



Guide

# Deduction Instruction Guide for ADP Workforce Now Next Gen

Published on  
May 13, 2021 5:20AM

Last modified  
Feb 16, 2022 5:20PM





## 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.

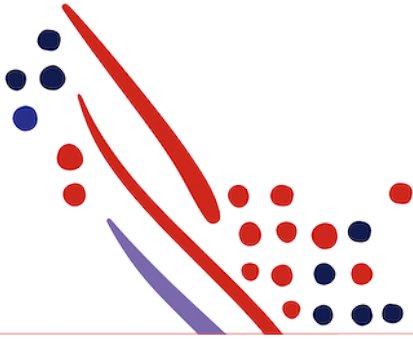
All other trademarks are the property of their respective owners.

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  
May 13, 2021 5:20AM

Last modified  
Feb 16, 2022 5:20PM



# Table of Contents

## Chapter 1

### About this API

Summary

What's New in this Guide?

**June 2021**

**May 2021**

About Voluntary Deduction Policies

Additional Information for Integration

Required Setup Steps

Postman Collection

## Chapter 2

### Retrieving Payroll Instructions

Use Case Description

API Usage

Application Scope

Supported Actors

Request Header Parameters

Sequence of Interactions

Data Dictionary

Responses

## Chapter 3

### Using the General Deduction Instruction Start API

Use Case Description

API Usage

Application Scope

Supported Actors

Request Header Parameters

Sequence of Interactions

Data Dictionary

Responses

## Chapter 4

### Changing a General Deduction Instruction

Use Case Description

API Usage  
Application Scope  
Supported Actors  
Request Header Parameters  
Sequence of Interactions  
Data Dictionary  
Responses

## Chapter 5

### Using the General Deduction Instruction Stop API

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

## Chapter 6

### Syncing Worker Deductions Using Notifications

## Chapter 7

### Frequently Asked Questions

Question 1: How does Goal Amount work for an employee?  
Question 2: What is the goalLimitAmount?  
Question 3: When is a deduction processed for an employee?  
Question 4: How do I keep deduction records in synch between ADP Workforce Now and my (partner) application?

## Chapter 8

### Fields Not Supported through API

## Chapter 9

### Known Issues and Limitations

USXXXXX: Cannot update a Deduction Status to Inactive or Active from the API

Impacted APIs  
Issue Description  
Suggested Workaround

USXXXXX: GET payroll instructions and deduction APIs are not supported for company loan.

Impacted APIs  
Issue Description  
Suggested Workaround

USXXXXX: GET payroll instruction API returns 500 server error instead of 400 bad request when Invalid aid is passed in the request.

Impacted APIs  
Issue Description  
Suggested Workaround

USXXXXX: Currently there is no error validation when unitCode value 'percent' is passed for the deduction which only supports 'Dollar' amount.

Impacted APIs

Issue Description

Suggested Workaround

USXXXXX: Currently there is no error validation when unitCode value 'percent' is passed for the deduction which only supports 'Dollar' amount.

Impacted APIs

Issue Description

Suggested Workaround

USXXXXX: Currently there is no error validation when starting a deduction which is already active.

Impacted APIs

Issue Description

Suggested Workaround

USXXXXX: Currently there is no error validation when admin Fee deduction policy is passed in the request before pre and post tax amount is not active in UI.

Impacted APIs

Issue Description

Suggested Workaround

USXXXXX: Currently there is no error validation when starting a deduction which configured as dollar amount but percent amount is passed in the request payload.

Impacted APIs

Issue Description

Suggested Workaround

USXXXXX: Currently deductions managed by benefits is not supported.

Impacted APIs

Issue Description

Suggested Workaround

USXXXXX: Currently meta API doesn't support any parameter which filters out deduction policy based on company code.

Impacted APIs

Issue Description

Suggested Workaround

USXXXXX: Currently meta API doesn't return any field which describes whether the deduction policy supports annual goal limit.

Impacted APIs

Issue Description

Suggested Workaround

# About this API

## Summary

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

- Start, stop, or change the payroll voluntary deduction setup for a worker.
- View all deduction instructions of a worker.
- Retrieve and send employee deductions to ADP (commonly used by Benefits Solutions).
- Capture taxable employer contributions. FUTURE
- Involuntary Withholding Orders (IWO or Garnishments) are currently not supported to be setup through this API. FUTURE
- Company Loans are currently not supported FUTURE

**Note:** Voluntary deductions are currently supported for ADP Workforce Now Next Gen US clients.

Access tokens need to be used based on the following clients:

- United States (U.S.)

## What's New in this Guide?

### June 2021

- GET payroll Instruction API is now also returns associate information with multiple positions. **Chapter 2.**
- Deduction STOP api will Inactive the specific deduction code, instead of complete deduction policy. **Chapter 5.**

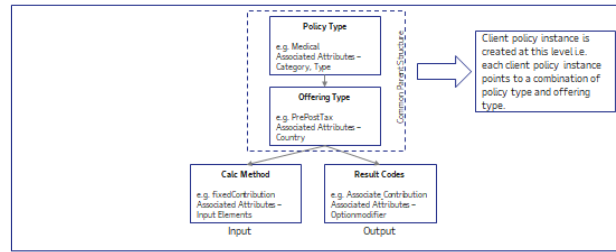
### May 2021

- This is the first version of the guide for ADP Workforce Now Next Gen. It provides details on the operation of voluntary deduction policies through the API.

## About Voluntary Deduction Policies

Workforce Now Next Gen enables a fully self service, policy based payroll setup, that can be completed by a client, without needing to contact ADP. A policy is an ADP defined set of statutory and functional rules to process a specific deduction such as medical or 401K. Policies have all the rules built into it them process a specific type of deduction. Policies are coarse grained and can process multiple deductions, employer contributions or imputed income within the same policy. For example: A medical policy can have a pre-tax deduction, a post tax deduction, a post tax admin fee, an employer contribution and multiple imputed income values.

There are currently 60+ voluntary deduction policies that are available to clients and each one of them has it's own structure and pay elements. Policies are structured constructs with multiple global categories as well as inputs and result codes. Image below depicts a typical policy structure. Each policy belongs to a parent category, has a policy type identifier, one or more offerings and multiple inputs and outputs. A medical policy has a category of 'Health and Welfare', policy type of 'Medical', offering of 'PrePostTax' and result codes as described above (Pre Tax Contr., Post Tax Contr., Admin Fee, ER Contr., Imputed Income(s)).



Copyright © 2019 ADP, LLC.



Clients have ability to select these policies from the library and configure them for use. This is called a client policy instance. Client can have multiple instances of the same policy type. For example: A client can have two medical policy instances, one to process deductions for their Aetna plan and another one to process Aflac.

Since policies are complex constructs, Workforce Now Next Gen has written a policy adapter that generates 'flat codes' from each client policy instance for policies to work with existing APIs. Leveraging the policy structure the adapter generates deduction codes that are unique for a clients setup. These deductions then can be setup using payroll instructions API. When an external application calls Workforce Now Next Gen APIs the policy adapter converts policy constructs into codes and responds to the caller. When an external application sends codes to WFN Next Gen for starting or stopping a deduction, adapter again converts them into policy constructs.

Voluntary Deduction Policies can be accessed within Workforce Now Next Gen by navigating to **Setup → Payroll → Policy Manager** and filtering on **Voluntary Deductions**. Here is an example of a medical policy instance. In this example, client has 3 separate policy instances.

Each policy instance is uniquely identified by Policy ID. Policy ID can be setup when a policy instance is created. However, once created, it cannot be changed. Codes generated within a policy instance have two portions - the policy ID and the actual result code within a policy. Here is an example of codes generated for the above policy:

These codes can be viewed or downloaded from the policy manager itself. These codes are also available through the API itself.

**These codes are not visible at the EE level. However there is a field corresponding to each code that is easily identifiable.**



## Note

1. **ER Contribution and Imputed Income setup is currently not supported through the API.**
2. **Workforce Now Next Gen doesn't use a deduction code for Direct Deposit. There is a separate API to setup direct deposits. No Company level setup is required.**
3. **Deduction codes are also not needed for School District, Medicare Surtax or any other taxes.**
4. **Involuntary Withholding Orders (IWO, formerly known as Liens) are not supported currently.**
5. **Wherever supported, same deduction code supports both \$ and % values. e.g. 401K. Unit Code parameter is a required field for the system to know if the code is carrying a \$ amount or %.**

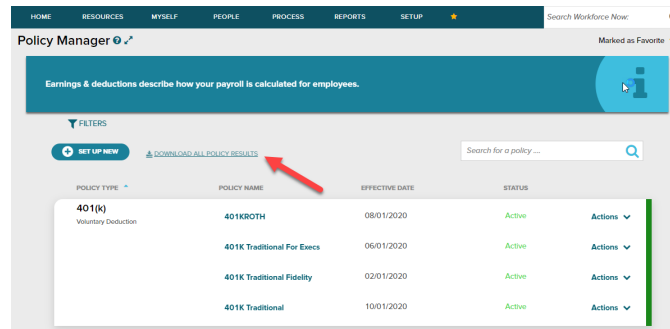
Additional Information for Integration



A voluntary deductions policy generally entails multiple deductions. Going by the previous example, a medical policy may have a pre-tax, a post-tax and an admin fee deduction. A client can start/stop/change any of these deductions individually through the API. If any of these deductions are active, the EE-Policy assignment will stay active. If all of them are stopped, the policy assignments will be marked inactive.

You may want to retrieve actual deductions resulted after the payroll run. The following are options you and the client could use:

- **API based integration** – Your application could use the Payroll Data Output APIs to retrieve payroll output data and get the deduction information for each employee after each payroll job.
- **Policy Manager UI or Result Code Download** - Download all possible result codes generated from clients' policy setup by going to policy manager (**Setup→Payroll→Policy Manager**) and clicking **Download All Policy Results**



Here is a **sample** of the download:

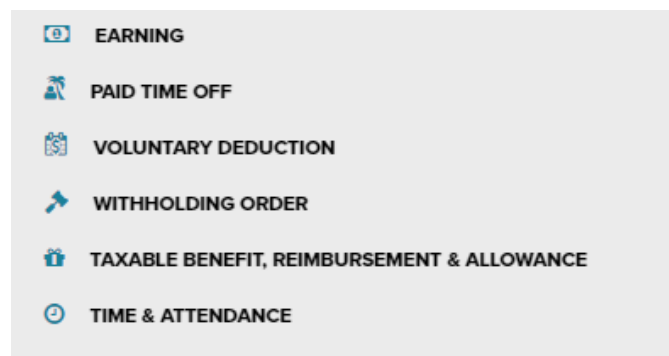
1	2	3	4	5	6	7
Policy Type	Offering Type	Policy Short Name	Result Name	Result Code		Pay Element Category
401(k) Loans	401(k) Loans	401K	401K - Employee Deduction	401K_ASSOCIATE_LOAN_COLLECTION		Voluntary Deduction
401(k) Roth	Roth	401KROTH	401KROTH - Employee Cash-up Contribution	401KROTH_CASHUP_ASSOCIATE_CONTRIBUTION		Voluntary Deduction
401(k) Roth	Roth	401KROTH	401KROTH - Employee Contribution	401KROTH_BASE_ASSOCIATE_CONTRIBUTION		Voluntary Deduction
401(k) Roth	Roth	401KROTH	401KROTH - Employee Cash-up Contribution	401KROTH_CASHUP_ORGANIZATION_CONTRIBUTION		Voluntary Deduction
401(k) Roth	Roth	401KROTH	401KROTH - Employee Contribution	401KROTH_BASE_ORGANIZATION_CONTRIBUTION		Voluntary Deduction
401(k) Roth	Roth	401KROTH	401KROTH - Employee Non Deductible Contribution - Roth	401KROTH_NON_DEDUCTIBLE_ORGANIZATION_CONTRIBUTION		Voluntary Deduction
401(k) Traditional	401(k) Traditional	401K	401K 10 Fidelity - Employee Cash-up Contribution	401K10FIDELITY_CASHUP_ASSOCIATE_CONTRIBUTION		Voluntary Deduction
401(k) Traditional	401(k) Traditional	401K	401K 10 Fidelity - Employee Contribution	401K10FIDELITY_BASE_ASSOCIATE_CONTRIBUTION		Voluntary Deduction
401(k) Traditional	401(k) Traditional	401K	401K 10 Fidelity - Employee Cash-up Contribution	401K10FIDELITY_CASHUP_ORGANIZATION_CONTRIBUTION		Voluntary Deduction
401(k) Traditional	401(k) Traditional	401K	401K 10 Fidelity - Employee Contribution	401K10FIDELITY_BASE_ORGANIZATION_CONTRIBUTION		Voluntary Deduction
401(k) Traditional	401(k) Traditional	401K	401K 10 Fidelity - Employee Non Deductible Contribution	401K10FIDELITY_NON_DEDUCTIBLE_ORGANIZATION_CONTRIBUTION		Voluntary Deduction
401(k) Traditional	401(k) Traditional	401K	401K - Employee Cash-up Contribution	401K_CASHUP_ASSOCIATE_CONTRIBUTION		Voluntary Deduction
401(k) Traditional	401(k) Traditional	401K	401K - Employee Contribution	401K_BASE_ORGANIZATION_CONTRIBUTION		Voluntary Deduction
401(k) Traditional	401(k) Traditional	401K	401K - Employee Cash-up Contribution	401K_CASHUP_ORGANIZATION_CONTRIBUTION		Voluntary Deduction
401(k) Traditional	401(k) Traditional	401K	401K - Employee Non Deductible Contribution	401K_NON_DEDUCTIBLE_ORGANIZATION_CONTRIBUTION		Voluntary Deduction
401(k) Traditional	401(k) Traditional	401K	401K - Employee Cash-up Contribution	401K_CASHUP_ASSOCIATE_CONTRIBUTION		Voluntary Deduction
401(k) Traditional	401(k) Traditional	401K	401K - Employee Contribution	401K_BASE_ORGANIZATION_CONTRIBUTION		Voluntary Deduction
401(k) Traditional	401(k) Traditional	401K	401K - Employee Cash-up Contribution	401K_CASHUP_ORGANIZATION_CONTRIBUTION		Voluntary Deduction
401(k) Traditional	401(k) Traditional	401K	401K - Employee Non Deductible Contribution	401K_NON_DEDUCTIBLE_ORGANIZATION_CONTRIBUTION		Voluntary Deduction
401(k) Traditional	401(k) Traditional	401K	401K - Employee Cash-up Contribution	401K_CASHUP_ASSOCIATE_CONTRIBUTION		Voluntary Deduction
401(k) Traditional	401(k) Traditional	401K	401K - Employee Contribution	401K_BASE_ORGANIZATION_CONTRIBUTION		Voluntary Deduction
401(k) Traditional	401(k) Traditional	401K	401K - Employee Cash-up Contribution	401K_CASHUP_ORGANIZATION_CONTRIBUTION		Voluntary Deduction

## Required Setup Steps

- **Policy Manager**

ADP Workforce Now Next Gen allows payroll setup to be done in form on earnings, deductions, withholding orders, taxable/reportable benefits and tax policies. There are largely two grouping of these policies:

**1. Business Policies → All of these can be setup under Setup→Payroll→Policy Manager. These can be sub-categorized in the following buckets.**





## Note

Time & Attendance is available only when time & attendance module is turned on for the client.

**2. Tax Policies** → All of these can be setup under **Setup→Payroll→Tax Jurisdictions**. These are Federal, State and Local tax policies for various tax jurisdictions in which the client operates.

- **Eligibility & Frequency** – Employees from multiple pay groups can be made eligible for the same policy. There is no need to setup separate policy for each pay group. You can also choose processing frequency of the policy for each pay group separately. Following frequencies are currently supported. For a given pay group, the entire policy gets processed with the same frequency, i.e. all the policy results have the same frequency of the processing within a pay group.

**Accumulators – Good News!** No need to setup separate accumulator codes. Each policy result gets accumulated automatically and is available in three time frames - Month-To-Date, Quarter-To-Date, Year-To-Date



## Tip

There are no reserved codes for % deductions (81-96 in legacy). Unit Code is a required field in payroll instructions api that will determine whether a value is a % value or \$ value. You can also send the same code with both the unit codes separately wherever it is supported. For example: 401K Base contribution can be setup with a % and with an additional flat \$ amount at the same time.

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

## Chapter 2

# Retrieving Payroll Instructions

## Use Case Description

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

This API exposes values found in the ADP Workforce Now UI by selecting **People > Pay > Payroll Profile > Voluntary Deductions**. You must provide the Associate Organization ID (AOID) of the desired worker in the AOID Uniform Resource Identifier (URI) parameter.



### Note

GET payroll Instruction API is now also returns associate information with multiple positions.



### Tip

To retrieve the ADP worker's unique Associate Object Identifier (AOID), your application can first retrieve all workers using GET /hr/v2/workers and creating the mapping between the two systems.

For more details, see the [Worker Information Management API Guide for ADP Workforce Now](#).

## API Usage

Method	URI	Description	Note
GET	<a href="#">/payroll/v1/workers/{aoid}/payroll-instructions</a>	Returns the list of all active payroll instructions for an employee until the current effective date.	ADP Workforce Now supports workers with multiple positions. When a worker has multiple positions, the Payroll Instructions API returns all active deductions and garnishments for all payroll file numbers for a given worker. Use the GET Workers API to retrieve employee position status.
GET	<a href="#">/payroll/v1/workers/{aoid}/payroll-instructions/{payroll-instruction-id}</a>	Returns deductions based on the <b>pay-distribution-id</b> . <b>Tip:</b> When an associate has more than one position, each position has a unique <b>payrollGroupCode</b> and <b>payrollFileNumber</b> . The user can retrieve all the payroll instructions for all positions. If the user wants to retrieve payroll instructions for only one position, the user can make use of the GET <a href="#">/payroll/v1/workers/{aoid}/payroll-instructions/{payroll-instruction-id}</a> . The API is called by passing the <a href="#">{payroll-instruction-id}</a> of the particular position.	
GET	<a href="#">/payroll/v1/workers/{aoid}/payroll-instructions?asOfDate={YYYY-MM-DD}</a>	Returns the list of all active deductions of an employee until the effective date provided in the <b>asOfDate</b> parameter.	
GET	<a href="#">/payroll/v1/workers/{aoid}/payroll-instructions?inactive=true</a>	Returns the list of all active and inactive deductions for an employee until the current effective date.	

GET	<a href="#">/payroll/v1/workers/{aoid}/payroll-instructions?asofDate={YYYY-MM-DD}&amp;inactive=true</a>	Returns the list of all active and inactive deductions for an employee until the effective date provided in the <b>asOfDate</b> parameter.
-----	---	--



## Note

Parameter 'Inactive=true' returns both Inactive and active deductions. There is a similar behavior in the UI.

## 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 the Deduction Instruction 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 as a practitioner. A practitioner can view, start, stop, or change a deduction. A system user is considered a practitioner.

## Request Header Parameters

None, in addition to the ADP standard header 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 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 Workforce Now UI by selecting **People > Payroll Profile > Voluntary Deductions**.

Payroll API Response Location	Field Name	Note
/payrollInstructions/generalDeductionInstructions	Not Displayed	
/payrollInstructions/generalDeductionInstructions/itemID	Not Displayed	Indicates the deduction Item ID. Each deduction has one item ID generated.
/payrollInstructions/generalDeductionInstructions/deductionCode/codeValue	Deduction Code	Refer to the policy overview section to understand next-gen deduction codes.
/payrollInstructions/generalDeductionInstructions/deductionRate/rateValue	Deduction Amount	Indicates the deduction amount.
/payrollInstructions/generalDeductionInstructions/deductionStartDate	Effective Date	Indicates the deduction effective date.
/payrollInstructions/generalDeductionInstructions/deductionGoal/goalLimitAmount/amountValue	Annual Goal	Indicates the Goal limit amount.
/payrollInstructions/generalDeductionInstructions/inactiveIndicator	Not Displayed	Indicates the status of the deduction. <ul style="list-style-type: none"> <li>• <b>True</b> indicates Inactive status.</li> <li>• <b>False</b> indicates active status.</li> </ul> <p>A voluntary deductions policy generally entails multiple deductions. Going by the previous example, a medical policy may have a pre-tax, a post-tax and an admin fee deduction. A client can start/stop/change any of these deductions individually through the API. If any of these deductions are active, the EE-Policy assignment will stay active. If all of them are stopped, the policy assignments will be marked inactive.</p>
/payrollInstructions/payrollFileNumber	Not Displayed	
/payrollInstructions/payrollGroupCode	Not Displayed	

## Responses

Response Code	Response Condition	message txt	GitHub Sample Request Payload	GitHub Sample Response Payload	Note
200 OK	Retrieves all the payroll deductions until the current effective date having multiple positions.		NA	<a href="#">payrollinstructions-aid-200.response.json</a>	
200 OK	Retrieves all the payroll deductions until the provided <b>asOfDate</b> parameter.		NA	<a href="#">payrollinstructions-asOfDate-200.response.json</a>	
200 OK	Retrieves all the payroll deductions until the provided <b>asOfDate</b> parameter.		NA	<a href="#">deductionsWithGoal-asOfDate-200.response.json</a>	
200 OK	Retrieves any specific deduction passing deduction <b>itemID</b> .		NA	<a href="#">Specificteduction-itemid-200.response.json</a>	
200 OK	Retrieves all the inactive deductions until the current <b>effectiveDate</b> .		NA	<a href="#">Inactivedeductions-currentdate-200.response.json</a>	
200 OK	Retrieves all the inactive deductions until the future <b>effectiveDate</b> .		NA	<a href="#">Inactivedeductions-futuredate-200.response.json</a>	
200 OK	Verifies the response when the employee does not have any deduction.		NA	<a href="#">empty-deductions-200.response.json</a>	
200 OK	Verifies the response for future date when the employee does not have any deductions.		NA	<a href="#">asOfDate-empty-200.response.json</a>	
204 No Content	Verifies the response when the employee does not get paid from ADP Workforce Now.		NA	NA	There will not be any response in this case.
400 Bad Request	Verifies the response when the <b>itemID</b> passed in the URI is invalid.		N/A	<a href="#">Invalid-ItemID-400.response.json</a>	
500 Internal Server Error	Verifies the response when the associate ID passed in the request URI is invalid.	"userMessage": {"messageTxt": "Label not found, missing following key:	NA	<a href="#">Invalid-associate-500.response.json</a>	

Exception in the  
requestHTTP 404 Not  
Found"} }



## Tip

To update deductions for a given worker AOID using the POST API (as described in [Chapter 3 - Use Case: Using the General Deduction Instruction Start API](#), [Chapter 4 - Use Case: Changing a General Deduction Instruction](#), and [Chapter 5 - Use Case: Using the General Deduction Instruction Stop API](#)) the payload needs to send a few values. These values can be obtained only by the GET `/payroll/v1/workers/{associateoid}/payroll-instructions` API. The following values need to be used from the response of the Payroll Instructions API:

- payrollFileNumber
- itemID

## Chapter 3

# Using the General Deduction Instruction Start API

## Use Case Description

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

## API Usage

Method	URI	Description	GitHub Sample Request Payload	GitHub Sample Response Payload
POST	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.start</a>	Starts worker general deduction instruction information.	<a href="#">deductionStart-200.request.json</a>	<a href="#">deductionStart-200.response.json</a>
GET	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.start/meta</a>	Returns an event metadata. <b>Note:</b> Sample payload can be built from the meta call.	NA	<a href="#">deduction-start-meta-200.response.json</a>



## Important

In the given payload ([deductionStart-200.request.json](#)), the following is the purpose of the **eventContext** and **transform** sections:

- **eventContext:** A set of keys, identifying the subject. In the payload, the **associateId** field is present under **eventContext**. The **associateId** identifies the subject.
- **transform:** Provides the values added or changed with respect to the subject keys defined in the **eventContext** section.

## Application Scope

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

**/payroll/payrollManagement/payrollInstructionManagement/workerGeneralDeductionInstructionManagement/workerGeneralDeductionInstruction**

## Supported Actors

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

## Request Header Parameters

None, in addition to the ADP standard header 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 **Worker-general-deduction-instruction** start meta.
2. The ADP API endpoint responds to your consumer application with the meta payload, which lists eligible deductions for the worker under the **/meta/data/transforms/payrollInstruction/generalDeductionInstruction/deductionCode/codeList** collection. Your consumer application needs to validate that the Deduction Code processed is returned in the codeList collection before building the payload for the next call. If the Deduction Code you need to start is not part of the collection, your application needs to provide user error handling and stop or go to the next step.
3. Your consumer application makes a request to the ADP API endpoint for the **Worker-general-deduction-instruction** start API with the payload.
4. The ADP API endpoint responds to the 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 Workforce Now UI by selecting **People > Pay > Deductions**.

Schema Location	ADP Workforce Now Field Name	Required (Y/N)?	Note
/events/data/eventContext	Not Displayed	N	
/events/data/eventContext/worker	Not Displayed	N	
/events/data/eventC	Not Displayed	Y	



ontext/worker/associateOID			
/events/data/eventContext/payrollInstruction	Not Displayed	N	
/events/data/eventContext/payrollInstruction/payrollGroupCode	Not Displayed	Y	The <b>payrollGroupCode</b> found in payroll. This is an Instruction GET call.
/events/data/eventContext/payrollInstruction/payrollGroupCode/codeValue	Not Displayed	Y	
/events/data/eventContext/payrollInstruction/payrollGroupCode/shortName	Not Displayed	Y	
/events/data/eventContext/payrollInstruction/payrollFileNumber	Not Displayed	Y	The <b>payrollFileNumber</b> found in payroll. This is an instruction GET call.
/events/data/effectiveDateTime	<b>Effective Date</b>	Y	
/events/data/payrollInstruction	Not Displayed	N	
/events/data/payrollInstruction/generalDeductionInstruction	Not Displayed	N	
/events/data/payrollInstruction/generalDeductionInstruction/deductionCode	Not Displayed	Y	
/events/data/payrollInstruction/generalDeductionInstruction/deductionCode/codeValue	Not Displayed	Y	
/events/data/payrollInstruction/generalDeductionInstruction/deductionCode/shortName	<b>Deduction Code</b>	Y	
/events/data/payrollInstruction/generalDeductionInstruction/deductionRate	Not Displayed	N	
/events/data/payrollInstruction/generalDeductionInstruction/deductionRate/rateValue	<b>Deduction Amount</b>	Y	
/events/data/payrollInstruction/generalDeductionInstruction/de	<b>Deduction Goal</b>	Y	

ductionGoal			
/events/data/payrollInstruction/generalDeductionInstruction/deductionGoal/goalLimitAmount	Not Displayed	N	
/events/data/payrollInstruction/generalDeductionInstruction/deductionGoal/goalLimitAmount/amountValue	<b>Deduction Goal Limit</b>	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](#).

Response Code	Response Condition	message Txt	GitHub Sample Request Payload	GitHub Sample Response Payload	Tips to Handle
200 OK	When adding a deduction without a deduction goal and the effective date as the current date.	NA	<a href="#">deduction-start-200.request.json</a>	<a href="#">deduction-start-200.response.json</a>	
200 OK	When adding a deduction with a deduction goal and the effective date as the current date.	NA	<a href="#">deduction-start-currentdate-200.request.json</a>	<a href="#">deduction-start-currentdate-200.response.json</a>	
200 OK	When adding a deduction with a deduction goal and effective date as the future date.	NA	<a href="#">deduction-start-futuredate-200.request.json</a>	<a href="#">deduction-start-futuredate-200.response.json</a>	
400 Bad Request	When trying to start a deduction but the goal limit amount is invalid.		<a href="#">deduction-start-Invalid-GoalLimitAmount-400.request.json</a>	<a href="#">deduction-start-Invalid-GoalLimitAmount-400.response.json</a>	
400 Bad Request	When the invalid deduction code is passed in the request.	{ "userMessage": {"messageTxt": "deductionCodeis invalid."}}}	<a href="#">deduction-start-400.invalid-deduction-code-request.json</a>	<a href="#">deduction-start-400.invalid-deduction-code-response.json</a>	Make sure a valid supported Deduction Code is passed in the request.
400 Bad Request	When there is an invalid <b>payrollGroup Code/codeValue</b> in the request.	{ "userMessage": {"messageTxt": "payrollGroupCode/co deValue is invalid." }}}	<a href="#">deduction-start-400.invalid-payroll-groupCode-request.json</a>	<a href="#">deduction-start-400.invalid-payroll-groupCode-response.json</a>	Make sure a valid <b>payrollGroupCode/codeValue</b> is passed in the request.

400 Bad Request	When there is an invalid <b>payrollFileNumber</b> in the request.	<pre>{"userMessage": {"messageTxt": "payrollFileNumber is invalid." }}</pre>	<a href="#">deduction-start-400.invalid-payroll-fileNumber-request.json</a>	<a href="#">deduction-start-400.invalid-payroll-fileNumber-response.json</a>	Make sure a valid <b>payrollFileNumber</b> is passed in the request.
400 Bad Request	When a <b>DeductionAmount</b> is passed as an invalid value in the request.	<pre>{"userMessage": {"messageTxt": "Deduction Amountfield must contain a non- zero value between -99,999,999.99 and 99,999,999.99 with a decimalprecision of 2 places." }}</pre>	<a href="#">deduction-start-400.invalid-deduction-amount-request.json</a>	<a href="#">deduction-start-400.invalid-deduction-amount-response.json</a>	Make sure you have entered the amount value correctly. <b>Note:</b> Deduction Amount must contain a non-zero value between -99,999,999.99 and 99,999,999.99 with a precision of 2.

## Chapter 4

# Changing a General Deduction Instruction

## Use Case Description

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

## API Usage

Method	URI	Description	GitHub Sample Request Payload	GitHub Sample Request Payload
POST	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.change</a>	Changes a worker's general deduction instruction information.	<a href="#">deduction-change-200.request.json</a>	<a href="#">deduction-change-200.response.json</a>
GET	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.change/meta</a>	Returns event metadata. <b>Note:</b> Sample payload can be built from the meta call.	NA	<a href="#">deduction-change-meta-200.response.json</a>



## Important

In the given payload ([deduction-change-200.request.json](#)), the following is the purpose of the **eventContext** and **transform** sections:

- **eventContext:** A set of keys, identifying the subject. In the payload, the **associateID** field is present under **eventContext**. The **associateID** identifies the subject.
- **transform:** Provides the values added or changed with respect to the subject keys defined in the **eventContext** section.

## Application Scope

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

**/payroll/payrollManagement/payrollInstructionManagement/workerGeneralDeductionInstructionManagement/workerGeneralDeductionInstructi**

## Supported Actors

Request Parameter roleCode Value	Usage
practitioner	Changes deduction instructions for workers.

## Request Header Parameters

None, in addition to the ADP standard header 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 **Worker-general-deduction-instruction** change meta.
2. The ADP API endpoint responds to your consumer application with the meta payload, which lists the eligible deductions for the worker under the **/meta/data/transforms/ /payrollInstruction/generalDeductionInstruction/deductionCode/codeList** collection. Your consumer application needs to validate the Deduction Code to be processed is returned in the codeList collection before building the payload for the next call. If the Deduction Code you need to start is not part of the collection, your application needs to provide user error handling and stop or go to step 3.
3. Your consumer application makes a request with the payload to the ADP API endpoint for the **Worker-general-deduction-instruction** change API.
4. The ADP API endpoint responds to the consumer with the details of the **Worker-general-deduction-instruction** change.

## Data Dictionary

Schema Location	Field Name	Required (Y/N)?	Note
/events/data/eventContext	Not Displayed	N	
/events/data/eventContext/worker	Not Displayed	N	
/events/data/eventContext/worker/associateOID	Not Displayed	Y	The <b>associateOID</b> will get this from the worker call API.
/events/data/eventC	Not Displayed	N	

ontext/payrollInstruction			
/events/data/eventContext/payrollInstruction/itemID	Not Displayed	Y	The <b>itemID</b> will get this from the payroll Instructions GET call.
/events/data/eventContext/payrollInstruction/payrollGroupCode	Not Displayed	Y	The <b>payrollGroupCode</b> will get this from the payroll Instructions GET call.
/events/data/eventContext/payrollInstruction/payrollGroupCode/codeValue	Not Displayed	Y	
/events/data/eventContext/payrollInstruction/payrollGroupCode/shortName	Not Displayed	Y	
/events/data/eventContext/payrollInstruction/payrollFileNumber	Not Displayed	Y	The <b>payrollFileNumber</b> will get this from the payroll Instruction GET call.
/events/data/eventContext/payrollInstruction/generalDeductionInstruction	Not Displayed	N	
/events/data/eventContext/payrollInstruction/generalDeductionInstruction/deductionCode	Not Displayed	Y	
/events/data/eventContext/payrollInstruction/generalDeductionInstruction/deductionCode/codeValue	Not Displayed	Y	
/events/data/transform/effectiveDateTime	<b>Effective</b>	Y	
/events/data/transform/payrollInstruction	Not Displayed		
/events/data/transform/payrollInstruction/generalDeductionInstruction	Not Displayed		
/events/data/transform/payrollInstruction/generalDeductionInstruction/deductionRate	Not Displayed	Y	
/events/data/transform/payrollInstruction/generalDeductionInstruction/deductionRate/rateValue	<b>Deduction Amount</b>	Y	
/events/data/transform	<b>Deduction Goal</b>	Y	

m/payrollInstruction/ generalDeductionInstr uction/deductionGoal			
/events/data/transfor m/payrollInstruction/g eneralDeductionInstr uction/deductionGoal/ goalLimitAmount	<b>Deduction Goal Limit</b>	Y	
/events/data/transfor m/payrollInstruction/ generalDeductionInstr uction/deductionGoal / goalLimitAmount/amo untValue	Not Displayed	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](#).

Response Code	Response Condition	messageTxt	Git Hub Sample Request Payload	Git Hub Sample Response Payload	Tips to Handle
200 OK	When changing a deduction, without a deduction goal, with effective date as current.	NA	<a href="#">deduction-change-200.request.json</a>	<a href="#">deduction-change-200.response.json</a>	
200 OK	When changing a deduction with deduction goal with effective date as current date.	NA	<a href="#">deduction-change-currentdate-200.request.json</a>	<a href="#">deduction-change-currentdate-200.response.json</a>	
200 OK	When changing a deduction with the deduction goal effective on a future date.	N/A	<a href="#">deduction-change-futuredate-200.request.json</a>	<a href="#">deduction-change-futuredate-200.response.json</a>	
400 Bad Request	When trying to change a deduction but the goal limit amount is invalid.		<a href="#">deduction-change-Invalid-GoalLimitAmount-400.request.json</a>	<a href="#">deduction-change-Invalid-GoalLimitAmount-400.response.json</a>	
400 Bad Request	When the invalid payroll <b>GroupCode/codeValue</b> is passed in the request.	{"userMessage": {"messageTxt": "payrollGroupCode/co deValue is invalid."}}	<a href="#">deduction-change-400.invalid-payroll-groupCode-request.json</a>	<a href="#">deduction-change-400.invalid-payroll-groupCode-response.json</a>	Make sure a valid <b>payrollGroupCode/codeValue</b> is passed in the request.
400 Bad Request	When the invalid <b>payrollFileNumber</b> is passed in the request.	{"userMessage": {"messageTxt": "payrollFileNumber is invalid."}}	<a href="#">deduction-change-400.invalid-payroll-fileNumber-request.json</a>	<a href="#">deduction-change-400.invalid-payroll-fileNumber-response.json</a>	Make sure a valid <b>payrollFileNumber</b> is passed in the request.

400 Bad Request	When the invalid itemID is passed in the request.	{"userMessage": {"messageTxt": "itemID is invalid."}}	<a href="#">deduction-change-400.invalid-itemID-request.json</a>	<a href="#">deduction-change-400.invalid-itemID-response.json</a>	Make sure a valid itemID is passed in the request. <b>Note:</b> The <b>itemID</b> is obtained from the Payroll Instructions API.
400 Bad Request	When an invalid <b>deductionCode</b> is passed in the request.	{"userMessage": {"messageTxt": "deduction Code is invalid."}}	<a href="#">deduction-change-400.invalid-deduction-code-request.json</a>	<a href="#">deduction-change-400.invalid-deduction-code-response.json</a>	Make sure a valid <b>deductionCode</b> is passed in the request.

## Chapter 5

# Using the General Deduction Instruction Stop API

## Use Case Description

This use case is used to stop a general deduction instruction for a single worker. A successful stop request changes the Deduction record from the **Active** to the **Inactive** state.



### Note

Deduction STOP api will Inactive the specific deduction code, instead of complete deduction policy

To view **Inactive** deductions in the ADP Workforce Now UI, select **People > Pay > Pay Profile** and follow these steps:

1. Find the worker record.
2. Click **Deductions**.
3. Select data from **Show as of**.
4. Switch **SHOW INACTIVE** to **YES**.

Your program could also retrieve **Inactive** records using APIs. For more information, see [Chapter 2 - Use Case: Retrieving Payroll Instructions \(Payroll Instructions API\)](#).

## API Usage

Method	URI	Description	GitHub Sample Request Payload	GitHub Sample Response Payload
GET	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.stop/meta</a>	Returns an event metadata. <b>Note:</b> Sample payload can be built from the meta call.	NA	<a href="#">deduction-stop-meta-200-response.json</a>
POST	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.stop</a>	Stops worker general deduction instruction information.	<a href="#">deduction-stop-200.request.json</a>	<a href="#">deduction-stop-200.response.json</a>



## Important

In the given payload ([deduction-stop-200.request.json](#)), the following is the purpose of the **eventContext** and **transform** sections:

- **eventContext**: A set of keys, identifying the subject. In the payload, the **associateID** field is present under **eventContext**. The **associateID** identifies the subject.
- **transform**: Provides the values added or changed with respect to the subject keys defined in the **eventContext** section.

## Application Scope

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

**/payroll/payrollManagement/payrollInstructionManagement/workerGeneralDeductionInstructionManagement/workerGeneralDeductionInstruction**

## Supported Actors

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

## Request Header Parameters

None, in addition to the ADP standard header 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 a **Worker-general-deduction-instruction Stop** meta.
2. The ADP API endpoint responds to your consumer application with the meta payload, which lists the eligible deductions for the worker under the **/meta/data/transforms/payrollInstruction/generalDeductionInstruction/deductionCode/codeList** collection. Your consumer application needs to validate the Deduction Code to be processed is returned in the **codeList** collection before building the payload for the next call. If the Deduction Code you need to start is not part of the collection, your application needs to provide user error handling and stop or go to the next step.
3. Your consumer application makes a request to the ADP API endpoint for the **Worker-general-deduction-instruction Stop** API with the payload.
4. The ADP API endpoint responds to the 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 Workforce Now UI by selecting **People > Pay > Deductions**.

Schema Location	ADP Workforce Now Field Name	Required (Y/N)?	Note
/events/data/eventContext	Not Displayed	N	
/events/data/eventContext/worker	Not Displayed	N	
/events/data/eventContext/worker/associateOID	Not Displayed	Y	The <b>associateOID</b> will get this from the Worker Call API.
/events/data/eventContext/payrollInstruction	Not Displayed	N	
/events/data/eventContext/payrollInstruction/itemID	Not Displayed	Y	The <b>itemID</b> will get this from the payroll Instructions GET call.
/events/data/eventContext/payrollInstruction/payrollGroupCode	Not Displayed	Y	The <b>payrollGroupCode</b> will get this in the payroll Instructions GET call.
/events/data/eventContext/payrollInstruction/payrollGroupCode/codeValue	Not Displayed	Y	
/events/data/eventContext/payrollInstruction/payrollGroupCode/shortName	Not Displayed	Y	
/events/data/eventContext/payrollInstruction/payrollFileNumber	Not Displayed	Y	The <b>payrollFileNumber</b> will get this in the payroll Instructions GET call.
/events/data/eventContext/payrollInstruction/generalDeductionInstruction	Not Displayed	Y	
/events/data/eventContext/payrollInstruction/generalDeductionInstruction/deductionCode	Not Displayed	Y	
/events/data/eventContext/payrollInstruction/generalDeductionInstruction/deductionCode/codeValue	Not Displayed	Y	
/events/data/eventContext/payrollInstruction/generalDeductionInstruction	Not Displayed	Y	

instruction/deduction Code/shortName			
/events/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](#).

Response Code	Response Condition	messageTxt	GitHub Sample Request Payload	GitHub Sample Response Payload	Tips to Handle
200 OK	When the request is processed successfully.	NA	<a href="#">deduction-stop-200.request.json</a>	<a href="#">deduction-stop-200.response.json</a>	
400 Bad Request	When the invalid <b>payrollFileNumber</b> is passed in the request.	{"userMessage": {"messageTxt": "payrollFileNumber is invalid."}}	<a href="#">deduction-stop-400.invalid-payroll-fileNumber-request.json</a>	<a href="#">deduction-stop-400.invalid-payroll-fileNumber-response.json</a>	Make sure a valid <b>payrollFileNumber</b> is passed in the request.
400 Bad Request	When the invalid <b>itemID</b> is passed in the request.	{"userMessage": {"messageTxt": "itemID is invalid."}}	<a href="#">deduction-stop-400.invalid-itemID-request.json</a>	<a href="#">deduction-stop-400.invalid-itemID-response.json</a>	Make sure a valid <b>itemID</b> is passed in the request.
400 Bad Request	When the <b>effectiveDate</b> is passed as previous year.	"userMessage": { "messageTxt": "effectiveDateTime is invalid." }	<a href="#">deduction-stop-400.invalid-effective-date-request.json</a>	<a href="#">deduction-stop-400.invalid-effective-date-response.json</a>	Make sure the <b>effectiveDate</b> is the same as or later than the position start date.

## Chapter 6

# Synching Worker Deductions Using Notifications

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

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

API	Scopes	Scopes to Subscribe	Sample Event Notification Response
<a href="#">/events/payroll/v2/worker-general-deduction-instruction.start</a>	<a href="#">/payroll/payrollManagement/payrollInstructionManagement/worker.generalDeductionInstructionManagement/workerGeneralDeductionInstruction.start</a>	<a href="#">/payroll/payrollManagement/payrollInstructionManagement/worker.generalDeductionInstructionManagement/workerGeneralDeductionInstruction.start.eventNotify.subscribe</a>	<a href="#">deduction-start-event-notification-http-200-response.json</a>

<a href="#">/events/payroll/v2/worker-general-deduction-instruction.change</a>	<a href="#">/payroll/payrollManagement/payrollInstructionManagement/workerGeneralDeductionInstructionManagement/workerGeneralDeductionInstruction.change</a>	<a href="#">/payroll/payrollManagement/payrollInstructionManagement/workerGeneralDeductionInstructionManagement/workerGeneralDeductionInstruction.change.eventNotify.subscribe</a>	<a href="#">deduction-change-event-notification-http-200-response.json</a>
<a href="#">/events/payroll/v2/worker-general-deduction-instruction.stop</a>	<a href="#">/payroll/payrollManagement/payrollInstructionManagement/workerGeneralDeductionInstructionManagement/workerGeneralDeductionInstruction.stop</a>	<a href="#">/payroll/payrollManagement/payrollInstructionManagement/workerGeneralDeductionInstructionManagement/workerGeneralDeductionInstruction.stop.eventNotify.subscribe</a>	<a href="#">deduction-stop-event-notification-http-200-response.json</a>

## Chapter 7

# Frequently Asked Questions

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

**Answer:** Most of the voluntary deduction goals are annual except for the loan goals (401K Loan, Company Loan) which are for the lifetime (duration of the loan). Annual goals are automatically reset at the beginning of the calendar year. Let's say an EE has an annual goal amount of \$500 for **Medical-PreTax** Amount. The employee reaches that goal in September. Then the **Medical-PreTax** deduction will stop for the year. Come January, the deduction will start again if the goal amount is still there on EE Payroll Profile.

## Question 2: What is the goalLimitAmount?

### Answer:

- **goalLimitAmount** – The maximum limit of a deduction. For example, if the deduction amount is \$10 and the GoalLimitAmount is \$100, then the deduction is applied in payroll until a total of \$100 is deducted.

## Question 3: When is a deduction processed for an employee?

### Answer:

- Case 1: If the deduction is added in current date, it will be effective from next payroll. Not from the current payroll.
- Case 2: If the deduction is added in future date, it will be effective from the next payroll from the date when it is active.

## Question 4: How do I keep deduction records in synch between ADP Workforce Now and my (partner) application?

### Answer:

1. To retrieve all deductions, including both future effective deductions, for both the **Active** and **Inactive** status, do the following (in order):
  - Retrieve all deductions for each employee using **asOfDate={effectiveDate} & inactive=true** for the enrollment. For more information, see [Chapter 2 - Use Case: Retrieving Payroll Instructions \(Payroll Instructions API\)](#).
  - For each deduction record in your (partner) system, compare the records retrieved from ADP Workforce Now. For example, the **inactiveIndicator**, **deductionStartDate**, and so on.
2. To START or CHANGE enrollment, do the following:
  - **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 3 - Use Case: Using the General Deduction Instruction Start API](#).

- **Case 2: What if there are deductions available but they're Inactive?** - The API doesn't handle updating the deduction status to **Active** or **Inactive**. It can only be completed on the ADP Workforce Now user interface (UI).
- **Case 3: What if deductions are available, are they the same deduction amount?** - Do nothing.
- **Case 4: 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 4 - Use Case: Changing a General Deduction Instruction](#).

3. To Stop enrollment, do the following:

- **Case 1: What if there are no effective deductions available?** Do nothing.
- **Case 2: What if deductions are available but they're Inactive?** Do nothing.
- **Case 3: What if the deductions are available but they're Active?** Stop the deduction by using POST .stop with the effective date. For more information, see [Chapter 4 - Use Case: Changing a General Deduction Instruction](#).

## Chapter 8

# Fields Not Supported through API

Field Name in WFN Cloud	Deduction Category	Deduction Policy Type
Principal Repayment Goal	Company Loan	Pre-calculated, per-period principal & interest amounts
Interest Goal	Company Loan	Pre-calculated, per-period principal & interest amounts
Imputed Income	Company Loan	Pre-calculated, per-period principal & interest amounts
Total Principal Amount	Company Loan	Per-period principal & interest amounts calculated by payroll
Company Interest Rate	Company Loan	Per-period principal & interest amounts calculated by payroll
Market Interest Rate	Company Loan	Per-period principal & interest amounts calculated by payroll
Per Period Repayment Amount	Company Loan	Per-period principal & interest amounts calculated by payroll
Number of Deductions	Company Loan	Per-period principal & interest amounts calculated by payroll
Employer Contribution	Accident & Critical illness	Pre & Post-Tax
Imputed Income For S-Corp	Accident & Critical illness	Pre & Post-Tax
Employer Contribution	Dental	Pre & Post-Tax
Imputed Income For Employee	Dental	Pre & Post-Tax
Imputed Income For S-Corp	Dental	Pre & Post-Tax

Imputed Income For Domestic Partner	Dental	Pre & Post-Tax
Imputed Income For Over Age Dependents	Dental	Pre & Post-Tax
Employer Contribution	Donation	Charitable Contributions
Employer Contribution	Group Term Life	The Employee
Benefit Coverage	Group Term Life	The Employee
Employer Contribution	Health Savings Account	Pre & Post-Tax
Imputed Income For Employee	Health Savings Account	Pre & Post-Tax
Imputed Income For S-Corp	Health Savings Account	Pre & Post-Tax
Imputed Income For Domestic Partner	Health Savings Account	Pre & Post-Tax
Imputed Income For Over Age Dependents	Health Savings Account	Pre & Post-Tax
Employer Contribution	Medical	Pre & Post-Tax
Imputed Income For Employee	Medical	Pre & Post-Tax
Imputed Income For S-Corp	Medical	Pre & Post-Tax
Imputed Income For Domestic Partner	Medical	Pre & Post-Tax
Imputed Income For Over Age Dependents	Medical	Pre & Post-Tax
Employer Contribution	Short Term Disability	Pre & Post-Tax
Imputed Income For S-Corp	Short Term Disability	Pre & Post-Tax
Employer Contribution	Vision	Pre & Post-Tax
Imputed Income For Employee	Vision	Pre & Post-Tax
Imputed Income For S-Corp	Vision	Pre & Post-Tax
Imputed Income For Domestic Partner	Vision	Pre & Post-Tax
Imputed Income For Over Age Dependents	Vision	Pre & Post-Tax

## Known Issues and Limitations

USXXXXX: Cannot update a Deduction Status to Inactive or Active from the API

### Impacted APIs

Method	URI	roleCode Value
GET	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.change/meta</a>	practitioner
POST	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.change</a>	practitioner

### Issue Description

The Deduction API cannot make a deduction status **Active** or **Inactive**. Only retrieve is possible through the payroll instruction GET API.

### Suggested Workaround

You can make the deduction status **Active** or **Inactive** using the ADP Workforce Now UI.

USXXXXX: GET payroll instructions and deduction APIs are not supported for company loan.

### Impacted APIs

Method	URI	roleCode Value
GET	<a href="#">/payroll/v1/workers/{aoid}/payroll-instructions</a>	practitioner
GET	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.start/meta</a>	practitioner
POST	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.start</a>	practitioner
GET	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.change/meta</a>	practitioner

POST	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.change</a>	practitioner
GET	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.stop/meta</a>	practitioner
POST	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.stop</a>	practitioner

## Issue Description

Currently company loan deductions cannot be retrieved or add/change/stop using APIs

## Suggested Workaround

It can be done using the ADP Workforce Now UI.

USXXXXX: GET payroll instruction API returns 500 server error instead of 400 bad request when Invalid aoid is passed in the request.

## Impacted APIs

Method	URI	roleCode Value
GET	<a href="#">/payroll/v1/workers/{aoid}/payroll-instructions</a>	practitioner

## Issue Description

When the aoid passed in the request end point is invalid, the api should return 400 bad request with proper error message. But the API is returning 500 Internal server error.

## Suggested Workaround

NA

USXXXXX: Currently there is no error validation when unitCode value 'percent' is passed for the deduction which only supports 'Dollar' amount.

## Impacted APIs

Method	URI	roleCode Value
POST	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.start</a>	practitioner

## Issue Description

For the deduction which only supports dollar amount but unitCode as percent is passed in the request.

The API should return validation error message stating Invalid UnitCode value, but the API is giving success response.

## Suggested Workaround

NA

USXXXXX: Currently there is no error validation when unitCode value 'percent' is passed for the deduction which only supports 'Dollar' amount.

## Impacted APIs

Method	URI	roleCode Value
POST	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.start</a>	practitioner

## Issue Description

For the deduction which only supports dollar amount but unitCode as percent is passed in the request.

The API should return validation error message stating Invalid UnitCode value, but the API is giving success response.

## Suggested Workaround

NA



USXXXXX: Currently there is no error validation when starting a deduction which is already active.

## Impacted APIs

Method	URI	roleCode Value
POST	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.start</a>	practitioner

## Issue Description

When try to add a deduction which is already active for an employee, the API should throw validation error with proper error message. But the API is giving success response.

## Suggested Workaround

NA

USXXXXX: Currently there is no error validation when admin Fee deduction policy is passed in the request before pre and post tax amount is not active in UI.

## Impacted APIs

Method	URI	roleCode Value
POST	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.start</a>	practitioner

## Issue Description

Validate the error message when admin Fee deduction policy is passed in the request before pre and post tax amount is not active in UI. The API should throw an error validation with proper message but it is currently giving success response and nothing is reflected in UI

## Suggested Workaround

NA

USXXXXX: Currently there is no error validation when starting a deduction which configured as dollar amount but percent amount is passed in the request payload.

## Impacted APIs

Method	URI	roleCode Value
POST	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.start</a>	practitioner

## Issue Description

This test case is when starting a deduction which supports only dollar amount value but percent amount is passed in the request payload  
API gives success response but its not reflects in UI

## Suggested Workaround

NA

USXXXXX: Currently deductions managed by benefits is not supported.

## Impacted APIs

Method	URI	roleCode Value
POST	<a href="#">/events/payroll/v2/worker-general-deduction-instruction.start</a>	practitioner

## Issue Description

Deductions managed by Benefits is currently not supported.

## Suggested Workaround

NA

USXXXXX: Currently meta API doesn't support any parameter which filters out deduction policy based on company code.

## Impacted APIs

Method	URI	roleCode Value
GET	/events/payroll/v2/worker-general-deduction-instruction.start/meta	practitioner
GET	/events/payroll/v2/worker-general-deduction-instruction.change/meta	

## Issue Description

Currently meta API doesn't support any parameter which filters out deduction policy based on company code.

## Suggested Workaround

Need to login to WFN Cloud and validate which deduction policy is supported to which company code.

USXXXXX: Currently meta API doesn't return any field which describes whether the deduction policy supports annual goal limit.

## Impacted APIs

Method	URI	roleCode Value
GET	/events/payroll/v2/worker-general-deduction-instruction.start/meta	practitioner
GET	/events/payroll/v2/worker-general-deduction-instruction.change/meta	

## Issue Description

Currently meta API doesn't return any field which describes whether the deduction policy supports annual goal limit.

## Suggested Workaround

Need to login to WFN Cloud and validate which deduction policy is supported for annual goal limit.