Ngenea Hub

Ngenea Hub harnesses to power of Ngenea to provide global workflows, enabling your data to to be where you need it, when you need it.

Installation

A deployment of Ngenea Hub comprises two main components:

- **Ngenea Hub** - the central management point, from which all tasks are managed
- **Ngenea Workers** - worker agents, installed on individual Ngenea servers, that execute the Ngenea tasks

Ngenea Workers are installed on one or more nodes in a *Site*, which typically represents a single Ngenea cluster or location.

The *Ngenea Hub* needs to be accesible from all *Ngenea Workers* on the following ports:

- `6379/tcp`
- `5672/tcp`

Installing Ngenea Hub

Ngenea Hub can be installed in a number of ways, use one of the methods described below.

CentOS / Redhat - Online Installation

Configure Docker Authentication

> **Note:** This step is not required on PixStor systems

Configure Docker authentication for `eurepo.arcapix.com`.

```
docker login eurepo.arcapix.com
```

Installing Ngenea Hub

Transfer the `ngenea-hub` rpm to the target system.

Install the `ngenea-hub` package via yum.

```
yum install ngenea-hub-<version>.rpm
```

Create an initial Ngenea Hub configuration file at `/etc/sysconfig/ngeneahub`. This file contains the credentials which will be required for deploying workers.

```
ngeneahubctl createconfig
```

Enable and start the ngeneahub service.

```
systemctl enable --now ngeneahub
```

Check the status of the service with

```
ngeneahubctl status
```

CentOS / Redhat - Offline installation

The `ngeneahub` service will attempt to pull the required docker images from the Ngenea software repository servers. In situations where this is not possible (due to network restrictions, for instance), the containers can be installed via additional RPM: `ngenea-hub-images`, available at the same location as the main RPM.

Once the RPMs are transferred to the target system, they can be installed using `rpm`.

```
rpm -ivh ngenea-hub-<version>.rpm ngenea-hub-images-<version>.rpm
```

Cloud Deployment

Coming soon: image-based deployment of Ngenea Hub in the cloud.

Container Native Deployment

It is possible to deploy Ngenea Hub using standard container management tools and processes.

Please contact us to discuss.

Ngenea Worker

Pre-requisite: Ngenea HSM Server software installed and configured

Configure the ngenea-worker yum repository.

```
# cat /etc/yum.repos.d/ngenea-worker.repo
[ngenea-worker]
skip_if_unavailable=1
gpgcheck=0
sslverify=1
baseurl=https://$apuser@eurepo.arcapix.com/pulp/repos/ngenea-worker/
name=Ngenea Worker repository
enabled=1
```

Install the `ngenea-worker` package via yum.

```
yum install ngenea-worker
```

Add the Ngenea Hub access details to `/etc/ngenea/ngenea-worker.conf`. The initial access credentials can be found in the `/etc/sysconfig/ngeneahub` configuration file on the Ngenea Hub server. By default, the `BROKER_PASSWORD` is used for both the `broker_url` and the `result_backend` passwords, and the `BROKER_USER` is used for the `broker_url` username.

Enable and start the `ngenea-worker@SITENAME.target` service. Replace `SITENAME` with the reference which will be used for this site.

```
systemctl enable --now ngenea-worker@SITENAME.target
```

Configuration

Initial Configuration

Once all services are up, create an admin user with:

```
ngeneahubctl adduser
```

Register an initial *site* (replacing `SITENAME`) (see Worker Installation for enabling a *site*'s worker agents).

```
ngeneahubctl addsite SITENAME
```

Log into the UI at http://server.address:8000.

Configuration

Settings

The main configuration file for Ngenea Hub is at `/etc/sysconfig/ngeneahub`. This is an environment file which holds the information required for connecting to the various backend services.

Mandatory Settings

Setting	Description
POSTGRES_DB	Internal database name
POSTGRES_USER	Internal database username
POSTGRES_PASSWORD	Internal database password
BROKER_USER	Queue username
BROKER_PASSWORD	Queue password

Optional settings

Setting	Description
RABBITMQ_USER	User to use when connecting to the rabbitmq broker. Overrides `BROKER_USER`.
RABBITMQ_PASSWORD	Password to use when connecting to the rabbitmq broker. Overrides `BROKER_PASSWORD`.
RABBITMQ_HOST	Address of the RabbitMQ broker service. Defaults to the container service address.
REDIS_PASSWORD	User to connect to the rabbitmq broker. Overrides `BROKER_PASSWORD`.
REDIS_HOST	Address of the Redis queue results store. Defaults to the container service address.
RABBITMQ_DEFAULT_USER	Username to use when initializing the RabbitMQ broker. Not used after the service has been initialised.
RABBITMQ_DEFAULT_PASS	Password to use when initializing the RabbitMQ broker. Not used after the service has been initialised. Overrides `BROKER_PASSWORD`.

Docker Compose configuration

The `docker-compose` file is stored in `/usr/share/ngeneahub/docker/docker-compose.yml`.

This can be extended by creating an override file at `/usr/share/ngeneahub/docker/docker-compose.override.yml`.

Usage

API

Reference

The API reference can be found at your Ngenea Hub install at `/api/docs/`. This section does not attempt to duplicate the reference, but instead provide some usage examples.

Authentication

Authentication to the API can be performed in one of three ways

- Session Authentication
- JWT Authentication
- Client Keys

The first two, are for interacting with the API interactivly and are therefore most likely not suitable for building automated workflows. On the other hand, client keys are valid until they are revoked and are more suitable for automation

Client Keys

Creating Client Keys

> **Note:** The UI does not currently support creating client keys and therefore have to be done via the API directly

Before we can authenticate using the client key, we need to temporarly authenticate using JWT to be able to create a client key.

To get a valid JWT access token using curl and `jq`:

```
JWT_TOKEN=$(curl -s -X POST 'http://example.com/api/auth/token/'  -H
'Accept: application/json'  -H 'Content-Type: application/json' -d
'{"username": "dfoster",  "password": "******"}' | jq -r .access)
```

This can now be used to create a client key:

```
 curl -s -X POST 'http://example.com/api/auth/clientkeys/'  -H 'Accep
t: application/json'  -H 'Content-Type: application/json'  -H "Author
ization: Bearer $TOKEN" -d '{"name": "my_automation_key"}'


{"url":"http://localhost:8000/api/auth/clientkeys/1/","id":1,"name":"
my_automation_key","api_key":"e1OVrtr3.smFadImw4chb615qN5RLCh0HHqIGyb
2bq"}
```

> **Warning:** This is the only time the client key will be visable, make sure it is recorded.

Using Client Keys

The key created in the previous section can now be used by setting the header `Auth orization: Api-Key YOUR_API_KEY` against an API endpoint. For example:

```
curl -s -X GET 'http://example.com:8000/api/jobs/'  -H 'Accept:
application/json'  -H 'Content-Type: application/json'  -H "Authoriza
tion: Api-Key pmS09XLj.3r1G557hnF0ixCabZktMy9SZwTPt24Ur" | jq
{
  "count": 0,
  "next": null,
  "previous": null,
  "results": [],
  "stats": {
    "type": {
      "migrate": 0,
      "premigrate": 0,
      "recall": 0
    },
    "state": {
      "SUCCESS": 0,
      "FAILURE": 0,
      "STARTED": 0,
      "PENDING": 0,
      "ERROR": 0
    },
    "created": {},
    "site": {}
  }
}
```

Submitting Workflow

Migrate

Using a Client Key stored in a environment variable `TOKEN`, the following is an example of migrating a file using curl.

```
curl -s -X POST 'http://example.com/api/file/workflow/'  -H 'Accept:
application/json'  -H 'Content-Type: application/json'  -H "Authoriza
tion: Api-Key $TOKEN"  -d '{"paths": ["/mmfs1/data/
sample_data.tgz"], "site": "dfoster1", "workflow": "migrate"}'
```

Monitoring and Management

systemd service

Ngenea Hub is controlled via the `ngeneahub` systemd service

ngeneahub cli tool

The `ngeneahub` tool can be used to manually stop/start the services, outside of systemd, for debugging

Docker containers

Ngenea Hub uses a collection of docker containers, which can be managed by standard Docker monitoring/management tools and processes:

Container Name	Description
`ngeneahub_app_1`	Web application
`ngeneahub_jobrefresh_1`	Maintenance task controller
`ngeneahub_db_1`	Application database (Postgres)
`ngeneahub_rabbitmq_1`	Task queue broker
`ngeneahub_redis_1`	Task results backend

Custom Workflows

Defining workflows

It's possible to define custom workflows which use pre-defined steps as building blocks to create your workflow.

Note: Custom workflows are not currently exposed via the UI. Use the API `/api/workflows/` endpoint to create custom workflows

A workflow definition requirements the following parameters:

name	description
`name`	The unique name for this workflow. For easy of submission again the API, this should not contain spaces.

name	description
`label`	The human readable name for this workflow, can contain spaces.
`icon`	A HTML representation of an icon to display in the UI. Font Awesome is useful here.
`steps`	A list of steps to perform. Described in more detail below.
`fields`	A list of runtime fields. Described in more detail below.

Steps

Steps are defined in JSON. Steps is a list of individual steps that will be perform serially.

Each step is a defined as a dict. The following fields are in defined

- `name` *[required]*: The name of the task to run, e.g. `dynamo.tasks.migrate`
- `site` *[optional]*: The name of the site to run against. Default: The site associated with the initial filelist provided to the workflow.

Any other fields are passed as `kwargs` to the task.

```
Example 1 - Send to a site
[
    {
        "name": "dynamo.tasks.migrate"
    },
    {
        "name": "dynamo.tasks.recall",
        "site": "london"
    }
]
```

Runtime fields

A workflow needs to be able to accept parameters as it submitted. Taking example #1 above, "london" doesn't want to be hardcoded as the destination site, as that would mean a new workflow would need to be defined for each possible destination.

Instead, fields can be defined, that in turn will need to be provided at workflow submission time. Fields are defined as a dict with the following keys:

- `name` *[required]*: The name of the field.
- `label` *[required]*: The fiendly name for this field, used for presenting in the UI
- `type` *[required]*: The type of the field. Valid options are:
    - `string` - a free text field
    - `int` - a free text field that will be validated a integer

- bool - a checkbox
- enum[enum_type] - A dropdown box representing a choice of option, populated from enum_type. enum_type can be one of the following
  - site - A list of all the sites Ngenea Hub has defined

Example 3 - Custom fields

```json
[
    {
        "name": "target_site",
        "label": "Site to migrate to",
        "type": "enum[site]"
    },
    {
        "name": "delete",
        "label": "Delete from source",
        "type": "bool"
    }
]
```

Back in the definition of a step, any value that is prefixed with a & will be used as a field name and the value replaced instead of a literal string.

The following example, modifies example #1 to use the custom field as defined in example #3:

Example 4 - Using custom fields

```json
[
    {
        "name": "dynamo.tasks.migrate"
    },
    {
        "name": "dynamo.tasks.recall",
        "site": "&target_site"
    }
]
```

Running Workflows

Note: Workflows that define runtime fields cannot be submitted via the UI currently.

A workflow is run via the `/api/file/workflow/` endpoint and expects the following parameters.

name	description
paths	A list of paths to perform the workflows against
site	The site to perform the workflow against

9

Reference

Workflow Steps

This section documents all the currently supported steps in ngenea hub. See Custom Workflows for guidance on how to use these steps in your own workflows.

dynamo.tasks.migrate

Migrates a list of files to a pre-defined remote target using Ngenea.

Argument	Type	Default	Description
premigrate	bool	False	retain the content of every migrated file and do not set the OFFLINE flag for the file.migrating.
stub_size	int	0	retain a segment of every migrated file starting from its beginning and having a specified approximate length in bytes.
overwrite	bool	False	overwrite remote objects if they already exist–do not create remote object instances with various UUID suffixes

dynamo.tasks.recall

Recalls a list of files to a pre-defined remote target using Ngenea.

Argument	Type	Default	Description
skip_hash	bool	False	If the recall should skip checking the hash of the file

dynamo.tasks.reverse_stub

Recalls a list of files to a pre-defined remote target using Ngenea.

Argument	Type	Default	Description
hydrate	bool	False	If the file should be premigrated instead of a regular stub
stub_size	int	0	The max file size before files will be stubbed for this task
skip_hash	bool	False	If the recall should skip checking the hash of the file
overwrite	bool	False	overwrite local files if they already exist.

dynamo.tasks.delete_paths_from_gpfs

Removes a list of files from a GPFS filesystem.

This step takes no additional arguments.

API

GET /auth/clientkeys/

    API endpoint for managing client keys

> **Query Parameters:**
> - **page** (*integer*) – A page number within the paginated result set. When not given, first page is retrieved by default.
> - **page_size** (*integer*) – Number of results to return per page. Page size parameter can be a number between `20` and `100`. For disabling pagination and retrieving all results, `0` should be given. When page size parameter is empty or `<20`, `20` results are returned by default. When page size parameter `>100`, `100` results are returned by default.
>
> **Status Codes:**
> - 200 OK –
>
> **Response JSON Object:**
> - **count** (*integer*) – (required)
> - **next** (*string*) –
> - **previous** (*string*) –
> - **results[].id** (*integer*) – (read only)
> - **results[].name** (*string*) – Name of the client key (required)
> - **results[].url** (*string*) – (read only)

POST /auth/clientkeys/

    API endpoint for managing client keys

> **Request JSON Object:**
> - **api_key** (*string*) – (read only)
> - **id** (*integer*) – (read only)
> - **name** (*string*) – Name of the client key (required)
> - **url** (*string*) – (read only)
>
> **Status Codes:**
> - 201 Created –
>
> **Response JSON Object:**
> - **api_key** (*string*) – (read only)
> - **id** (*integer*) – (read only)
> - **name** (*string*) – Name of the client key (required)
> - **url** (*string*) – (read only)

GET /auth/clientkeys/{id}/

    API endpoint for managing client keys

> **Parameters:**
> - **id** (*string*) –
>
> **Status Codes:**
> - 200 OK –
>
> **Response JSON Object:**
> - **id** (*integer*) – (read only)
> - **name** (*string*) – Name of the client key (required)
> - **url** (*string*) – (read only)

PATCH /auth/clientkeys/{id}/

API endpoint for managing client keys

Parameters:
- **id** (*string*) –

Request JSON Object:
- **id** (*integer*) – (read only)
- **name** (*string*) – Name of the client key (required)
- **url** (*string*) – (read only)

Status Codes:
- 200 OK –

Response JSON Object:
- **id** (*integer*) – (read only)
- **name** (*string*) – Name of the client key (required)
- **url** (*string*) – (read only)

DELETE /auth/clientkeys/{id}/

API endpoint for managing client keys

Parameters:
- **id** (*string*) –

Status Codes:
- 204 No Content –

POST /auth/token/

Takes a set of user credentials and returns an access and refresh JSON web token pair to prove the authentication of those credentials.

Request JSON Object:
- **password** (*string*) – (required)
- **username** (*string*) – (required)

Status Codes:
- 201 Created –

Response JSON Object:
- **password** (*string*) – (required)
- **username** (*string*) – (required)

POST /auth/token/refresh/

Takes a refresh type JSON web token and returns an access type JSON web token if the refresh token is valid.

Request JSON Object:
- **refresh** (*string*) – (required)

Status Codes:
- 201 Created –

Response JSON Object:
- **refresh** (*string*) – (required)

POST /auth/token/verify/

Verifies that the token is not expired AND the token owner exists in the database AND the token owner is an active user.

Request JSON Object:
- **token** (*string*) – (required)
- **type** (*string*) – Token type e.g: *access* or *refresh* (required)

Status Codes:
- 201 Created –

Response JSON Object:
- **token** (*string*) – (required)
- **type** (*string*) – Token type e.g: *access* or *refresh* (required)

GET /file/

Retrieves list of files under given path for given site.

Query Parameters:
- **path** (*string*) – Target directory path
- **site** (*string*) – Site name
- **details** (*boolean*) – Show details of children objects

Status Codes:
- 200 OK –

GET /file/test/

API endpoint for managing files.

Status Codes:
- 200 OK –

POST /file/workflow/

Performs a workflow on a list of files

Request JSON Object:
- **fields** (*object*) –
- **paths[]** (*string*) –
- **site** (*string*) – Site name (required)
- **workflow** (*string*) – Workflow name (required)

Status Codes:
- 201 Created –

Response JSON Object:
- **fields** (*object*) –
- **paths[]** (*string*) –
- **site** (*string*) – Site name (required)
- **workflow** (*string*) – Workflow name (required)

GET /filestatustypes/

API endpoint for managing file status types.

Query Parameters:
- **page** (*integer*) – A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (*integer*) – Number of results to return per page. Page size parameter can be a number between `20` and `100`. For disabling pagination and retrieving all results, `0` should be given. When page size parameter is

empty or `<20`, `20` results are returned by default. When page size parameter `>100`, `100` results are returned by default.

Status Codes:
- 200 OK –

Response JSON Object:
- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[].background_color** (*string*) – (required)
- **results[].key** (*string*) – (required)
- **results[].label** (*string*) – (required)
- **results[].text_color** (*string*) – (required)
- **results[].url** (*string*) – (read only)

GET /filestatustypes/{key}/

API endpoint for managing file status types.

Parameters:
- **key** (*string*) –

Status Codes:
- 200 OK –

Response JSON Object:
- **background_color** (*string*) – (required)
- **key** (*string*) – (required)
- **label** (*string*) – (required)
- **text_color** (*string*) – (required)
- **url** (*string*) – (read only)

PATCH /filestatustypes/{key}/

API endpoint for managing file status types.

Parameters:
- **key** (*string*) –

Request JSON Object:
- **background_color** (*string*) – (required)
- **label** (*string*) – (required)
- **text_color** (*string*) – (required)

Status Codes:
- 200 OK –

Response JSON Object:
- **background_color** (*string*) – (required)
- **label** (*string*) – (required)
- **text_color** (*string*) – (required)

GET /jobs/

API endpoint for managing jobs.

Query Parameters:
- **page** (*integer*) – A page number within the paginated result set. When not given, first page is retrieved by default.

- **page_size** (*integer*) – Number of results to return per page. Page size parameter can be a number between `20` and `100`. For disabling pagination and retrieving all results, `0` should be given. When page size parameter is empty or `<20`, `20` results are returned by default. When page size parameter `>100`, `100` results are returned by default.
- **created** (*string*) – Jobs are filtered for this period. Leave `null` for displaying jobs in all times.
- **jobtype** (*string*) – Job type
- **state** (*string*) – Job state
- **owner** (*string*) – Job owner's username

Status Codes:
- 200 OK –

Response JSON Object:
- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[].created** (*string*) – Time of the job creation
- **results[].dir_walk_complete** (*string*) – (read only)
- **results[].id** (*integer*) – (read only)
- **results[].jobtype** (*string*) – (required)
- **results[].num_dir_tasks** (*string*) – (read only)
- **results[].numfailedfiles** (*integer*) –
- **results[].numfiles** (*integer*) –
- **results[].numprocessedfiles** (*integer*) –
- **results[].numskippedfiles** (*integer*) –
- **results[].owner** (*string*) – (read only)
- **results[].site** (*string*) – Site Name (required)
- **results[].state** (*string*) –
- **results[].url** (*string*) – (read only)

POST /jobs/
API endpoint for managing jobs.

Request JSON Object:
- **created** (*string*) – Time of the job creation
- **dir_walk_complete** (*string*) – (read only)
- **id** (*integer*) – (read only)
- **jobtype** (*string*) – (required)
- **num_dir_tasks** (*string*) – (read only)
- **numfailedfiles** (*integer*) –
- **numfiles** (*integer*) –
- **numprocessedfiles** (*integer*) –
- **numskippedfiles** (*integer*) –
- **owner** (*string*) – (read only)
- **paths** (*string*) – (read only)
- **site** (*string*) – Site Name (required)
- **state** (*string*) –
- **tasks[].job** (*string*) – (read only)
- **tasks[].numfailedfiles** (*integer*) –
- **tasks[].numfiles** (*integer*) –

- **tasks[].numprocessedfiles** (*integer*) –
- **tasks[].numskippedfiles** (*integer*) –
- **tasks[].state** (*string*) – (required)
- **tasks[].taskid** (*string*) – Job task ID (required)
- **tasks[].tasktype** (*string*) – (required)
- **tasks[].url** (*string*) – (read only)
- **url** (*string*) – (read only)

- 201 Created –

Response JSON Object:
- **created** (*string*) – Time of the job creation
- **dir_walk_complete** (*string*) – (read only)
- **id** (*integer*) – (read only)
- **jobtype** (*string*) – (required)
- **num_dir_tasks** (*string*) – (read only)
- **numfailedfiles** (*integer*) –
- **numfiles** (*integer*) –
- **numprocessedfiles** (*integer*) –
- **numskippedfiles** (*integer*) –
- **owner** (*string*) – (read only)
- **paths** (*string*) – (read only)
- **site** (*string*) – Site Name (required)
- **state** (*string*) –
- **tasks[].job** (*string*) – (read only)
- **tasks[].numfailedfiles** (*integer*) –
- **tasks[].numfiles** (*integer*) –
- **tasks[].numprocessedfiles** (*integer*) –
- **tasks[].numskippedfiles** (*integer*) –
- **tasks[].state** (*string*) – (required)
- **tasks[].taskid** (*string*) – Job task ID (required)
- **tasks[].tasktype** (*string*) – (required)
- **tasks[].url** (*string*) – (read only)
- **url** (*string*) – (read only)

GET /jobs/recent/

Retrieves last N jobs as recent jobs. N = 5 by default (defined in dynamohub/settings/base.py).

Query Parameters:
- **page** (*integer*) – A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (*integer*) – Number of results to return per page. Page size parameter can be a number between `20` and `100`. For disabling pagination and retrieving all results, `0` should be given. When page size parameter is empty or `<20`, `20` results are returned by default. When page size parameter `>100`, `100` results are returned by default.

Status Codes:
- 200 OK –

Response JSON Object:
- **count** (*integer*) – (required)

- **next** (*string*) –
- **previous** (*string*) –
- **results[].created** (*string*) – Time of the job creation
- **results[].dir_walk_complete** (*string*) – (read only)
- **results[].id** (*integer*) – (read only)
- **results[].jobtype** (*string*) – (required)
- **results[].num_dir_tasks** (*string*) – (read only)
- **results[].numfailedfiles** (*integer*) –
- **results[].numfiles** (*integer*) –
- **results[].numprocessedfiles** (*integer*) –
- **results[].numskippedfiles** (*integer*) –
- **results[].owner** (*string*) – (read only)
- **results[].paths** (*string*) – (read only)
- **results[].site** (*string*) – Site Name (required)
- **results[].state** (*string*) –
- **results[].tasks[].job** (*string*) – (read only)
- **results[].tasks[].numfailedfiles** (*integer*) –
- **results[].tasks[].numfiles** (*integer*) –
- **results[].tasks[].numprocessedfiles** (*integer*) –
- **results[].tasks[].numskippedfiles** (*integer*) –
- **results[].tasks[].state** (*string*) – (required)
- **results[].tasks[].taskid** (*string*) – Job task ID (required)
- **results[].tasks[].tasktype** (*string*) – (required)
- **results[].tasks[].url** (*string*) – (read only)
- **results[].url** (*string*) – (read only)

GET /jobs/{id}/

API endpoint for managing jobs.

Parameters:
- **id** (*string*) –

Status Codes:
- 200 OK –

Response JSON Object:
- **created** (*string*) – Time of the job creation
- **dir_walk_complete** (*string*) – (read only)
- **id** (*integer*) – (read only)
- **jobtype** (*string*) – (required)
- **num_dir_tasks** (*string*) – (read only)
- **numfailedfiles** (*integer*) –
- **numfiles** (*integer*) –
- **numprocessedfiles** (*integer*) –
- **numskippedfiles** (*integer*) –
- **owner** (*string*) – (read only)
- **paths** (*string*) – (read only)
- **site** (*string*) – Site Name (required)
- **state** (*string*) –
- **tasks[].job** (*string*) – (read only)
- **tasks[].numfailedfiles** (*integer*) –
- **tasks[].numfiles** (*integer*) –

- **tasks[].numprocessedfiles** (*integer*) –
- **tasks[].numskippedfiles** (*integer*) –
- **tasks[].state** (*string*) – (required)
- **tasks[].taskid** (*string*) – Job task ID (required)
- **tasks[].tasktype** (*string*) – (required)
- **tasks[].url** (*string*) – (read only)
- **url** (*string*) – (read only)

PATCH /jobs/{id}/

API endpoint for managing jobs.

Parameters:
- **id** (*string*) –

Request JSON Object:
- **created** (*string*) – Time of the job creation
- **dir_walk_complete** (*string*) – (read only)
- **id** (*integer*) – (read only)
- **jobtype** (*string*) – (required)
- **num_dir_tasks** (*string*) – (read only)
- **numfailedfiles** (*integer*) –
- **numfiles** (*integer*) –
- **numprocessedfiles** (*integer*) –
- **numskippedfiles** (*integer*) –
- **owner** (*string*) – (read only)
- **paths** (*string*) – (read only)
- **site** (*string*) – Site Name (required)
- **state** (*string*) –
- **tasks[].job** (*string*) – (read only)
- **tasks[].numfailedfiles** (*integer*) –
- **tasks[].numfiles** (*integer*) –
- **tasks[].numprocessedfiles** (*integer*) –
- **tasks[].numskippedfiles** (*integer*) –
- **tasks[].state** (*string*) – (required)
- **tasks[].taskid** (*string*) – Job task ID (required)
- **tasks[].tasktype** (*string*) – (required)
- **tasks[].url** (*string*) – (read only)
- **url** (*string*) – (read only)

Status Codes:
- 200 OK –

Response JSON Object:
- **created** (*string*) – Time of the job creation
- **dir_walk_complete** (*string*) – (read only)
- **id** (*integer*) – (read only)
- **jobtype** (*string*) – (required)
- **num_dir_tasks** (*string*) – (read only)
- **numfailedfiles** (*integer*) –
- **numfiles** (*integer*) –
- **numprocessedfiles** (*integer*) –
- **numskippedfiles** (*integer*) –
- **owner** (*string*) – (read only)

- **paths** (*string*) – (read only)
- **site** (*string*) – Site Name (required)
- **state** (*string*) –
- **tasks[].job** (*string*) – (read only)
- **tasks[].numfailedfiles** (*integer*) –
- **tasks[].numfiles** (*integer*) –
- **tasks[].numprocessedfiles** (*integer*) –
- **tasks[].numskippedfiles** (*integer*) –
- **tasks[].state** (*string*) – (required)
- **tasks[].taskid** (*string*) – Job task ID (required)
- **tasks[].tasktype** (*string*) – (required)
- **tasks[].url** (*string*) – (read only)
- **url** (*string*) – (read only)

DELETE /jobs/{id}/

API endpoint for managing jobs.

Parameters:
- **id** (*string*) –

Status Codes:
- 204 No Content –

GET /jobs/{id}/files/

Retrieves the files related with a job, with their execution status.

Parameters:
- **id** (*string*) –

Status Codes:
- 200 OK –

Response JSON Object:
- **created** (*string*) – Time of the job creation
- **dir_walk_complete** (*string*) – (read only)
- **id** (*integer*) – (read only)
- **jobtype** (*string*) – (required)
- **num_dir_tasks** (*string*) – (read only)
- **numfailedfiles** (*integer*) –
- **numfiles** (*integer*) –
- **numprocessedfiles** (*integer*) –
- **numskippedfiles** (*integer*) –
- **owner** (*string*) – (read only)
- **paths** (*string*) – (read only)
- **site** (*string*) – Site Name (required)
- **state** (*string*) –
- **tasks[].job** (*string*) – (read only)
- **tasks[].numfailedfiles** (*integer*) –
- **tasks[].numfiles** (*integer*) –
- **tasks[].numprocessedfiles** (*integer*) –
- **tasks[].numskippedfiles** (*integer*) –
- **tasks[].state** (*string*) – (required)
- **tasks[].taskid** (*string*) – Job task ID (required)

- **tasks[].tasktype** (*string*) – (required)
- **tasks[].url** (*string*) – (read only)
- **url** (*string*) – (read only)

POST /jobs/{id}/resubmit/

Resubmits the job with given id. If the job is not finished yet, this action will not have an effect.

Parameters:
- **id** (*string*) –

Status Codes:
- 201 Created –

GET /sites/

API endpoint for managing sites.

Query Parameters:
- **page** (*integer*) – A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (*integer*) – Number of results to return per page. Page size parameter can be a number between `20` and `100`. For disabling pagination and retrieving all results, `0` should be given. When page size parameter is empty or `<20`, `20` results are returned by default. When page size parameter `>100`, `100` results are returned by default.

Status Codes:
- 200 OK –

Response JSON Object:
- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[].id** (*integer*) – (read only)
- **results[].name** (*string*) – Site Name (required)
- **results[].url** (*string*) – (read only)

POST /sites/

API endpoint for managing sites.

Request JSON Object:
- **id** (*integer*) – (read only)
- **name** (*string*) – Site Name (required)
- **url** (*string*) – (read only)

Status Codes:
- 201 Created –

Response JSON Object:
- **id** (*integer*) – (read only)
- **name** (*string*) – Site Name (required)
- **url** (*string*) – (read only)

GET /sites/{id}/

API endpoint for managing sites.

Parameters:
- **id** (*string*) –

Status Codes:
- 200 OK –

Response JSON Object:
- **id** (*integer*) – (read only)
- **name** (*string*) – Site Name (required)
- **url** (*string*) – (read only)

PATCH /sites/{id}/

API endpoint for managing sites.

Parameters:
- **id** (*string*) –

Request JSON Object:
- **id** (*integer*) – (read only)
- **name** (*string*) – Site Name (required)
- **url** (*string*) – (read only)

Status Codes:
- 200 OK –

Response JSON Object:
- **id** (*integer*) – (read only)
- **name** (*string*) – Site Name (required)
- **url** (*string*) – (read only)

DELETE /sites/{id}/

API endpoint for managing sites.

Parameters:
- **id** (*string*) –

Status Codes:
- 204 No Content –

GET /tasks/

API endpoint for viewing tasks.

Query Parameters:
- **page** (*integer*) – A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (*integer*) – Number of results to return per page. Page size parameter can be a number between `20` and `100`. For disabling pagination and retrieving all results, `0` should be given. When page size parameter is empty or `<20`, `20` results are returned by default. When page size parameter `>100`, `100` results are returned by default.

Status Codes:
- 200 OK –

Response JSON Object:
- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[].job** (*string*) – (read only)

- **results[].numfailedfiles** (*integer*) –
- **results[].numfiles** (*integer*) –
- **results[].numprocessedfiles** (*integer*) –
- **results[].numskippedfiles** (*integer*) –
- **results[].state** (*string*) – (required)
- **results[].taskid** (*string*) – Job task ID (required)
- **results[].tasktype** (*string*) – (required)
- **results[].url** (*string*) – (read only)

GET /tasks/{taskid}/

API endpoint for viewing tasks.

Parameters:
- **taskid** (*string*) –

Status Codes:
- 200 OK –

Response JSON Object:
- **job** (*string*) – (read only)
- **numfailedfiles** (*integer*) –
- **numfiles** (*integer*) –
- **numprocessedfiles** (*integer*) –
- **numskippedfiles** (*integer*) –
- **paths** (*string*) – (read only)
- **results** (*string*) – (read only)
- **state** (*string*) – (required)
- **taskid** (*string*) – Job task ID (required)
- **tasktype** (*string*) – (required)
- **url** (*string*) – (read only)

GET /users/

API endpoint for managing users.

Query Parameters:
- **page** (*integer*) – A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (*integer*) – Number of results to return per page. Page size parameter can be a number between `20` and `100`. For disabling pagination and retrieving all results, `0` should be given. When page size parameter is empty or `<20`, `20` results are returned by default. When page size parameter `>100`, `100` results are returned by default.

Status Codes:
- 200 OK –

Response JSON Object:
- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[].date_joined** (*string*) –
- **results[].email** (*string*) –
- **results[].first_name** (*string*) –
- **results[].id** (*integer*) – (read only)

- **results[].is_active** (*boolean*) – Designates whether this user should be treated as active. Unselect this instead of deleting accounts.
- **results[].last_login** (*string*) –
- **results[].last_name** (*string*) –
- **results[].url** (*string*) – (read only)
- **results[].username** (*string*) – Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only. (required)

POST /users/

API endpoint for managing users.

Request JSON Object:
- **email** (*string*) –
- **first_name** (*string*) –
- **last_name** (*string*) –
- **password** (*string*) – (required)
- **username** (*string*) – Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only. (required)

Status Codes:
- 201 Created –

Response JSON Object:
- **email** (*string*) –
- **first_name** (*string*) –
- **last_name** (*string*) –
- **password** (*string*) – (required)
- **username** (*string*) – Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only. (required)

GET /users/{username}/

API endpoint for managing users.

Parameters:
- **username** (*string*) –

Status Codes:
- 200 OK –

Response JSON Object:
- **date_joined** (*string*) –
- **email** (*string*) –
- **first_name** (*string*) –
- **id** (*integer*) – (read only)
- **is_active** (*boolean*) – Designates whether this user should be treated as active. Unselect this instead of deleting accounts.
- **last_login** (*string*) –
- **last_name** (*string*) –
- **url** (*string*) – (read only)
- **username** (*string*) – Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only. (required)

PATCH /users/{username}/

API endpoint for managing users.

Parameters:

- **username** (*string*) –

Request JSON Object:

- **email** (*string*) –
- **first_name** (*string*) –
- **last_name** (*string*) –
- **password** (*string*) –

Status Codes:

- 200 OK –

Response JSON Object:

- **email** (*string*) –
- **first_name** (*string*) –
- **last_name** (*string*) –
- **password** (*string*) –

DELETE /users/{username}/

API endpoint for managing users.

Parameters:

- **username** (*string*) –

Status Codes:

- 204 No Content –

POST /users/{username}/activate/

Activates user account with given username.

Parameters:

- **username** (*string*) –

Status Codes:

- 201 Created –

POST /users/{username}/deactivate/

Deactivates user account with given username.

Parameters:

- **username** (*string*) –

Status Codes:

- 201 Created –

GET /workflows/

API endpoint for viewing workflows.

Query Parameters:

- **page** (*integer*) – A page number within the paginated result set. When not given, first page is retrieved by default.
- **page_size** (*integer*) – Number of results to return per page. Page size parameter can be a number between `20` and `100`. For disabling pagination and retrieving all results, `0` should be given. When page size parameter is empty or `<20`, `20` results are returned by default. When page size parameter `>100`, `100` results are returned by default.

- 200 OK –

Response JSON Object:
- **count** (*integer*) – (required)
- **next** (*string*) –
- **previous** (*string*) –
- **results[].fields** (*object*) –
- **results[].icon** (*string*) – HTML Representation of icon
- **results[].id** (*integer*) – (read only)
- **results[].label** (*string*) – Friendly name of the workflow (required)
- **results[].name** (*string*) – Name of Workflow (required)
- **results[].steps** (*object*) – (required)

POST /workflows/

API endpoint for viewing workflows.

Request JSON Object:
- **fields** (*object*) –
- **icon** (*string*) – HTML Representation of icon
- **id** (*integer*) – (read only)
- **label** (*string*) – Friendly name of the workflow (required)
- **name** (*string*) – Name of Workflow (required)
- **steps** (*object*) – (required)

Status Codes:
- 201 Created –

Response JSON Object:
- **fields** (*object*) –
- **icon** (*string*) – HTML Representation of icon
- **id** (*integer*) – (read only)
- **label** (*string*) – Friendly name of the workflow (required)
- **name** (*string*) – Name of Workflow (required)
- **steps** (*object*) – (required)

GET /workflows/{id}/

API endpoint for viewing workflows.

Parameters:
- **id** (*string*) –

Status Codes:
- 200 OK –

Response JSON Object:
- **fields** (*object*) –
- **icon** (*string*) – HTML Representation of icon
- **id** (*integer*) – (read only)
- **label** (*string*) – Friendly name of the workflow (required)
- **name** (*string*) – Name of Workflow (required)
- **steps** (*object*) – (required)

DELETE /workflows/{id}/

API endpoint for viewing workflows.

Parameters:
  • **id** (*string*) –

Status Codes:
  • 204 No Content –

Contact

Ngenea Hub is provided and supported by:

Pixit Media: https://pixitmedia.com/contact-us/

Arcastream: https://www.arcastream.com/contact/

Changelog

1.1.0

  • DYNAMOHUB-187 `api/users/` endpoint throwing errror when a username containing "." exists
  • DYNAMOHUB-186 "failed to acquire a DMAPI lock EXCL immediately; keeping trying…" is treated as an error
  • DYNAMOHUB-185 Job reporting not providing correct counts
  • DYNAMOHUB-184 Resubmit button should not be enabled when not usable
  • DYNAMOHUB-177 Using UI to trigger workflow fails
  • DYNAMOHUB-171 Attempting to post to /api/file/workflow with only a token causes 500
  • DYNAMOHUB-170 Bug when a directory is in intermediate selection state and closed
  • DYNAMOHUB-124 Send data between different sites via the UI
  • DYNAMOHUB-115 Rework ngenea output parsing
  • DYNAMOHUB-103 Rebuild UI using React
  • DYNAMOHUB-188 Improve stat task to make diretory handling explict
  • DYNAMOHUB-176 step argument: skip-check-hash
  • DYNAMOHUB-175 step argument: overwrite-remote
  • DYNAMOHUB-174 step argument: overwrite-local
  • DYNAMOHUB-173 step argument: stub-size
  • DYNAMOHUB-166 React 404 should not change URL
  • DYNAMOHUB-164 Support file delete workflow
  • DYNAMOHUB-147 Recent jobs list should show the most recent jobs, nevermind what state they are in
  • DYNAMOHUB-84 Job reporting scaling improvements

1.0.4

Thu Jun 10 2021

  • [DYNAMOHUB-148] Fix bug where sub-directories can fail to be processed

26

- [DYNAMOHUB-142] recursive_action failures report as success
- [DYNAMOHUB-151] Allow overriding the site on a per-step basis
- [DYNAMOHUB-150] Submission time arguments to workflow support
- [DYNAMOHUB-136] Validate runtime fields
- [DYNAMOHUB-156] Show verbose error information on failure
- [DYNAMOHUB-155] New task: reverse stub
- [DYNAMOHUB-152] Report task types with more meaningful names

1.0.3

Wed May 19 2021

- [DYNAMOHUB-144] Fixed an issue with submitting workflows with Client Key
- [DYNAMOHUB-133] Fixed warnings generated from auto field
- [DYNAMOHUB-140] Refactor existing ngenea hub tasks to message passing format
- [DYNAMOHUB-137] Support static arguments to a step
- [DYNAMOHUB-131] Provide existing actions as default workflows
- [DYNAMOHUB-134] enable easy access to dbshell
- [DYNAMOHUB-132] Workflows are now defined dynamically.
- [DYNAMOHUB-141] Allow users to create (API) client keys

1.0.2

Mon Apr 26 2021

- [DYNAMOHUB-90] Expose JWT Login API endpoints
- [DYNAMOHUB-94] Add Swagger API Documentation
- [DYNAMOHUB-85] Validate that site names do not end with the suffix we use to identify queue types
- [DYNAMOHUB-83] Allow middle-click/right-click to open navbar links in new tabs
- [DYNAMOHUB-81] Ship ngeneahub cli tool as a venv with docker-compose included
- [DYNAMOHUB-56] Improve performance of job page by doing pagination server-side
- [DYNAMOHUB-72] Package up rabbitmq and redis in the deployment
- [DYNAMOHUB-65] Provide systemd unit

1.0.1

Fri Mar 26 2021

- Initial Release

License

Ngenea Hub is licensed under the ArcaPix EULA: https://www.arcapix.com/licenses/EULA.txt

```
ArcaPix EULA
January 2021
http://www.arcapix.com/licenses/EULA.txt
Copyright 2021 ArcaPix Limited

This end user licence agreement (EULA) is a legal agreement between
you (the entity or individual who is using the software) and us (as
applicable either Pixit Media Limited, company number 07298805 or
ArcaStream Ltd, company number 08346283) in respect of (as
applicable):

* the computer software described in the order form, proof of
entitlement (POE), invoice or other document linking to this EULA
(in each case as issued by or agreed in writing with us), or which
otherwise incorporates or is governed by this EULA and the data
supplied with that software (collectively, Software);

* the application programming interface described in the order form,
proof of entitlement (POE), invoice or other document linking to
this EULA (in each case as issued by or agreed in writing with us),
or which otherwise incorporates or is governed by this EULA and the
data supplied with that application programming interface, in each
case whether provided on a standalone basis or alongside the
Software (collectively, API) and

* printed materials and electronic documents associated with that
Software or API (Documents).

The Software, API and Documents are collectively the "Work" and such
term includes each of the Software, API and Documents as applicable.

We license use of the Work to you on the basis of this EULA. We do
not sell the Work to you. We remain the owners of the Work at all
times. By using the Work, or otherwise by clicking "accept" or
otherwise indicating acceptance of this EULA, you confirm you accept
the terms of this EULA. If you do not accept this EULA you may not
use the Work.

Any services we provide, including but not limited to maintenance,
support, development and hosting, will be under separate terms of
business, but any software we provide to you incidentally in the
course of those services, will also be governed by this EULA unless
expressly stated that other licence terms will apply.

You should print a copy of this EULA for future reference.

1 Grant and scope of licence
```

1.1 In consideration of payment by you of the agreed licence fee and you agreeing to abide by the terms of this EULA, we grant to you a non-exclusive, non-transferable licence to use the Work on the terms of this EULA for the duration of your subscription. Your subscription will only be valid during the period for which you have a valid POE from us to use the Software and when your subscription expires this EULA will automatically terminate without the need for notice. Any termination of this EULA will also terminate your subscription and your POE will be invalidated.

1.2 You may download, install and use the Software for your own internal business purposes only on the systems, either physical or virtual detailed in the accompanying POE as identified in your purchase order (if applicable) or otherwise approved by us or our authorised representatives.

1.3 You may not use the Work for the purposes of making its functionality available to third parties as a service, whether directly or indirectly, without our express written agreement.

1.4 You may access and use the API solely for the purposes of

1.4.1 internally developing applications which communicate and interoperate with software or systems detailed in (and for the purposes detailed in) the accompanying POE as identified in your purchase order (if applicable) or otherwise approved by us or our authorised representatives; and

1.4.2 making calls to the systems or software permitted under clause 1.4.1, subject to any limits detailed in the accompanying POE as identified in your purchase order (if applicable) or otherwise agreed with us or our authorised representatives.

1.5 Your display and use information received through the API or data derived from that information is in each case subject to any limits detailed in the accompanying POE as identified in your purchase order (if applicable) or otherwise agreed with us or our authorised representatives).

1.6 You may use any Documents in support of the use permitted under condition 1.2 and make copies of the Documents as are reasonably necessary for their lawful use.

1.7 This EULA does not grant you permission to use the trade names, trademarks, service marks, or product names of us or our contributors or licensors, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of any "licence" files.

2 Restrictions

2.1 Except as expressly set out in this EULA or as permitted by any local law, you undertake:

2.1.1 not to copy the Work except where such copying is incidental to normal use of the Software, or where it is strictly necessary for the purpose of back-up or operational security;

2.1.2 not to rent, lease, sub-license, loan, translate, merge, adapt, vary or modify the Work;

2.1.3 not to make alterations to, or modifications of, the whole or any part of the Work, nor permit the Work or any part of them to be combined with, or become incorporated in, any other programs or other documentation as applicable, other than as expressly permitted in writing by us;

2.1.4 not to disassemble, decompile, reverse-engineer or create derivative works based on the whole or any part of the Software or API (except as expressly permitted by us in writing or clearly provided for within the functionality of the Software or any accompanying API we provide) nor attempt to do any such thing except to the extent that (by virtue of section 296A of the Copyright, Designs and Patents Act 1988) such actions cannot be prohibited because they are essential for the purpose of achieving inter-operability of the Software or API with another software program, and provided that the information obtained by you during such activities:

2.1.4.1 is used only for the purpose of achieving inter-operability of the Software or API with another software program; and

2.1.4.2 is not unnecessarily disclosed or communicated without our prior written consent to any third party; and

2.1.4.3 is not used to create any software which is substantially similar to the Software or API;

2.1.5 to keep all copies of the Work secure and to maintain accurate and up-to-date records of the number and locations of all copies of the Work;

2.1.6 to supervise and control use of the Work and ensure that the Work are only used by your employees (or such other individuals or entities are you may be expressly permitted in writing by us to allow to access or use the Work) in accordance with the terms of this EULA;

2.1.7 to include our copyright notice on and any "licence" text files in all entire and partial copies you make of the Work on any medium, however you may not use any component parts of the Work outside of or separately from the Work;

2.1.8 not to provide or otherwise make available the Work in whole or in part (including but not limited to program listings, object and source program listings, object code and source code), in any form to any person other than your employees without prior written consent from us; and

2.1.9 to comply with all applicable technology control or export laws and regulations.

2.2 Without prejudice to the restrictions in this EULA on copying, modifying or creating derivative works from the Work, where you (or someone on your behalf) creates (solely or in conjunction with others, and whether in object or source code form) any software or other work which is based on or derived from the Work (Derivative Work) in breach of this EULA or otherwise, then in consideration of the sum of 1 GBP (receipt and sufficiency of which you acknowledge), you hereby:

2.2.1 assign to us (by way of present assignment of future rights) all intellectual property rights in such Derivative Work and waive (and shall procure a waiver of) all moral rights arising under the Copyright, Designs and Patents Act 1988 in relation to the Derivative Work and, so far as is legally possible, any broadly equivalent rights that may exist in any territory of the world; and

2.2.2 In the event that any rights in such Derivative Work are not assigned to us pursuant to clause 2.2.1, you hereby grant to us an exclusive, royalty-free, worldwide, transferrable, irrevocable, perpetual licence (together with the right to grant sub-licences) to use in any manner as we determine, any such Derivative Work.

2.3 For the avoidance of doubt, for the purposes of clause 2.2, Derivative Work shall not include works which merely link or bind by name an existing third party application to the interfaces of the Software or API but does include works which are created to integrate with, or to be processed using, the interface of the Software or any API which we provide.

2.4 You agree not to (by your act or omission) do, or permit to be done, any act that will or may weaken, damage or be detrimental to the Work or any of our intellectual property rights or our or any of our contributors or licensors' rights in such, or seek to register any rights in the Work or any part of it or seek to commence litigation against any third party in respect of any intellectual property infringement in relation to the Work or any part of it.

3 Intellectual property rights

3.1 You acknowledge that all intellectual property rights in the Work anywhere in the world belong to us or our licensors or contributors, that rights in the Work are licensed (not sold) to you, and that you have no rights in, or to, the Work other than the right to use them in accordance with the terms of this EULA.

3.2 You acknowledge (unless explicitly agreed in writing by us) that you have no right to have access to the Software in source code form.

4 Liability

4.1 You acknowledge that the Work has not been developed to meet your individual requirements, including any particular cybersecurity requirements you might be subject to under law or otherwise, and that it is therefore your responsibility to ensure that the facilities and functions of the Software and API as described in the Documents meet your requirements.

4.2 We only supply the Work for internal use by your business, and you agree not to use the Work for any other purposes unless expressly permitted in writing by us.

4.3 We shall not in any circumstances whatever be liable to you, whether in contract, tort (including negligence), breach of statutory duty, or otherwise, arising under or in connection with this EULA for:

4.3.1 loss of profits, sales, business, or revenue;

4.3.2 business interruption;

4.3.3 loss of anticipated savings;

4.3.4 loss or corruption of data or information or any loss arising from misconfiguration or incorrect implementation or use of any API;

4.3.5 loss of business opportunity, goodwill or reputation;

where any of the losses set out in condition 4.3.1 to condition 4.3.5 are direct or indirect; or

4.3.6 any special, indirect or consequential loss, damage, charges or expenses.

4.4 Other than the losses set out in condition 4.3 (for which we are not liable), our maximum aggregate liability under or in connection with this EULA whether in contract, tort (including negligence) or otherwise, shall in all circumstances not exceed a sum equal to the Licence Fee paid in the 12 months prior to the event first giving rise to any liability. This maximum cap does not apply to condition 4.5.

4.5 Nothing in this EULA shall limit or exclude our liability for:

4.5.1 death or personal injury resulting from our negligence;

4.5.2 fraud or fraudulent misrepresentation;

4.5.3 any other liability that cannot be excluded or limited by English law.

4.6 Save as required by applicable law or agreed to in writing, we provide the Work on an "AS IS" basis, without conditions, warranties, representations or other terms of any kind, either express or implied (and any such implied conditions, warranties,

representations or other terms, whether implied by statute, common law or otherwise, are excluded to the fullest extent permitted by law), including, without limitation, any conditions, warranties, representations or other terms relating to title, non-infringement, merchantability, or fitness for a particular purpose. You are solely responsible for determining the appropriateness of using the Work and for any configuration or interface necessary for you to effectively use the Work and assume any risks associated with your exercise of permissions under this EULA.

4.7 Without prejudice to clause 4.6, where the API interacts with any software or system which is not provided by us, we are not responsible and shall have no liability in any way for such software or system.

5 Termination

5.1 We may terminate this EULA immediately by written notice to you if you commit a breach of this EULA which you fail to remedy (if remediable) within 14 days after the service of written notice requiring you to do so. Without prejudice to our rights under this clause 5.1, your rights under this EULA will terminate automatically without the need for notice if you commit a material breach of any of the terms of this EULA.

5.2 On termination for any reason:

5.2.1 all rights granted to you under this EULA shall cease;

5.2.2 you must immediately cease all activities authorised by this EULA; and

5.2.3 you must immediately and permanently delete or remove the Work from all computer equipment in your possession, and immediately destroy or return to us (at our option) all copies of the Work then in your possession, custody or control and, in the case of destruction, certify to us that you have done so.

6 Communications between us

6.1 We may update the terms of this EULA at any time on notice to you in accordance with this condition 6. Your continued use of the Work following the deemed receipt and service of the notice under condition 6.3 shall constitute your acceptance to the terms of this EULA, as varied. If you do not wish to accept the terms of the EULA (as varied) you must immediately stop using and accessing the Work on the deemed receipt and service of the notice.

6.2 If we have to contact you, we will do so by email or by pre-paid post to the address you provided in accordance with your order for or registration of the Work.

6.3 Note that any notice:

6.3.1 given by us to you will be deemed received and properly served 24 hours after it is first posted on our website, 24 hours after an email is sent, or three days after the date of posting of any letter; and

6.3.2 given by you to us will be deemed received and properly served 24 hours after an email is sent, or three days after the date of posting of any letter.

6.4 In proving the service of any notice, it will be sufficient to prove, in the case of posting on our website, that the website was generally accessible to the public for a period of 24 hours after the first posting of the notice; in the case of a letter, that such letter was properly addressed, stamped and placed in the post to the address of the recipient given for these purposes; and, in the case of an email, that such email was sent to the email address of the recipient given for these purposes.

7 Events outside our control

7.1 We will not be liable or responsible for any failure to perform, or delay in performance of, any of our obligations under this EULA that is caused by an Event Outside Our Control. An Event Outside Our Control is defined below in condition 7.2.

7.2 An Event Outside Our Control means any act or event beyond our reasonable control, including without limitation failure of public or private telecommunications networks.

7.3 If an Event Outside Our Control takes place that affects the performance of our obligations under this EULA:

7.3.1 our obligations under this EULA will be suspended and the time for performance of our obligations will be extended for the duration of the Event Outside Our Control; and

7.3.2 we will use our reasonable endeavours to find a solution by which our obligations under this EULA may be performed despite the Event Outside Our Control.

8 Third Party Software

8.1 Any part or component of the Software which has been contributed or created by any third party (including any open-source software) and which is not owned by us (Third Party Software) shall be deemed to be incorporated within the Software for the purposes of this EULA (except where expressly provided to the contrary) and use of the Third Party Software shall be subject to (and you shall comply with) such additional terms as relate to such Third Party Software from time to time (Third Party Additional Terms), and such Third Party Additional terms shall take precedence over this EULA in relation to such Third Party Software. You shall indemnify and hold us harmless against any loss or damage which we may suffer or incur as a result of your breach of any Third Party Additional Terms howsoever

arising, and we may treat your breach of any Third Party Additional Terms as a material breach of this EULA.

8.2 For the avoidance of doubt, the performance of, and any issues caused by or arising from, any Third Party Software shall be considered an Event Outside Our Control and (without prejudice to the provisions of this EULA in relation to warranties regarding the Software generally) all Third Party Software is provided on an "AS IS" basis and without conditions, warranties, representations or other terms of any kind, either express or implied (and any such implied conditions, warranties, representations or other terms, whether implied by statute, common law or otherwise, are excluded to the fullest extent permitted by law), including, without limitation, any conditions, warranties, representations or other terms relating to title, non-infringement, merchantability, or fitness for a particular purpose.

9 Other important terms

9.1 We may transfer our rights and obligations under this EULA to another organisation, but this will not affect your rights or our obligations under this EULA.

9.2 You may only transfer your rights or your obligations under this EULA to another person if we agree in writing.

9.3 This EULA and any document expressly referred to in it constitutes the entire agreement between us and supersedes and extinguishes all previous agreements, promises, assurances, warranties, representations and understandings between us, whether written or oral, relating to its subject matter. You agree that you shall have no remedies in respect of any statement, representation, assurance or warranty (whether made innocently or negligently) that is not set out in this EULA or any document expressly referred to in it. You agree that you shall have no claim for innocent or negligent misrepresentation or negligent misstatement based on any statement in this EULA or any document expressly referred to in it.

9.4 If we fail to insist that you perform any of your obligations under this EULA, or if we do not enforce our rights against you, or if we delay in doing so, that will not mean that we have waived our rights against you and will not mean that you do not have to comply with those obligations. If we do waive a default by you, we will only do so in writing signed by us, and that will not mean that we will automatically waive any later default by you.

9.5 Each of the conditions of this EULA operates separately. If any court or competent authority decides that any of them are unlawful or unenforceable, the remaining conditions will remain in full force and effect.

9.6 This EULA, its subject matter and its formation (and any non-contractual disputes or claims) are governed by English law. We both

irrevocably agree to the exclusive jurisdiction of the courts of England and Wales.