

# Staff Data Upload/Import REST Interfaces

V1.0 01/11/2022

The following are the REST APIs for UCOATS staff data upload/import process. The staff data file format (data element specifications <https://uclahs.app.box.com/file/687936747332?s=h3j1bfqvvtb5r48gjmsxnscy96elumnw>) is the same as the one under staff import under App/Admin.

<b>IMP.10 STAFF_IMPORT - Submit request to upload/import staff data</b>
Endpoint: /import/app/staff/upload
<ul style="list-style-type: none"><li>• Method: POST (multipart/form-data)</li><li>• Input: file_upload – the staff import file fiscal_year - 2018-2019, 2019-2020, 2020-2021 Fiscal year can be</li><li>• Output: R10.0 , R20.0, R70.0 (See API Responses below)</li></ul>
<b>IMP.20 STAFF_IMPORT_CHECK - Check on status of staff upload/import request</b>
Endpoint: / import/app/staff/status/{upload_id}
<ul style="list-style-type: none"><li>• Method: GET</li><li>• 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.</li><li>• Output: R20.0, R30.0, R40.0, R50.0, R60.0, R70.0 (See API Responses below)</li></ul>
<b>IMP.30 FACT_IMPORT_EXCEPT Return annotated exception file for the upload/import request</b>
EndPoint: /import/app/staff/exceptionFile/{upload_id}
<ul style="list-style-type: none"><li>• NOTES: only available when the file upload request status is “complete_with_exceptions” or “complete_successful”.</li><li>• Method: GET</li><li>• 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.</li><li>• 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)</li></ul>

## 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 staff records. Use the webservice to pull the annotated error file.
- \* response\_body has three attributes:
  - upload\_id - Upload ID
  - records\_failed - The number of staff records that did not pass validation.
  - total\_import - The number of staff records that were added/updated in the system.

```
{  
  "response_code":"COMPLETE_WITH_EXCEPTIONS",  
  "response_message":"Some staff 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 staff records successfully.
- \* response\_body has two attributes:
  - upload\_id - Upload ID
  - total\_import - The number of staff 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}  
}
```

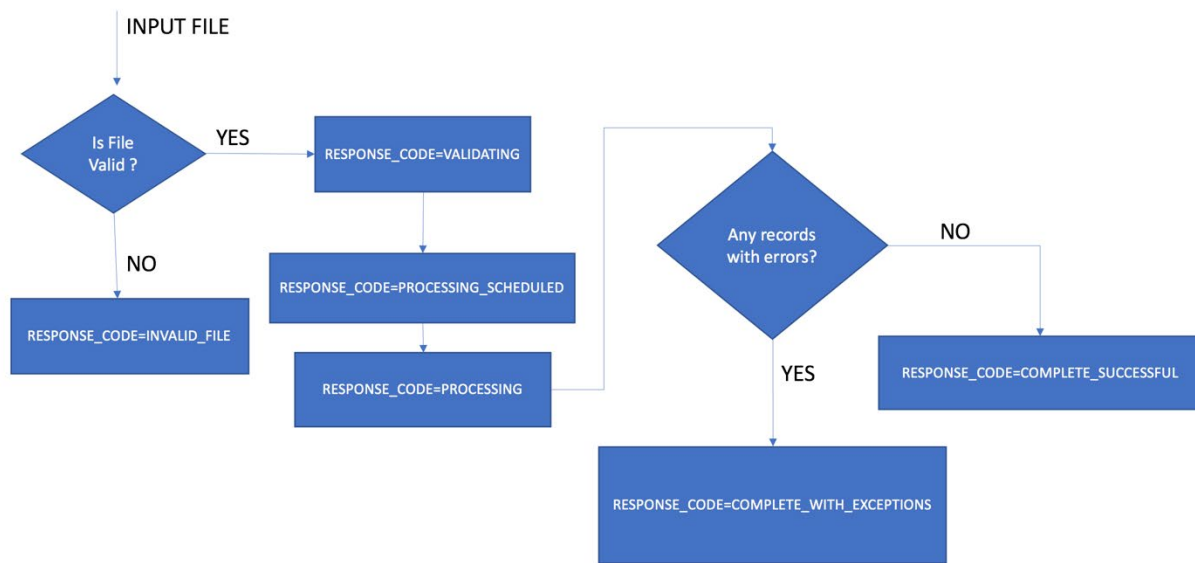
#### R70.0 UPLOAD\_STATUS = INVALID

- \* 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": {}  
}
```

## IMPORT STATUS FLOW



## 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 staff 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 webservice.

Step 1: Upload file

```
curl -X POST -H "Authorization: bearer {user_token}" -F "file_upload=@mystaff-file.tsv -F "fiscal_year=2020-2021" -k https://{ws_urlstub}/import/app/staff/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/staff/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/staff/status/123

Return:

```
{"response_code":"COMPLETE_WITH_EXCEPTIONS","response_message":"Some staff 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/staff/exceptionFile/last -o "{error_file_name}"
```

Option B:

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

Return:

File