{
  "dataReturned": true,
  "fileName": "test.txt",
  "fileTags": [
    "ow",
    "now",
    "brown",
    "cow"
  ],
  "hash": "556942229",
  "id": 311,
  "isDirectory": false,
  "modifiedDate": "2014-04-16T23:13:24 UTC",
  "path": "/test.txt",
  "revision": "8510209611e",
  "size": 34,
  "success": true
}
```

## Remove Tags

```
curl -X PUT \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  -H 'Content-Type: application/json' \
  -d @remove-tags.json \
  'https://console.cloud-elements.com/elements/api-v1/document/removeTags'
```

**remove-tags.json**

```
{
  "tags": "ow|now",
  "file": {
    "path": "/test.txt"
  }
}

{
  "dataReturned": true,
  "fileName": "test.txt",
  "fileTags": [
    "brown",
    "cow"
  ],
  "hash": "556942229",
  "id": 311,
  "isDirectory": false,
  "modifiedDate": "2014-04-16T23:13:24 UTC",
  "path": "/test.txt",
  "revision": "8510209611e",
  "size": 34,
  "success": true
}
```

## Usage

```
curl -X GET \
  -H 'Authorization: Element 852e8bad78b4eab8cea6e912cfad6b4b' \
  -H 'Content-Type: application/json' \
  'https://console.cloud-elements.com/elements/api-v1/document/usage'
```

```
{
  "dataReturned": true,
  "shared": 831732517,
  "success": true,
  "total": 11005853696,
  "used": 2393065113
}
```

```
{
  "dataReturned": true,
  "fileName": "test.txt",
  "fileTags": [
    "brown",
    "cow"
  ],
  "hash": "556942229",
  "id": 311,
  "isDirectory": false,
  "modifiedDate": "2014-04-16T23:13:24 UTC",
  "path": "/test.txt",
  "revision": "8510209611e",
  "size": 34,
  "success": true
}
```

## **Appendix**

### **Create a new Dropbox platform application**

In this section, we will use Dropbox as the example to set up your application with the endpoint.

Via a web browser, go to <https://www.dropbox.com/developers/apps/create>.

Select the Dropbox API app option, select the appropriate options and enter a name for your application.

For this example, let's assume that the URL for your application is <https://www.demonstrab.ly>.

## Create a new Dropbox Platform app

What type of app do you want to create?

<input type="radio"/>  <b>Drop-ins app</b> Chooser or Saver	<input checked="" type="radio"/>  <b>Dropbox API app</b> Sync API, Datastore API, or Core API
---	---

What type of data does your app need to store on Dropbox?

Files and datastores

Datastores only

Can your app be limited to its own, private folder?

Yes — My app only needs access to files it creates.

No — My app needs access to files already on Dropbox.

What type of files does your app need access to?

Specific file types — My app only needs access to certain file types, like text or photos.

All file types — My app needs access to a user's full Dropbox. Only supported via the [Core API](#).

Provide an app name, and you're on your way.

Demonstrab.ly

[Create app](#)

On the next screen, you'll be required to enter a callback URL from the endpoint. This URL will be in your application's address space. You will be required to retrieve some information returned on this URL by the endpoint. The details of handling the callback are elsewhere in this document.

The screenshot shows the 'Details' tab of a Dropbox application named 'Demonstrab.ly'. The interface includes several sections with labels and values, and buttons for actions.

Demonstrab.ly	
Settings	Details
Status	Development <span>Apply for production</span>
Development users	Only you <span>Enable additional users</span>
Permission type	Full Dropbox
App key	h3pb6nzqojwpxkq
App secret	weuythx5hp4ji7
OAuth redirect URIs	<p>https://www.demonstrab.ly/authz</p> <input type="text" value="https://"/> <span>Add</span> <small>http:// allowed only for localhost URIs</small>
Drop-ins domains	<input type="text" value="example.com"/> <span>Add</span> <small>If using <a href="#">Drop-ins</a> on a website, the domain of that site.</small>
Datstores	<a href="#">Browse datstores</a>
Delete app	<span>Delete app</span>

For our example, we'll use a callback URL of <https://www.demonstrab.ly/authz>. Enter this URL on the following screen at Dropbox.

Please make a note of the "App key" and "App secret" assigned by Dropbox for the Demonstrab.ly application.

### Create a new Box platform application

In this section, we will use Box as the example to set up your application with the endpoint.

Via a web browser, go to <https://app.box.com/developers/services/edit/>.

Enter the name of your application, select the appropriate options and click "Create Application".

For this example, let's assume that the URL for your application is <https://www.demonstrab.ly>.

box DEVELOPERS



### Create a Box Application

Demonstrab.ly

- Box Content**  
Access the content management features available in the Box Web App and extend them for use in your own application. [Learn More](#)
- Box View**  
Convert PDF and Office documents to HTML for easy display in web and mobile applications. [Learn More](#)

[Create Application](#)

By clicking 'Create Application', you agree to the terms of service for:  
[Box Content](#) | [Box View](#)

After receiving confirmation that your application is created, click "Configure your application".

Box Content Application Created



Success!

Your application has been successfully created! Take a look at the [getting started guide](#) or the [API documentation](#) for guidance.

Configure your application

On the next screen, you'll be required to enter a callback URL from the endpoint. This URL will be in your application's address space. You will be required to retrieve some information returned on this URL by the endpoint. The details of handling the callback are elsewhere in this document.

**box** DEVELOPERS

Editing Demonstrab.ly

---

### General Information

Application name:	<input type="text" value="Demonstrab.ly"/>	Ex: MyBoxApp
Application description:	<input style="height: 40px;" type="text"/>	Ex: MyBoxApp is an online productivity suite
Support email:	<input type="text"/>	Ex: support@myboxapp.com
Website URL (optional):	<input type="text"/>	Ex: http://myboxapp.com
Content API Access Only:	<input checked="" type="radio"/>	This key can only call the Box Content API
View API Access Only:	<input type="radio"/>	This key can only call the Box View API

---

### OAuth2 Parameters

client_id:	fd5w2064yvraeturlikd8u3pteoj7s1r	client_id as specified in the OAuth2 spec
client_secret:	<input type="text" value="FEIv1do7PUHExO3rwwhX8fkMqWkYLFF"/>	client_secret as specified in the OAuth2 spec (leave blank to reset)
redirect_uri:	<input type="text" value="https://demonstrab.ly/authz"/>	redirect_uri as specified in the OAuth2 spec
Scopes:	<input checked="" type="checkbox"/> Read and write all files and folders <input type="checkbox"/> Manage an enterprise	Enter the set of scopes you request users to authorize for your app
Developer token:	You do not currently have a developer token.	Developer tokens allow you to use the Box API to access your personal Box account.
	<input type="button" value="Create a developer token"/>	

For our example, we'll use a callback URL of <https://www.demonstrab.ly/authz>. Enter this URL on the following screen at Box.

Please make a note of the "client\_id" and "client\_secret" assigned by Box for the Demonstrab.ly application.

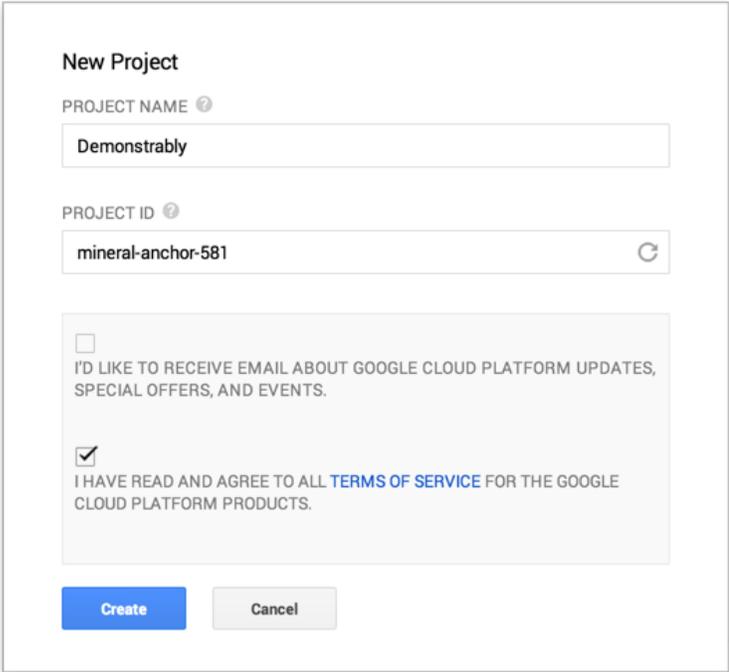
## Create a new Google Drive platform application

In this section, we will use Google Drive as the example to set up your application with the endpoint. Further details on this process are documented at <https://developers.google.com/drive/web/enable-sdk>.

Via a web browser, go to <https://console.developers.google.com>.

Click on "Create Project". Enter a title and project ID (or use the auto-generated ID), and click "Create".

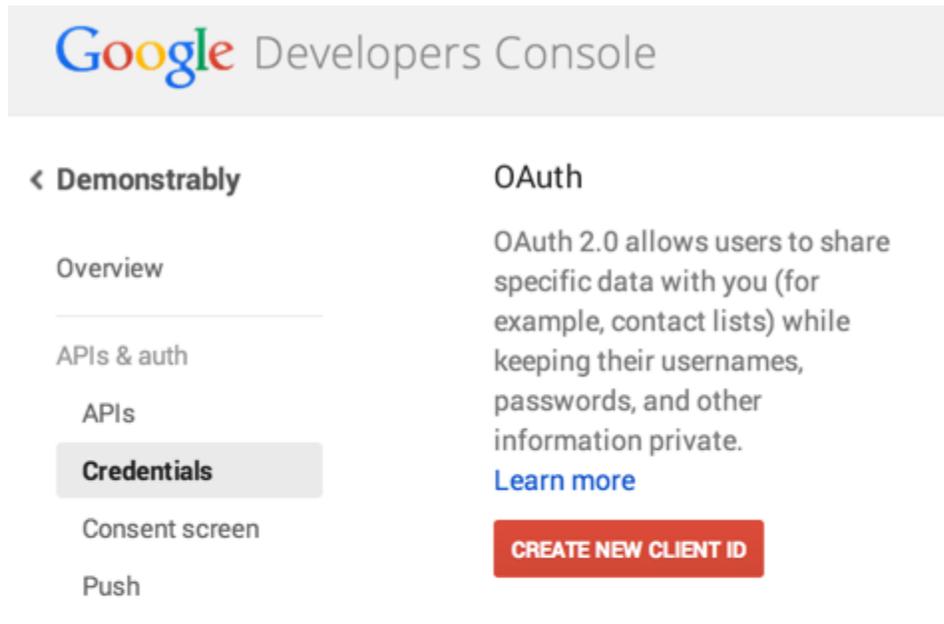
For this example, let's assume that the URL for your application is <https://www.demonstrab.ly>.



The screenshot shows the Google Developers Console interface. On the left, there is a sidebar with navigation options: "Projects" (highlighted), "Billing", and "Account settings". The main content area displays a "New Project" dialog box. The dialog has the following fields and options:

- PROJECT NAME**: A text input field containing "Demonstrably".
- PROJECT ID**: A text input field containing "mineral-anchor-581" with a refresh icon on the right.
- I'D LIKE TO RECEIVE EMAIL ABOUT GOOGLE CLOUD PLATFORM UPDATES, SPECIAL OFFERS, AND EVENTS.
- I HAVE READ AND AGREE TO ALL [TERMS OF SERVICE](#) FOR THE GOOGLE CLOUD PLATFORM PRODUCTS.
- At the bottom, there are two buttons: "Create" (blue) and "Cancel" (grey).

After receiving confirmation that your application is created, click "APIs & auth" and then "Credentials". In the OAuth Section, click "Create New Client ID".



On the next screen, you'll be required to enter a callback URL from the endpoint. This URL will be in your application's address space. You will be required to retrieve some information returned on this URL by the endpoint. The details of handling the callback are elsewhere in this document.

For our example, we'll use a callback URL of <https://www.demonstrably.ly/authz>. Enter this URL on the following screen on the Google Developers Console. Enter the required information and click "Create Client ID".

## Create Client ID

### APPLICATION TYPE

- Web application**  
Accessed by web browsers over a network.
- Service account**  
Calls Google APIs on behalf of your application instead of an end-user. [Learn more](#)
- Installed application**  
Runs on a desktop computer or handheld device (like Android or iPhone).

### AUTHORIZED JAVASCRIPT ORIGINS

Cannot contain a wildcard (http://\*.example.com) or a path (http://example.com/subdir).

### AUTHORIZED REDIRECT URI

Needs to have a protocol, no URL fragment, and no relative paths

After a few moments, the Client ID information for your application will appear on the console, as shown below.

**Client ID for web application**

Client ID	696471275446-27opde4jh9ubo6h1b9n84k46tkbav0jv.apps.googleusercontent.com
Email address	696471275446-27opde4jh9ubo6h1b9n84k46tkbav0jv@developer.gserviceaccount.com
Client secret	5qUB7Kj78b9On-Iuqjht659E
Redirect URIs	https://demonstrab.ly/authz
Javascript Origins	https://demonstrab.ly

[Edit settings](#)
[Download JSON](#)
[Delete](#)

Please make a note of the "Client ID" and "Client secret" assigned by Google for the Demonstrab.ly application.

## Create a new OneDrive platform application

In this section, we will use OneDrive as the example to set up your application with the endpoint.

Via a web browser, go to <https://account.live.com/developers/applications/create>.

Enter a name for your application and click "I accept".

Microsoft account Developer Center

Brian Rothhaar | Sign out

[Home](#) [My apps](#) [Docs](#) [Downloads](#) [Support](#)

My applications

## Enable your application to use Microsoft accounts

This site will allow your web-based Android and iOS applications to authenticate users via Microsoft accounts.

**If you want to register an application for Windows 8.1 or Windows Phone 8.1, go to the [Windows Store Dashboard](#) instead.**

Provide the name of your application that users will see.

Application name\*

Use letters, digits, and underscores only. 129-character limit.

Language\*

Select your application's primary language.

Clicking **I accept** means that you agree to the Microsoft services [terms of use](#). Read [Privacy & Cookies](#).



For this example, let's assume that the URL for your application is <https://www.demonstrab.ly>.

On the next screen, click "API Settings" and enter a callback URL from the endpoint. This URL will be in your application's address space. You will be required to retrieve some information returned on this URL by the endpoint. The details of handling the callback are elsewhere in this document.

## Demonstrab.ly

**Settings**

- Basic Information
- API Settings**
- App Settings
- Localization

Mobile or desktop client app:  
 Yes  No

Restrict JWT issuing:  
 Yes  No

**We recommend enabling enhanced redirection security for your application.**

Enhanced redirection security:  
 Enabled  Disabled (Deprecated)

Root domain:

Redirect URLs:

Mobile client applications use a different OAuth 2.0 authentication flow. Selecting Yes will allow mobile clients to use this authentication flow. [Learn More](#)

Limits the issuing of JSON Web Tokens (JWT) for your domain to exclusively this application.

To improve security, your application should specify the exact redirect URLs to be used; all new applications are now required to do this. With enhanced redirection security enabled, you will need to specify both a root domain and specific redirection URLs under that domain.

All redirect URLs must fall within your root domain. This is also the domain that other apps will use when they request a JWT for your app on Windows (such as www.contoso.com).

Specify the exact URL you will be redirecting users to (such as http://www.contoso.com/redirect). You must provide at least one redirect URL.

For our example, we'll use a callback URL of <https://www.demonstrab.ly/authz>. Enter this URL and click "Save".

Click on "App Settings" to view OAuth keys.

# Demonstrab.ly

## Settings

Basic Information

API Settings

**App Settings**

Localization

To protect your app's security, [Windows Push Notification Services \(WNS\)](#) and [services using Microsoft account](#) use client secrets to authenticate the communications from your server.

Client ID:  
00000004011FC33

This is a unique identifier for your application.

Client secret (v1):  
um-EV7rBUKsMbLeTJ5LoGioIa3L09-J4

For security purposes, don't share your client secret with anyone.

If your client secret has been compromised or your organization requires that you periodically change client secrets, create a new client secret here. After you create a new client secret, both the old and the new client secrets will be accepted until you activate the new secret.

[Create a new client secret](#)

**Note:** Please wait 24 hours before you activate your new client secret, because the old client secret won't work after you activate the new one.

Please make a note of the "Client ID" and "Client secret" assigned by Microsoft for the Demonstrab.ly application.

## Create a new SharePoint platform application

In this section, we will use Microsoft SharePoint as the example to set up your application with the endpoint.

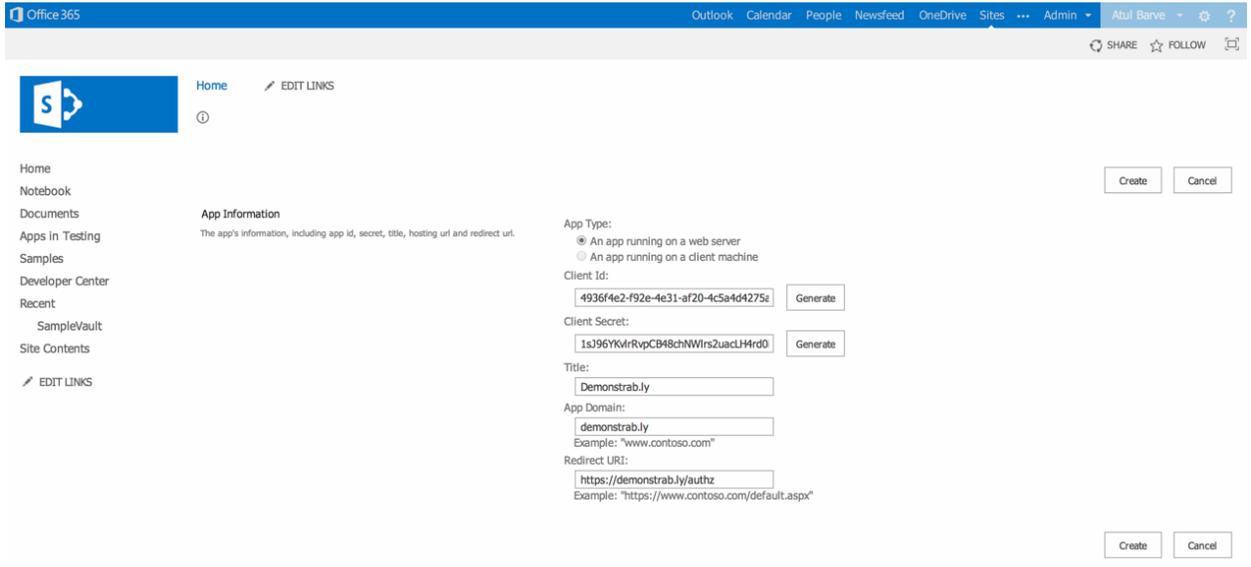
Via a web browser, go to [http://yourSharePointServerName/\\_layouts/15/appregnew.aspx](http://yourSharePointServerName/_layouts/15/appregnew.aspx).

Select the "An app running on a web server" option, select the appropriate options and enter a name for your application.

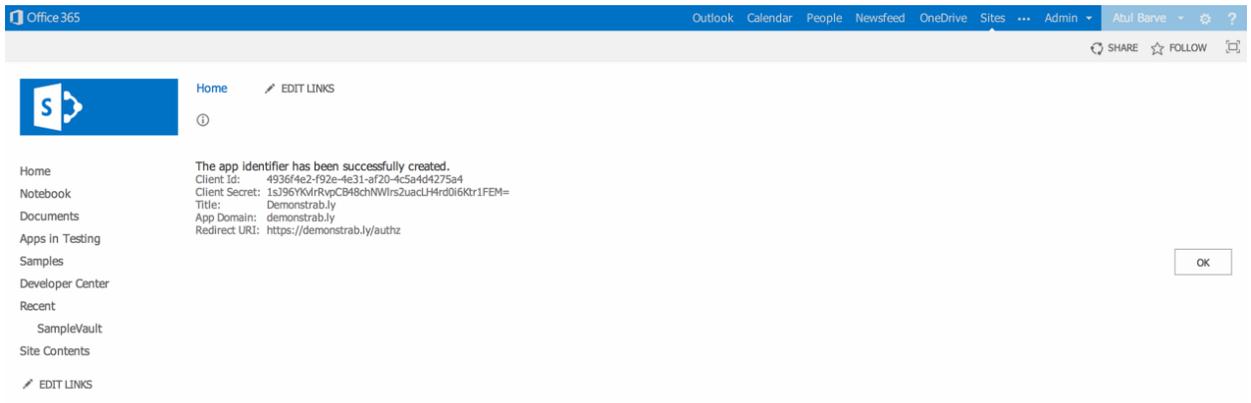
For this example, let's assume that the URL for your application is <https://www.demonstrab.ly>. You'll also be required to enter a callback URL from the endpoint. This URL will be in your application's address space. You will be required to retrieve some information returned on this URL by the endpoint. The details of handling the callback are elsewhere in this document.

For our example, we'll use a callback URL of <https://www.demonstrab.ly/authz>. Enter this URL on the App information screen in SharePoint.

Click the "Generate" buttons for both the "Client Id" and "Client Secret" fields. Finally, click the "Create" button to register the app with your SharePoint instance.



On the next page, you should receive confirmation that your app was created successfully, as shown below.



Please make a note of the "Client Id" and "Client Secret" assigned by SharePoint for the Demonstrab.ly application.