Faculty Data Upload/Import REST Interfaces

V1.1 06/10/2020

The following are the REST APIs for UCOATS faculty data upload/import process. The faculty data file format (data element specifications https://uclahs.box.com/s/ekvd7iaqsqxnd48ix1iw6wtx2c9dn19n) is the same as the one under faculty import under App/Admin.

IMP.10 FACULTY_IMPORT - Submit request to upload/import faculty data
Endpoint: /import/app/faculty/upload
• Method: POST (multipart/form-data) • Input: file_upload – the faculty import file fiscal_year - 2018-2019, 2019-2020, 2020-2021 Fiscal year can be • Output: R10.0 , R20.0, R70.0 (See API Responses below)

IMP.20 FACULTY_IMPORT_CHECK - Check on status of faculty upload/import request
Endpoint: / import/app/faculty/status/{upload_id}
• Method: GET • Input: upload_id – the identifier for the file upload. Set upload_id = last to get status of the last upload to the system regardless of fiscal year. • Output: R20.0, R30.0, R40.0, R50.0, R60.0, R70.0 (See API Responses below)

IMP.30 FACT_IMPORT_EXCEPT Return annotated exception file for the upload/import request
EndPoint: /import/app/faculty/exceptionFile/{upload_id}
• NOTES: only available when the file upload request status is “complete_with_exceptions” or “complete_successful”. • Method: GET • Input: upload_id – the identifier for the file upload. Returns the state of the last upload if upload id is not passed in. Set upload_id = last to get status of the last upload to the system regardless of fiscal year. • Output: Subset of rows from the import file that could not processed. There is an additional column added that describes the exception associated with rows. The purpose of this file is for the sender to update the file and resubmit for processing. , R70.0 (See API Responses below)

REST API Responses

The REST responses data payload is as follows -

```
{
  "response_code": "",
  "response_message" : "",
  "response_body": ""
```

}

- response_code provides the status information about the import request
- response_message provides a message describing the response code.
- response_body a response_code specific structured response. Not all responses will include the response_body.

R10.0 UPLOAD_STATUS = INVALID_FILE

* There is an issue with the file. General issues are the file extensions and file is not in tab delimited format.
* response_body could be empty or not exist

Sample

```
{
  "response_code":"INVALID_FILE",
  "response_message":"File needs to have an extension .tsv or .txt and be in tab delimited",
  "response_body": {}
}
```

R20.0 UPLOAD_STATUS = VALIDATING

* File has been received and system is performing row checks
* response_body has one attribute:
  upload_id – upload_request_id

Sample

```
{
  "response_code":"VALIDATING",
  "response_message":"System is currently validating rows in the file",
  "response_body": {
     "upload_id":2342
   }
}
```

R30.0 UPLOAD_STATUS = PROCESSING_SCHEDULED

* Import has been scheduled for file.
* response_body
  upload_id – upload request identifier

Sample

```
{
  "response_code":"PROCESSING_SCHEDULED",
  "response_message":"Import file request is in queue",
  "response_body": {"upload_id":2342}
}
```

R40.0 UPLOAD_STATUS = PROCESSING

* Import file is being processed
* response_body has one attribute
  upload_id - update request identifier


```
{
 "response_code":"PROCESSING",
 "response_message":"Import File is being processed",
 "response_body": {"upload_id":2342}
}
```


R50.0 UPLOAD_STATUS = COMPLETE_WITH_EXCEPTIONS

* The system is unable to process some faculty records. Use the webservice to pull the annotated error file.
* response_body has three attributes:
  upload_id - Upload ID
  records_failed - The number of faculty records that did not pass validation.
  total_import - The number of faculty records that were added/updated in the system.


```
{
 "response_code":"COMPLETE_WITH_EXCEPTIONS",
 "response_message":"Some faculty records could not be processed",
 "response_body": {"upload_id":2342, "records_failed":3, "total_imported":197}
}
```


R60.0 UPLOAD_STATUS = COMPLETE_SUCCESSFUL

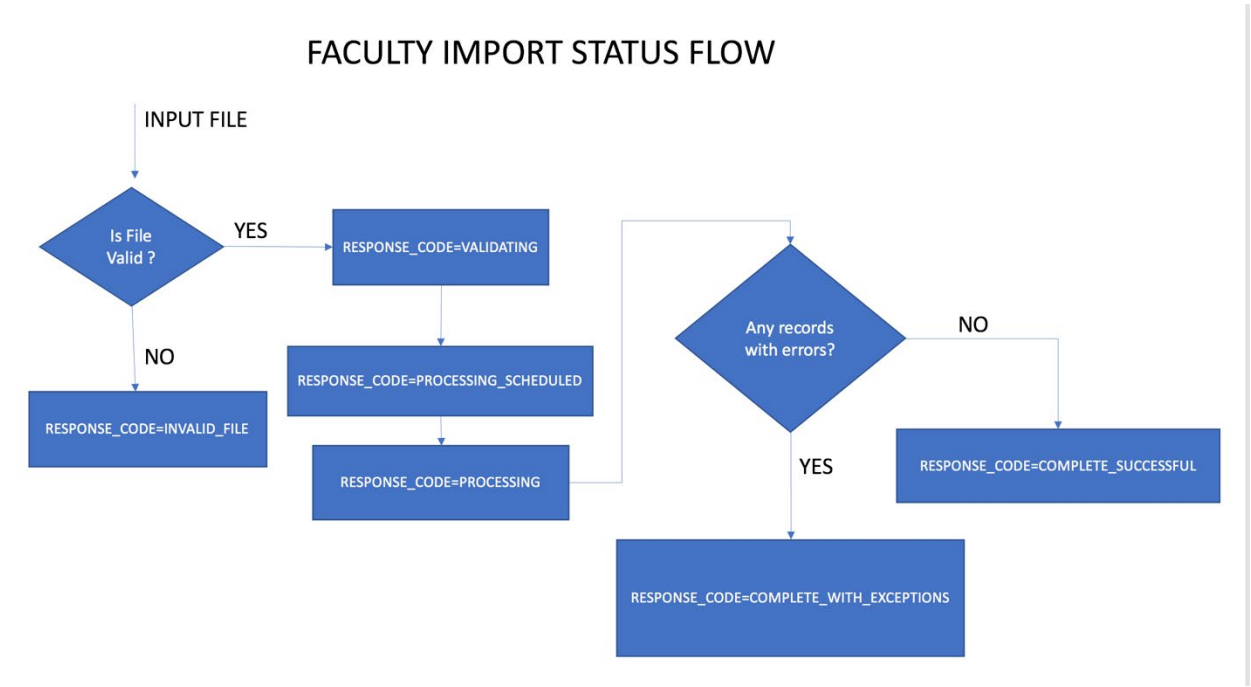
* The system processed all faculty records successfully.
* response_body has two attributes:
  upload_id - Upload ID
  total_import - The number of faculty records that were added/updated in the system.


```
{
 "response_code":"COMPLETE_SUCCESSFUL",
 "response_message":"All records were processed successfully",
 "response_body": {"upload_id":2342, "total_import":200}
}
```


R70.0 UPLOAD_STATUS = INVALID

* response_message – describes the reason for the request to be invalid. This could be a response due to the request for an exception file that is not ready or an invalid upload_id.
* response_body - response_body could be empty or not exist.

```
{
  "response_code":"INVALID",
  "response_message":"Import exception file not ready.  Import status has to be in either complete_with_exception or complete_successful",
  "response_body": {}
}
```

Upload file Notifications

The system will send an email notification (associated with the upload account) on receipt of the file and when the import processing has completed (import status = complete_successful or complete_with_exceptions).

Modes of Import process:

1. Single file processing
2. Multiple file processing

Processing Mode

Single File processing:

1. Send request for import faculty file (WS.10) and check that the response_status for VALIDATING
2. Wait for email of import process has completed or run a daily check with upload_id = last
3. If email returned or daily check returns R60.0 response status, then all records were imported; No action is needed. If a R50.0 response is returned call WS.30 to get the exception file. Address the issues specified in the extra column and reupload the modified exception for processing.

Multiple File Processing –

The system is able to handle multiple import requests. The system process one import file (First in First out) at a time until all of the requests are completed. This requires the external system to maintain the upload_ids associated for each request in order to check on the status for the correct one

CURL Example:

Request a token to access the webservices.

Step 1: Upload file

curl -X POST -H "Authorization: bearer {user_token}" -F "file_upload=@myfaculty-file.tsv -F "fiscal_year=2020-2021" -k https://{ws_urlstub}/import/app/faculty/upload

Return:

{"response_code":"VALIDATING","response_message":"System is currently validating rows in the file","response_body":{"upload_id":"123"} }

Step 2: Status check

Option A: Check the status of last upload request (assumes one upload at a time)

curl -X GET -H "Authorization: bearer {user_token}" -k https://{ws_urlstub}/import/app/faculty/status/last

Option B: Get upload status using the upload_id

curl -X GET -H "Authorization: bearer {user_token}" -k https://{ws_urlstub}/import/app/faculty/status/123

Return:

{"response_code":"COMPLETE_WITH_EXCEPTIONS","response_message":"Some faculty records could not be
processed","response_body":{"upload_id":"123","records_failed":"7","total_imported":"3"} }

Step 3: Get exception file – records that could not be processed.

Option A: Get file with the rows with exceptions

curl -X GET -H "Authorization: bearer {user_token}" -k https://{ws_urlstub}
/import/app/faculty/exceptionFile/last -o "{error_file_name}"

Option B:

curl -X GET -H "Authorization: bearer {user_token}" -
k https://{ws_urlstub}/import/app/faculty/exceptionFile/123 -o "{error_file_name}"

Return:

File