

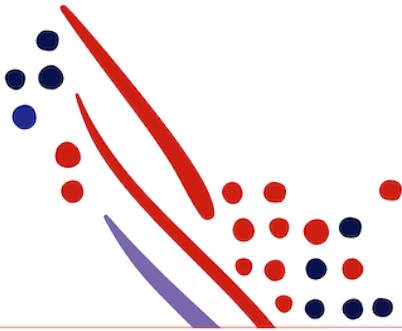


Guide

Digital Signature API guide for ADP iHCM

Published on
Apr 13, 2022, 08:46 AM

Last modified
Nov 26, 2024, 02:45 PM



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 © 2024 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
Apr 13, 2022, 08:46 AM

Published on
Nov 26, 2024, 02:45 PM

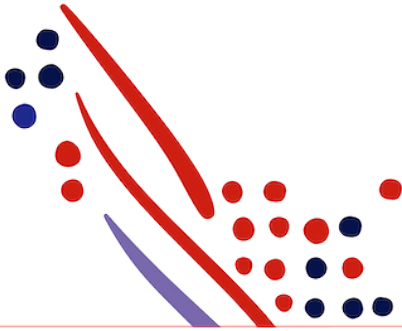


Table of Contents

Chapter 1

Introduction

- Flow Diagram
- Digital Signature Request Creation
- What's New in this guide

Chapter 2

API 2 & 4 – Get & delete the events

Chapter 3

API 3 – Get event notification callback content ("package")

- Summary
- Request Header Information
- Supported OData Query Options
- API Response
- Data Dictionary

Chapter 4

API 5 – Get documents to be signed, reviewed or for information

- Summary
- Request Header Information
- Supported OData Query Options
- API Response

Chapter 5

API 6 – Get document metadata

- Summary
- Request Header Information
- Supported OData Query Options
- API Response

Chapter 6

API 7 – Post event about the signature process status

- Summary
- Request Header Information
- Supported OData Query Options
- API Response
- Data Dictionary

Chapter 7

API 8 – Post signed documents in specified location

Summary

Request Header Information

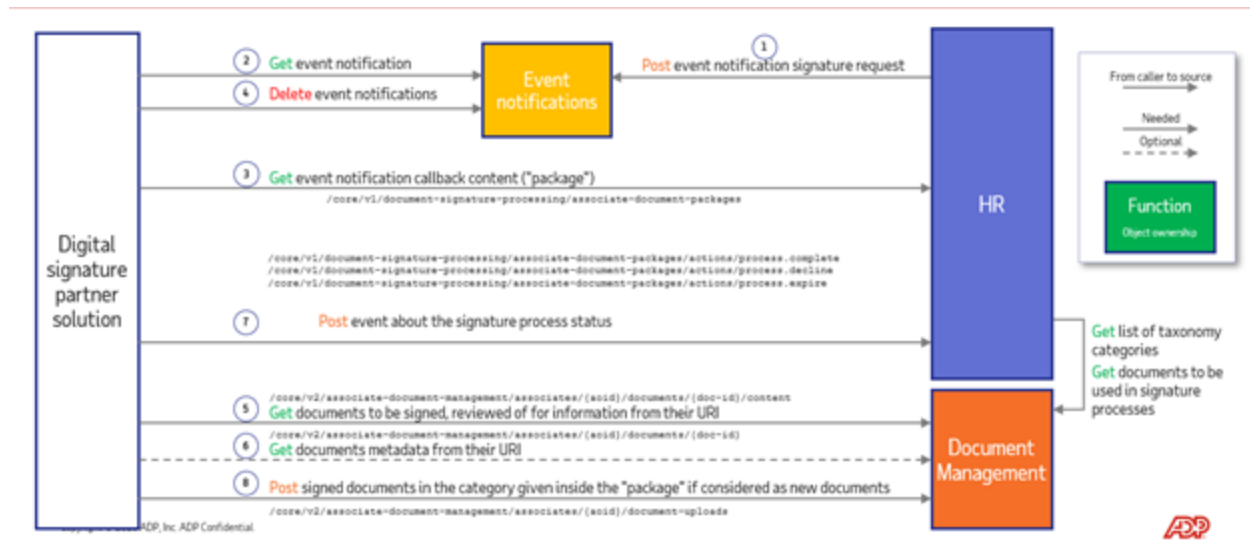
Supported OData Query Options

Data Dictionary

Introduction

IHCM has introduced a new Digital Signature feature to allow expert users to request a digital signature to employees for one or more documents, such as contracts and policy documents, then store the signed document safely in iHCM. This document explains the flow involved for this process and describes the APIs used to complete this process.

Flow Diagram

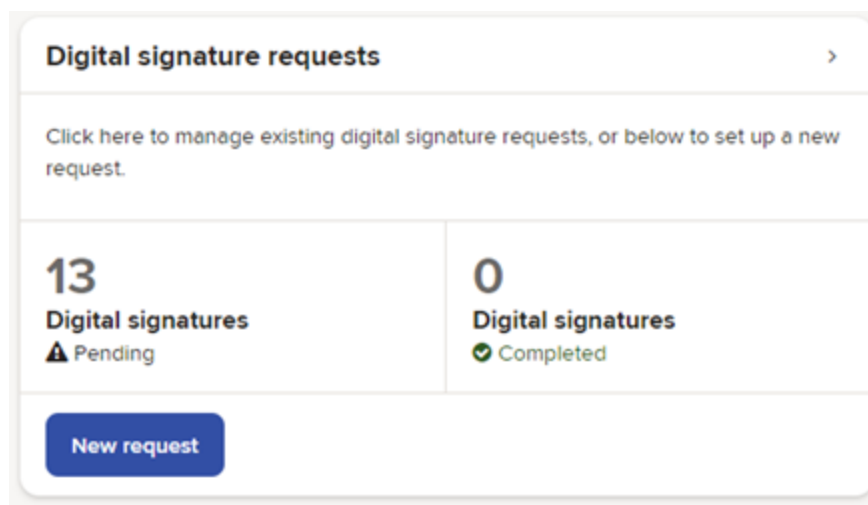


Digital Signature Request Creation

To request a digital signature from an employee, the expert user must be logged into iHCM. They then must use the navigation below to create or view a Digital Signature Request:

Navigation: Expert > Documents > Digital Signature Request

The user will be able to see the Digital Signature tile. Clicking on this tile will show the existing digital sign requests. To create a new request, click on the "New request" button within the tile.



Clicking on the "New request" button will open a wizard with 3 steps:

1 Add details 2 Add documents 3 Add Signatories

Package Name *

Demo Package

Description

Demo Package

Expiry Date

31/01/2022

Cancel

In step 1 the user must enter a package name, description and package request expiry date, then click on Next.

1 Add details 2 Add documents 3 Add Signatories

Select Employee *

1002 - Fernan Amil

Search Document name

<input type="checkbox"/>	Invoice for Deolive HCM Employee Documents/Sickness	Upload date 17 Mar 2020
<input type="checkbox"/>	DE604344-Fixed HCM Employee Documents/Sickness	Upload date 27 Nov 2019

Cancel Back Next

In Step 2 "Select employee" the user chooses to whom they want to send the digital sign request. After selecting the employee all the documents will be listed. The user then selects the documents that they would like to be signed as part of the digital signature process. They can select any number of documents, there is no limitation, they then click on Next.

1 Add details 2 Add documents 3 Add Signatories

+ Add

<input checked="" type="radio"/>	Fernan Amil Reference number 1002
<input type="radio"/>	Ambati Rayudu Reference number 777
<input type="radio"/>	Anderson SG Reference number 1039

Cancel Back Submit

In step 3, the system will show the signatories list in signatory order. The first signatory will be populated automatically, this is always the employee from whom they are requesting the digital signature. The user can add or remove more signatories by clicking on the "+ Add" button or remove by using the "Remove" button.

After adding all the required details, the user can submit the request. iHCM will save this information and an event notification will be posted to the Marketplace for the new request.

Posting an event notification to Marketplace (API 1 in the diagram) is internal to ADP. Once the event notification is posted to Marketplace, the partners will be notified via Marketplace about this request.

What's New in this guide

In this section we will announce any new revisions to the current set of Address APIs.

	Date	Description	Section
1		Creation	
2	3rd April 2024	Last name Prefix added for NL Context	API 3 – Get event notification callback content ("package")

Chapter 2

API 2 & 4 – Get & delete the events

Please refer to the [ADP Marketplace Event Notifications document](#).

Chapter 3

API 3 – Get event notification callback content ("package")

Summary

This API is used when the partner wants to fetch the digital signature package details. Its response includes all the information such as details about the package, signatories, and documents. It should be invoked with Package-Id as the query parameter. The callback uri is provided by the event notification messages Marketplace API.

Method	Uniform Resource Identifier (URI)	Description	Response Code
Get	/core/v1/document-signature-processing/associate-document-packages with a package ID in parameter	The partner application retrieves the "package" containing the information needed to start a signature process	200 OK 400 Bad Request 401 Unauthorized 403 Forbidden 404 Not Found 500 Internal Server Error 503 Service Unavailable

Request Header Information

There are no request header parameters, in addition to the ADP standard header parameters.

Supported OData Query Options

There are no OData options to consider for this use case.

API Response

[Digital Signature Example](#)

Data Dictionary

Schema Location	Description
documentPackage/packageId/id	Identifier of the package
documentPackage/packageSignByDate	Date when the signature process must be finished
documentPackage/eSignatureRequester/associateOID	Identifier of the requester
documentPackage/eSignatureRequester/roleCode	Role of the requester
documentPackage/eSignatureRequester/formattedName	Usually the first and last name of the requester
documentPackage/eSignatureRequestedDate	Date when the process was triggered
documentPackage/reminder/trigger/duration	Duration after which a reminder must be sent to a signatory if the signatory has not responded to the invitation to sign
documentPackage/reminder/recurrenceIndicator	Information to tell the partner that other reminders must be sent after the first one
documentPackage/reminder/recurrencePattern	This allows to set a recurrence pattern about reminders to be sent to signatories. A simple pattern such as sending a reminder after a few days is advised, knowing that the signature process has an end date defined in packageSignByDate attribute
documentPackage/documents	Array of documents that are part of the signature process
documentPackage/documents[]/documentID	Unique identifier of a document
documentPackage/documents[]/documentName	Name of the document

documentPackage/documents[]/document Title	Title of the document
documentPackage/documents[]/categoryC ode/code	Category code The label corresponding to this code could be retrieved with /core/v2/associate-document- management/document-categories or /core/v2/organization-document-management/document- categories requests
documentPackage/documents[]/subCateg oryCode/code	Subcategory code The label corresponding to this code could be retrieved with /core/v2/associate-document- management/document-categories or /core/v2/organization-document-management/document- categories requests
documentPackage/documents[]/document Ranking	Rank of the document in the signature process
documentPackage/documents[]/eSignatur es	List of the signatures to get on the document
documentPackage/documents[]/eSignatur es[]/signatureTypeCode/code	Type of signature to apply to the document
documentPackage/documents[]/eSignatur es[]/signatory/person/ personName/givenName	First name of the signatory
documentPackage/documents[]/eSignatur es[]/signatory/person/ personName/familyName	Last name of signatory
documentPackage/documents[]/eSignatures[]/sign atory/person/ legalName/familyNamePrefix	Last name prefix of signatory (NL Country ONLY)
documentPackage/documents[]/eSignatur es[]/signatory/person/ communication/emails[]/emailUri	Email used to contact the signatory
documentPackage/documents[]/eSignatur es[]/signatory/person/ communication/mobiles[]/formattedNumb er	Mobile number used to contact the signatory
documentPackage/documents[]/eSignatur es[]/signatureRanking	Order in which signatories will sign
documentPackage/documents[]/eSignatur es[]/links[]	Link to where the signed document can be posted if a copy is made available to the signatory
documentPackage/documents[]/eSignatur es[]/links[]/linkID	Type of document (enum: PROOD, ORIGINAL, SIGNED). This is either the link to the ORIGINAL or the endpoints to post the PROOF and SIGNED documents

documentPackage/documents[]/eSignatures[]/links[]/href	<p>URI to send a signed document</p> <p>Note: Documents can be sent to the document management system or to the user by email after being signed.</p> <p>In case of an email, the URI should be in the "mailto" form and the payloadArguments lines in this table can be ignored.</p> <p>If the signed document must be stored, use the URI to contain the pointer to where to put it in the document management system. The payloadArguments lines are used in this context when the document management system is ADP Document Cloud.</p>
documentPackage/documents[]/eSignatures[]/links[]/payloadArguments[]	<p>Elements of context that may be needed for uploading a new document into the Document Management system such as:</p> <ul style="list-style-type: none"> • documentName • categoryCode • subCategoryCode <p>...</p>
documentPackage/documents[]/eSignatures[]/links[]/payloadArguments[]/argumentName	Name of the metadata field
documentPackage/documents[]/eSignatures[]/links[]/payloadArguments[]/argumentValue	Value of the metadata field
documentPackage/documents[]/links[]	Direct link to the document to sign in the document management system, to where the signed document and the proof file must go
documentPackage/documents[]/links[]/linkID	Type of link to use for the documents
documentPackage/documents[]/links[]/href	<p>URI to get or send a document</p> <p>The payloadArguments lines are used in this context when the document management system is ADP Document Cloud.</p>
documentPackage/documents[]/links[]/payloadArguments[]	<p>Elements of context that may be needed for uploading a new document into the Document Management system such as:</p> <ul style="list-style-type: none"> • documentName • categoryCode • subCategoryCode
documentPackage/documents[]/links[]/payloadArguments[]/argumentName	Name of the metadata field
documentPackage/documents[]/links[]/payloadArguments[]/argumentValue	Value of the metadata field

Chapter 4

API 5 – Get documents to be signed, reviewed or for information

Summary

This API is used when we want to fetch the document content to start the signature process. The document fetch URL can be prepared based on response of API 3. The partner will fetch the document content with this API and will store this on the partner side for further digital signing process.

Method	Uniform Resource Identifier (URI)	Description	Response Code
Get	/core/v2/associate-document-management/associates/{aoid}/documents/{doc-id}/content	The partner application retrieves the document content using the API. This URL can be prepared based on the package API response specified in API 3. The navigation below needs to refer in API 3. documentPackage > documents [] > links[] > href Make sure we need to use links.linkID type is "Original"	200 OK 400 Bad Request 401 Unauthorized 403 Forbidden 404 Not Found 500 Internal Server Error 503 Service Unavailable

Request Header Information

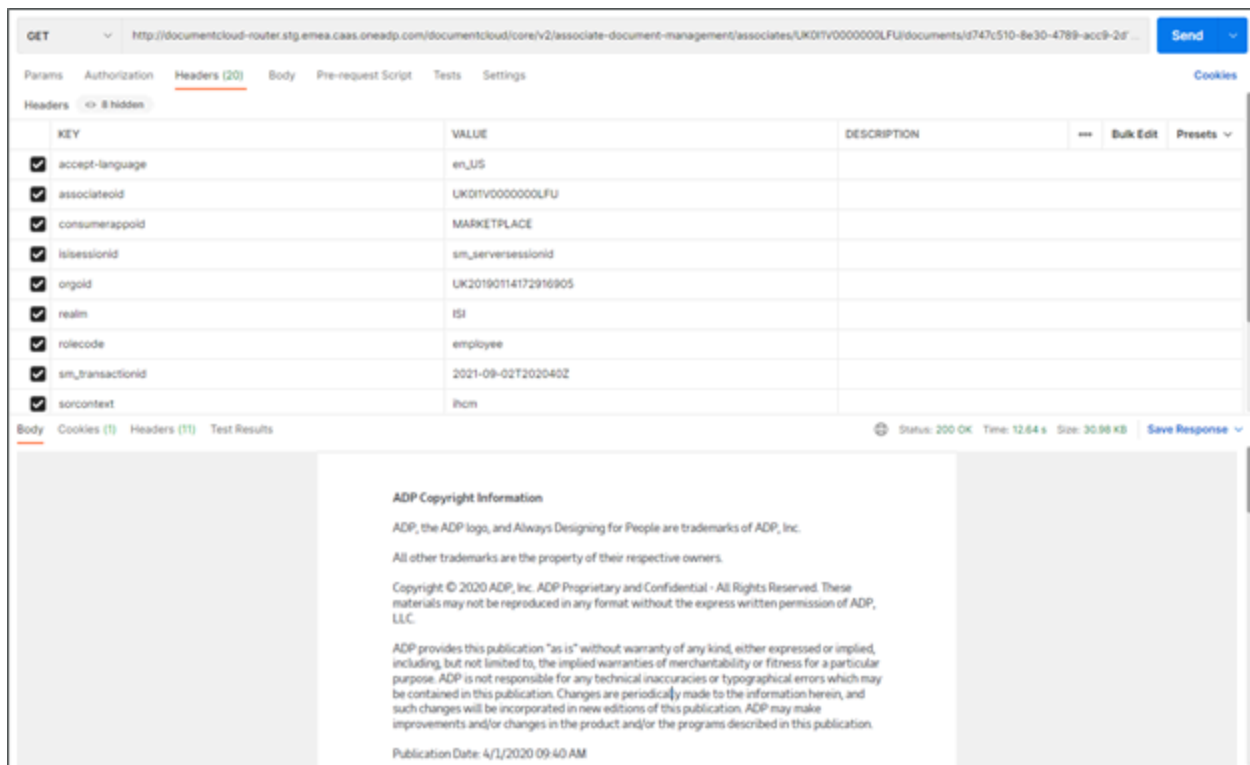
There are no request header parameters, in addition to the ADP standard header parameters.

Supported OData Query Options

There are no OData options to consider for this use case.

API Response

The response of this API will be document byte content. A sample API postman request & response is shown below.



Chapter 5

API 6 – Get document metadata

Summary

API 6 is used when we want to fetch the document properties and metadata. API 5 will provide only the document content; this API can be used to get all document properties (like Document Name, Type, size etc.).

Method	Uniform Resource Identifier (URI)	Description	Response Code
Get	<code>/core/v2/associate-document-management/associates/{aoid}/documents/{doc-id}</code>	<p>The partner application retrieves the document properties from this API. This URL can be prepared based on the package API response. Use the navigation below from API response.</p> <p>documentPackage > documents [] > links[] > href</p> <p>Make sure we need to use links.linkID type is "Original"</p>	200 OK 400 Bad Request 401 Unauthorized 403 Forbidden 404 Not Found 500 Internal Server Error 503 Service Unavailable

Request Header Information

There are no request header parameters, in addition to the ADP standard header parameters.

Supported OData Query Options

There are no OData options to consider for this use case.

API Response

<https://github.com/marketplace-esi/postman-samples/blob/master/ihcm/Document-Sign/success/Retrieve-document-metadata-properties-http-200-response.json>

Chapter 6

API 7 – Post event about the signature process status

Summary

This API is used when a package status changes on the partner application. The status can be communicated to the HR system with this API. This end point will be called with different package status (expire, decline, or complete) based on status.

The signature process can be updated with different states:

- **Complete:** The process is successful, and all documents have been processed by the signatories.
In this case, the partner must send a POST on `/core/v1/document-signature-processing/associate-document-packages/actions/process.complete`
- **Decline:** One of the signatories refused to sign one of the documents from the package and then the signature process fails.
In this case, the partner must send a POST on `/core/v1/document-signature-processing/associate-document-packages/actions/process.decline`
- **Expired:** The signature process exceeds the allocated time, and it is cancelled automatically on the partner side.
In this case, the partner must send a POST on `/core/v1/document-signature-processing/associate-document-packages/actions/process.expired`

Below is the sample text that needs to be sent as part of request. Here the package-id is the mandatory argument. Package-Id argument specifies to which package the status need to updated.

<https://github.com/marketplace-esi/postman-samples/tree/master/ihcm/Document-Sign/success/Post-package-status-payload-details.json>

Method	Uniform Resource Identifier (URI)	Description	Response Code
Post	<code>core/v1/document-signature-processing/associate-document-packages/actions/{status}</code>	Partner will call this API when the package status is updated in the partner application. Status (expired/decline/complete) will be passed in a query string based on the package status.	200 OK 400 Bad Request 401 Unauthorized 403 Forbidden 404 Not Found 500 Internal Server Error 503 Service Unavailable

Request Header Information

There are no request header parameters, in addition to the ADP standard header parameters.

Supported OData Query Options

There are no OData options to consider for this use case.

API Response

<https://github.com/marketplace-esi/postman-samples/tree/master/iHCM/Document-Sign/success/Update-package-status-http-200-response.json>

Data Dictionary

Schema Location	Description
actionID	Any unique string will be accepted.
actionMessage.messageText	Value of this property is always "complete"
actionMessage.messageArguments[0].argumentName	Value of this property is always "Packageid"
actionMessage.messageArguments[0].argumentValue	Value of this property should be the package id that we wanted to update the status for.
Links[0].href	This should be the URI appended with the package ID. /core/v1/document-signature-processing/associate-document-packages?package-id={0} In this format {0} should be replaced with package id that we are requesting to change the package status

Chapter 7

API 8 – Post signed documents in specified location

Summary

This API is used to upload the signed & proof documents to the HR system. This API will be called mostly at the end of a digital signing process and will be used to upload the document. The location for the document to be uploaded can be determined from the package API response (API 3). The package API also provides the payload to invoke this API.

Method	Uniform Resource Identifier (URI)	Description	Response Code
Post	/core/v2/associate-document-management/associates/{aoid} / document-uploads	Partner will call this API once the digital signature process has finished. The signed document must be uploaded using this API. Metadata can be retrieved with API 3	200 OK 400 Bad Request 401 Unauthorized 403 Forbidden 404 Not Found 500 Internal Server Error 503 Service Unavailable

When using this API form-data type must be used which contains document properties as json and the file byte content. The payload for this API looks like this:

KEY	VALUE	DESCRIPTION
uploadfile	Github procedure for postman-samples.pdf	
json	{ "documentName": "DocumentFileName.pdf", "documentTitle": "DocumentDisplayName", "documentTypeCode": { "code": "Employee" }, "categoryCode": { "code": "Incom-ec-ed" }, "subCategoryCode": { "code": "ec-ed-employee-details" }, "keywords": "", "author": { "associateOID": "UKDK3R9323645PX7", "roleCode": { "code": "Employee" }, "formattedName": "TEST USER" } }	Description

This form-data body will have 2 keys, 'uploadfile' and 'json'. The uploadfile key has as a value the binary file. The json key will have as a value the JSON formatted document properties. A sample JSON can we found using the url below.

<https://github.com/marketplace-esi/postman-samples/tree/master/ihcm/Document-Sign/success/Document-upload-payload-details.json>

The document upload location can be fetched from API 3 response using below path:

documentPackage > documents [] > links[] > href

SIGNED -- Location to upload the signed document

PROOF -- Location to upload the audit document

Document properties (like file name, display name, category etc..) can be identified using the path below:

documentPackage > documents[] > eSignatures[] > links[] > payloadArguments[]

Elements of context that may be needed for uploading a new document into the Document Management system such as:

- documentName
- categoryCode
- subCategoryCode

Request Header Information

There are no request header parameters, in addition to the ADP standard header parameters.

Supported OData Query Options

There are no OData options to consider for this use case.

Data Dictionary

Schema Location	Description
documentName	Name of the file.
documentTitle	Display Name of the file.
documentTypeCode.code	Always fixed values "Employee"
categoryCode.code	Category code, we will receive this value in API 3

subCategoryCode.code	Subcategory code, we will receive this value in API 3
Keywords	File search help key words. In general, use the name of the file.
autor.associateOID	AOID of the Employee to whom this document needs to send. This Id will be found in API 3.
autor.roleCode	Always fixed values "Employee"
autor.formattedName	Display name of the employee. Please refer to API 3 for this value.