Guide

ADP DataCloud Identity, Employment and Income Verification API

Published on
Jun 16, 2021 10:11AM

Last modified
Jun 27, 2023 12:30AM

ADP Copyright Information

These materials may not be reproduced in any format without the express written permission of ADP, Inc. ADP provides this publication "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. ADP is not responsible for any technical inaccurancies or typographical errors which may be contained in this publication. Changes are periodically made to the information herein, and such changes will be incorporated in new editions of this publication. ADP may make improvements and/or changes in the product and/or the programes described in this publication.

Published on
Jun 16, 2021 10:11AM

Last modified
Jun 27, 2023 12:30AM

Table of Contents

Chapter 1

Onboarding Steps

- Summary
- Step 1: Introduction and Overview
- Step 2: Business Term Sheet
- Step 3: Sandbox Agreement
- Step 4: Letter of Intent (LOI)
- Step 5: Development Credentialing
- Step 6: FCRA Questionnaire
- Step 7: Global Security Organization (GSO), Third-Party Risk Assessment (TPRA), & Privacy Reviews
- Step 8: Master Services Agreement (MSA) & Statement of Work (SOW)
- Step 9: Production Credentialing
- Step 10: Billing / Invoicing Set Up

Chapter 2

API Overview

- Summary
- Identity Verification
- Employment Verification
- Employment & Income Verification (YTD)
- Employment & Income Verification (Detailed)
- v1 API Overview
- v2 API Overview

Chapter 3

Recommendations API

- Overview
- Batch Process
  - 1. Submit Job
    - 1a. Input file in CSV format
    - 1b. Config file in JSON format
    - 1c. Submitting the files
  - 2. Get Job Status (API) (swagger spec)
    - Input
    - Response Returns
    - Example Responses

- **3. Retrieve Output (SFTP)**
  - **Response Format**
- **Real-Time Process**
  - **1. Set up SSL Certificate**
  - **2. Request Authentication Token**
  - **3. Make API Call**
- **Response Body**
  - **Response Dictionary**
  - **Response Schema**
  - **Watchouts**
- **Q&A**

Chapter 4

Batch Processes

- **Overview**
- **PGP Encryption - Public Encryption Keys**
- **1. Submit Job**
  - **1a. Input file in CSV format**
  - **1b. Config file in JSON format**
  - **1c. Submitting the files**
- **2. Get Job Status (API) (swagger spec)**
  - **Input**
  - **Response Returns**
  - **Example Responses**
  - **Additional Notes**
- **3. Retrieve Output (SFTP)**
  - **Response Format**
  - **Watchouts**
- **Q&A**

Chapter 5

API Rate Limits

- **Rate Limits**

Chapter 6

API Production Support

- **Summary**
- **Emergency Connectivity Issues**
- **Data Disputes**
- **General Questions**

Chapter 1

Onboarding Steps

Summary

The purpose of this page is to share the necessary steps to onboard the ADP DataCloud Employment and Income Verification APIs from development to production. It serves as a guide to inform what are the major milestones and required steps in the journey, so that you may best prepare to bring your integration to market.

Pre-requisite: before any of the below steps can be taken, the following 2 pre-requisites must be completed:

1. Register your organization via our partner program form, if not done so previously
2. Sign the mutual non-disclosure agreement (MNDA). The MNDA will be provided by your ADP business development contact. The information in this guide is ADP proprietary and Confidential Information.

Step 1: Introduction and Overview

With the MNDA signed, you'll now be able to get further acquainted with our API service. Typically, this involves arranging a call between ADP and your teams to dive into the details.

Step 2: Business Term Sheet

As part of the business development effort, ADP will present a business term sheet. The business term sheet is a mostly nonbinding agreement setting forth the basic terms and conditions under which a partnership alliance or other business relationship will be made. It serves as a template to develop more detailed legally binding documents (MSA/SOW). The term sheet also helps ADP understand the different use cases and market verticals your organization wishes to use ADP data for.

Step 3: Sandbox Agreement

Your ADP business development contact will then provide an API Sandbox Agreement to be signed. The ADP API Sandbox Agreement allows the creation of developer credentials so that your team can test the APIs using dummy test data and begin development.

Step 4: Letter of Intent (LOI)

Your ADP business development contact will also provide an LOI, which is a mostly non-binding agreement that records the nonbinding intention of ADP and your organization to move forward with discussions and potential negotiations related to formalizing a partnership or other business relationship.

Step 5: Development Credentialing

Once the Sandbox Agreement and LOI are signed, your developer account can be created (you will receive an automated email with login details) and you will be able to have your development credentials fully set up (via the Partner Self Service portal: **adpapps.adp.com/self-service**). The Partner Self Service Portal is a convenient place where you can view documentation, generate SSL Certificates, and generate development credentials. You can view the API specification, and even download a Postman Collection to make calling our APIs as easy as possible.

In addition, your ADP Technical Advisor will help set up the following:

1. Set up a secure file sharing site to which onboarding documents can be shared, such as dummy test inputs that provide different types of data responses to depict various scenarios, and other supporting documentation to help your team with development.
2. Set up a private Slack Channel between development teams to answer any technical questions related to the integration. Your ADP Technical Advisor will provide a link to your development team to join Slack.
3. Finish the setup of your development credentials so that you may try out the APIs for development purposes.

Development credentials are composed of a client ID and client secret. In addition to the client ID and client secret, you must generate a mutual SSL certificate (via the Partner Self-Service portal) in order to successfully connect to our API endpoints.

Step 6: FCRA Questionnaire

As part of the legal review process, potential partners are required to complete the FCRA Questionnaire to help ADP gauge your organization's position on and compliance with potentially applicable regulatory requirements. This questionnaire will be provided by your business development contact. The ADP legal team will also review any use case, process flow or interactive demonstrations to gain understanding of your services and data flows. It is helpful if you can provide any documentation or demos of how ADP data will be used in your product or services.

Step 7: Global Security Organization (GSO), Third-Party Risk Assessment (TPRA), & Privacy Reviews

All partners of ADP are required to undergo a formal security and risk assessment, faciliated by our GSO team. Partners will be asked to complete required security and risk questionnaires. This process can take 4-8 weeks to complete. Your ADP business development contact will help kick off this process.

The privacy review process includes a questionnaire used by the Global Data Privacy and Governance team to identify and evaluate privacy risks associated with the processing of personally identifiable information/personal data (PI).

Step 8: Master Services Agreement (MSA) & Statement of Work (SOW)

At this stage, ADP will provide an MSA and SOW for review. Once the MSA/SOW is signed by both parties, production credentials can be fully registered and used.

Step 9: Production Credentialing

Once the MSA/SOW is signed, production credentials can be fully registered and set up. The production credentials are a new set of client ID and client secret which you can generate from the Partner Self-Service Portal, however they will not work until the MSA/SOW are signed. When the production credentials are fully registered by the ADP Technical Advisor after the MSA/SOW is signed, you will gain access to live production data by using those production credentials. The existing mutual SSL certificate can be used with production credentials.

Step 10: Billing / Invoicing Set Up

Finally, your ADP Technical Advisor will ask for the contact info of your lead business development member, to whom invoices and billing will be addressed to. ADP will then set up an account for that business development member and send an invitation to login to finish registration. Afterwards, the billing/invoicing will be set up.

The Partner Self Service Portal also contains a link to your production API Usage. Here, you can quickly view how many API calls were made each month, including the ability to download all the transctions for each month.

Chapter 2

API Overview

Summary

The ADP DataCloud Employment and Income Verification APIs are composed of 5 endpoints, returning varying levels of data fit for different types of use cases. In this chapter, you'll find an overview of the endpoints and data they return, so that it may help you choose the right endpoint that best serves your use cases.

Identity Verification

basic/identity **API** - Obtains simple signal match (true or false) to indicate whether a person exists in the ADP database, for confirming their identity

Example Use Cases

- Account creation (i.e. online banking, cell phone account, etc.) – confirm an individual's identity before new accounts are opened by checking if ADP has data relating to the individual

Employment Verification

basic/employment **API** - Obtains employment history for an individual (e.g. verification of employment)

Example Use Cases

- Pre-employment screening – check if individual is employed, and obtain history of employment

Employment & Income Verification (YTD)

enhanced/employment **API** - Obtains employment history + annual pay summary (YTD for current year and up to past 3 full years) for an individual

Example Use Cases

- Pre-employment screening for certain jobs (check jurisdiction restrictions) – check if individual is employed, obtain history of employment, and YTD income
- Credit card – verification of employment and basic income summary to help confirm stated income and employment
- Offers or Pre-qualification – to help target offers and pre-qualifications to potential customers

---

Employment & Income Verification (Detailed)

basic/employment-income **API** - Obtains employment history, annual pay summary (YTD), and income history (pay period details) for the current year to date plus the last 3 full years, if available. Best used for mortgage, auto loans, personal loans where income history is needed, but deductions are not necessary.

Example Use Cases

- Auto loans, tenant screenings, personal loans, mortgage – check employment history and detailed pay period history for up to 3 years

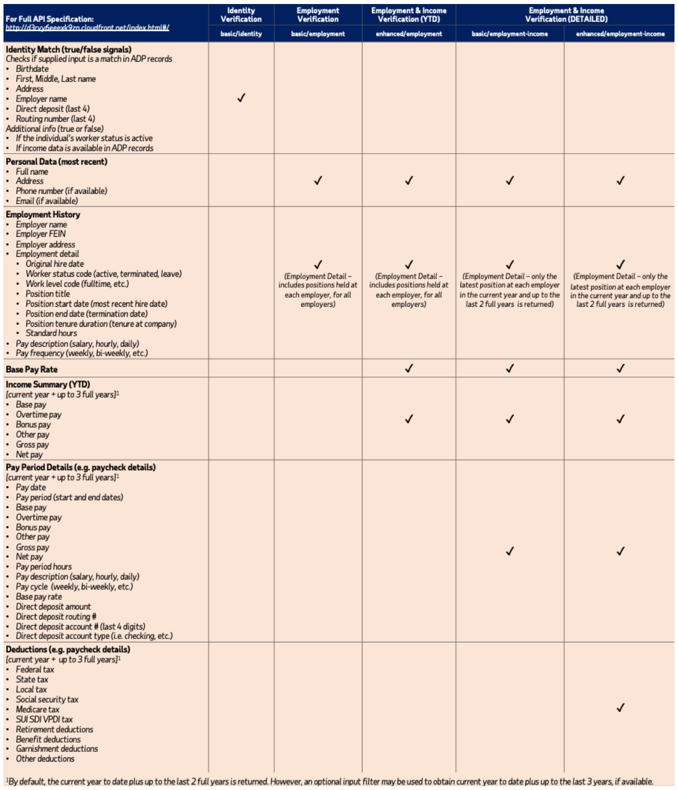
enhanced/employment-income **API** - Obtains everything in basic/employment-income API, PLUS payroll deductions information

Example Use Cases

- Mortgage, government services – check employment history and detailed pay period history (with deductions information) for up to 3 years, for a more comprehensive data set

---

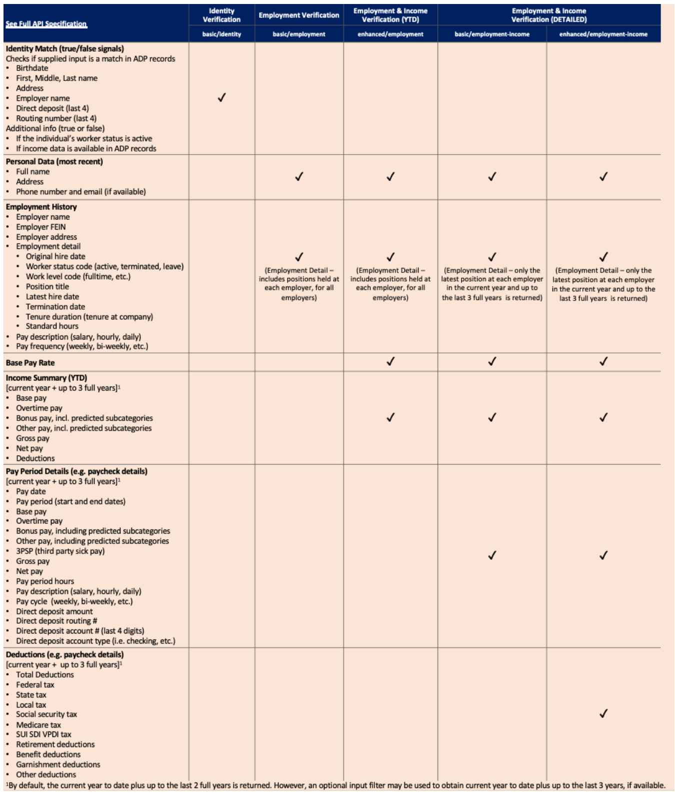
v1 API Overview



For Full API Specification: http://d3rvy6eeexk9zn.cloudfront.net/index.html#/	Identity Verification	Employment Verification	Employment & Income Verification (YTD)	Employment & Income Verification (DETAILED)	
	basic/identity	basic/employment	enhanced/employment	basic/employment-income	enhanced/employment-income
Identity Match (true/false signals) *Checks if supplied input is a match in ADP records* • *Birthdate* • *First, Middle, Last name* • *Address* • *Employer name* • *Direct deposit (last 4)* • *Routing number (last 4)* *Additional info (true or false)* • *If the individual's worker status is active* • *If income data is available in ADP records*	✓				
Personal Data (most recent) • *Full name* • *Address* • *Phone number (if available)* • *Email (if available)*		✓	✓	✓	✓
Employment History • *Employer name* • *Employer FEIN* • *Employer address* • *Employment detail* • *Original hire date* • *Worker status code (active, terminated, leave)* • *Work level code (fulltime, etc.)* • *Position title* • *Position start date (most recent hire date)* • *Position end date (termination date)* • *Position tenure duration (tenure at company)* • *Standard hours* • *Pay description (salary, hourly, daily)* • *Pay frequency (weekly, bi-weekly, etc.)*		✓ *(Employment Detail – includes positions held at each employer, for all employers)*	✓ *(Employment Detail – includes positions held at each employer, for all employers)*	✓ *(Employment Detail – only the latest position at each employer in the current year and up to the last 2 full years is returned)*	✓ *(Employment Detail – only the latest position at each employer in the current year and up to the last 2 full years is returned)*
Base Pay Rate			✓	✓	✓
Income Summary (YTD) *[current year + up to 3 full years][1]* • *Base pay* • *Overtime pay* • *Bonus pay* • *Other pay* • *Gross pay* • *Net pay*			✓	✓	✓
Pay Period Details (e.g. paycheck details) *[current year + up to 3 full years][1]* • *Pay date* • *Pay period (start and end dates)* • *Base pay* • *Overtime pay* • *Bonus pay* • *Other pay* • *Gross pay* • *Net pay* • *Pay period hours* • *Pay description (salary, hourly, daily)* • *Pay cycle (weekly, bi-weekly, etc.)* • *Base pay rate* • *Direct deposit amount* • *Direct deposit routing #* • *Direct deposit account # (last 4 digits)* • *Direct deposit account type (i.e. checking, etc.)*				✓	✓
Deductions (e.g. paycheck details) *[current year + up to 3 full years][1]* • *Federal tax* • *State tax* • *Local tax* • *Social security tax* • *Medicare tax* • *SUI SDI VPDI tax* • *Retirement deductions* • *Benefit deductions* • *Garnishment deductions* • *Other deductions*					✓

[1] By default, the current year to date plus up to the last 2 full years is returned. However, an optional input filter may be used to obtain current year to date plus up to the last 3 years, if available.

v2 API Overview



See Full API Specification	Identity Verification	Employment Verification	Employment & Income Verification (YTD)	Employment & Income Verification (DETAILED)	
	basic/identity	basic/employment	enhanced/employment	basic/employment-income	enhanced/employment-income
Identity Match (true/false signals) Checks if supplied input is a match in ADP records • Birthdate • First, Middle, Last name • Address • Employer name • Direct deposit (last 4) • Routing number (last 4) Additional info (true or false) • If the individual's worker status is active • If income data is available in ADP records	✓				
Personal Data (most recent) • Full name • Address • Phone number and email (if available)		✓	✓	✓	✓
Employment History • Employer name • Employer FEIN • Employer address • Employment detail • Original hire date • Worker status code (active, terminated, leave) • Work level code (fulltime, etc.) • Position title • Latest hire date • Termination date • Tenure duration (tenure at company) • Standard hours • Pay description (salary, hourly, daily) • Pay frequency (weekly, bi-weekly, etc.)		✓ (Employment Detail – includes positions held at each employer, for all employers)	✓ (Employment Detail – includes positions held at each employer, for all employers)	✓ (Employment Detail – only the latest position at each employer in the current year and up to the last 3 full years is returned)	✓ (Employment Detail – only the latest position at each employer in the current year and up to the last 3 full years is returned)
Base Pay Rate			✓	✓	✓
Income Summary (YTD) [current year + up to 3 full years][1] • Base pay • Overtime pay • Bonus pay, incl. predicted subcategories • Other pay, incl. predicted subcategories • Gross pay • Net pay • Deductions			✓	✓	✓
Pay Period Details (e.g. paycheck details) [current year + up to 3 full years][1] • Pay date • Pay period (start and end dates) • Base pay • Overtime pay • Bonus pay, including predicted subcategories • Other pay, including predicted subcategories • 3PSP (third party sick pay) • Gross pay • Net pay • Pay period hours • Pay description (salary, hourly, daily) • Pay cycle (weekly, bi-weekly, etc.) • Direct deposit amount • Direct deposit routing # • Direct deposit account # (last 4 digits) • Direct deposit account type (i.e. checking, etc.)				✓	✓
Deductions (e.g. paycheck details) [current year + up to 3 full years][1] • Total Deductions • Federal tax • State tax • Local tax • Social security tax • Medicare tax • SUI SDI VPDI tax • Retirement deductions • Benefit deductions • Garnishment deductions • Other deductions					✓

[1] By default, the current year to date plus up to the last 2 full years is returned. However, an optional input filter may be used to obtain current year to date plus up to the last 3 years, if available.

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.

The Recommendations API is available via the Batch process or Real-Time process, depending on what's authorized in your contract. You can refer to the chapter on Batch Processing for more information.

Batch Process

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. **Refer to the Response Body section below for full sample response.**

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

...	...

Real-Time Process

1. Set up SSL Certificate

Once the Sandbox agreement or MSA has been signed, you will be provided corresponding credentials to the ADP Marketplace (dev/test or prod credentials). In the Self-Service Portal, go to Request a Certificate and generate a new SSL Certificate.

2. Request Authentication Token

When ready to start using either the IAT or Prod environment, the Partner will have to retrieve the authentication token. The token lasts 60 minutes, but best practice is to refresh token before the full 60 minutes. The Recommendations API requires the encoding of the token to be be a JWT bearer token instead of the default Oauth bearer token. Thus, the Partner mustexplicitly request this token type from the authorization endpoint with the "Accept" header:

Request

```
curl -X POST \ 'https://accounts.adp.com/auth/oauth/v2/token?grant_type=client_credentials&client_id=<client_id>&client_secret=<client_secret>' \
--header 'Accept: application/at+jwt'
```

Response

```
{ "access_token": "string2", "expires_in": "3600", "token_type": "Bearer" }
```

3. Make API Call

You may now make your API call.

Sample Input Request:

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

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
totalAnnualRenumerationAmount	YTD Gross
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.

Chapter 4

Batch Processes

Overview

For large volume transactions that don't require real-time responses (for example, if there is a need to verify employment & income for a million individuals), we have an asynchronous process that can facilitate these bulk transactions as an alternative to funneling them through our real-time APIs. There are 3 main steps to this Batch process:

1. **Submit job** - Submit a job via SFTP by uploading an input csv file along with a config json file to a secure location. The input must come in a pair (csv and json).
2. **Get job status** - Ping our job status API to find out the status of the job, and whether the job is complete
3. **Retrieve output** - Retrieve the output via SFTP.

Depending on the size of the input, processing of jobs can take up to 48hrs.

Important

PGP Encryption - Public Encryption Keys

- **Input:** The csv input file **must** be PGP encrypted. Because ADP will be reading these files, ADP will be supplying the public PGP encryption key to the partner.
- **Output:** Output files **must** be PGP encrypted. Because the partner will be reading these files, the partner must supply their public PGP encryption key to ADP.

Refer to the OpenPGP standard: https://datatracker.ietf.org/doc/html/rfc4880.

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) - For the full list of available filters, refer to Swagger
    - "paidWithin" (string-enum)
        - Valid values: "30", "90", "180", "365", null
        - null is equivalent to the filter not being used
    - "workerStatus" (string-enum)
        - Valid values: "Active", "Terminated", "All", null
        - "Active" signifies workerStatus being either Active or Leave
        - null is equivalent to the filter not being used
    - "payUnitTypeIndicator" (boolean)
        - Value values: true, false, null
        - null is equivalent to the filter not being used

```
{
    {
    "requestorInfo": {
      "marketSegmentCode": "Batch",
      "verifierName": "ABC Bank"
    },
    "associateConsentReceivedIndicator": true,
    "verificationType": "enhanced-employment-income-standard-v2",
    "responseFilters": {
        "paidWithin": "30",
        "workerStatus": "Active",
        "payUnitTypeIndicator": 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", ...}
...	...

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.

Chapter 5

API Rate Limits

Rate Limits

- 50 concurrent calls
- 3000 calls per minute

Chapter 6

API Production Support

Summary

In this chapter, we'll outline our API support process, and provide details on how to reach our support teams once you have gone live (i.e. production). In general there are 3 types of support:

- **Emergency connectivity issues:** these are situations when there is loss of connectivity with our API service
- **Data disputes:** these are dispute questions that may come from your end-consumers, which may require further investigation with ADP
- **General questions:** any non-urgent questions related to the service

We will go through each of these categories and how to reach our support teams

Emergency Connectivity Issues

While we maintain robust health monitoring and alerting mechanisms, should issues arise such as a loss of connectivity, please email our 24-hour support:

dm.api-support@adp.com

This email initiates a process which will notify our internal engineering and support teams via phone, text and email so that we respond in a timely manner. Our technical team will respond shortly and reach out if we have any further questions

Data Disputes

When end-consumers file a data dispute, as a first step we recommend to direct end-consumers to first check with their employer to see if the issue can be resolved with the employer. For example, if there is a data discrepancy, end-consumers should first check with their employer to see if this is not a mistake made by the employer.

If the end-consumer has checked with their employer, and wants to continue with the data dispute and it is determined that further investigation is necessary with ADP, please send an email to our ADP Employment and Income Verification Consumer Support team at DMConsumerSupport@adp.com with the following details:

- Partner's company name
- Partner support contact name
- Partner support email contact
- Partner support phone contact
- ADP-CorrelationID (the unique identifier in the header of the API response serviced for the individual)
- Date and time of API request
- Dispute reason

Once the email is sent, you will receive an automated acknowledgement reply with a ticket number so you can use for future reference to track the case. An ADP service representative may reach out (referencing the ticket number) if there are any additional questions, and will respond when the issue is resolved.

Our response time for data disputes is within 10 business days. Our service representative will reply with the resolution, and provide any further comments as necessary.

General Questions

Any general questions or inquiries related to the service should be directed to your ADP product development or business development contacts.