Quick reference for Aiida integration

Background

This document is a quick reference guide on how to authenticate, upload and interpret a document, as well as how to retrieve and update its values.

For the entire available API, please see api.aiida.io.

Guide

Authentication

Client

POST /authenticate/client

The response from successful authentication (status code 200) will return an authorization token. This token needs to be passed in the header ('authorization') for all other calls to the webservice.

Example request:

```
{
  "client_id": <string>,
  "client_secret": <string>
}
```

Example response:

```
{
  "access_token": "eyJhbG...",
  "expires_in": 3600,
  "token_type": "Bearer"
}
```

User

POST /authenticate/user

When authenticating as a user, the response will include a **set-cookie** header with the token.

Example request:

```
{
  "email": <string>,
  "password": <string>
}
```

Example response:

```
{
  "id": <uuid string>,
  "tos_accepted": <boolean>
}
```

Get workspaces

Before you can start uploading items and what not, you will need to check your workspaces for which IDs to reference.

GET /workspaces?include=collections,workflows

Example response:

```
[
  {
    "name": <string>,
    "created_at": "2020-03-11T08:48:12.682Z",
    "updated_at": "2020-03-11T08:48:12.682Z",
    "id": <uuid string>,
    "collections": [
      {
        "id": <uuid string>,
        "created_at": "2020-03-18 16:57:29.076141",
        "updated_at": "2020-09-15 11:48:53.918164",
        "name": <string>,
        "workspace_id": <uuid string>,
        "settings": {
          "autobook": true
        },
        "accounts": [],
        "cost_centers": [],
        "projects": [],
        "currencies": [],
        "voucher_series": []
      }
```

```
  ],
  "workflows": [
    {
      "name": <string>,
      "created_at": "2020-03-11T08:48:12.682Z",
      "updated_at": "2020-03-11T08:48:12.682Z",
      "id": <uuid string>,
      "workspace_id": <uuid string>,
      "collection_id": <uuid string>,
      "columns": [
        {
          "edit_items": true,
          "delete_items": true,
          "previous": {
            "name": <string>,
            "type": <string>,
            "workflow_id": <uuid string>,
            "index": 0,
            "created_at": "2020-03-11T08:48:12.682Z",
            "updated_at": "2020-03-11T08:48:12.682Z",
            "id": <uuid string>,
            "settings": {}
          },
          "next": {
            "name": <string>,
            "type": <string>,
            "workflow_id": <uuid string>,
            "index": 0,
            "created_at": "2020-03-11T08:48:12.682Z",
            "updated_at": "2020-03-11T08:48:12.682Z",
            "id": <uuid string>,
            "settings": {}
          },
          "name": <string>,
          "subscribed_items": 0,
          "type": <string>,
          "workflow_id": <uuid string>,
          "index": 0,
          "created_at": "2020-03-11T08:48:12.682Z",
          "updated_at": "2020-03-11T08:48:12.682Z",
          "id": <uuid string>,
          "settings": {}
        }
      ],
      "settings": {}
    }
  ]
```

```
  }
]
```

Get Fields

You will also need to download the fields that the eventual values belong to.

GET /fields?workflow_id={workflow_id}

Example response:

```
[
  {
    "field_type_id": <uuid string>,
    "required": true,
    "shared": true,
    "name": <string>,
    "created_at": "2020-03-11T08:48:12.682Z",
    "updated_at": "2020-03-11T08:48:12.682Z",
    "id": <uuid string>,
    "field_group_id": <uuid string>,
    "display_name": <string>
  }
]
```

Update accounts

PUT /collections/{collection_id}/accounts

Used to set the list of accounts you are using for the given collection. Account number predictions (see “**Get document accounting values**”) will always be one of these values.

Example request:

```
[
  {
    "number": <string>,
    "description": <string>,
    "project": <string>,
    "project_settings": <string>,
    "cost_center": <string>,
    "cost_center_settings": <string>
  }
]
```

Example response:

```
[
  {
    "id": <uuid string>,
    "created_at": "2020-10-09T09:42:27.939Z",
    "updated_at": "2020-10-09T09:42:27.939Z",
    "collection_id": <uuid string>,
    "number": <string>,
    "description": <string>,
    "project": <string>,
    "project_settings": <string>,
    "cost_center": <string>,
    "cost_center_settings": <string>
  }
]
```

Update cost centers

PUT /collections/{collection_id}/cost_centers

Used to set the list of cost centers you are using for the given collection. Cost center code predictions (see "**Get document accounting values**") will always be one of these values.

Example request:

```
[
  {
    "description": <string>,
    "code": <string>
  }
]
```

Example response:

```
[
  {
    "id": <uuid string>,
    "created_at": "2020-10-09T09:54:19.641Z",
    "updated_at": "2020-10-09T09:54:19.641Z",
    "collection_id": <uuid string>,
    "code": <string>,
    "description": <string>
  }
```

```
]
```

Update projects

PUT /collections/{collection_id}/projects

Used to set the list of projects you are using for the given collection. Project number predictions (see "**Get document accounting values**") will always be one of these values.

Example request:

```
[
  {
    "project_number": <string>,
    "description": <string>
  }
]
```

Example response:

```
[
  {
    "id": <uuid string>,
    "created_at": "2020-10-09T09:55:58.800Z",
    "updated_at": "2020-10-09T09:55:58.800Z",
    "collection_id": <uuid string>,
    "project_number": <string>,
    "description": <string>
  }
]
```

Upload and start interpretation of a document

POST /items

Set **mime_type** with a standard mime type, **file_content_base64** with base 64 content, **name** (could be whatever you choose but filename or another identifier may help tracking errors) and **workflow_id** to upload a file. (**document_type** is unused, can be any string)

*You can substitute **workflow_id** with a **collection_id**, if you want automatic classification.*

Example request:

```
{
  "document_type": "",
  "file_content_base64": <base64 string>,
  "mime_type": "application/pdf",
  "name": <string>,
  "workflow_id": <uuid string>
}
```

Example response:

```
{
  "id": <uuid string>,
  "created_at": "2020-03-19 08:21:10.969269",
  "updated_at": "2020-03-19 08:21:10.969278",
  "name": <string>,
  "error": <string> / null,
  "values": [],
  "state": "processing",
  "document_type": "",
  "user_id": <uuid string> / null,
  "column_id": <uuid string>,
  "collection_id": <uuid string> / null,
  "original_file_id": <uuid string> / null,
  "locked_by_user_id": <uuid string> / null
}
```

Poll document to determine when values has been interpreted

GET /items/{item_id}

Check the "**state**" key. If an item has the state…

- "**processing**" means that the interpretation is not yet finished.
- "**Interpreted**" means that it's considered as valid, and the values are ready to be used.
- "**incomplete**" then the document is interpreted but not all fields are valid. For more detailed information, look at the values "**valid**" state to see what field(s) was wrong.
- "**error**" that means that some error happened so the document could not be interpreted

Example response:

```
{
  "id": <uuid string>,
  "created_at": "2020-03-10T14:45:13.012Z",
  "updated_at": "2020-03-10T14:45:13.012Z",
```

```
  "column_id": <uuid string>,
  "collection_id": <uuid string> / null,
  "original_file_id": <uuid string> / null,
  "locked_by_user_id": <uuid string> / null,
  "state": "interpreted",
  "name": <string>,
  "page_count": 1,
  "comment_count": 0,
  "user_id": <uuid string> / null,
  "error": <string> / null,
  "document_type": "",
}
```

Get document field values

Once the item has finished processing, it should include a "**values**" array.
Use "**field_id**" to identify what field the value belongs to. Key "**value**" contains the formatted value.

Example response:

```
{
  "id": <uuid string>,
  "created_at": "2020-03-10T14:45:13.012Z",
  "updated_at": "2020-03-10T14:45:13.012Z",
  "column_id": <uuid string>,
  "collection_id": <uuid string> / null,
  "original_file_id": <uuid string> / null,
  "locked_by_user_id": <uuid string> / null,
  "state": "interpreted",
  "name": <string>,
  "page_count": 1,
  "comment_count": 0,
  "user_id": <uuid string> / null,
  "error": <string> / null,
  "document_type": <string>,
  "values": [
    {
      "id": <uuid string>,
      "valid": true,
      "count": 0,
      "value": <string>,
      "field_id": <uuid string>,
      "location": [{ "y": 0, "x": 0 }],
      "page_number": 0,
      "confidence": 0,
```

```
      "extracted": [<string uuids>]
      "column_index": 0
    }
  ]
}
```

Set document field values

PUT /items/{item_id}/values

In the case where some of the values are incorrect, you might want to correct them so they are trained on, to improve future results. This can be done by sending an array with the updated values.

Set "**field_id**" to identify field and set "**user_value**" to set actual value.

Example request:

```
[
  {
    "user_value": <string>,
    "field_id": <uuid string>,
  }
]
```

Get document accounting values

GET /items/{item_id}/accountings

This is used to get account, cost center and project predictions. Note that since cost centers and projects are specific for each collection, you need to provide training data for these before you can get any predictions, see “**Updating document accounting values**”.

Example response:

```
[
  {
    "id": <uuid string>,
    "account": <string>,
    "cost_center": <string>,
    "project": <string>,
    "is_accrual": <boolean>,
    "transaction_information": <string>,
    "debit": <string>,
    "credit": <string>
```

```
  }
]
```

Updating document accounting values

PUT /items/{item_id}/accountings

In the case where some of the accounting values are incorrect, you might want to correct them so they are trained on, to improve future results. This can be done by sending an array with the updated accounting values.

Update incorrect entries while keeping the “**id**”, or remove them from the list. Add new entries without any “**id**”.

Example request:

```
[
  {
    "id": <uuid string>,
    "transaction_information": <string>,
    "debit": <string>,
    "project": <string>,
    "credit": <string>,
    "account": <string>,
    "is_accrual": <boolean>,
    "cost_center": <string>
  }
]
```