



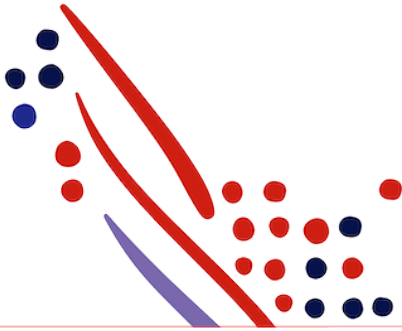
Guide

# ESI Marketplace Worker Payroll Documents API Catalog for ADP GlobalView

Published on  
Nov 28, 2023, 07:17 AM

Last modified  
Mar 11, 2025, 08:38 AM





## 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.

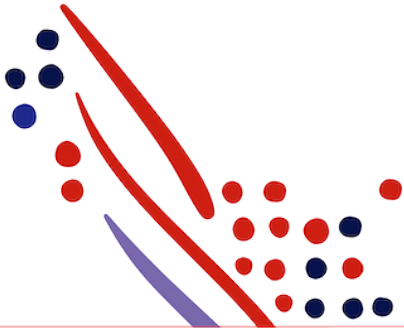
All other trademarks are the property of their respective owners.

Copyright © 2025 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  
Nov 28, 2023, 07:17 AM

Published on  
Mar 11, 2025, 08:38 AM



# Table of Contents

## Chapter 1

### About This Guide

Summary

Process Overview

What's New in this Guide?

## Chapter 2

### Worker Pay Statements API Guide

## Chapter 3

### Worker Tax Statements API Guide

# About This Guide

## Summary

The ESI Marketplace API Catalog for ADP GlobalView lists APIs supported by ADP GlobalView. It's used to help understand application programming interface (API) availability and evaluate work effort for your use case.

## Process Overview

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

	Actor	Task Description
1	Client practitioner	Set up the Practitioner in GlobalView to get the access to the client's worker data, pay and Tax Statements
2	Employee	Must have the ADP log-on credentials and SSO must be established to retrieve their own data

## What's New in this Guide?

In this section, we will announce any new revisions to the GlobalView Application Programming Interface (APIs) and this guide.

# Worker Pay Statements API Guide

## About this API

### Summary

The Pay Statements API Guide for ADP GlobalView describes how your application can access and manage pay statement information of workers in the ADP GlobalView application. It helps you to determine which use case best meets your business needs. It also guides you through the implementation of each use case.

### Overview

GlobalView offers APIs to retrieve the list of pay statements available for a specific employee. Your application can then retrieve details related to the pay statement, as well as the PDF pay statement for a specific pay period.

### Required Setup Steps

To support the B2B use case, a Communication User (Practitioner) must be created in GlobalView to provide access to the client's pay slips via your application. Otherwise, employees must have ADP log-on credentials and SSO must be established to retrieve their own pay statements. All payroll and payslip set-up must be in place and documents must be available in the CMS. The Consumer App set-up must be complete.

### Before You Begin

ADP provides the following Pay Statement APIs to stay in synch with an ADP application and do the following:

- Retrieve Pay statement list for a given employee.
- Retrieve Pay statement details for a given employee based on pay Id.
- Retrieve pay PDF for a specific pay period based on pay Id and Document Id.

API controls are in place allowing you to select or filter the requested data based on your needs.

### Exceptions

The Pay Statement APIs currently return pay statements for only the host country when an employee is on a global assignment and for only the latest assignment where an employee has been transferred between legal entities or rehired on a new Personnel number.

### 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

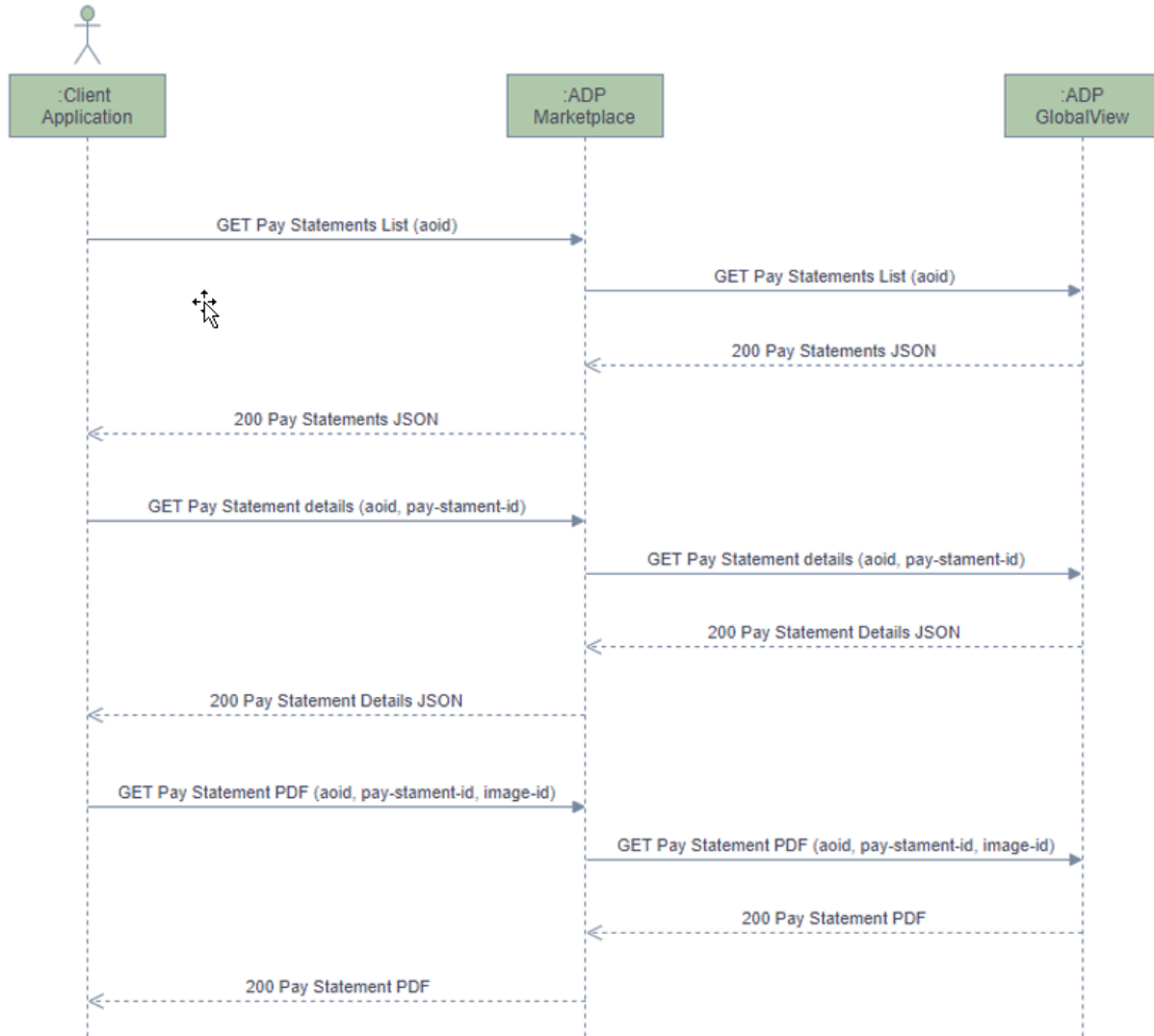
make successful API calls.

The following canonical needs to be added to your application scope to enable this use case:  
`/payroll/payrollManagement/payStatementManagement/payStatementViewing/payStatement.read`

### Sequence of Interactions

The below sequence diagram shows the sequence of actions between client application and GlobalView application.

1. Client application makes an API call to retrieve the Pay statements list by passing the associate-oid of the user.
2. The marketplace receives the request and then forwards the call to GlobalView which returns the pay statements list response json.
3. Client Application then makes an API call to get the pay statement details by providing the associate-oid and pay statement-id.
4. The marketplace receives the request and then forwards the call to GlobalView which returns the pay statement details json.
5. Client Application then makes an API call to get the pay statement Image by providing the associate-oid and pay statement-id / image id.
6. The marketplace receives the request and then forwards the call to GlobalView which returns the pay statement PDF.



### Error Message Format

The application returns an error message in a common format which has all the information about the problem. Below is the format of the Error message.

```
{
  "response": {
    "resourceUri": {
      "href": "/payroll/v1/workers/{aoid}/pay-statements" ----- URI of request
    },
    "applicationCode": {
      "code": "100-001", ----- Error Message Code
      "message": "User AOID NOT FOUND", ----- Error Message
      "typeCode": "error"
    }
  }
}
```

```

    },
    "methodCode": "GET", ----- HTTP Verb used
    "responseCode": "404" ----- HTTP Status Code
  }
}

```

## Responses

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

Response Code	Response Condition
200 OK	Retrieves list of pay statements
400 Bad Request	Bad request, e.g. incorrect query parameter
401 Unauthorized	The user is not authorized to access the API or retrieve the requested data
404 Not Found	The resource cannot be found, e.g. an invalid associate-oid
405 Method Not Allowed	An incorrect HTTP method has been used, e.g. POST instead of GET
412 Precondition Failed	Required header is not provided
413 Payload Too Large	The requested payload is too large to be managed
429 Too Many Requests	The consumer has sent more requests than is allowed by the server.
500 Internal Server error	Internal Server error
503 Service Unavailable	The service is not available

## Use Case Retrieving Pay Statements List

### Before You Begin

The Pay statements List API is used to retrieve the pay statements of an employee.

Each pay statement in the pay statement list has the following information

- statementImageURI
- payDetailUri
- payDate
- grossPayAmount
- netPayAmount

### API Usage

Method	Uniform Resource Identifier (URI)	Description
GET	/payroll/v1/workers/{associate-oid}/pay-statements	Retrieve the available payslips of an employee

### Query Parameters

Parameter	Description
-----------	-------------

numberoflastpaydates	Maximum pay statements which should be returned based on a request.
----------------------	---

### Use Case: Retrieving Pay Statement Details

#### Before You Begin

The Pay statement Details API is used to retrieve the pay statement details for a particular pay statement of an employee.

This API returns the following information:

- payPeriod
- payDate
- grossPayAmount
- netPayAmount
- earnings
- deductions

#### API Usage

Method	Uniform Resource Identifier (URI)	Description
GET	/payroll/v1/workers/{associate-oid}/pay-statements/{pay-statement-id}	Retrieve the details related to a pay statement of an employee

**associate-oid:** The associate-oid of the employee as registered in the ADP registry

**pay-statement-id:** The pay-statement-id of a specific pay statement.

### Use Case: Retrieving Pay Statement Image

#### Before You Begin

The Pay statement image API is used to retrieve the PDF image of a specific pay statement of an employee. The client application needs to specify the pay-statement-id and image-id for which PDF image needs to be retrieved.

#### API Usage

Method	Uniform Resource Identifier (URI)	Description
GET	/payroll/v1/workers/{associate-oid}/pay-statements/{pay-statement-id}/images/{image-id}.{image-extension}	Retrieve the PDF image of a specific pay statement for an employee

**associate-oid:** The associate-oid of the employee as registered in the ADP registry

**pay-statement-id:** The pay-statement-id of a specific pay statement.

**image-id:** The image-id of the pay statement

**image-extension:** The image extension. As of now only pdf is supported.

The pay-statement-id and image-id can be found part of imageStatementUri part of pay statement list response. Sample below

*/payroll/v1/workers/E11Q932K5FPEK0HF/pay-statements/3535373C3D3536353539353535353635/images/35393535353536353535373C3D2635.pdf*

### Frequently Asked Questions (FAQs)

**Question 1:** I see junk characters when I make a pay statement PDF call from POSTMAN

**Answer:** The pay statement PDF returns the PDF image of the pay statement. You need to save this to your workstation as a PDF and then view it.

## Chapter 3

# Worker Tax Statements API Guide

#### About this API

## Summary

The Worker Tax Statements API Guide for ADP GlobalView describes how your application can access and manage tax statement information of workers in the ADP GlobalView application. It helps you to determine which use case best meets your business needs. It also guides you through the implementation of each use case.

## Overview

GlobalView offers APIs to retrieve the list of tax statements available for a specific employee. Your application can then retrieve details related to the tax statement, as well as the PDF tax statement for a specific tax year.

## Required Setup Steps

To support the B2B use case, a Communication User (Practitioner) must be created in GlobalView to provide access to the client's tax forms via your application. Otherwise, employees must have ADP log-on credentials and SSO must be established to retrieve their own tax statements. All year-end and tax form set-up needs to be in place and tax forms must be present in the CMS. The Consumer App set-up must be complete.

Features ZOPRV and ZOMCP need to be configured to control the return of tax forms from prior assignments and other countries.

## Before You Begin

ADP provides the following Tax Statement APIs to stay in synch with an ADP application and do the following:

- Retrieve Tax statement list for a given employee.
- Retrieve Tax statement details for a given employee based on tax statement Id.
- Retrieve tax PDF for a specific year based on tax statement Id and image Id.

API controls are in place allowing you to select or filter the requested data based on your needs.

## 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:

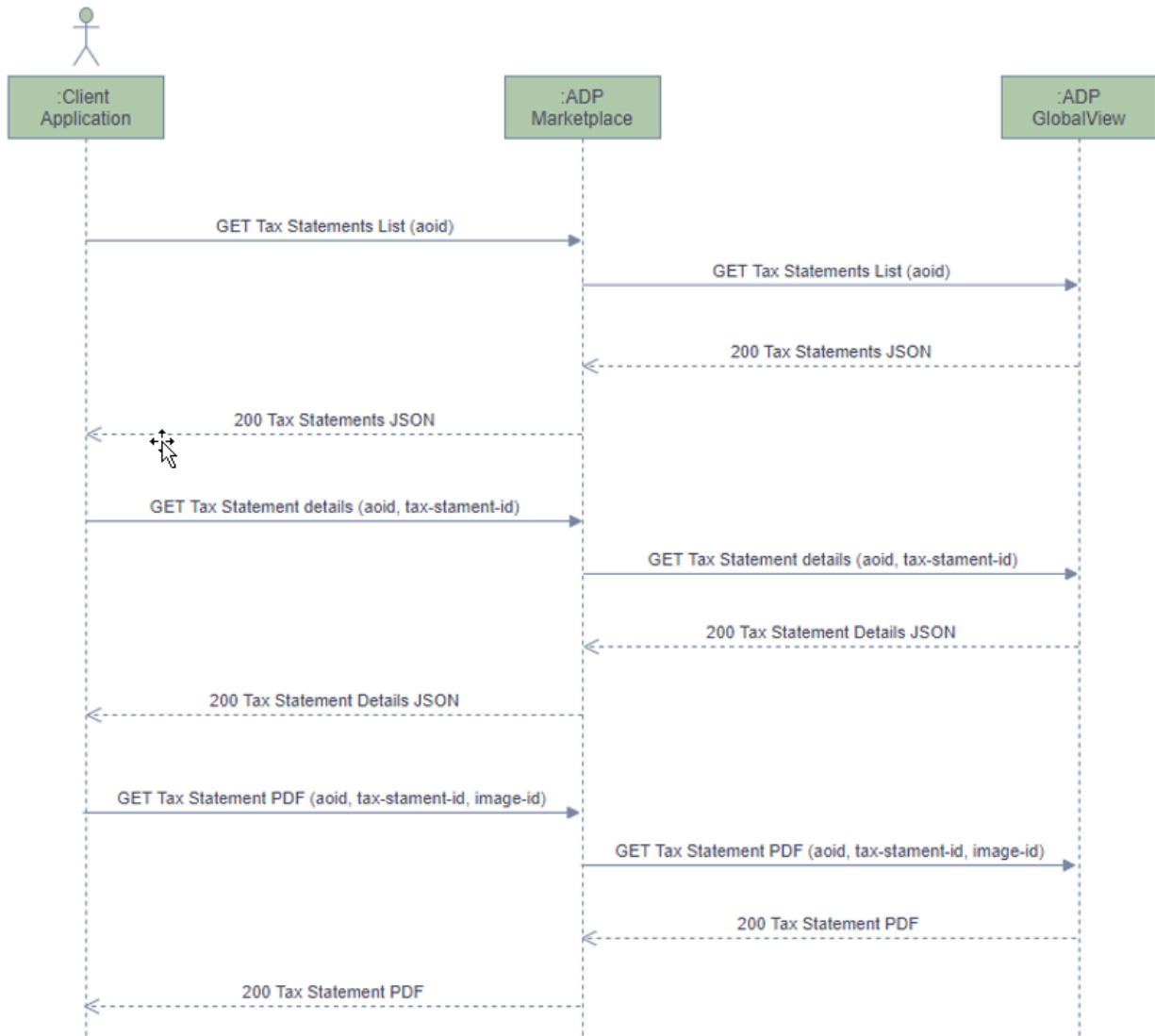
`/payroll/payrollManagement/taxStatementManagement/taxStatementViewing/taxStatement.read`

## Sequence of Interactions

The below sequence diagram shows the sequence of actions between client application and GlobalView application.

1. Client application makes an API call to retrieve the tax statements list by passing the associate-oid of the user.
2. The marketplace place receives the request and then forwards the call to GlobalView which returns the tax statements list response json
3. Client Application then makes an API call to get the tax statement details by providing the associate-oid and tax statement-id.
4. The marketplace receives the request and then forwards the call to GlobalView which returns the tax statement details json.
5. Client application then makes an API call to retrieve the tax statement Image by providing the associate-oid and tax statement-id/ document id.
6. The marketplace place receives the request and then forwards the call to GlobalView which returns the tax statements details/ tax statement PDF.





## Error Message Format

The application returns an error message in a common format which has all the information about the problem. Below is the format of the Error message.

```

{
  "response": {
    "resourceUri": {
      "href": "/payroll/v1/workers/{aoid}/worker-tax-statements" ----- URI of request
    },
    "applicationCode": {
      "code": "100-001", ----- Error Message Code
      "message": "User AOID NOT FOUND", ----- Error Message
      "typeCode": "error"
    },
    "methodCode": "GET", ----- HTTP Verb used
    "responseCode": "404" ----- HTTP Status Code
  }
}

```

## Responses

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

Response Code	Response Condition
200 OK	Retrieves list of tax statements

400 Bad Request	Bad request, e.g. incorrect query parameter
401 Unauthorized	The user is not authorized to access the API or retrieve the requested data
404 Not Found	The resource cannot be found, e.g. an invalid associate-oid
405 Method Not Allowed	An incorrect HTTP method has been used, e.g. POST instead of GET
412 Precondition Failed	Required header is not provided
413 Payload Too Large	The requested payload is too large to be managed
429 Too Many Requests	The consumer has sent more requests than is allowed by the server.
500 Internal Server error	Internal Server error
503 Service Unavailable	The service is not available

## Use Case: Retrieving Tax Statements List

### Before You Begin

The Tax statements List API is used to retrieve the tax statements of an employee.

Each tax statement in the tax statement list has the following information

- statementID
- statementName
- statementUri
- statementImageUri
- statementClassification
- statementYear
- form
- jurisdiction

### API Usage

Method	Uniform Resource Identifier (URI)	Description
GET	/payroll/v1/workers/{associate-oid}/tax-statements	Retrieve the available tax statements of an employee

### Query Parameters

Parameter	Description
numStatementYr	The number of statement years to return.

## Use Case: Retrieving Tax Statement Details

### Before You Begin

The Tax statements Details API is used to retrieve the details of a tax statement of an employee.

Each tax statement in the tax statement list has the following information

- statementID
- statementName

- statementYear
- form
- jurisdiction
- workerName

#### API Usage

Method	Uniform Resource Identifier (URI)	Description
GET	/payroll/v1/workers/{associate-oid}/tax-statements/{tax-statement-id}	Retrieve the available details of a tax statement of an employee

**associate-oid:** The associate-oid of the employee as registered in ADP registry

**tax-statement-id:** The tax-statement-id of a specific tax statement.

The tax-statement-id can be found part of StatementUri part of pay statement list response. Sample below

**statementUri**:{

**"href":**"/payroll/v1/workers/E17XDZ7ZKXV3Z05Z/tax-statements/353B3535353B3535373735363D5F57645D4649556452353B644B4E584846514B4E484D4A3535GV

#### Use Case: Retrieving Tax Statement Image

##### Before You Begin

The Tax statement image API is used to retrieve the PDF image of a specific tax statement of an employee. The client application needs to specify the tax-statement-id and image-id for which PDF image needs to be retrieved.

#### API Usage

Method	Uniform Resource Identifier (URI)	Description
GET	/payroll/v1/workers/{associate-oid}/tax-statements/{tax-statement-id}/images/{image-id}.{image-extension}	Retrieve the PDF image of a specific tax statement

**associate-oid:** The associate-oid of the employee as registered in ADP registry

**tax-statement-id:** The tax-statement-id of a specific tax statement.

**image-id:** The image-id of the tax statement

**image-extension:** The image extension. As of now only pdf is supported.

The tax-statement-id and image-id can be found part of imageStatementUri part of pay statement list response. Sample below

**statementImageUri**:{

**"href":**"/payroll/v1/workers/E17XDZ7ZKXV3Z05Z/tax-statements/353B3535353B3535373735363D5F57645D4649556452353B644B4E584846514B4E484D4A3535GV/images/005056B875CF1EDB82AF14B075FE85E3.pdf"

}

#### Frequently Asked Questions (FAQs)

**Question 1:** I see junk characters when I make a pay statement PDF call from POSTMAN

**Answer:** The pay statement PDF returns the PDF image of the pay statement. You need to save this to your workstation as a PDF and then view it.