

## Chapter 2

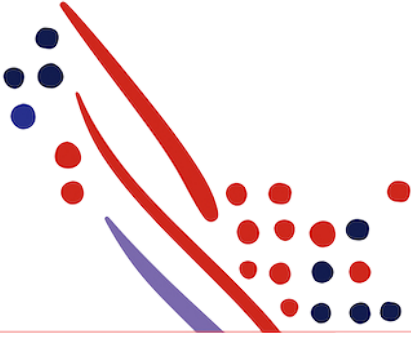
# Use Case: Publishing an Assessment Package

from Job Applicant External Screening APIs  
(Assessment APIs) Guide for ADP Workforce Now  
Guide

Published on  
Jun 17, 2020 4:58PM

Last modified  
Apr 27, 2023 3:50PM





## 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.

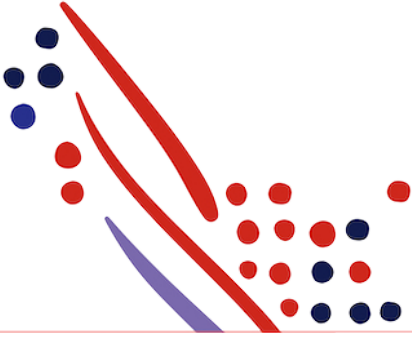
All other trademarks are the property of their respective owners.

Copyright © 2023 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  
Jun 17, 2020 4:58PM

Last modified  
Apr 27, 2023 3:50PM



# Chapter Contents

## Chapter 2

### **Use Case: Publishing an Assessment Package**

- Use Case Description
- API Usage
- Application Scope
- Supported Actors
- Sequence of Interactions
- Data Dictionary
- Responses

## Chapter 2

# Use Case: Publishing an Assessment Package

## Use Case Description

This use case describes how to use the Publish Package API to publish and update an assessment package in ADP Workforce Now.

For more information refer [Job Applicants for ADP WFN](#).

## API Usage

Method	Uniform Resource Identifier (URI)	Description
POST	/events/staffing/v1/job-applicant.external-screening.packages.modify	Publishing or updating an assessment package.

## Application Scope

The canonical URI corresponding to the Job Applicant External Screening 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:

**/staffing/staffingManagement/positionSeekerManagement/applicantManagementExternal/externalScreening.packages.modify**

## 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	Is Supported?	Usage when Supported
practitioner	Y	Publishes job assessment packages.

## Sequence of Interactions

1. Your consumer application makes a **/events/staffing/v1/job-applicant.external-screening.packages.modify** request the ADP API endpoint.

- The ADP API endpoint responds to the consumer application concerning the details of **/events/staffing/v1/job-applicant.external-screening.packages.modify**

## Data Dictionary

**Important:** The ADP Workforce Now user interface (UI) is not yet implemented.

Schema Location	Field Name	Is it Required in the Request Payload?	Note
/events/data/eventContext/screeningTypeCode/codeValue		Y	
/events/data/eventContext/agencyCode		Y	Vendor Name
/events/data/eventContext/agencyCode/codeValue		Y	Code Value = ASSESSMENT
/events/data/transform/externalScreeningPackages		Y	
/events/data/transform/externalScreeningPackages/screeningPackageID		Y	Should be unique for each package.
/events/data/transform/externalScreeningPackages/packageName		Y	Can be anything, per the user's preference.
/events/data/transform/externalScreeningPackages/packageDescription		N	
/events/data/transform/externalScreeningPackages/packageStatusCode		Y	
/events/data/transform/externalScreeningPackages/packageStatusCode/codeValue		Y	Code Value = ASSESSMENT
/events/data/transform/externalScreeningPackages/packageAmount		Y	
/events/data/transform/externalScreeningPacka		Y	

ges/packageAmount/amountValue			
/events/data/transform/externalScreeningPackages/packageAmount/currencyCode		N	
/events/data/transform/externalScreeningPackages/screeningLinkEffectiveDate		Y	
/events/data/transform/externalScreeningPackages/screeningLinkExpirationDate		Y	

## Responses

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

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

For more information refer [Job Applicants for ADP WFN](#).

Response Code	Condition	messageTxt	Tips to Handle
200 OK	Publishing an assessment package.		
200 OK	Publishing more than one assessment packages.		
200 OK	Updating published package details.		
200 OK	While publishing a package without <b>packageAmount</b> .		
400 Bad Request	While publishing a package without <b>screeningTypeCode</b> .	"userMessage": { "messageTxt": "Bad Request - screeningTypeCode should not be blank" }	
400 Bad Request	While publishing a package with invalid <b>screeningTypeCode</b> .	"userMessage": { "messageTxt": "Bad Request - screeningTypeCode should not be blank or should be ASSESSMENT" }	Default <b>screeningTypeCode</b> is Assessment.
		"userMessage": {	

400 Bad Request	While publishing a package without <b>agencyCode</b> .	"messageTxt": "Bad Request - agencyCode should not be blank" }	
400 Bad Request	While publishing a package without <b>screeningPackageID</b> .	"userMessage": { "messageTxt": "Bad Request - [screeningPackageID should not be blank against agency_code: Agency1]" }	
400 Bad Request	While publishing a package without <b>packageName</b> .	"userMessage": { "messageTxt": "Bad Request - [packageName should not be blank against screeningPackageID: Package1]" }	
400 Bad Request	While publishing a package without <b>packageStatusCode</b> .	"userMessage": { "messageTxt": "Bad Request - [packageStatusCode should not be blank against screeningPackageID: Package1]" }	
400 Bad Request	While publishing a package without <b>screeningLinkEffectiveDate</b> .	"userMessage": { "messageTxt": "Bad Request - [screeningLinkEffectiveDate should not be blank against screeningPackageID: Package1]" }	
400 Bad Request	While publishing a package with blank <b>screeningLinkExpirationDate</b> format.	"userMessage": { "messageTxt": "Bad Request - [screeningLinkExpirationDate should not be blank against screeningPackageID: Package1]" }	
500 Internal Server Error	While trying to publish multiple packages at a time with the same <b>screeningPackageID</b> .	"userMessage": { "messageTxt": "Internal Server Error - BackgroundScreeningDao: saveScreeningPackages: Exception occurred while persisting Screening data: BackgroundScreeningDao: saveScreeningPackages: Screening data not persisted" }	Screening Package ID should be unique for every package.