Chapter 3

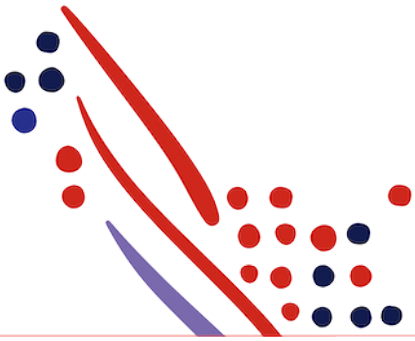
Recommendations API

from ADP DataCloud Identity, Employment and Income Verification API Guide

Published on
Jun 16, 2021 10:11AM

Last modified
Jun 20, 2023 11:50AM

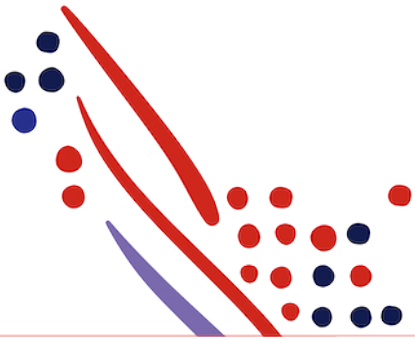
Always Designing for People™



ADP Copyright Information

Published on
Jun 16, 2021 10:11AM

Last modified
Jun 20, 2023 11:50AM



Chapter Contents

Chapter 3

Recommendations API

Overview

The Recommendations API is intended for use by partners who want to improve their customer experiences by using their end users' data to tailor their experiences and provide recommendations when directed by the user to do so.

Currently, the Recommendations API is only available via the Batch process. You can refer to the chapter on Batch Processing for more information.

---

1. Submit Job

2 files need to be submitted via SFTP: 1) input file in csv format and 2) config file in json format.

1a. Input file in CSV format

The input file is a CSV file that includes all the individuals for which verifications are needed. This file must include only the following columns unless otherwise agreed upon with ADP: id, birthDate, uid

- id
  - Format: ######### (SSN with no space or dash)
- birthDate
  - Format: YYYY-MM-DD (i.e. 1900-01-01)
- uid
  - 16 character length unique ID

Example File Structure

id	birthDate	uid
#########	YYYY-MM-DD	ab211827-825b-447d-9d27-088b9079ed10
#########	YYYY-MM-DD	ab211827-825b-447d-9d27-088b9079ed10

The input file must:

- have the header and the field format specifically as identified above.
- not exceed 7M rows.
- be in csv format and gzip compressed *before* PGP encryption
- contain newline separators delineating new rows, commas to indicate new field values and the values must not be quoted with quotations.

1b. Config file in JSON format

The filename of this JSON file must match the employee record filename, or else the input data file will not be processed.

The following fields must be included:

- "requestorInfo" (object)
    - "marketSegmentCode" (string) - You must use "Batch" if using the batch process.
    - "verifierName" (string)
- "associateConsentReceivedIndicator" (boolean)
- "verificationType" (string-enum)
    - Valid values: ["enhanced-employment", "basic-employment-income", "enhanced-employment-income", "basic-employment-income-standard-v2", "enhanced-employment-income-standard-v2"]
- "responseFilters" (object)
    - "ytdReturnedIndicator" (boolean)
        - null is equivalent to false
        - true will return the remunerationSummary block

If you want to use different response filter values for different sets of data, you must submit them as two different jobs. Each job must have one JSON and csv pair. As a result, if you want to use two different response filter values, you must submit two different pairs, each pair consisting of a JSON and csv.

```
{
   {
   "requestorInfo": {
     "marketSegmentCode": "Batch",
     "verifierName": "ABC Bank"
   },
   "associateConsentReceivedIndicator": true,
   "verificationType": "recommendations",
   "responseFilters": {
       "ytdReturnedIndicator": true
   }
}
```

1c. Submitting the files

Submit the two files using naming convention below:

- **<filename>.csv.gz.pgp file**
    - The input file must be OpenPGP encrypted using the provided DataCloud public OpenPGP key, using a public key the ADP team will provide to the partner
    - The filename must be unique. If the filename is not unique (for instance, conflicts with an already previously submitted file), the file will not be processed and the status API with the filename as the job ID will return a 404 error. We recommend suffixing the filename with a timestamp to ensure uniqueness
- **<filename>.json file**
    - **The config JSON file included in the multipart request mentioned in the previous section must be included as a JSON file (cannot be PGP encrypted or GZIP compressed)**

There will be an approximate 15-minute delay between when the files are submitted and when the status API will begin returning a "received" response.

Both the input_filename.json and the input_filename.csv.gz.pgp files must be included in the root directory titled "toADP" or else the files will not be processed.

ADP will provide 2 SFTP accounts to the partner: one for production files and one for test files.

- **Example directory structure:**
    - toADP/**filename_1.csv.gz.pgp**
    - toADP/**filename_1.json**
    - toADP/**filename_2.csv.gz.pgp**
    - toADP/**filename_2.json**
- **Example incorrect dir structures:**

- Missing JSON file containing job metadata
    - toADP/**filename_1.csv.gz.pgp**
- Duplicate filenames
    - toADP/**filename_1.csv.gz.pgp**
    - toADP/**filename_1.json**
    - toADP/**filename_1.csv.gz.pgp**
    - toADP/**filename_1.json**
- Unmatched filenames
    - toADP/filename_1.csv.gz.pgp
    - toADP/filename_2.json
- Unencrypted data file
    - toADP/filename.csv.gz
    - toADP/filename.json

---

2. Get Job Status (API) (swagger spec)

A separate REST API (HTTPS GET) will help retrieve the status of the submitted job

Input

GET https://verifications.adp.com/core/v2/associate-verifications/processing-jobs/{job-id}/summary

"{job-id}" in the above URL will be the filename of the .csv that was part of the input, minus the extension.

```
curl --location --request GET 'https://verifications.adp.com/core/v2/associate-verifications/processing-jobs/filename_1/summary' \
--header 'Authorization: Bearer <insert_auth_token_here>'
```

Response Returns

- The filename (original file name submitted as an attachment in the input submission request).
  Response file naming structure: <input_filename>.<matched|unmatched|error>.<page_number>.csv.gz.pgp
- Job status
- Number of matched records
- Number of unmatched records
- Number of error records (invalid inputs, etc)

Example Responses

Scenario 1: Job is submitted, but not yet processed

```
{
   "processingJobId": "filename_1",
   "processingStatus": "received"
}
```

Scenario 2: Job is processed and the output files are delivered (no error file)

```
{
 "processingJobId": "filename_1",
 "totalItemCount": 1000000,
 "processingCounts": {
  "errorCount": 0,
  "successCount": 1000000,
  "matchedRecordCount": 500000,
  "unmatchedRecordCount": 500000
 },
 "processingStatus": "completed",
 "messages": [
     {
      "messageTypeCode": "matched",
      "resourceID": "filename_1.matched.0.csv.gz.pgp",
      "itemCount": 500000
     },
     {
      "messageTypeCode": "unmatched",
      "resourceID": "filename_1.unmatched.0.csv.gz.pgp",
      "itemCount": 500000
     }
   ]
}
```


Scenario 3: Job is processed and the output files are delivered (with error file and multiple matched response files)


```
{
 "processingJobId": "filename_1",
 "totalItemCount": 1000000,
 "processingCounts": {
  "errorCount": 2,
  "successCount": 999998,
  "matchedRecordCount": 500004,
  "unmatchedRecordCount": 499998
 },
 "processingStatus": "completed",
 "messages": [
     {
      "messageTypeCode": "matched",
      "resourceID": "filename_1.matched.0.csv.gz.pgp",
      "itemCount": 500000
     },
{
      "messageTypeCode": "matched",
      "resourceID": "filename_1.matched.1.csv.gz.pgp",
      "itemCount": 4
     },
     {
      "messageTypeCode": "unmatched",
      "resourceID": "filename_1.unmatched.0.csv.gz.pgp",
      "itemCount": 499994
     },
     {
      "messageTypeCode": "error",
      "resourceID": "filename_1.error.0.csv.gz.pgp",
      "itemCount": 2
     }
   ]
}
```


Scenario 4: Job not found (response code = 404)

```
{
  "_confirmMessage": {
    "messageDateTime": "2022-08-09",
    "requestID": "f74cc059-6411-4722-95bf-f5a94be69a3a",
    "processingStatus": "not found",
    "messages": [
      {
        "messageCode": 404,
        "messageTypeCode": "error",
        "messageText": "Job not found with job id: 123"
      }
    ]
  }
}
```


Additional Notes

- The processingJobID will hold the value of the input file name sans the suffix. For example, if the input filename pair is filename_1.csv.gz.pgp & filename_1.json, processingJobID in the job status API would be "filename_1". Thus, to ensure uniqueness in the filenames and job ids, **we recommend using the current epoch timestamp as the filename value**
- Please note that the filename format in the response is: "<input_filename>.<file_status>.<file_number>.csv.gz.pgp" (output files are pgp encrypted using the partner provided)
- <file_status> can be either matched, unmatched, or error. Essentially, all 200 HTTP codes go into the matched file, all 404 HTTP codes into an unmatched file, and all other records (i.e. 400 HTTP codes) go into the error file
- <file_number> is incremented by 1 for each <file_status> depending on the number of the files. Index starts at 0. There can be multiple numbers of files for each file type: matched, unmatched, error
- The partner must poll for each job status
- An error, matched, or unmatched file is only generated when there is a value in it. For example, if an input file yields 0 matches, then a matched file will not be generated. Conversely, if an input file yields 100% match, then an unmatched file will not be generated

---

3. Retrieve Output (SFTP)

Output response retrieval will always be SFTP (fromADP folder) once the Get Job Status API's (mentioned in the previous section) "processingStatus" field is marked as "completed" and the HTTP response code of the status API returns an HTTP 200 code. Partner will:

- log in to SFTP to retrieve the response file(s)
- transfer the file from the SFTP site to a destination, and then have to use the valid private key to decrypt the PGP file
- decompress the gzip-compressed response files.

Response Format

- Each new entry/record/row in the response file is delineated with a newline character
- Each field is dilineated with a pipe character, i.e. '|'
- The header row is contained in the response file
- The "uid" field contains the uid value from the input file
- The "responseData" field contains the output verification data that corresponds to the "uid" value provided in the input file and is a JSON encoded string

uid	responseData

db214827-825b-447d-9d27-088b9079ed10	**{"governmentID": {"id": "xxx-xx-xxxx", ...}**
...	...

Response Body

Response Dictionary

systemRefreshDate	date of latest data refresh
employerName	employer name
legalEntityID	employer legal ID, i.e. federal employer ID number (FEIN)
employerAddress	address of employer
workerStatusCode	employee status (A - Active, T - Terminated, L - Leave)
workLevelCode	employment type - free-text string (e.g., Full-time, part-time, regular, temp, contractor)
positionTitle	job title
latestHireDate	date the employee was most recently hired at company (This can be interepreted as original hire date unless the employee had left the company and returned, in which case, the latestHireDate only reflects the return date)
terminationDate	date the employee is no longer at company (This can be voluntary or involuntary termination. This will will be null if the employee is still with teh company)
standardHours	hours entered into payroll system by employer
remunerationSummary	YTD breakdown that is only returned with input parameter selected
baseRemunerationAmount	YTD Base
remunerationTypeCode: Bonus	YTD Bonus
remunerationTypeCode: Overtime	YTD Overtime
remunerationTypeCode: Other	YTD Other
totalAnnualRenumerati	YTD Gross

onAmount	
netPayYTDAmount	YTD Net
payDate	date of pay
payPeriod: startDate	starting date of pay period
payPeriod: endDate	ending date of pay period
grossPayAmount	gross pay for this pay period
netPayAmount	net pay for this pay period
basePayAmount	base pay for this pay period
overtimePayAmount	overtime pay for this pay period
bonusPayAmount	bonus pay for this pay period
otherPayAmount	other pay for this pay period
payPeriodHours	total hours for this pay period (may differ from standardHours)
remunerationBasisCode	pay description (Salary, Daily, Hourly)
payCycleCode	pay frequency • code (corresponds to the name, respectively, as in M = Monthly, S = Semimonthly, etc.) - M, S, B, W, D, BO, BE, Q, SA, A, 2, 4, 5 ◦ name - Monthly, Semimonthly, Biweekly, Weekly, Daily, Quarterly, Semiannual, Annual, Every 2.6 wks, Every 4 wks, Every 5.2 wks
basePayRate	regular pay rate
payUnitType	unit of regular pay rate (Pay Period, Day, Hour)

Response Schema

```
{
  "employmentHistory": [
    {
      "systemRefreshDate": "date",
      "employerName": "string",
      "legalEntityID": {
        "legalEntityID": "string"
      },
      "employerAddress": {
        "lineOne": "string",
        "lineTwo": "string",
        "lineThree": "string",
        "cityName": "string",
        "subdivisionCode": {
          "name": "string"
        },
        "countryCode": "string",
        "postalCode": "string"
      },
      "workerStatusCode": {
        "code": "string",
        "name": "string"
      },
      "workLevelCode": {
        "code": "string",
        "name": "string"
      },
      "positionTitle": "string",
      "latestHireDate": "date",
      "terminationDate": "string",
      "standardHours": integer,
      "remunerationSummary":
        {
          "baseRemunerationAmount": {
            "amount": number,
            "currencyCode": "string"
          },
          "additionalRemunerations": [
            {
              "remunerationTypeCode": "string",
              "remunerationAmount": {
                "amount": number,
                "currencyCode": "string"
              }
            }
          ],
          "totalAnnualRemunerationAmount": {
            "amount": number,
            "currencyCode": "string"
          },
          "netPayYTDAmount": {
            "amount": number,
            "currencyCode": "string"
          }
        },
      "latestPayment":
        {
          "payDate": "date",
          "payPeriod": {
            "startDate": "date",
            "endDate": "date"
          },
          "payPeriodHours": number,
          "payAmount": {
            "grossPayAmount": {
              "amount": number,
              "currencyCode": "string"
            },
            "netPayAmount": {
              "amount": number,
              "currencyCode": "string"
            },
            "basePayAmount": {
              "basePay": {
                "amount": number,
                "currencyCode": "string"
              }
            },
```

```
            "overtimePayAmount": {
              "overtimePay": {
                "amount": number,
                "currencyCode": "string"
              }
            },
            "bonusPayAmount": {
              "bonusPay": {
                "amount": number,
                "currencyCode": "string"
              }
            },
            "otherPayAmount": {
              "otherPay": {
                "amount": number,
                "currencyCode": "string"
              }
            }
          },

          "remunerationBasisCode": {
            "name": "string"
          },
          "basePayRate": {
            "amount": number,
            "currencyCode": "string"
            "payUnitType": "string"
          },
          "payCycleCode": {
            "code": "string",
            "name": "string"
          },
          "deductions": [
            {
              "deductionCode": {
                "name": "string"
              },
              "deductionAmount": {
                "amount": number,
                "currencyCode": "string"
              }
            }
          ]
        }
    }
  ]
}
```


Watchouts

- The retention period / lifespan of files on an SFTP site is 2 weeks from the onset of the initial file delivery time. After the retention period is up, the files will be purged
- The partner may read each file a maximum of 99 times. Once the limit is reached, the file will be inaccessible
- If a timeout occurs, partner will have to retry the download from the beginning
- Based on the size of the output file, the matched response file may be split into multiple files. The current limit is a compressed output file size of 5gb, which translates to approximately 3M matched records.
- Any duplicate records in the output file will remain as is, there will be no de-duplication within the processing of the file. The ordering of records is not guaranteed to be maintained

Q&A

How do you retrieve job information?

- Job information, such as the job status, number of invalid records, number of matched records, etc. will be available via a job status HTTP endpoint. The ingestor will have to poll for this endpoint in order identify when a job has completed processing. See Section 2.

How can we recover failed transfers?

- If a file download fails, the entire file will have to be re-downloaded. There is a max file read limit that is set to 99, so if there are 99 read operations performed on a file, it will become inaccessible. Additionally, any file on the SFTP site will be purged 2 weeks from the ingestion time. Once the files are purged, if the partner wants to retrieve results, they must submit a new job, which will count as a new transaction.

What endpoint URL and protocol should we use for SFTP ingestion?

- Please use filetransfer1.adp.com as your host on port 22.

What happens if a job status is stuck in "received" status?

- There are multiple reasons why a job might be stuck in "received" status. One reason is that the file is still being processed/queue'd for processing and the output files have not been generated. A rare case is that the file is unable to be processed. Reach out to your ADP product contacts if you have any concerns about a particular job.

What happens if a particular job is not found, even after submitting a job via SFTP and waiting the required 15 minute time period for a job status to be populated?

- First, check to make sure the Job ID is correct. If submitted via API, the Job ID is the GUID returned within the response payload of the submit API, and should be used to query a job status. If the job was submitted via SFTP, the filename (minus the extension) should be used as the "job_id" field within the status API.
- If any of the following requirements are not met, the job status may not be populated and thus the job will not be processed:
    - The filename of the submitted file is not unique (i.e. a previously submitted file contains the same name as the newly submitted file), or does not contain a ".csv.gz.pgp" extension
    - The corresponding JSON file containing the metadata of the job must also be submitted via SFTP and must contain the same filename as the file containing employee records, with the following additional requirements:
        - This file must not be GZIP compressed
        - This file must not be OpenPGP encrypted
        - The filename must contain a ".json" extension
    - If the JSON file is not processed successfully for any other reason, i.e. not properly formatted, contains missing fields, or invalid field values, the job may not be processed successfully.