



Guide

Leave Request API Guide for ADP Workforce Now

Published on
Aug 07, 2020 10:32AM

Last modified
May 20, 2022 3:49AM



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
Aug 07, 2020 10:32AM

Last modified
May 20, 2022 3:49AM

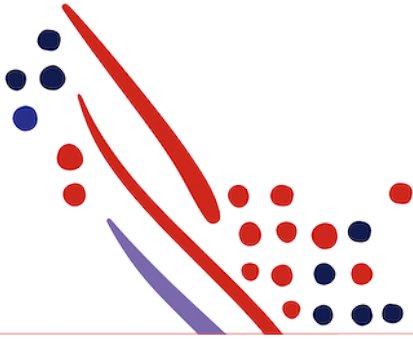


Table of Contents

Chapter 1

About this API

Summary

What's New in this Guide?

May 2022

Supported Product Version and Customer Base

Process Overview

Chapter 2

Use Case: Retrieving an Employee's Leaves

Use Case Description

API Usage

Application Scope

Supported Actors

Request Header Parameters

Sequence of Interactions

Data Dictionary

Responses

Chapter 3

Use Case: Applying Leave Absence for an Employee

Use Case Description

API Usage

Application Scope

Supported Actors

Request Header Parameters

Sequence of Interactions

Data Dictionary

Responses

Chapter 4

Use Case: Changing an Employee's Leave of Absence

Use Case Description

API Usage

Application Scope

Supported Actors
Request Header Parameters
Sequence of Interactions
Data Dictionary
Responses

Chapter 5

Use Case: Returning an Employee's Leave of Absence

Use Case Description
API Usage

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

Responses

Chapter 6

Use Case: Cancelling an Employee's Leave of Absence

Use Case Description
API Usage

Application Scope
Supported Actors

Request Header Parameters
Sequence of Interactions
Data Dictionary

Responses

Chapter 7

Known Issues and Limitations

Issue 1: Effective date can't be updated when changing the leave information using the leave change API

Impacted APIs
Issue Description
Suggested Workaround

Issue 2: Leave request for a terminated employee is not possible through the API

Impacted APIs
Issue Description
Suggested Workaround

Issue 3: Event notification isn't supported

Impacted APIs
Issue Description
Suggested Workaround

About this API

Summary

This API guide covers the Leave Request Application Programming Interface (API), which assists in processing an employee's leave requests, changes, and return and leave cancellations. Each API is categorized as a use case.

This API also limited with known issues, which are listed under [Chapter 7 - Known Issues and Limitations](#).

What's New in this Guide?

This is the initial version of the Leave Request API Guide for ADP Workforce Now. In this section, we will announce any new revisions to the Leave request API and this guide.

May 2022

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

Supported Product Version and Customer Base

All U.S. clients are on the most current version of the Leave Request API for ADP Workforce Now.

Process Overview

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

	Actor	Task Description
1	Client Practitioner	Triggers the API-based data exchange between your application and ADP Workforce Now.
2	Your application	Terminates the work assignment of an employee.
3	Client Practitioner	Verifies the information in the ADP Workforce Now user interface (UI) by selecting People > Employment > Employment Profile > Status .

Use Case: Retrieving an Employee's Leaves

Use Case Description

This use case is about retrieving an employee's leaves using a GET API.

API Usage

Worker Leaves

Method	Uniform Resource Identifier (URI)	Description
GET	/hr/v2/workers/{aoid}/leaves	This retrieves the list of leaves for an employee.

Application Scope

The canonical URI corresponding to the Leave Request 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:

**/hr/workerInformationManagement/workerManagement/workerLeaveManagement/workerLeave.
read**

Supported Actors

ADP APIs have built-in data entitlement and access control. The response contains data based on the **roleCode** parameter value in an API request. For example, a person could have two roles defined in an ADP application, a Practitioner who manages all employees, and an employee. Your application will use the Practitioner **roleCode** to request the data as the Practitioner and the **roleCode** of the employee.

Request Parameter roleCode Value	Supported (Y/N)?
Practitioner	Y
Employee	N
Manager	N
Admin	Y

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 the **worker leaves**.
2. The ADP API endpoint responds to the consumer application with the processing result.

Data Dictionary

The fields listed in the following table can be found in the ADP WorkForce Now UI by selecting **People > Employment > Employment Profile > Status section**.

Schema Location	Field Name	Note
Schema Location	Field Name	Note
workerLeaves/associateOID		
workerLeaves/workerID		
workerLeaves/workerID/idValue	NA	
workerLeaves/workAssignmentID	NA	
workerLeaves/leaves		
workerLeaves/leaves/itemID		This field is currently not supported.
workerLeaves/leaves/leaveAbsence		
workerLeaves/leaves/leaveAbsence/startDateTime	Leave of Absence Start Date	
workerLeaves/leaves/leaveAbsence/expectedEndTime	Leave of Absence Expected Return Date	
workerLeaves/leaves/leaveAbsence/leaveDuration		This field is currently not supported.
workerLeaves/leaves/leaveAbsence/leaveStatus		This field is currently not supported.
workerLeaves/leaves/leaveAbsence/leaveTypeCode		

workerLeaves/leaves/leaveAbsence/leaveTypeCode/codeValue	Leave Reason	
workerLeaves/leaves/leaveAbsence/leaveSubTypeCode		This field is currently not supported.
workerLeaves/leaves/leaveAbsence/paymentStatusCode		
workerLeaves/leaves/leaveAbsence/paymentStatusCode/codeValue	Paid Leave of Absence	
workerLeaves/leaves/leaveAbsence/comment		This field is currently not supported.
workerLeaves/leaves/leaveAbsence/links		This field is currently not supported.
workerLeaves/leaves/leaveReturn		
workerLeaves/leaves/leaveReturn/returnDateTime	Leave Return Date	
workerLeaves/leaves/leaveReturn/notificationReceivedDateTime		This field is currently not supported.
workerLeaves/leaves/leaveReturn/returnToWorkIndicator		This field is currently not supported.
workerLeaves/leaves/leaveReturn/returnStatus		
workerLeaves/leaves/leaveReturn/returnStatus/returnStatusCode		
workerLeaves/leaves/leaveReturn/returnStatus/returnStatusCode/codeValue		
workerLeaves/leaves/leaveReturn/returnStatus/deniedIndicator		This field is currently not supported.
workerLeaves/leaves/leaveReturn/returnStatus/deniedReasonCode		This field is currently not supported.
workerLeaves/leaves/leaveReturn/certifications		This field is currently not supported.
workerLeaves/leaves/lea		This field is currently not supported.

veReturn/comment		
workerLeaves/leaves/leaveReturn/links		This field is currently not supported.

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](#).

Worker Leaves

Response Code	Condition	messageTxt	Tips to Handle
200 OK	Worker leaves when a valid Associate Object ID (AOID) is passed.		Fields with null values are not supported fields and need to be ignored.
400 Bad Request	Unsuccessful response when Invalid AOID is passed.	<pre>"userMessage": { "messageTxt": "JSON is invalid." }</pre>	

Chapter 3

Use Case: Applying Leave Absence for an Employee

Use Case Description

This use case is to describe how to request a leave for an employee using leave absent API.

API Usage

Worker Leaves

Method	URI	Description
POST	/events/hr/v1/worker.leave.absence.request	This API is to request for leave for an employee. In the request payload, this API expects associateID and work assignment ID , which can be fetched from worker API call.
GET	/events/hr/v1/worker.leave.absence.request/meta	

Application Scope

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

**/hr/workerInformationManagement/workerManagement/workerLeaveManagement/
worker.leave.absence.request**

Supported Actors

Request Parameter roleCode Value	Supported (Y/N)?
Practitioner	Y
Employee	N
Manager	N
Admin	Y

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 for the **leave absent** meta to the ADP API endpoint.
2. The ADP API endpoint responds to your consumer application with the meta payload. Your consumer application processes the meta payload to validate data included in the payload in step 3 and prompts the user to fix any validation issues. Otherwise, compose the payload for the **POST leave absent** request.
3. Your consumer application makes a **leave absent** request the ADP API endpoint.
4. The ADP API endpoint responds to the consumer application concerning the details of **worker leave request**.

Data Dictionary

The fields listed in the following table can be found in the ADP WorkForce Now UI by selecting **People > Employment > Employment Profile > Status section**.

Schema Location	Field Name	Required (Y/N)?	Note
events/data/eventContext/associateOID	NA	Y	

events/data/eventContext/workAssignmentID	NA	Y	
events/data/transform/effectiveDateTime	Pick an effective date.	Y	
events/data/transform/workerLeave			
events/data/transform/workerLeave/leaveAbsence			
events/data/transform/workerLeave/leaveAbsence/startDateTime	Leave of Absence Start Date	Y	
events/data/transform/workerLeave/leaveAbsence/expectedEndDateTime	Leave of Absence Expected Return Date	Y	
events/data/transform/workerLeave/leaveAbsence/leaveTypeCode			
events/data/transform/workerLeave/leaveAbsence/leaveTypeCode/codeValue	Leave Reason	Y	
events/data/transform/workerLeave/leaveAbsence/paymentStatusCode			
events/data/transform/workerLeave/leaveAbsence/paymentStatusCode/codeValue	Paid Leave of Absence	Y	
events/data/transform/workerLeave/leaveAbsence/comment			This is not currently supported.
events/data/transform/workerLeave/leaveAbsence/comment/textValue	NA		This is not currently supported.

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](#).

Worker Leaves

Response Code	Condition	messageTxt
200 Success	Success response when request for leave for a worker.	
400 Bad Request	Unsuccessful response when Invalid associate ID is passed.	{ "messageType": "error", "message": "Employee position with id G3MFT49T8KFAWMM5999 doesn't exist", "contextType": "GLOBAL" }
400 Bad Request	Unsuccessful response when Invalid date format is passed.	{ "messageType": "error", "message": "Invalid Date Format", "contextType": "GLOBAL" }
400 Bad Request	Unsuccessful response when Invalid leave type code is passed.	{ "messageType": "error", "message": "Termination reason code EDU1 is invalid", "contextType": "GLOBAL" }
400 Bad Request	Unsuccessful response when Invalid payment status is passed.	{ "messageType": "error", "message": "PAID LOA is invalid", "contextType": "GLOBAL" }
400 Bad Request	Unsuccessful response when Invalid work assignment ID is passed.	{ "messageType": "error", "message": "Employee position with id 59658833 doesn't exist", "contextType": "GLOBAL" }
400 Bad Request	Unsuccessful response when end date is earlier than start date.	{ "messageType": "error", "message": "End Date has to be after or same as Start Date!", "contextType": "GLOBAL" }
400 Bad Request	Unsuccessful response when leave already exists.	{ "messageType": "error", "message": "Leave record already exist and can not create new leave.", "contextType": "GLOBAL" }

Chapter 4

Use Case: Changing an Employee's Leave of Absence

Use Case Description

This use case describes how to change an employee's leave information.

API Usage

Worker Leaves

Method	URI	Description
POST	/events/hr/v1/worker.leave.change	This API is to change leave for an employee. In the request payload, this API expects associateID and work assignment ID , which can be fetched from worker API call.
GET	/events/hr/v1/worker.leave.change/meta	

Application Scope

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

**/hr/workerInformationManagement/workerManagement/workerLeaveManagement/
worker.leave.change**

Supported Actors

Request Parameter roleCode Value	Supported (Y/N)?
Practitioner	Y
Employee	N
Manager	N
Admin	Y

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 for **leave change meta** to the ADP API endpoint.

2. The ADP API endpoint responds to your consumer application with the meta payload. Your consumer application processes the meta payload to validate data included in the payload in step 3 and prompts the user to fix any validation issues. Otherwise, compose the payload for the **POST leave change** request.
3. Your consumer application makes a **leave change** request the ADP API endpoint.
4. The ADP API endpoint responds to the consumer application concerning the details of **worker changed leave**

Data Dictionary

The fields listed in the following table can be found in the ADP WorkForce Now UI by selecting **People > Employment > Employment Profile > Status section**.

Schema Location	Field Name	Required (Y/N)?	Note
events/data/eventContext/associateOID	NA	Y	
events/data/eventContext/workAssignmentID	NA	Y	
events/data/transform/effectiveDateTime	Pick an effective date.	Y	
events/data/transform/workerLeave/leaveAbsence/startDateTime	Leave of Absence Start Date	Y	
events/data/transform/workerLeave/leaveAbsence/expectedEndDateTime	Leave of Absence Expected Return Date	Y	
events/data/transform/workerLeave/leaveAbsence/leaveTypeCode/codeValue	Leave Reason	Y	
events/data/transform/workerLeave/leaveAbsence/paymentStatusCode/codeValue	Paid Leave of Absence	Y	
events/data/transform/workerLeave/leaveAbsence/comment/textValue	NA		Currently this field is not supported.

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	Condition	messageTxt
200 Success	Success response when changing an employee's leave.	
400 Bad Request	Unsuccessful response when passing an invalid AOID.	{ "messageType": "error", "message": "Employee position with id G3MFT49T8KFAWMM51243 doesn't exist", "contextType": "GLOBAL" }
400 Bad Request	Unsuccessful response when passing an invalid date format.	{ "messageType": "error", "message": "Invalid Date Format", "contextType": "GLOBAL" }
400 Bad Request	Unsuccessful response when passing an invalid effective date.	{ "messageType": "error", "message": "Effective Date is not the same as Effective Date used to create Leave Request.", "contextType": "GLOBAL" }
400 Bad Request	Unsuccessful response when passing an invalid leaveTypeCode .	{ "messageType": "error", "message": "Termination reason code EDU01 is invalid", "contextType": "GLOBAL" }
400 Bad Request	Unsuccessful response when passing an invalid paymentStatus .	{ "messageType": "error", "message": "Invalid Payment Status Code: Should be YES or NO.", "contextType": "GLOBAL" }
400 Bad Request	Unsuccessful response when passing an invalid workAssignment ID .	{ "messageType": "error", "message": "Employee position with id 59658833_10264 doesn't exist", "contextType": "GLOBAL" }

Chapter 5

Use Case: Returning an Employee's Leave of Absence

Use Case Description

This use case describes the employee's leave return request.

API Usage

Worker Leaves

Method	URI	Description
POST	/events/hr/v1/worker.leave.return.request	This API is to change an employee's leave. In the request payload, this API expects associateID and work assignment ID , which can be fetched from worker API call.
GET	/events/hr/v1/worker.leave.return.request/meta	

Application Scope

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

**/hr/workerInformationManagement/workerManagement/workerLeaveManagement/
worker.leave.return.request**

Supported Actors

Request Parameter roleCode Value	Supported (Y/N)?
Practitioner	Y
Employee	N
Manager	N
Admin	Y

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 for **leave return meta** to the ADP API endpoint.

2. The ADP API endpoint responds to your consumer application with the meta payload. Your consumer application processes the meta payload to validate data included in the payload in step 3 and prompts the user to fix any validation issues. Otherwise, compose the payload for the **POST leave return** request.
3. Your consumer application makes a **leave return** request the ADP API endpoint.
4. The ADP API endpoint responds to the consumer application concerning the details of **worker leave return request**.

Data Dictionary

The fields listed in the following table can be found in the ADP WorkForce Now UI by selecting **People > Employment > Employment Profile > Status section**.

Schema Location	Field Name	Required (Y/N)?	Note
Schema Location	Field Name	Required (Y/N)?	Note
events/data/eventContext/associateOID	NA	Y	
events/data/eventContext/workAssignmentID	NA	Y	
events/data/eventContext/leaveStartDate	Leave of Absence Start Date	Y	
events/data/transform/eventReasonCode/codeValue	Leave Reason	Y	
events/data/transform/leaveReturn/returnDateTime	Leave Return Date	Y	
events/data/transform/leaveReturn/returnStatus/returnStatusCode/codeValue		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](#).

Worker Leaves

Response Code	Condition	messageTxt

200 Success	Success response when leave return request is done.	
400 Bad Request	Unsuccessful response when passing an invalid date format.	<pre>{ "messageType": "error", "message": "Invalid Date Format", "contextType": "GLOBAL" }</pre>
400 Bad Request	Unsuccessful response when passing an invalid leave reason code.	<pre>{ "messageType": "error", "message": "Termination reason code ERs is invalid", "contextType": "GLOBAL" }</pre>
400 Bad Request	Unsuccessful response when passing an invalid return status.	<pre>{ "messageType": "error", "message": "Invalid status code. Please enter value as 'A'", "contextType": "GLOBAL" }</pre>

Chapter 6

Use Case: Cancelling an Employee's Leave of Absence

Use Case Description

This use case describes the leave cancel request.

API Usage

Worker Leaves

Method	Uniform Resource Identifier (URI)	Description
POST	/events/hr/v1/worker.leave.cancel	This API is to change leave for an employee. In the request payload, this API expects associateID and work assignment ID which can be fetched from worker API call.
GET	/events/hr/v1/worker.leave.cancel/meta	

Application Scope

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

**/hr/workerInformationManagement/workerManagement/workerLeaveManagement/
worker.leave.cancel**

Supported Actors

Request Parameter roleCode Value	Supported (Y/N)?
Practitioner	Y
Employee	N
Manager	N
Admin	Y

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 for **leave cancel meta** to the ADP API endpoint.
2. The ADP API endpoint responds to your consumer application with the meta payload. Your consumer application processes the meta payload to validate data included in the payload in step 3 and prompts the user to fix any validation issues. Otherwise, compose the payload for the **POST leave cancel** request.
3. Your consumer application makes a **leave cancel** request the ADP API endpoint.
4. The ADP API endpoint responds to the consumer application concerning the details of **worker leave cancel request**.

Data Dictionary

The fields listed in the following table can be found in the ADP WorkForce Now UI by selecting **People > Employment > Employment Profile > Status section**.

Schema Location	Field Name	Is Required (Y/N)?	Note
events/data/eventContext/associateOID	NA	Y	
events/data/eventContext/workAssignmentID	NA	Y	
events/data/eventContext/leaveStartDate	Leave of Absence Start 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](#).

Worker Leaves

Response Code	Condition	messageTxt
200 Success	Success response when an employee's leave is cancelled.	
400 Bad Request	Unsuccessful response when passing an invalid AOID.	<pre>{ "messageType": "error", "message": "Employee position with id G3MFT49T8KFAWMM5 doesn`t exist", "contextType": "GLOBAL" }</pre>
400 Bad Request	Unsuccessful response when passing an invalid leave start date.	<pre>{ "messageType": "error", "message": "The employee position does not exist. Please check the payload! ", "contextType": "GLOBAL" }</pre>
400 Bad Request	Unsuccessful response when passing an invalid work assignment ID.	<pre>{ "messageType": "error", "message": "Employee position with id G3MFT49T8KFAWMM5 doesn`t exist", "contextType": "GLOBAL" }</pre>

Chapter 7

Known Issues and Limitations

Issue 1: Effective date can't be updated when changing the leave information using the leave change API

Impacted APIs

Method	URI	roleCode Value
POST	/events/hr/v1/worker.leave.change	Practitioner

Issue Description

Using the leave change API, it's not possible to update the effective date. The API returns a validation error.

Suggested Workaround

To achieve this, the consumer first needs to cancel the existing leave and then apply it again with a new effective date.

Issue 2: Leave request for a terminated employee is not possible through the API

Impacted APIs

Method	URI	roleCode Value
POST	/events/hr/v1/worker.leave.cancel	Practitioner

Issue Description

The API can't move an employee to the leave status when the employee is terminated.

The API will return the following error validation:

```
{
  "messageType": "error",
  "message": "Employee position with id 61967145N is already terminated",
  "contextType": "GLOBAL"
}
```

Suggested Workaround

This can be done through the ADP Workforce Now UI.

Issue 3: Event notification isn't supported

Impacted APIs

Method	URI	roleCode Value
POST	/events/hr/v1/worker.leave.change	

POST	/events/hr/v1/worker.leave.absence.request	
POST	/events/hr/v1/worker.leave.return.request	
POST	/events/hr/v1/worker.leave.cancel	Practitioner

Issue Description

The leave retrieve, change, and cancel and return APIs don't support event notification.

Suggested Workaround

There are no suggested workarounds available.