

Chapter 1

About this API

from Payroll Data Input API Guide for ADP Vantage
HCM Guide

Published on
Mar 10, 2020 5:18PM

Last modified
Feb 16, 2022 3:28PM





ADP Copyright Information

ADP, the ADP logo, and Always Designing for People are trademarks of ADP, Inc.

Windows is a registered trademark of the Microsoft Corporation.

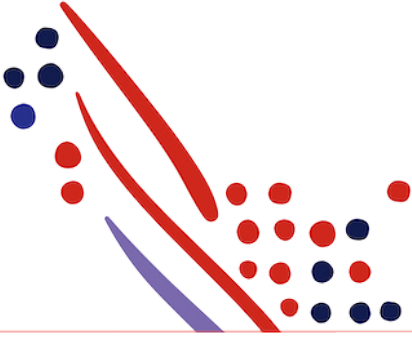
All other trademarks are the property of their respective owners.

Copyright © 2022 ADP, Inc. ADP Proprietary and Confidential - All Rights Reserved. These materials may not be reproduced in any format without the express written permission of ADP, Inc.

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 inaccuracies 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 programmes described in this publication.

Published on
Mar 10, 2020 5:18PM

Last modified
Feb 16, 2022 3:28PM



Chapter Contents

Chapter 1

About this API

Summary

What's New in this Guide?

July 2020

May 2020

Process Overview

Required Setup Steps

Pay Data Entry Views Setup in ADP Vantage HCM

Pay Data Types Handled by the Payroll Data Input API

Postman Collection

Chapter 1

About this API

Summary

ADP Vantage HCM enables users to create a Pay Data batch for a payroll cycle to include the following:

- Hours and Earnings
- Deductions
- Memos
- Taxes
- Temporary Cost Numbers
- Temporary Department Numbers
- Allocations

The Payroll Data Input Application Programming Interface (API) enables you to build a data connector application for clients to transmit data sourced in your application to a specific Pay Data batch in ADP Vantage HCM.



Note

(Developer Note) The Payroll Data Input API call is an **Asynchronous call**, which means when you try to add a Pay Data batch through the API, it throws a **202 Accepted** as a status code along with another uniform resource locator (URL) provided at the end of the response payload. On hitting this URL, it will throw a **200 OK** as a status code along with a response payload, which displays a generic message stating if the Import Pay Data process is passed or failed.

- If the following message is displayed: **The 'Import Pay Data' process is successful.**, it means the pay data batch is successfully added into ADP Vantage HCM on the **Pay Data Entry** page.
- If the following message is displayed: **The 'Import Pay Data' process failed.**, it means there is a mistake in the request payload and the pay data batch didn't get added into ADP Vantage HCM.

What's New in this Guide?

In this section, revisions to the Payroll Data Input API and this guide will be announced.

July 2020

Added a Note under [Chapter 2 - Use Case: Adding Entries in a Pay Data Batch](#).

May 2020

Updated the **Data Dictionary** within the **Batch and Employee Information** section in [Chapter 2 - Adding Entries in a Pay Data Batch](#).

Process Overview

The following table shows an illustration of how your data connector application would be used by a client.

	Actor	Task Description
1	Payroll practitioner	Selects the payroll group code, employees, and payroll file numbers to be imported.
2	Payroll practitioner	Enters the Pay Data inputs and initiating the Pay Data import.
3	Your application	Automatically creates a new batch and imports Pay Data in ADP Vantage HCM.
4	Payroll practitioner	Reviews the import result on the Pay Data Entry page in the ADP Vantage HCM user interface (UI) by selecting Process > Payroll > Pay Data Entry .
5	Payroll practitioner	Takes actions to balance and adjust, as needed.

Required Setup Steps

The Payroll Data Input API supports all the following codes:

- Earnings
- Deductions
- Memos
- Taxes
- Cost Numbers
- Payroll Department Numbers
- Allocations

Clients need to make sure the codes used by your application are added and configured in the ADP Vantage HCM UI by selecting **Setup > Field Definitions & Mappings > Setup Information > Payroll**.

The following table lists the records in your data connector application and how they correspond to the fields on the **Setup Information** page under **Payroll** in the ADP Vantage HCM UI.

Records	Fields in ADP Vantage HCM UI
Earning Codes	Earnings
Deduction Codes	Deductions
Taxes	Tax Setup
Memo Codes	Memo
Cost Number Codes	Cost Number
Department Codes	Payroll Departments
Shift Qualifier Codes	Shift Qualifier
Rate codes	Additional Rates

Pay Data Entry Views Setup in ADP Vantage HCM

The column names appearing on the **Pay Data Entry** page when a pay data batch is created should be set up in the **Pay Data Entry Views** within in the ADP Vantage HCM UI by selecting **Process > Payroll > Pay Data Entry Views**.

The following columns should be present under the **Selected Columns** of the **Pay Data Entry Views** page. If not present, move the fields under **Available Columns** to **Selected Columns**.

- File Number (File #)
- FLSA Workweek (Workweek)
- Pay Number (Pay #)
- ALA Occurrence (ALA#)
- Rate Code
- Shift
- Temp Cost Number (Temp Cost #)
- Temp Department (Dept)
- Employee Name (Name)
- Hours/Earnings 1(Reg Erns)
- Hours/Earnings 2(O/T Erns)
- Hours/Earnings 3(Reg Hrs)
- Hours/Earnings 4(O/T Hrs)
- Temp Rate
- Deductions
- Memos
- Taxes Federal (Federal Tax)
- Taxes Local 2 (Local 2 Tax)
- Taxes State 2 (State 2 Tax)
- Taxes State (State Tax)
- Taxes Local (Local Tax)
- State2 Tax Code (State 2 Tax Cd)
- State Tax Code (State Tax Cd)
- Other Period Begin Date
- Other Period End Date



Note

When the </events/payroll/v1/pay-data-input.add> API is used to add a Pay Data Batch, the **Pay Data Entry** view will be automatically set up according to the input fields provided through the Payroll Data Input API.

Pay Data Types Handled by the Payroll Data Input API

The following table lists the pay data transmitted through the Payroll Data Input API to ADP Vantage HCM.

Type	Common Uses	Schema Location
Regular and Overtime Hours and Earnings	Sends regular and overtime earnings to ADP Vantage HCM.	/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/earningInputs
Coded Hours and Earnings	Sends client-defined and other earnings to ADP Vantage HCM.	/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/earningInputs
Deductions	Sends deductions to ADP Vantage HCM, including out of box deductions and client-defined deductions.	/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/ deductionInputs

Allocation Position	Sends the allocation position details to ADP Vantage HCM. Specifies the number of the allocations associated with an entry. This is for Automatic Labor Allocation (ALA).	/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/payAllocation
Cost Number	Sends the cost number details to ADP Vantage HCM. Specifies the temporary cost number.	/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/payAllocation
Department Number	Sends the department details to ADP Vantage HCM. Specifies the temporary department number.	/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/payAllocation
Memos	Sends the memo details to ADP Vantage HCM.	/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/memoInputs
Taxes(Federal,State & Local)	Sends Federal, State (Lived In and Worked In), Local (Lived In and Worked In) taxes to ADP Vantage HCM.	/payDataInput/payeePayInputs/payrollProfilePayInputs/payInputs/taxInputs

Postman Collection

Postman allows you to import a collection of APIs, created by others, so you can try them out. For more information on Postman, see [Making Your First API Call Using Postman](#).

To download API collections for the Payroll Data Input API from the ADP GitHub library and import them to Postman, go to [Payroll Data Input API Postman Collection](#).