Quick reference guide

Aiida accounting workflow integration

Rev. 1.3

Background

This document is a quick reference guide on how to set up a new collection, create account workflow, setup account workflow and get account information.

Documentation

Full API documentation can be found at https://api.aiida.io

Steps

Create a new collection for a customer

POST /collections

Request:

```
{
  "workspace_id": <WORKSPACE ID>,
  "name": <YOUR COLLECTION NAME>,
  "settings": {
    "accounting": true,
    "autobook": true
  }
}
```

Create new fields and workflows using a template

POST /apply_template

Request:

```
{
  "collection_id": <COLLECTION ID>,
  "template_name": "api-standard"
}
```

NOTE: This creates all the standard fields. If you use other or less fields you need to explicitly create or delete these fields. **Important** especially when setting values (training).

Fill accounts for collection

PUT /collections/{collection_id}/accounts

Request:

```
[
  {
    "number": <ACCOUNT NUMBER (four digits)>,
    "description": <DESCRIPTION OR NAME OF ACCOUNT>
  }
]
```

Note: For collections using cost centers or projects there exists additional parameters, see full api documentation page.

Set predefined accounts

NOTE: This only works for supplier invoices

PUT /collections/{collection_id}/predefined_accounts

Request:

```
[
  {
    "account": <ACCOUNT NUMBER (four digits)>,
    "name": "Total"|"VAT"|"Rounding"
  }
]
```

Accounts must first be uploaded with PUT /collections/{collection_id}/accounts. In this step, standard accounts are assigned for total, VAT, and rounding.

Get accountings after finished uploading item

GET /items/{item_id}/predefined_accountings
Gets the entries for the predefined accounts, possibly with debit and credit assigned
GET /items/{item_id}/accountings

Gets the predicted accounts and corresponding entries with debit / credit.
Response:

```
[
  {
    "id": <ACCOUNTING ID>,
    "account": <ACCOUNT NUMBER (four digits)>,
    "debit": <AMOUNT>,
    "credit": <AMOUNT>,
    "created_at": <DATE>,
    "updated_at": <DATE>,
    "cost_center": <STRING>,
    "project": <STRING>,
    "transaction_information": <STRING>,
    "is_accrual": false,
    "item_id": <ITEM ID>
  }
]
```

Note: This call can be done after the item has received the state “Interpreted” or “Incomplete”.

Train accounts

PUT /items/{item_id}/accountings

Request:

```
[
  {
    "debit": <YOUR AMOUNT>,
    "credit": <YOUR AMOUNT>,
    "transaction_information": "string",
    "is_accrual": false,
    "account": <YOUR ACCOUNT NUMBER (four digits)>
  }
]
```

Note: This must be done before setting field values (PUT /items/{item_id}/values)
Note: The PUT request is done on the collection, and therefore the complete list of accountings will be overwritten by the input values here.