Hilenium Api Guide Documentation

Release 1

Matthew Clarkson

Oct 26, 2017

Contents

Hilenium is a performance website hosting company. Think of us a race tuners for your website.

This document describes the API for the now defunct workflow application. It is no longer supported.

CHAPTER 1

Table of Contents

API Overview

The Hilenium public API follows REST conventions and uses HTTP verbs and response codes to ensure you can securely access resources without requiring complex libraries.

Cross-origin resource sharing is supported. All responses are in JSON format.

Note: While we do keep track of requests made with each API key, we do not *currently* impose any rate limits or restrictions on the number of requests that can be made.

Authentication

Authentication is conducted using **HTTP Basic** over **HTTPS**. You must provide a valid API key as the username for each request. A password is not required. API keys are available via your account page.

Tip: API keys are **private** and should never be published in client side code.

Example cURL request

```
curl https://api.hilenium.com/v1/users.json -u yoursecureapikey:
```

Example HTTP request

```
GET /v1/users.json HTTP/1.1
Host: https://api.hilenium.com
Authorization: Basic yoursecureapikey
Accept: application/json
```

Note: An API key is associated with an individual user and only provides access to resources and actions the user is permitted to perform. In most cases the API key for your integration should be associated with a user who has administrator access to your account.

Data Formats

All responses are provided in JSON format and the type is self-evident.

Strings

Unless otherwise noted, `name` and `title` is restricted to 3-50 characters, `description` must be between 3-1,000 characters and other strings cannot be longer than 10,000 characters.

Dates

Dates are in ISO 8601 format, for example `2014-02-12T15:19:21+00:00`.

Form Errors

When using `POST` or `PATCH` to submit data, errors will be returned mapped to the specific fields, for example:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
  "errors": {
    "email": [
      "The email is already used."
    ]
    "profile": {
      "first_name": [
        "First name cannot be longer than 50 characters."
      ]
    }
  }
}
```

HTTP Methods

The following HTTP methods are available via the API:

- `GET` Request a resource or array of resources.
- `PATCH` Modify an existing resource (`PUT` is not used).
- `POST` Create a new resource
- `DELETE` Delete a resource.

Base Url

The base url for all requests is `https://api.hilenium.com/v1`

Note: Not all methods are permitted or are available for all resources and users.

Response Codes

Standard HTTP response codes are used. Anything in the `2XX` range indicates a successful response and the `4XX` range indicates there is a problem with your request.

The most common response codes are:

Success Codes

- `200` Successful response.
- `201` Resource created successfully.
- `204` Resource updated successfully, no content is included in the response.

Failure Codes

- `401` The request requires authentication.
- `403` The user is not permitted to perform the action.
- `404` Resource could not be found.

Resources

The following Hilenium objects are available via the API.

Users

Users are individuals who have access to Hilenium. You can only access users in your organization.

Property	Type	Description
id	int	Unique identifier
email	string	Email address
group_name	string	The group the user is in (determines system permissions).
active	bool	Identifies if the user is enabled (must explicitly set to true for the user to have access)
created	date-time	Date created
modified	date-time	Date last modified
profile.first_name	string	First name
profile.last_name	string	Last name
profile.initials	string	2-3 characters that identifies the user in comments
profile.job_title	string	Job title

List Users

GET /users.json

Returns all users in the organization.

Query Parameters

- **offset** – Default is 0
- **limit** – Default is 30

Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json

  [
    {
      "id": 1234,
      "email": "user1@yourorganisation.com",
      "group_name": "Administrator",
      "active": true,
      "profile": {
        "first_name": "User",
        "last_name": "One",
        "initials": "U1",
        "job_title": "Marketing Manager",
      }
    },
    {
      "id": 1235,
      "email": "user2@yourorganisation.com",
      "group_name": "Editor",
      "active": true,
      "profile": {
        "first_name": "User",
        "last_name": "Three",
        "initials": "U1",
        "job_title": "Marketing Assistant",
      }
    }
  ]
```

Create a User

POST /users.json

Creates a new user in the organisation.

Request JSON Object

- **email** (*string*) – Unique email address
- **group_name** (*string*) – Name of the group the user is in
- **active** (*boolean*) – Active (default is true)
- **profile.first_name** (*string*) – First name
- **profile.last_name** (*string*) – Last name
- **profile.initials** (*string*) – Initials/short identifier (3 char max)

- **profile.job_title** (*string*) – Job title

Example Response

```
HTTP/1.1 201 OK
Content-Type: application/json

    {
      "id": 1236,
      "email": "user2@yourorganisation.com",
      "group_name": "Editor",
      "active": true,
      "profile": {
        "first_name": "New",
        "last_name": "User",
        "initials": "NU",
        "job_title": "New User",
      }
    }
```

Retrieve a User

GET /users/[id].json

Returns a single user by their id.

Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json

    {
      "id": 1234,
      "email": "user1@yourorganisation.com",
      "group_name": "Administrator",
      "active": true,
      "profile": {
        "first_name": "User",
        "last_name": "One",
        "initials": "U1",
        "job_title": "Marketing Manager",
      }
    }
```

Update a User

PATCH /users/[id].json

Updates a user. You only need include the properties you wish to update in the JSON object.

Example Response

```
HTTP/1.1 204 OK
```

Delete a User

You cannot currently delete users in Hilenium. Instead set their ‘active’ status to ‘false’.

Groups

Note: This is a read-only resource. You cannot create or edit groups.

Groups define the permissions user's have in your system.

Property	Type	Description
id	int	Unique identifier
name	string	Name of the group
description	string	Short description of the permissions available to the user

List Groups

GET /groups.json

Returns all groups.

Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json

  [
    {
      "id": 1,
      "name": "Administrator",
      "description": "Full system access"
    }
  ]
```

Retrieve a Group

GET /groups/[id].json

Returns a single group by its id.

Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json

    {
      "id": 1,
      "name": "Administrator",
      "description": "Full system access"
    }
```

Organizations

Note: You can only access your own organization. Organizations cannot be deleted or created.

Property	Type	Description
name	string	Name
address.street	string	Street Line 1
address.street2	string	Street Line 2
address.city	string	City
address.region_name	string	State, Province or Region
address.postal_code	string	Zip / Post Code
address.country	string	Two letter ISO country code
website	string	Valid URL
phone	string	Primary telephone number
email	string	Primary email address
email_billing	string	Billing email address. Defaults to primary email.
created	datetime	Date created
modified	datetime	Date last modified

Retrieve your Organization

`GET /organizations.json`

Returns the user's organization.

Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json

    {
        "name": "My Organization Name",
        "address": {
          "street": "Street 1",
          "street2": "Street 2",
          "city": "City",
          "region_name": "Region name",
          "postal_code": "Zip code",
          "country": "US"
        },
        "website": "http://www.myorganizationname.com",
        "phone": "555-5555",
        "email": "admin@myorganizationname.com",
        "email_billing": "billing@myorganizationname.com"
    }
```

Update your Organization

`PATCH /organizations.json`

Updates the user's organization. Only include the properties you wish to update in the JSON object.

Example Response

```
HTTP/1.1 204 OK
```

Note: You must be a company administrator to update your organization.

Projects

Property	Type	Description
id	int	Unique Identifier
created	date-time	Date created
modified	date-time	Date last modified
title	string	Title (name)
description	string	Project Description
created_by	string	Name of the user who created the project (fallback is email address)
job_code	string	Your internal code for the project
ownership	string	Internal (created by your company) or External (invited by another company)
num_comments	int	Number of associated comments
_thumbnails	href	Links to project thumbnails in various sizes (the last image added or a default placeholder)

List Projects

GET /projects.json

Returns a summary of projects the user has been added to, including those they have been invited to from other organisations.

Query Parameters

- **offset** – Default is 0
- **limit** – Default is 30

Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json

  [
    {
      "id": 1234,
      "created": "2014-11-04T00:29:01+00:00",
      "modified": null,
      "title": "Example Project",
      "description": "This is an example marketing project",
      "created_by": "admin@yourorganisation.com",
      "job_code": "Your Unique Project Code",
      "ownership": "internal",
      "_thumbnails": {
        "small": {
          "href": "http://thecdn.com/small/thumbnail.png"
        },
        "medium": {
          "href": "http://thecdn.com/medium/thumbnail.png"
        },
        "large": {
          "href": "http://thecdn.com/large/thumbnail.png"
        }
      }
     }
  ]
```

Create a Project

POST /projects.json

Creates a new project for the user.

Request JSON Object

- **name** (*string*) – Project name (50 char)
- **description** (*string*) – The project description (1,000 char)
- **job_code** (*string*) – Internal project identifier (20 char)

Example Response

```
HTTP/1.1 201 OK
Content-Type: application/json

   {
      "id": 1234,
      "created": "2014-11-04T00:29:01+00:00",
      "modified": null,
      "title": "Example Project",
      "description": "This is an example marketing project",
      "created_by": "admin@yourorganisation.com",
      "job_code": "Your Unique Project Code",
      "ownership": "internal",
      "_thumbnails": {
        "small": {
          "href": "http://thecdn.com/small/thumbnail.png"
        },
        "medium": {
          "href": "http://thecdn.com/medium/thumbnail.png"
        },
        "large": {
          "href": "http://thecdn.com/large/thumbnail.png"
        }
      }
    }
```

Note: When a project is created the user is automatically added as a project member.

Retrieve a Project

GET /projects/[id].json

Returns a project summary by id.

Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json

    {
      "id": 1234,
      "created": "2014-11-04T00:29:01+00:00",
      "modified": null,
      "title": "Example Project",
      "description": "This is an example marketing project",
      "created_by": "admin@yourorganisation.com",
      "job_code": "Your Unique Project Code",
      "ownership": "internal",
      "_thumbnails": {
        "small": {
          "href": "http://thecdn.com/small/thumbnail.png"
        },
        "medium": {
          "href": "http://thecdn.com/medium/thumbnail.png"
        },
        "large": {
          "href": "http://thecdn.com/large/thumbnail.png"
        }
      }
    }
```

Update a Project

PATCH /projects/[id].json

Updates a project. Only include the properties you wish to update in the JSON object.

Example Response

```
HTTP/1.1 204 OK
```

Delete a Project

DELETE /projects/[id].json

Example Response

```
HTTP/1.1 204 OK
```

Note: You must be either a company administrator or project creator to delete a project.

Warning: Deleting a project is permanent. All project files, comments and assets are deleted. This cannot be reversed.

Project Users

Project users are people who have access to a project. They can be internal or external to your Hilenium account.

List Project Users

GET /projects/[id]/users.json

Returns an array of users who have access to a project.

Query Parameters

- **offset** – Default is 0
- **limit** – Default is 30

Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json

  [
    {
      "id": 1234,
      "email": "user1@yourorganisation.com",
      "group_name": "Administrator",
      "active": true,
      "profile": {
        "first_name": "User",
        "last_name": "One",
        "initials": "U1",
        "job_title": "Marketing Manager",
        "bio": "A passionate marketing manager who loves integrated campaigns",
        "work_phone": "555-1234",
        "mobile_phone": null
      }
    },
    {
      "id": 1235,
      "email": "user2@yourorganisation.com",
      "group_name": "Editor",
      "active": true,
      "profile": {
        "first_name": "User",
        "last_name": "Three",
        "initials": "U1",
        "job_title": "Marketing Assistant",
        "bio": "Marketing assistant who loves football",
        "work_phone": "555-1235",
        "mobile_phone": null
      }
    }
  ]
```

Add A User to Project

POST /projects/[id]/users.json

Adds a user to a project by their email address or existing user id. Invited users receive a system generated email.

Request JSON Object

- **user** (*string*) – A user id (preferred) OR
- **email** (*string*) – An email address

Example Response

```
HTTP/1.1 201 OK
Content-Type: application/json
```

Warning: When users are invited by email, the email address is checked to see if the account already exists. If an existing user is found, they are added to the project. However, if the user does not have a pre-existing account, a separate account with full administrator access is created for them. For this reason, it is advised that you **only invite people from other companies to collaborate on your projects by email**. Users from your organisation should be invited by their Hilenium user id.

Remove a User from a Project

DELETE /projects/[id]/users/[id].json

Removes a user from a project.

Example Response

```
HTTP/1.1 204 OK
Content-Type: application/json
```

Note: The project creator cannot be removed from a project.

Assets

Note: The API only provides read only access to project assets. They cannot be created via the API.

Three types of assets are associated with projects:

- Files - Any type of file that has been uploaded to a projects, such as images or documents.
- Captured Urls - These are image 'snapshots' of urls or bookmarks that have added to a project.
- Copy - HTML Text that has been added to the project for collaboration, including marked up revisions.

Object Properties

All assets share the following properties.

Property	Type	Description
id	int	Unique identifier
name	string	Name
description	string	Description of the object
num_comments	int	Number of associated comments
created_by	string	Name of the creator (falls back to email address)
created	datetime	Date created
modified	datetime	Date last modified

Files have the following additional properties.

Property	Type	Description
type	string	The 'guessed' MIME type
_links	href	Links to the uploaded file
_thumbnail	href	Links to file thumbnails

Urls have the following additional properties.

Property	Type	Description
url	string	The url that was recorded
_links	href	Links to the uploaded file
_thumbnail	href	Links to url thumbnails

Copy objects have the following additional properties.

Property	Type	Description
text	string	Current text

Retrieve Files

GET v1/projects/[id]/files.json

Returns files uploaded to the projects.

Query Parameters

- **offset** – Default is 0
- **limit** – Default is 30

Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json

    [
      {
        "id": 1234,
        "name": "File name",
        "created": "2014-11-04T00:29:01+00:00",
        "modified": null,
        "description": "File description",
        "type": "image/png",
        "num_comments": 12,
        "created_by": "admin@yourorganisation.com",
         "_thumbnails": {
            "small": {
              "href": "https://thecdn.com/small/54581ddd80d84.png"
            },
            "medium": {
              "href": "https://thecdn.com/medium/54581ddd80d84.png"
            },
            "large": {
              "href": "https://thecdn.com/large/54581ddd80d84.png"
            }
         }
      }
    ]
```

Retrieve Captured Urls

GET /projects/[id]/urls.json

Returns urls added to a project

Query Parameters

- **offset** – Default is 0
- **limit** – Default is 30

Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json

  [
    {
      "id": 1234,
      "name": "Url name",
      "created": "2014-11-04T00:29:01+00:00",
      "modified": null,
      "description": "Url description",
      "url": "http://www.theurl.com",
      "type": "image/png",
      "created_by": "admin@yourorganisation.com",
       "_thumbnails": {
          "small": {
            "href": "https://thecdn.com/small/54581ddd80d84.png"
          },
          "medium": {
            "href": "https://thecdn.com/medium/54581ddd80d84.png"
          },
          "large": {
            "href": "https://thecdn.com/large/54581ddd80d84.png"
          }
       }
     }
  ]
```

Retrieve Copy

GET /projects/[id]/copy.json

Returns copy added to a project

Query Parameters

- **offset** – Default is 0
- **limit** – Default is 30

Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json

  [
    {
      "id": 1234,
      "name": "Copy name",
      "created": "2014-11-04T00:29:01+00:00",
      "modified": null,
      "description": "Url description",
      "text": "This is the latest version of the copy",
      "created_by": "admin@yourorganisation.com"
     }
  ]
```

Retrieve Copy Revisions

GET /copy/[id]/revisions.json

Returns the revision history for copy, including the marked up changes

Query Parameters

- **offset** – Default is 0
- **limit** – Default is 30

Example Response

```
HTTP/1.1 200 OK
Content-Type: application/json

  [
      {
        "id": 3,
        "created": "2014-11-17T05:28:36+00:00",
        "modified": null,
        "modified_by": "theuser@yourdomain.com",
        "text": "This is the current copy",
        "text_diff": "Marked up difference compared to previous version"
      }
  ]
```

HTTP Routing Table