



Guide

Deduction API Guide for ADP Vantage HCM

Published on
Mar 31, 2020 4:09PM

Last modified
Nov 02, 2022 9:09AM



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 31, 2020 4:09PM

Last modified
Nov 02, 2022 9:09AM

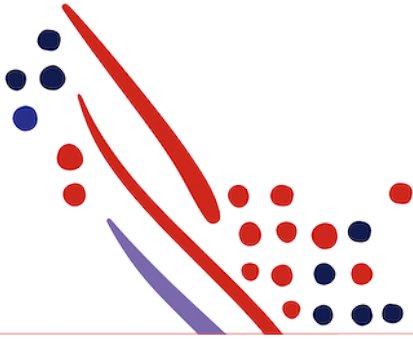


Table of Contents

Chapter 1

About this API

Summary

What's New in this Guide?

October 2022

September 2021

July 2020

June 2020

Employee Deductions Tasks

About Employee Deductions

Viewing the Deductions Summary

Required Setup Steps

Postman Collection

Chapter 2

Use Case: Retrieving Payroll Instructions (Payroll Instruction API)

Use Case Description

API Usage

Application Scope

Supported Actors

Request Header Parameters

Other Supported Parameters

Sequence of Interactions

Data Dictionary

Responses

Chapter 3

Use Case: Returning a List of Available Deduction Codes

Use Case Description

API Usage

Application Scope

Supported Actors

Request Header Parameters

Supported OData Query Options

Other Supported Parameters

Sequence of Interactions

Data Dictionary
Responses

Chapter 4

Use Case: Starting a General Deduction Instruction

Use Case Description
API Usage
Application Scope
Supported Actors
Request Header Parameters
Supported OData Query Options
Other Supported Parameters
Sequence of Interactions
Data Dictionary
Responses

Chapter 5

Use Case: Changing a General Deduction Instruction

Use Case Description
API Usage
Application Scope
Supported Actors
Request Header Parameters
Supported OData Query Options
Other Supported Parameters
Sequence of Interactions
Data Dictionary
Responses

Chapter 6

Use Case: Stopping a General Deduction Instruction

Use Case Description
API Usage
Application Scope
Supported Actors
Request Header Parameters
Supported OData Query Options
Other Supported Parameters
Sequence of Interactions
Data Dictionary
Responses

Chapter 7

Use Case: Synchronizing Deduction Event Notifications

Chapter 8

Fields Not Supported by this API

Chapter 9

Frequently Asked Questions

Question 1 : How does Goal Amount work for an employee?

Question 2: What is the difference between Goal Amount, Balance, and Amount to Date?

Question 3: If the Deduction is already applied for an employee, how do you adjust the Overall Goal Amount?

Question 4: How do I keep deduction records in sync with ADP Vantage HCM and the My\ [partner\] application?

Chapter 10

Known Issues and Limitations

Issue 1: codeList response doesn't specify if a Deduction Code supports dollar amount or percentage amount

Impacted APIs

Issue Description

Suggested Workaround

Issue 2: GET pay instructions for an Item ID through an API is not supported

Impacted APIs

Issue Description

Suggested Workaround

Issue 3: Deduction Start API doesn't display effectiveDateTime field in response to when it is sent through request payload

Impacted APIs

Issue Description

Suggested Workaround

Issue 4: Deduction start returns success response when there is no effectiveDate field in the request payload but, according to the meta, it's mandatory

Impacted APIs

Issue Description

Suggested Workaround

Issue 5: Additional fields in the GET response which aren't mapped to the ADP Vantage HCM UI

Impacted APIs

Issue Description

Suggested Workaround

Issue 6: Event notifications aren't supported when deductions are invoked through the API

Impacted APIs

Issue Description

Suggested Workaround

Issue 7: API isn't throwing any error if a percentage deduction is added without passing the Unit Code field in the request payload

Impacted APIs

Issue Description

Suggested Workaround

Issue 8: There is no facility to stop the deduction at any required date other than the effectiveDate when it's created

Impacted APIs

Issue Description

Suggested Workaround

Issue 9: STOP API is deleting the whole deduction record instead of stopping the deduction

Impacted APIs

Issue Description

Suggested Workaround

About this API

Summary

The Deduction Application Programming Interfaces (APIs) are used to do the following:

- Start, stop or change the payroll deduction setup for a worker.
- View all deduction instructions of a worker.
- Retrieve and send employee deductions to ADP (commonly used by Benefits Solutions).



Important

Currently, ADP Vantage HCM doesn't support the configuration of special calculations for deductions. However, the user has an option of defining the calculation in a text box while setting-up the deduction codes. This information is passed to ADP service delivery through a Service Request (SR) ticket. The Service Consultant would then manually configure the special calculation for that deduction in AutoPay.

What's New in this Guide?

October 2022

- Updated API endpoints and samples to link to API explorer.

September 2021

- Updated few github links.

July 2020

- Updated the last two conditions of the **Responses** section in [Chapter 4 - Use Case: Starting a General Deduction Instruction](#)
- Updated the last three conditions of **Responses** section in [Chapter 5 - Use Case: Changing a General Deduction Instruction](#)
- Removed the following issues from [Chapter 10 - Known Issues and Limitations](#), since they've been resolved:
 - **Issue 1:** codeList GET API is not working as expected
 - **Issue 4:** Deduction Start API doesn't display effectiveDateTime field in response to when it is sent through request payload
 - **Issue 12:** Deduction Start and Change APIs return 500 Internal Server error instead of 400 Bad Request when an invalid goal amount is passed through the request payload
 - **Issue 13:** Deduction Start, Change, and Stop API's return a 500 Internal Server error instead of a 400 Bad Request when an invalid date format is passed through the request payload
- Rearranged the Issue numbers in order and updated the **Suggested Workaround** under **codeList response doesn't specify if a Deduction Code supports dollar amount or percentage amount**

- Updated the **Required Setup Steps** section in this chapter.
- Updated the **Use Case Description** in [Chapter 3 - Use Case: Returning a List of Available Deduction Codes](#) is updated with an Important notice.
- Updated the **API Usage** section in [Chapter 4 - Use Case: Starting a General Deduction Instruction](#) with filter support for the Start meta.
- Added the **Other Supported Parameters** in [Chapter 4 - Use Case: Starting a General Deduction Instruction](#).
- Updated the table in [Chapter 7 - Use Case: Synchronizing Deduction Event Notifications](#) with **GitHub Sample Responses**.
- Under [Chapter 10 - Known Issues and Limitations](#):
 - Updated **CodeList response doesn't specify if a Deduction Code supports dollar amount or percentage amount** with **Suggested Workaround**.
 - Added **Meta states Unit Code field as optional for percent deductions where it is mandatory**.

Employee Deductions Tasks

- Add employee Deductions
- Change employee Deductions
- Delete employee Deductions

About Employee Deductions

Employee deductions and goal limit are set up to specify an amount to be deducted from an employee's regular pay. You can set up a goal limit to stop the deduction when the limit is reached.

You can also adjust the deducted amount using **Amount to Date** towards the goal limit.



Important

These deductions are non-benefit deductions.

Viewing the Deductions Summary

A summary of the deductions scheduled to be taken from an employee's paycheck is viewed as of a specific date within ADP Vantage HCM by selecting **People > Pay & Taxes > Deductions**.

If you are using ADP Vantage HCM Payroll with an external third-party HR solution, the deductions summary may not be available or used.



Important

There is a restriction for deduction codes which will be supported through the Deduction API. Which means, not all deductions available in the ADP Vantage HCM application are retrieved through the API and not all deductions which are available in the user interface (UI) can be added through the API.

There is a codeList API which will return the list of deduction codes which are supported through the API.

The code list only returns the unused deductions, which can be added. This means the deductions added already will not be returned in the codeList response. For more information, see [Chapter 3 - Use Case: Returning a List of Available Deduction Codes](#).

Required Setup Steps

- Before you can use employee deductions, deduction codes must be set up for the pay group in the ADP Vantage HCM UI by selecting **Setup > Field Definitions & Mappings > Setup Information > Payroll > Deductions**.
- Before trying to retrieve unused deductions for an employee using CodeList Uniform Resource Identifier (URI) mentioned in [Chapter 3 - Use Case: Returning a List of Available Deduction Codes](#), the API user should make sure the **EE Flag** should be enabled for the employee in ADP Vantage HCM.

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 Deduction API from the ADP GitHub library and import them to Postman, go to [Deduction API Postman Collection](#).

Chapter 2

Use Case: Retrieving Payroll Instructions (Payroll Instruction API)

Use Case Description

This use case returns the list of payroll instructions which the requester is authorized to view for a given worker.

The Payroll Instruction API exposes values found in the ADP Vantage HCM UI by selecting **People > Pay & Taxes > Deductions**. You must provide the Associate Object Identifier (AOID) of the desired worker in the AOID Uniform Resource Identifier (URI) parameter.



Important

To get the associate ID, the consumer needs to do a Worker API call.

API Usage

For API related information please refer [Worker Payroll Instructions](#) under API Explorer.

Method	URI	Description
GET	<code>/payroll/v1/workers/{associateoid}/payroll-instructions</code>	Returns all the active deductions for an associate.
GET	<code>/payroll/v1/workers/{associateoid}/payroll-instructions?effectiveDate={YYYY-MM-DD}</code>	Returns the deductions based on the effective date.
GET	<code>/payroll/v1/workers/{associateoid}/payroll-instructions/meta</code>	Returns response schema information.

Application Scope

The canonical URI corresponding to the API needs to be added in the Consumer Application Registry (CAR) for the subscription following which a user can access this API and make successful API calls.

The following canonical needs to be added to your application scope to enable this use case:

**/payroll/payrollManagement/payrollInstructionManagement/workerPayrollInstructionManagement/
workerPayrollInstructions.read**

Supported Actors

Request Parameter roleCode Value	Usage
Practitioner	Retrieves payroll instructions. A practitioner can view, start, stop, or change a deduction. A system user is considered a practitioner.

Request Header Parameters

Parameter Name	Required (Y/N)	Usage	Value	Sample
Accept	Y	For this API, this is the mandatory header parameter	application/json	Accept: application/json

Other Supported Parameters

Parameter Name	Required (Y/N)	Usage	Value	Sample
effectiveDate	N	Fetches the record based on the date passed in the request.	YYYY-MM-DD	effectiveDate:YYYY-MM-DD

Sequence of Interactions

1. Your consumer application makes a request to the ADP API endpoint to GET payroll-instructions for a worker.
2. The ADP API endpoint responds to your consumer application with single worker payroll information.

Data Dictionary

The fields listed in the following table can be found in the ADP Vantage HCM UI by selecting **People > Pay & Taxes > Deductions**.

Schema Location	Field Name in ADP Vantage HCM	Note
/payrollInstructions/itemID	Not Displayed	Indicates the number of employee's positions in a company. For example, if itemID is 0 , the employee has only one position.
/payrollInstructions/generalDeductionInstructions/itemID	Not Displayed	Indicates the itemID is similar to code Value .
/payrollInstructions/generalDeductionInstructions/actions/operationID	Not Displayed	Important: These fields are implemented for MyADP consumers.
/payrollInstructions/generalDeductionInstructions/actions/actionTypeCode	Not Displayed	
/payrollInstructions/generalDeductionInstructions/actions/confirmationRequiredIndicator	Not Displayed	
/payrollInstructions/generalDeductionInstructions/actions/commentAllowedIndicator	Not Displayed	
/payrollInstructions/generalDeductionInstructions/actions/defaultIndicator	Not Displayed	
/payrollInstructions/generalDeductionInstructions/actions/links	Not Displayed	
/payrollInstructions/generalDeductionInstructions/actions/links/href	Not Displayed	
/payrollInstructions/generalDeductionInstructions/actions/links/rel	Not Displayed	
/payrollInstructions/generalDeductionInstructions/actions/links/method	Not Displayed	
/payrollInstructions/generalDeductionInstructions/deductionCode	Not Displayed	
/payrollInstructions/g	CODE	Indicates the deduction code.

eneralDeductionInstructions/deductionCode/codeValue		
/payrollInstructions/generalDeductionInstructions/deductionCode/longName	Not Displayed	Indicates the name of the deduction.
/payrollInstructions/generalDeductionInstructions/deductionRate	Not Displayed	
/payrollInstructions/generalDeductionInstructions/deductionRate/rateValue	AMOUNT/ PERCENT	This schema supports two different fields: AMOUNT and PERCENT . Deductions specific to the amount (dollar) and percentage are set at the client level.
/payrollInstructions/generalDeductionInstructions/deductionRate/currencyCode	Not Displayed	Indicates the currency code of the deduction amount given.
/payrollInstructions/generalDeductionInstructions/deductionStartDate	Effective Date	Indicates the effective date of the deduction.
/payrollInstructions/generalDeductionInstructions/deductionGoal	Not Displayed	
/payrollInstructions/generalDeductionInstructions/deductionGoal/goalID	GOAL #	Indicates a goal number. Valid values are 1-9.
/payrollInstructions/generalDeductionInstructions/deductionGoal/goalBalanceAmount	Not Displayed	
/payrollInstructions/generalDeductionInstructions/deductionGoal/goalBalanceAmount/amountValue	BALANCE	Indicates the balance amount that still needs to be deducted to reach the goal limit.
/payrollInstructions/generalDeductionInstructions/deductionGoal/goalBalanceAmount/currencyCode	Not Displayed	Indicates the currency code of the balance amount.
/payrollInstructions/generalDeductionInstructions/deductionGoal/goalLimitAmount	Not Displayed	
/payrollInstructions/generalDeductionInstructions/deductionGoal/goalLimitAmount/amountValue	GOAL AMOUNT	Indicates the goal limit amount value.

/payrollInstructions/generalDeductionInstructions /deductionGoal/goalLimitAmount/currencyCode	Not Displayed	Indicates the currency code of the goal amount.
/payrollInstructions/payrollRegionCode	Not Displayed	
/payrollInstructions/payrollRegionCode/codeValue	Not Displayed	
/payrollInstructions/payrollRegionCode/longName	Not Displayed	
/payrollInstructions/payrollGroupCode	Not Displayed	
/payrollInstructions/payrollGroupCode/codeValue	Not Displayed	Indicates the payroll group code of an employee.
/payrollInstructions/payrollGroupCode/longName	Not Displayed	Indicates the name of the payroll group to which the associate belongs.
/payrollInstructions/payrollFileNumber	Not Displayed	Indicates the unique payroll file number of an associate.

Responses

You may encounter exceptions outside your common success scenarios. You must account for these exceptions during your initial development.

For more information, see [API Common Exceptions and Tips for Handling](#).

For API related information please refer [Worker Payroll Instructions](#) under API Explorer.

Response Code	Condition	messageTxt	Tips to Handle
200 OK	Retrieves all the payroll deductions successfully without passing the effective date.	NA	
200 OK	Successful response when the past effective date is passed in the request.	NA	
200 OK	Successful response when future effective date is passed in the request.	NA	
204 No Content	GET call when there are no deductions available for an employee.	NA	In this case, there will not be any response.
400 Bad Request	When the AOID is invalid.	"userMessage": {"codeValue": "3-49", "messageTxt": "You do not have the	Make sure you give a valid AOID.

		authorization to access the server. Contact your system administrator. (3,49)\n"	
400 Bad Request	When the effective date is invalid.	"userMessage": {"codeValue": "40-10", "messageTxt": "The date you entered is invalid. Enter a valid date (-As Of Date). (40,10)\n"}	Make sure you are using a valid effective date format.

Chapter 3

Use Case: Returning a List of Available Deduction Codes

Use Case Description

This use case returns the list of all available deduction codes the requester is authorized to view for a given worker.

You must provide the AOID and Pay Group of the desired worker in the URI parameter.



Important

To retrieve all the unused deductions for a particular associate, make sure the **EE Flag** is enabled for the associate in ADP Vantage HCM.

API Usage

For API related information please refer [Worker Payroll Instructions](#) under API Explorer.

Method	URI	Description	GitHub Sample Request Payload
GET	/codelists/payroll/v3/payroll-instruction-management/unused-ess-general-deduction-code-values?effectiveDate={YYYY-MM-DD}&aoid={aoid}&paygroup={paygroup}	Returns all the unused deductions for an associate.	deduction-instructions-codelist-200-response.json

Application Scope

The following canonical needs to be added to your application scope to enable this use case:

/payroll/payrollManagement/payrollInstructionManagement/codeListViewing/codelist.read

Supported Actors

Request Parameter roleCode Value	Usage
Practitioner	Retrieves the payroll codeList instructions as a practitioner. A practitioner can view, start, stop, or change a deduction. A system user is considered a practitioner.

Request Header Parameters

Parameter Name	Required (Y/N)	Usage	Value	Sample
Accept	Y	For this API, this is the mandatory header parameter.	application/json	Accept: application/json

Supported OData Query Options

There are no supported OData Query Options.

Other Supported Parameters

Parameter Name	Required (Y/N)	Usage	Value	Sample
effectiveDate	Y	Lists all the unused deductions effective from this date.	effectiveDate	effectiveDate:2019-06-23
aoid	Y	Lists all the unused deductions under this associate.	associateID	aoid: G3R9MDBS40828RS9
paygroup	Y	Lists all the unused deductions of the associate under this pay group only.	paygroup	paygroup:VK2

Sequence of Interactions

The following are the sequence of interactions to return a list of available deduction codes:

1. Your consumer application makes a request to the ADP API endpoint to GET codeList-instructions for a worker.
2. The ADP API endpoint responds to your consumer application with single worker payroll codeList information.

Data Dictionary

The fields listed in the following table can be found in the ADP Vantage HCM UI by selecting **People > Pay & Taxes > Deductions**.

Schema Location	Field Name in ADP Vantage HCM	Note
/codeLists/codeListTitle	Not Displayed	
/codeLists/listItems	Not Displayed	
/codeLists/listItems/codeValue	CODE	Indicates the code of the deduction.
/codeLists/listItems/longName	Not Displayed	

Responses

You may encounter exceptions outside your common success scenarios. You must account for these exceptions during your initial development.

For more information, see [API Common Exceptions and Tips for Handling](#).

For API related information please refer [Worker Payroll Instructions](#) under API Explorer.

Response Code	Condition	messageTxt	Tips to Handle
200 OK	When there are unused deductions available for an employee.	NA	
204 No Content	When there are no unused deductions available for an employee.	NA	
204 No Content	When the Pay Group ID is invalid.	NA	Make sure you send a valid Pay Group ID. This can be retrieved from GET payroll instructions of an associate.
403 Forbidden	When the AOID is invalid.	"userMessage": {"codeValue": "3-49", "messageTxt": "You do not have the authorization to access the server. Contact your system administrator. (3,49)\n"}	Make sure you send a valid AOID. This can be retrieved using GET Workers API call.

Chapter 4

Use Case: Starting a General Deduction Instruction

Use Case Description

This use case is used to start a general deduction instruction for a worker.

Whenever a new deduction is added and if there are already existing deductions in the past, then they get carry forward.

API Usage

For API related information please refer [Worker Payroll Instructions](#) under API Explorer.

Method	URI	Description	Note
POST	/events/payroll/v2/worker-general-deduction-instruction.start	Starts worker general deduction instruction information.	
GET	/events/payroll/v2/worker-general-deduction-instruction.start/meta	Returns the event meta data. Note: Sample payload can be built from the meta call.	
GET	/events/payroll/v2/worker-general-deduction-instruction.start/meta? deductionCode= {deductionCode}&payrollGroupCode= {payGroup}&associateOID={aoid}	Returns the meta data depending upon the deduction Code and the associateOID . <ul style="list-style-type: none">• deductionCode can be obtained from un-used deductions codeList URI• AOID and payGroup can be obtained from "hr/v2/workers" call.	All the three parameters are mandatory. <ul style="list-style-type: none">• If the deduction code passed in the query parameter is percentage deduction, unit code field will be displayed in the meta response.• If the deduction code is dollar deduction, unit code won't be displayed in the meta response.

Application Scope

The following canonical needs to be added to your application scope to enable this use case:

**/payroll/payrollManagement/payrollInstructionManagement/
workerGeneralDeductionInstruction Management/workerGeneralDeductionInstruction.start**

Supported Actors

Request Parameter roleCode Value	Usage
Practitioner	Starts deduction instructions for workers.

Request Header Parameters

Parameter Name	Required (Y/N)	Usage	Value
Content-Type	Y	application/json	Content-Type: application/json

Supported OData Query Options

There are no supported OData Query Options.

Other Supported Parameters

The following parameters are only supported for the Deduction Start Meta API and should be combined.

For example:

/events/payroll/v2/worker-general-deduction-instruction.start/meta?
deductionCode={deductionCode}&payrollGroupCode={payGroup}&associateOID={aoid}

Parameter Name	Required (Y/N)	Usage	Value	Sample
deductionCode	Y	Returns the events meta information based on the associateOID , associate's payGroup and the type of deductionCode .	deductionCode	ADV
aoid	Y		associateOID	aoid: G3R9MDBS40828RS9
paygroup	Y		paygroup	paygroup:VK2

Sequence of Interactions

The following are the sequence of interactions to start a general deduction instruction:

1. Your consumer application makes a request to the ADP API endpoint to POST new deduction code for a worker.
2. The ADP API endpoint responds to your consumer application about the details of the **Worker-general-deduction-instruction** start.

Data Dictionary

The fields listed in the following table can be found in the ADP Vantage HCM UI by selecting **People > Pay & Taxes > Deductions**.

Schema Location	Field Name in ADP Vantage HCM	Is Required (Y/N)	Note
/data/eventContext/worker/associateOID	Not Displayed		The AOID can be retrieved from GET workers API call.
/data/transforms/effectiveDateTime	Effective Date or START DATE	N	If this field is not passed, the deduction is added to current date.
/data/transforms/payrollInstruction/generalDeductionInstruction/deductionCode	Not Displayed		
/data/transforms/payroll	Goal #	N	Indicates the goal number. Valid values

Instruction/generalDeductionInstruction/deductionGoal/goalID			are 1-9.
/data/transforms/payrollInstruction/generalDeductionInstruction/deductionCode/codeValue	CODE	Y	The codeValue can be retrieved from codeList API call.
/data/transforms/payrollInstruction/generalDeductionInstruction/deductionRate/rateValue	AMOUNT / PERCENT	N	If this field is not given the rate value is considered zero.
/data/transforms/payrollInstruction/generalDeductionInstruction/deductionRate/currencyCode	Not Displayed		
/data/transforms/payrollInstruction/generalDeductionInstruction/deductionRate/unitCode	Not Displayed		
/data/transforms/payrollInstruction/generalDeductionInstruction/deductionRate/unitCode/codeValue	Not Displayed		
/data/transforms/payrollInstruction/generalDeductionInstruction/deductionGoal/goalLimitAmount/amountValue	GOAL AMOUNT	N	If this field is not given the goal amount value is considered zero.
/data/transforms/payrollInstruction/generalDeductionInstruction/deductionGoal/goalLimitAmount/currencyCode	Not Displayed		

Responses

You may encounter exceptions outside your common success scenarios. You must account for these exceptions during your initial development.

For more information, see [API Common Exceptions and Tips for Handling](#).

For API related information please refer [Worker Payroll Instructions](#) under API Explorer.

Response Code	Condition	messageTxt	Tips to Handle
---------------	-----------	------------	----------------

201 Created	Adds a deduction code with a dollar amount for the current date.	NA	The deduction, as a dollar or percentage, needs to be set up at the client level.
201 Created	Adds a deduction code with a dollar amount for the current pay period.	NA	
201 Created	Adds a deduction code with a dollar amount for the next pay period.	NA	
201 Created	Adds a deduction code with a dollar amount for the future date.	NA	
201 Created	Adds a deduction code with an already existing goalID .	NA	When the goalID you are choosing for the deduction already exists in ADP Vantage HCM, then the API, by default, selects the next suitable goalID for the deduction you are trying to add and enters it into ADP Vantage HCM.
201 Created	Adds a deduction code with a percentage amount for the current pay period.		
201 Created	Adds a deduction code with a percentage amount for the next pay period.		
201 Created	Adds a deduction code with percentage amount for the future date.		
201 Created	Adds a deduction code without passing an effective date in the request payload.	NA	The deduction will be added to the system current date.
201 Created	Adds a deduction code without passing a goal limit amount.	NA	In this case the goal limit amount is taken as zero.
201 Created	Adds a deduction code without passing a deduction rate.	NA	In this case, the deduction rate is considered as zero.
400 Bad Request	Unsuccessful response when the AOID is invalid.	"userMessage": {"codeValue": "3-49", "messageTxt": "You do not have the authorization to access the server. Contact your system administrator. (3,49)\n"}	Make sure to use a valid AOID in the request payload.
400 Bad Request	Unsuccessful response when the deduction code is invalid.	"userMessage": {"codeValue": "15-11", "messageTxt": "You have entered an invalid value for EE General Deductions- Deduction Code. Select a valid value from the list provided. (15,11)\n"}	Make sure to use a valid deduction code in the request payload.
400 Bad Request	Unsuccessful response when the deduction goal amount is negative.	"userMessage": {"codeValue": "6000-373", "messageTxt": "Goal Amount cannot be negative. Change the value. (6000,373)\n"}	Goal amount should always be positive.
400 Bad Request	Unsuccessful response while trying to start an already existing deduction for an associate.	"userMessage": {"codeValue": "54-7", "messageTxt": "The row already exists. (54,7)\n"}	
400 Bad Request	Unsuccessful response when the deduction rate is invalid.	"messageTxt": "Cannot construct instance of double from String value 'aaa': not a valid Double value	

400 Bad Request	Unsuccessful response when the deduction goal amount is invalid.	"messageTxt": "Cannot construct instance of double from String value 'aaa': not a valid Double value	
-----------------	--	--	--

Chapter 5

Use Case: Changing a General Deduction Instruction

Use Case Description

This use case changes a general deduction instruction for a single worker.

API Usage

For API related information please refer [Worker Payroll Instructions](#) under API Explorer.

Method	URI	Description
POST	/events/payroll/v2/worker-general-deduction-instruction.change	Changes a worker's general deduction instruction information.
GET	/events/payroll/v2/worker-general-deduction-instruction.change/meta	Returns event metadata. Note: Sample payload can be built from the meta call.

Application Scope

The following canonical needs to be added to your application scope to enable this use case.

API	Canonical URI
Change deductions	/payroll/payrollManagement/payrollInstructionManagement/workerGeneralDeductionInstructionManagement/ workerGeneralDeductionInstruction.change

Supported Actors

Request Parameter roleCode Value	Usage
Practitioner	Retrieves payroll instructions. A practitioner can view, start, stop, or change a deduction. A system user is considered a practitioner.

Request Header Parameters

Parameter Name	Required (Y/N)	Usage	Value
Content-Type	Y	application/json	Content-Type:application/json

Supported OData Query Options

There are no supported OData Query Options.

Other Supported Parameters

There are no other supported parameters.

Sequence of Interactions

The following are the steps shown in the previous diagram:

1. Your consumer application makes a request to the ADP API endpoint for the Worker-general-deduction-instruction change API.
2. The ADP endpoint responds to your consumer application with the details of the Worker-general-deduction-instruction change.

Data Dictionary

The fields listed in the following table can be found in the ADP Vantage HCM UI by selecting **People > Pay & Taxes > Deductions**.

Schema Location	Field Name in ADP Vantage HCM	Is Required (Y/N)	Note
/data/eventContext/worker/associateOID	Not Displayed	Y	The AOID can be retrieved from GET workers API call.
/data/eventContext/payrollInstruction/generalDeductionInstruction/itemID	CODE	Y	The ItemID can be retrieved from codelist API call.
/data/transforms/effectiveDateTime	Effective Date or START DATE	Y	
/data/transforms/payrollInstruction/generalDeductionInstruction/deductionRate/rateValue	AMOUNT / PERCENT	Y	
/data/transforms/payrollInstruction/	Not Displayed		

generalDeductionInstruction/deductionRate/unitCode			
/data/transforms/payrollInstruction/generalDeductionInstruction/deductionRate/currencyCode	Not Displayed		
/data/transforms/payrollInstruction/generalDeductionInstruction/deductionRate/unitCode	Not Displayed		
/data/transforms/payrollInstruction/generalDeductionInstruction/deductionRate/unitCode/ codeValue	Not Displayed		
/data/transforms/payrollInstruction/generalDeductionInstruction/goal/goalID	Goal #	N	Indicates the goal number. Valid values are 1-9.
/data/transforms/payrollInstruction/generalDeductionInstruction/goal/goalLimitAmount/amountValue	GOAL AMOUNT	Y	
/data/transforms/payrollInstruction/generalDeductionInstruction/goal/goalLimitAmount/currencyCode	Not Displayed	N	

Responses

You may encounter exceptions outside your common success scenarios. You must account for these exceptions during your initial development.

For more information, see [API Common Exceptions and Tips for Handling](#).

For API related information please refer [Worker Payroll Instructions](#) under API Explorer.

Response Code	Condition	messageTxt	Tips to Handle
200 OK	Changes a deduction having a dollar amount for the current pay period.	NA	The deduction, as a dollar or percentage, needs to be set up at the client level.
200 OK	Changes a deduction having dollar amount for the next pay period.	NA	

200 OK	Changes a deduction having dollar amount for the future date.	NA	
200 OK	Changes a deduction having a percentage amount for the current pay period.		
200 OK	Changes a deduction having a percentage amount for the next pay period.		
200 OK	Changing a goalID for deduction code, when the goalID is already assigned with other deduction.		With the goalID , you are choosing the deduction already exists within ADP Vantage HCM. Then, by default, the API selects the next suitable goalID for the deduction you are trying to add and insert into ADP Vantage HCM.
200 OK	Changes the deduction having a percentage amount for the future date.		
200 OK	Changes a deduction passing a negative deduction rate.	NA	This case is used when any deduction is applied incorrectly and you want to refund the deducted amount. This negative deduction amount gets added into the payroll of an associate.
400 Bad Request	Unsuccessful response when the associateOID is invalid.	"userMessage": {"codeValue": "3-49", "messageTxt": "You do not have the authorization to access the server. Contact your system administrator. (3,49)\n"}	
400 Bad Request	Unsuccessful response when the item ID is invalid.	"userMessage": {"codeValue": "54-8", "messageTxt": "Unable to find the expected row. (54,8)\n"}	The item ID is nothing but the code value of the deduction for which you want to make changes.
500 Internal Server Error	Unsuccessful response when the deduction rate value is invalid.	"userMessage": {"codeValue": "0-0", "messageTxt": "Cannot construct instance of double from String value 'abc': not a valid Double value }	
500 Internal Server Error	Unsuccessful response when the deduction goal limit is invalid.	"userMessage": {"codeValue": "0-0", "messageTxt": "Cannot construct instance of double from String value 'ab': not a valid Double value }	
500 Internal Server Error	Unsuccessful response when the effective date format is invalid.	"userMessage": {"codeValue": "0-0", "messageTxt": "Cannot construct instance of java.util.Date from String value '28-09-2019': not a valid representation (error: Failed to parse Date value '28-09-2019': Cannot parse date '\\\"28-09-2019\\\": not compatible with any of standard forms (\\\"yyyy-MM-dd'T'HH:mm:ss.SSSZ\\\", \\\"yyyy-MM-dd'T'HH:mm:ss.SSS'Z'\\\", \\\"EEE, dd MMM yyyy HH:mm:ss zzz\\\", \\\"yyyy-MM-dd\\\"))}	

Use Case Description

This use case is used to stop a general deduction instruction for a single worker.

API Usage

For API related information please refer [Worker Payroll Instructions](#) under API Explorer.

Method	URI	Description
POST	/events/payroll/v2/worker-general-deduction-instruction.stop	Stops worker general deduction instruction information.
GET	/events/payroll/v2/worker-general-deduction-instruction.stop/meta	Returns an event metadata. Note: Sample payload can be built from the meta call.

Application Scope

The following canonical needs to be added to your application scope to enable this use case:

/payroll/payrollManagement/payrollInstructionManagement/workerGeneralDeductionInstructionManagement/workerGeneralDeductionInstruction.stop

Supported Actors

Request Parameter roleCode Value	Usage
Practitioner	Stops deduction instructions for workers.

Request Header Parameters

Parameter Name	Required (Y/N)	Usage	Value
Content-Type	Y	application/json	Content-Type:application

Supported OData Query Options

There are no supported OData Query Options.

Other Supported Parameters

There are no other supported parameters.

Sequence of Interactions

The following are the steps shown in the previous diagram:

1. Your consumer application makes a request to the ADP API endpoint for the Worker-general-deduction-instruction Stop API with the payload.
2. The ADP endpoint responds to your consumer application about the details of the Worker-general-deduction-instruction Stop.

Data Dictionary

The fields listed in the following table can be found in the ADP Vantage UI by selecting **People > Pay & Taxes > Deductions**.

Schema Location	Field Name in ADP Vantage HCM	Is Required (Y/N)
/data/eventContext/worker/associateOID	Not Displayed	Y
/data/eventContext/payrollInstruction/itemID	CODE	Y
/data/transform/effectiveDateTime	Effective date	Y

Responses

You may encounter exceptions outside your common success scenarios. You must account for these exceptions during your initial development.

For more information, see [API Common Exceptions and Tips for Handling](#).

For API related information please refer [Worker Payroll Instructions](#) under API Explorer.

Response Code	Condition	messageTxt	Tips to Handle
200 OK	Stops a deduction with an effective date.	NA	If the deduction is stopped, we will get a 200 OK response, but will not get any response payload.
400 Bad Request	Unsuccessful response when the item ID is invalid.	"userMessage": {"codeValue": "54-8", "messageTxt": "Unable to find the expected row. (54,8)\n"}	The item ID is nothing but the code value of the deduction which you want to stop.
403 Forbidden	Unsuccessful response when the AOID is invalid.	"userMessage": {"codeValue": "3-49", "messageTxt": "You do not have the authorization to access the server."}	

		Contact your system administrator. (3,49)\n"} "userMessage": { "codeValue": "0-0", "messageTxt": "Cannot construct instance of java.util.Date from String value '28-09-2019': not a valid representation (error: Failed to parse Date value '28-09-2019': Cannot parse date \"28-09-2019\": not compatible with any of standard forms (\"yyyy-MM- dd'T'HH:mm:ss.SSSZ\", \"yyyy-MM- dd'T'HH:mm:ss.SSS'Z'\", \"EEE, dd MMM yyyy HH:mm:ss zzz\", \"yyyy-MM-dd\") } Follow the {YYYY-MM-DD} format.	
500 Internal Server Error	Unsuccessful response when the effective date is invalid.		

Chapter 7

Use Case: Synchronizing Deduction Event Notifications

Use Case: Synchronizing Deduction Event Notifications

ADP applications trigger notifications when data is updated, and your application could subscribe interested notifications to synchronize data.

For more information about ADP Notifications, see the [ADP Event APIs and Event and Event Notification Guide](#).

These event notifications are generated only when the changes are made through the ADP Vantage HCM UI.

For API related information please refer [Worker Payroll Instructions](#) under API Explorer.

API	Scopes	Scopes to Subscribe
/events/payroll/v1/worker-general-deduction-instruction.start	/payroll/payrollManagement/payrollInstructionManagement/workerGeneralDeductionInstruction Management /workerGeneralDeductionInstruction.start	/payroll/payrollManagement/payrollInstructionManagement/workerGeneralDeductionInstruction Management /workerGeneralDeductionInstruction.start.eventNotify.subscribe
/events/payroll/v2/worker-general-deduction-instruction.change	/payroll/payrollManagement/payrollInstructionManagement/workerGeneralDeductionInstruction Management /workerGeneralDeductionInstruction.change	/payroll/payrollManagement/payrollInstructionManagement/workerGeneralDeductionInstruction Management /workerGeneralDeductionInstruction.change.eventNotify.subscribe
/events/payroll/v2/worker-general-deduction-instruction.stop	/payroll/payrollManagement/payrollInstructionManagement/workerGeneralDeductionInstruction Management /workerGeneralDeductionInstruction.stop	/payroll/payrollManagement/payrollInstructionManagement/workerGeneralDeductionInstruction Management /workerGeneralDeductionInstruction.stop.eventNotify.subscribe

Chapter 8

Fields Not Supported by this API

Field in ADP Vantage HCM	Description
AMOUNT TO DATE	Displays the total amount deducted until the current pay cycle. This field is not supported through the Deduction API.
BALANCE	Displays the remaining amount to deducted. This field is not supported through the Deduction API.

Chapter 9

Frequently Asked Questions

Question 1 : How does Goal Amount work for an employee?

Answer: If the goal amount is added to a deduction (for example, **D - DENTAL**), then the deduction **DENTAL** will be applied in payroll until the deduction amount reaches the goal limit amount. After it reaches that amount, the deduction will not be applied from the next payroll (**GOAL #**). There are total eight goals (1 to 8) for every employee. Each goal is restricted to only one deduction. For example, if the goal #: 1 is applied in deduction **DENTAL** then the same **Goal ID** cannot be applied to any other deduction.

Question 2: What is the difference between Goal Amount, Balance, and Amount to Date?

Answer:

- **Goal Amount** - The maximum limit of a deduction. For example, if the deduction amount is \$10 and the **Goal Amount** is \$100, then the deduction is applied in payroll until a total of \$100 is deducted.
- **Balance** - The remaining amount which is to be still deducted from the **Goal Amount**.
- **Amount to Date** - The amount deducted until the current pay cycle.

Question 3: If the Deduction is already applied for an employee, how do you adjust the Overall Goal Amount?

Answer: If the deduction is already running during every payroll without any goal amount but now the user wants to add a goal amount, then the user should include **Amount to Date** if it's from the ADP Vantage HCM UI. This means the amount entered in this field has already been deducted.

After this is completed, the **New Balance** amount in the ADP Vantage HCM UI will be displayed in **Balance**, which will be auto-calculated by deducting the amount given in the **Amount to Date** from actual **Goal Amount**.

For example, if Deduction **DENTAL** is already running in every payroll and a total of \$50 is deducted, the total **Goal Amount** is \$500.

If the user wants to adjust the deduction goal limit amount, then \$50 is entered in **Amount To Date**. After this is completed, the **New Goal Balance** is \$450, which is automatically calculated.

This can only be done through the UI as API does not support **Amount to Date** and **Balance** fields.

Question 4: How do I keep deduction records in sync with ADP Vantage HCM and the My\{partner\} application?

Answer:

- To retrieve all deductions, including both future effective deductions, for both the **Active** and **Inactive** status, do the following:
 1. Retrieve all deductions for each employee for the enrollment. For more information, see [Chapter 2 - Use Case: Retrieving Payroll Instructions \(Payroll Instruction API\)](#).
 2. For each deduction record in your (partner) system, compare the records retrieved from ADP Vantage HCM. For example, the **inactiveIndicator**, **deductionStartDate**, and so on.
- To START or CHANGE enrollment, do the following:
 1. Case 1: What if there are no effective deductions available? – Add the deduction by using POST **.start** with the effective date and amount. For more information, see [Chapter 4 - Use Case: Starting a General Deduction Instruction](#).
 2. Case 2: What if deductions are available but have the same deduction amount? – Do nothing.
 3. Case 3: What if deductions are available and there are different deduction amounts? – Update the deduction by using POST **.change** with the effective date and amount. For more information, see [Chapter 5 - Use Case: Changing a General Deduction Instruction](#).
- To Stop enrollment, do the following:
 1. Case 1: What if there are no effective deductions available? – Do nothing.
 2. Case 2: What if deductions are available? – Stop the deduction using POST **.stop** using effective date and item ID. For more information, see [Chapter 6 - Use Case: Stopping a General Deduction Instruction](#).

Chapter 10

Known Issues and Limitations

Issue 1: codeList response doesn't specify if a Deduction Code supports dollar amount or percentage amount

Impacted APIs

Method	URI	roleCode Value
GET	/codelists/payroll/v3/payroll-instruction-management/unused-ess-general-deduction-code-values?effectiveDate={YYYY-MM-DD}&paygroup={paygroup}&aoid={aoid}	Practitioner

Issue Description

The GET codeList instructions don't specify whether a deduction code is for a dollar or percentage amount.

Suggested Workaround

Consumer need to do META call using below query parameter to understand the deduction type.

For example:

**"/events/payroll/v2/worker-general-deduction-instruction.start/meta?
deductionCode=T1&payrollGroupCode=VK2&associateOID=G3R9MDBS40828RS9"**

- If the deduction code passed in the query parameter is percentage deduction then the meta response should return Unit code field
- If the deduction code passed in the query parameter is dollar deduction then it should not return unit code field in the meta response
- If no query parameter passed in the end point, the response will still show the Unit code (which is the current behaviour)

Issue 2: GET pay instructions for an Item ID through an API is not supported

Impacted APIs

Method	URI	roleCode Value
GET	/payroll/v1/workers/{aoid}/worker-general-deduction-instructions/{payroll-Instruction-ID}	Practitioner

Issue Description

This API returns only the specific payroll instruction deduction based on the ID passed in the end point.

Suggested Workaround

There are no workarounds available.

Issue 3: Deduction Start API doesn't display effectiveDateTime field in response to when it is sent through request payload

Impacted APIs

Method	URI	roleCode Value
POST	/events/payroll/v2/worker-general-deduction-instruction.start	Practitioner

Issue Description

When a deduction is added, along with effective date, through the request payload using the Deduction Start API, the API response doesn't show any **effectiveDateTime** field.

Suggested Workaround

There are no workarounds available.

Issue 4: Deduction start returns success response when there is no effectiveDate field in the request payload but, according to the meta, it's mandatory

Impacted APIs

Method	URI	roleCode Value
GET	/payroll/v1/workers/{aoid}/payroll-instructions/meta	Practitioner
POST	/events/payroll/v2/worker-general-deduction-instruction.start	Practitioner

Issue Description

In this case, the API takes the current date as the **effectiveDate** and adds the record for the current system date.

Suggested Workaround

There are no workarounds available.

Issue 5: Additional fields in the GET response which aren't mapped to the ADP Vantage HCM UI

Impacted APIs

Method	URI	roleCode Value
GET	/payroll/v1/workers/{aoid}/payroll-instructions	Practitioner

Issue Description

The payroll instruction GET response has few additional fields which are not mapped with ADP Vantage HCM UI.

Field	Description
operationID	These fields are not mapped to the ADP Vantage HCM UI. Important: These fields are implemented for a MyADP consumer.
actionTypeCode	
confirmationRequiredIndicator	
commentAllowedIndicator	
defaultIndicator	

Suggested Workaround

There are no workarounds available.

Issue 6: Event notifications aren't supported when deductions are invoked through the API

Impacted APIs

Method	URI	roleCode Value
POST	/events/payroll/v2/worker-general-deduction-instruction.start	Practitioner
POST	/events/payroll/v2/worker-general-deduction-instruction.change	Practitioner
POST	/events/payroll/v2/worker-general-deduction-instruction.stop	Practitioner

Issue Description

Whenever a deduction is invoked through the above API's, the event notification is not generated.

Suggested Workaround

There are no workarounds available.

Issue 7: API isn't throwing any error if a percentage deduction is added without passing the Unit Code field in the request payload

Impacted APIs

Method	URI	roleCode Value
POST	/events/payroll/v2/worker-general-deduction-instruction.start	Practitioner
POST	/events/payroll/v2/worker-general-deduction-instruction.change	Practitioner
POST	/events/payroll/v2/worker-general-deduction-instruction.stop	Practitioner

Issue Description

If a percentage valued deduction is added or changed using START and CHANGE API's without sending the **"unitCode" : {"codeValue" : "percent"}** field in the request payload it gives a successful response and it's reflecting in the ADP Vantage HCM UI.

Suggested Workaround

There are no workarounds available.

Issue 8: There is no facility to stop the deduction at any required date other than the effectiveDate when it's created

Impacted APIs

Method	URI	roleCode Value
POST	/events/payroll/v2/worker-general-deduction-instruction.stop	Practitioner

Issue Description

In the ADP Vantage HCM UI the user can delete a deduction at any specified date. But, through the API the user is unable to delete a deduction with a random date in future.

For example, consider a deduction called **M-Medical** is added through the API for a current date as the **effectiveDate** through the ADD Deductions API. If the user wants to stop that deduction in the future, giving some future date as the **effectiveDate**, the STOP API returns an error stating it can't find the deduction.

In the ADP Vantage HCM UI, the same process returns a successful response and deletes or stops the deductions for that specified date.

Suggested Workaround

There are no workarounds available.

Issue 9: STOP API is deleting the whole deduction record instead of stopping the deduction

Impacted APIs

Method	URI	roleCode Value
POST	/events/payroll/v2/worker-general-deduction-instruction.stop	Practitioner

Issue Description

In the ADP Vantage HCM UI, when you attempt to delete a deduction from an associate's deductions list by specifying any date, regardless of the date when the deduction is started, it gets deleted from the date selected. But, it won't get erased or deleted permanently from previous records.

In the API, the functionality is not the same. When a user tries to stop a deduction using the STOP Deduction Instructions API by giving an effective date, the API deletes, or erases, the whole row of deduction records from the ADP Vantage HCM UI for the date returned in the **effectiveDate** field through the payload.

Suggested Workaround

There are no workarounds available.