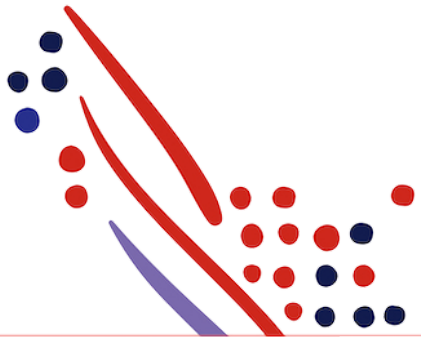
Guide

Understanding the Data Connector App Authorization Process

Published on
Nov 07, 2019 11:07PM

Last modified
Aug 06, 2021 9:09AM

Always Designing for People™



ADP Copyright Information

Published on
Nov 07, 2019 11:07PM

Last modified
Aug 06, 2021 9:09AM

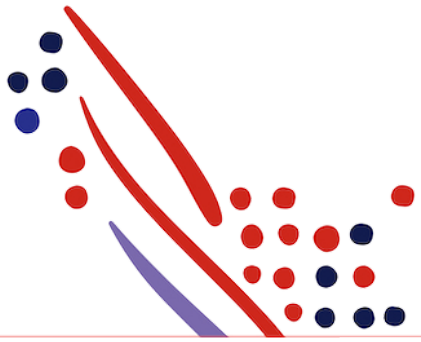


Table of Contents

Chapter 1

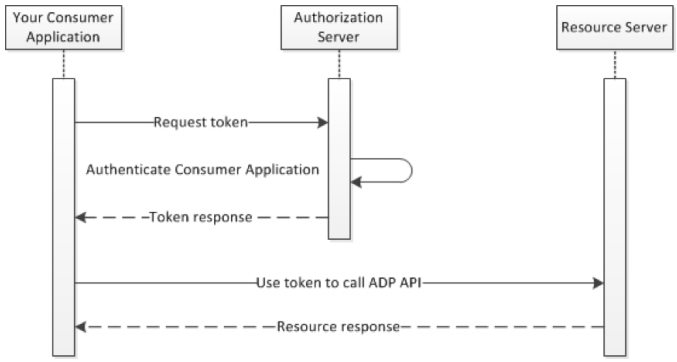
Introduction to the Data Connector App Authorization Process

Data Connectors are system-to-system applications that do not require end-user involvement. ADP uses the OAuth 2.0 authorization framework to permit applications to access ADP client data. OAuth provides a secure mechanism to grant resource (data) access to applications using access tokens on behalf of clients. This guide describes the OAuth *Client Credentials* grant used for system-to-system authorization.

The basic authorization flow required to access ADP APIs with the Client Credentials grant:

1. Your consumer application contacts the ADP token endpoint to get an access token.
2. ADP authenticates your consumer application, verifies the validity and provides an access token to your application.
3. Your consumer application uses the access token provided by ADP to access ADP APIs.

The following figure illustrates the authorization flow.



ADP provides developer libraries that you can use to take care of many of the implementation details of authorizing and gaining access to ADP APIs. If you choose not to use a library, follow the instructions in the next article, which describes the flow that underlies the available libraries.

Prerequisites

You must obtain the following from ADP in order to implement the Client Credentials grant with ADP:

- Signed Certificate
- Client Credentials

Chapter 2

Request an Access Token from ADP

Your application can request an access token by sending an HTTPS POST request to the token endpoint:
https://accounts.adp.com/auth/oauth/v2/token

The request must include the following parameters in the POST body:

Parameter	Description
grant_type	REQUIRED. Must be set to the value "client_credentials".

client_id	REQUIRED. The consumer application's account identifier, assigned during account registration or at secret reset.
client_secret	REQUIRED. The consumer application's account password, assigned during account registration or at secret reset.

In general, your consumer application should pass the client_id and client_secret parameters in the HTTP Authorization header using the HTTP Basic authentication scheme (or other designated scheme). The client_id and client_secret must be separated by a single colon (":") character and encoded within a base64-encoded string, as required by IETF RFC 2617.

Your consumer application must:

- Send the request with the X.509 certificate provided during registration.
- Pass all parameters in a URL-encoded format with UTF-8 character encoding as specified by the HTTP header Content-Type: application/x-www-form-urlencoded.

The actual request might look like the following example:

```
POST /auth/oauth/v2/token HTTP/1.1
    Host: accounts.adp.com
    Authorization: Basic QURQVGFibGV0OnRoZXRhYmxIdHBhc3N3b3Jk
    Content-Type: application/x-www-form-urlencoded
    grant_type=client_credentials
```

A successful response to this request contains the following fields in a JSON array:

Parameter	Description
access_token	The access_token parameter is set to the value of the access token issued by the ADP authorization service in exchange for the authorization code.
token_type	Identifies the type of token returned. At this time, this field always has the value Bearer.
expires_in	The expires_in parameter is set to the time remaining in the token's life (in seconds). For example, the value "3600" indicates that the access token will expire in one hour.

Chapter 3

Use an Access Token to Access ADP Data

After your consumer application obtains an access token from ADP, your application can use the access token to invoke ADP APIs to retrieve ADP data. Your application must:

- Pass the access token in the request using the Authorization request header field with the Bearer HTTP authorization scheme.
- Send the request with the X.509 certificate provided during registration.

The following shows an example request, with line breaks and spaces for readability.

```
GET /hr/v2/workers HTTP/1.1
    Host: api.adp.com
    Authorization: Bearer 024ded5f831d4483a9c606710026b09b
    Accept: application/json
```

Last updated: Aug 6, 2021