



Guide

# Work Assignment Retire API Guide for Vantage

Published on  
Jan 05, 2021 5:47AM

Last modified  
Feb 16, 2022 3:30PM





## 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.

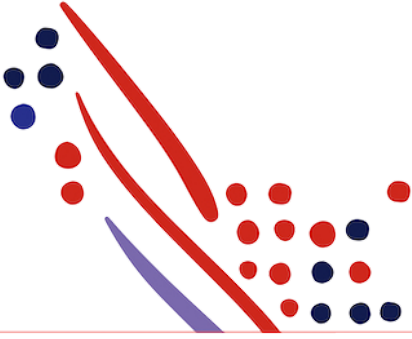
All other trademarks are the property of their respective owners.

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  
Jan 05, 2021 5:47 AM

Last modified  
Feb 16, 2022 3:30 PM



# Table of Contents

## Chapter 1

### **Use Case: Updating Work Assignment Status of an Employee**

Use Case Description

API Usage

Application Scope

Supported Actors

Request Header Parameters

Sequence of Interactions

Responses

Data Dictionary

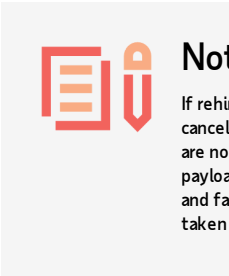
# Use Case: Updating Work Assignment Status of an Employee

## Use Case Description

This use case updates the employee work assignment status as Retired.

## API Usage

Method	Uniform Resource Identifier (URI)	Description	GitHub Sample Request Payload	GitHub Sample Response Payload
POST	/events/hr/v1/worker.work-assignment.retire	Updates work assignment status for an employee.	<a href="#">work-assignment-retire-employee-http-200-request.json</a>	<a href="#">work-assignment-retire-employee-http-200-response.json</a>
GET	/events/hr/v1/worker.work-assignment.retire/meta	Returns event metadata information.	NA	<a href="#">work-assignment-retire-meta-http-200-response.json</a>



**Note**  
 If rehireEligibleIndicator and cancelAutomaticPayIndicator are not passed in request payload, default values true and false respectively will be taken into consideration.

## 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:

**/hr/workerInformationManagement/workerManagement/workAssignmentManagement/worker.workAssignment.retire**

## Supported Actors

Request Parameter roleCode Value	Usage
Practitioner	Updates work status for an employee as a practitioner. A system user is considered a practitioner.

## Request Header Parameters

None, in addition to the ADP standard header parameters.

## Sequence of Interactions

1. Your consumer application makes a request for **/hr/v1/worker.work-assignment.retire/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 /hr/v1/worker.work-assignment.retire**
3. Your consumer application makes a **/hr/v1/worker.work-assignment.retire** request the ADP API endpoint.
4. The ADP API endpoint responds to the consumer application concerning the details of **/hr/v1/worker.work-assignment.retire**

## Responses

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

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

Response Code	Condition	messageTxt	GitHub Sample Request Payload	GitHub Sample Response Payload	Tips to Handle
200 OK	Update work status as retired for an employee with valid data.	NA	<a href="#">work-assignment-retire-employee-http-200-request.json</a>	<a href="#">work-assignment-retire-employee-http-200-response.json</a>	
200 OK	Update work status as retired for an employee with empty <b>LastWorkedDate</b>	NA	<a href="#">update-work-assignment-retire-with-empty-last-working-date-http-200-request.json</a>	<a href="#">update-work-assignment-retire-with-empty-last-working-date-http-200-response.json</a>	When LastworkedDate is empty then effective date value will be update in lastworkeddate field
200 OK	Update work status as retired for an employee with empty <b>reason code</b>	NA	<a href="#">update-work-assignment-retire-with-empty-reason-code-http-200-request.json</a>	<a href="#">update-work-assignment-retire-with-empty-reason-code-http-200-response.json</a>	
200 OK	Update work status as retired for an employee with empty <b>TerminationNoticeDate</b>	NA	<a href="#">update-work-assignment-retire-with-empty-termination-notice-date-http-200-request.json</a>	<a href="#">update-work-assignment-retire-with-empty-termination-notice-date-http-200-response.json</a>	
400 Bad Request	Update work status as retired for an employee	<b>"userMessage":</b> { <b>"messageTxt":</b> "You	<a href="#">update-work-assignment-retire-</a>	<a href="#">update-work-assignment-retire-</a>	

	with <b>invalid reason code</b>	have entered an invalid value for Employee Job Data-Reason Code. Select a valid value from the list provided" }	with-invalid-reasonCode-codeValue-http-400-request.json	with-invalid-reasonCode-codeValue-http-400-response.json	
400 Bad Request	Update work status as retired for an employee with <b>invalid effective date</b>	"userMessage": { "messageTxt": "Can not construct instance of java.util.Date from String value '06-15-2020': not a valid representation (error: Failed to parse Date value '06-15-2020': Can not parse date \"06-15-2020\": 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\\\")\\n\" }"	update-work-assignment-retire-with-invalid-effectiveDate-http-400-request.json	update-work-assignment-retire-with-invalid-effectiveDate-http-400-response.json	
400 Bad Request	Update work status as retired for an employee with <b>effective date earlier than last worked date</b>	"userMessage": { "messageTxt": "The termination or retirement date must be on or after the last date worked. (1000,299)\\n\" }"	update-work-assignment-retire-with-effectiveDate-earlier-than-lastWorkedDate-http-400-request.json	update-work-assignment-retire-with-effectiveDate-earlier-than-lastWorkedDate-http-400-response.json	
400 Bad Request	Update work status as retired for an employee with <b>invalid LastworkedDate</b>	"userMessage": { "messageTxt": "The date you entered is invalid. Enter a valid date (-Date Last Worked). (40,10)\\n\" }	update-work-assignment-retire-with-invalid-lastWorkedDate-http-400-request.json	update-work-assignment-retire-with-invalid-lastWorkedDate-http-400-response.json	
400 Bad Request	Update work status as retired for an employee with <b>invalid TerminationNoticeDate</b>	"userMessage": { "messageTxt": "The date you entered is invalid. Enter a valid date (-Resignation Notice Date). (40,10)\\n\" }	update-work-assignment-retire-with-invalid-rehireEligibleIndicator-http-400-request.json	update-work-assignment-retire-with-invalid-rehireEligibleIndicator-http-400-response.json	
400 Bad Request	Update work status as retired for an employee with <b>invalid RehireEligibleIndicator</b>	"userMessage": { "messageTxt": "Can not construct instance of boolean from String value 'tes': only \\\"true\\\" or \\\"false\\\" recognized\\n\" }"	update-work-assignment-retire-with-invalid-rehireEligibleIndicator-http-400-request.json	update-work-assignment-retire-with-invalid-rehireEligibleIndicator-http-400-response.json	
400 Bad Request	Update work status as retired for an employee with <b>invalid cancelAutomationPayIndicator</b>	"userMessage": { "messageTxt": "Can not construct instance of boolean from String value 'abc': only \\\"true\\\" or	update-work-assignment-retire-with-invalid-cancelAutomaticPayInd	update-work-assignment-retire-with-invalid-cancelAutomaticPayInd	

		<pre>\\"false\\" recognized\\n" }</pre>	<a href="#">icator-http-400-request.json</a>	<a href="#">icator-http-400-response.json</a>	
403 Bad Request	Update the ReportsTo details for an invalid employee in the API URI.	<pre>"userMessage": { "messageTxt": "You do not have the authorization to access the server. Contact your system administrator." }</pre>	<a href="#">update-work-assignment-retire-with-invalid-employee-http-403-request.json</a>	<a href="#">update-work-assignment-retire-with-invalid-employee-http-403-response.json</a>	

## Data Dictionary

Resources listed in the following table can be accessed in the ADP ADP Vantage user interface (UI) by selecting **People > Employment > Job Information**→**Job Dates**.

Schema Location	Field Name	Is Required (Y/N)	Note
/effectiveDateTime	<b>Effective Date</b>	Y	Effective date must be on or after the last date worked.
/eventReasonCode/code Value	<b>Reason</b>	N	
/worker/workAssignment/lastWorkedDate	<b>Date Last worked</b>	N	If LastworkedDate is not passed then effective date value will be update in Date last worked field. field
/worker/workAssignment/terminationNoticeDate	<b>Resignation Notice Date</b>	N	
/worker/workAssignment/rehireEligibleIndicator	<b>Eligible For Rehire</b>	N	If rehireEligibleIndicator is not passed in payload then default value: true will be updated in eligible for Rehire flag.
/worker/workAssignment/terminatePaymentInstruction/cancelAutomaticPayIndicator	<b>Not Displayed</b>	N	If rehireEligibleIndicator is not passed in payload then default value: false will be considered.