

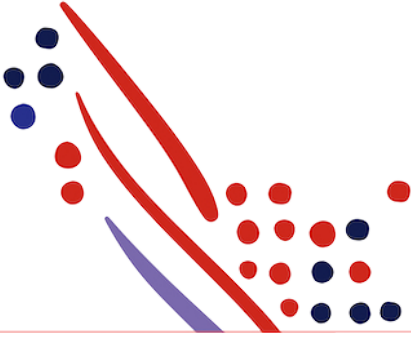
## Chapter 3

# Synchronous Subscription Events Workflow

from ADP Marketplace Integration - Separate  
Credentials (OAuth2) Guide

Published on  
Jul 22, 2021 4:08PM

Last modified  
Mar 30, 2023 9:53AM



## 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.

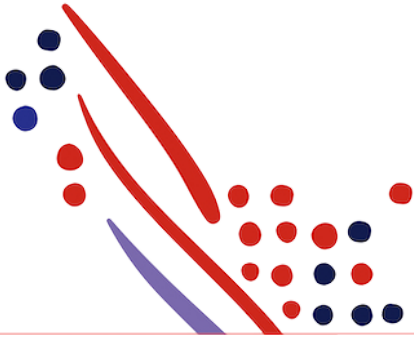
All other trademarks are the property of their respective owners.

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 programs described in this publication.

Published on  
Jul 22, 2021 4:08PM

Last modified  
Mar 30, 2023 9:53AM



# Chapter Contents

## Chapter 3

### **Synchronous Subscription Events Workflow**

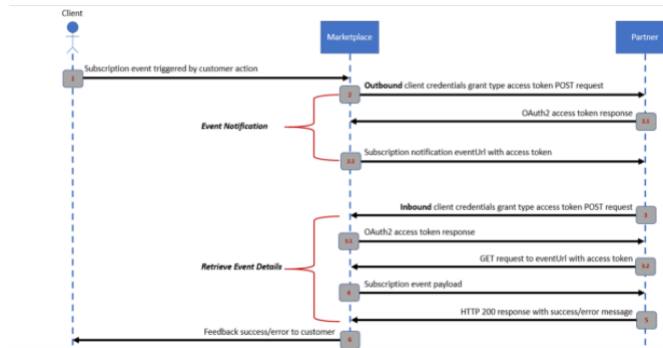
Chapter 3

# Synchronous Subscription Events Workflow

Use the appropriate domain depending on which Marketplace you are developing for:

ADP Marketplace U.S. Integration (<https://apps.adp.com>)

ADP Marketplace Canada Integration (<https://ca.apps.adp.com>)



+

1. An event is triggered by a customer action (e.g., subscribing to an application). This creates a SUBSCRIPTION\_ORDER event identified by a URL, e.g., <https://apps.adp.com/api/integration/v1/events/12345>.
2. ADP Marketplace will authenticate to your OAuth2 server's token endpoint and make a POST request using the client credentials grant type flow
  1. Your OAuth2 server should respond back with access token
  2. ADP Marketplace will utilize the access token to authenticate to the respective subscription event endpoint via a GET request and will providing you with the eventUrl which has the subscription details. As an example, the request will look like the following (GET <https://partnerapplication.com/subscription/create?eventUrl=https://apps.adp.com/api/integration/v1/events/12345>)
- Partner application initiates a POST request to ADP Marketplace's token endpoint:
  1. US Marketplace = <https://apps.adp.com/oauth2/token>  
Canada Marketplace = <https://ca.apps.adp.com/oauth2/token>.
  2. The request must be authenticated via basic authentication using the generated **Inbound** Client ID and Client Secret

Header	Query Parameter	Description
	grant_type	REQUIRED. The grant_type parameter must be set to the value <b>client_credentials</b> .
	scope	REQUIRED. The scope parameter must be set to <b>ROLE_APPLICATION</b>
Authorization		REQUIRED. The Authorization header will authenticate with the <b>Inbound Credentials</b> client id and client secret with basic authentication.
Accept		REQUIRED (one of the following). The Accept header must be set to the value <b>application/json</b> or <b>application/xml</b> depending on which format your application is expecting.
Content-Type		REQUIRED (one of the following). The Content-Type header must be set to the value

3. ADP Marketplace calls the Outbound Credentials endpoint to retrieve an access\_token from your application

4. Partner application initiates a GET request to the eventUrl provided from the subscription notification, including sending the access token in the Authorization: Bearer token header

By default eventUrl response will be in XML, utilize the Accept: application/json header if you wish to receive a JSON response. The partner application must not respond to the initial subscription notification until after a response is received from the eventUrl.

5. The subscription payload response would look something like the following

```
{
  "type": "SUBSCRIPTION_ORDER",
  "marketplace": {
    "partner": "ADP",
    "baseUrl": "https://apps.adp.com"
  },
  "applicationUuid": null,
  "flag": "DEVELOPMENT",
  "creator": {
    "uuid": "30a0c21c-e5f7-451d-a16f-149eac7061c2",
    "openId": "https://apps.adp.com/openid/id/30a0c21c-e5f7-451d-a16f-149eac7061c2",
    "email": "John.Doe@adp.com",
    "firstName": "John",
    "lastName": "Doe",
    "language": "en",
    "locale": "en-US",
    "address": null,
    "attributes": null
  },
  "payload": {
    "user": null,
    "company": {
      "uuid": "699da487-d17c-403e-90b7-aac7fb3e1d48",
      "externalId": "FFFFFFFFFFFFFFFF",
      "name": "ADP",
      "email": null,
      "phoneNumber": null,
      "website": null,
      "country": "US"
    },
    "account": null,
    "addonInstance": null,
    "addonBinding": null,
    "order": {
      "editionCode": "Trial + Monthly",
      "addonOfferingCode": null,
      "pricingDuration": "MONTHLY",
      "items": [],
      "customAttributes": []
    },
    "notice": null,
    "configuration": {
      "organizationOID": "FFFFFFFFFFFFFFFF",
      "applicationID": null,
      "associateOID": "636898NGDZCHEW8G"
    }
  },
  "returnUrl": null,
  "links": [
    {
      "rel": "oidcClient",
      "href": "https://apps.adp.com/api/account/v2/subscriptions/fa51b3a6-6198-4ed4-bf59-b5e418c43b0d/oidcClient"
    }
  ]
}
```

For more subscription related event examples please click [Here](#) to download.

6. Your application creates/updates account information for the subscriber details. Then, your application returns a JSON response back to the Marketplace event, and passes the following values depending on the below scenarios. The two variants of the status notifications are "success":true, or "success": "false". Here are sample JSON response documents:

#### Success

```
{  
  "accountIdentifier":"<OrganizationOID>",  
  "success":true  
}
```

#### False

```
{  
  "success": "false",  
  "errorCode": "USER_ALREADY_EXISTS",  
  "message": "Optional message about the user already existing on ISV"  
}
```



#### Info

The partner application should always return a 200 status with properly formatted JSON or XML as mentioned above. **Do not return a 500 or 404 status.**

**Note:** Once you have responded to the ADP Marketplace with either of the 200 Status response examples above you will no longer be able to pull the event payload. Please ensure you have consumed the event payload before responding to the ADP Marketplace.

In the event of an error, partner application should return only the following predefined error codes listed [here](#).

7. If a failure is sent the client will be notified via email as well as you the partner. Below are examples of the partner facing failure email as well as the client facing failure email.

#### Client Facing Email:

**Your Employee Contact Database for: ADP Workforce Now® purchase failed**

Hello Paul,

Your purchase of Employee Contact Database for: ADP Workforce Now® on the ADP Marketplace could not be completed because of a temporary error with the partner's integration.


No charge was made to your account.

Please try your purchase again later or contact Partnerx2 if this issue persists by visiting the solution's [Support Page](#)


---

Sincerely,  
The ADP Marketplace team


**Explore more HR solutions**




[Recruiting and Onboarding](#)




[Time and Labor Management](#)




[Financial Wellness](#)




[Productivity & Collaboration](#)



Build your HCM ecosystem.  
Choice, flexibility and security.  
All on ADP Marketplace.





[apps.adp.com](https://apps.adp.com)

**Partner Facing Email (will be enabled End of May 2022):**

**An event could not reach your endpoint.**

Hi Paul,

An ADP client has tried to purchase or update a subscription to Employee Contact Database for: ADP Workforce Now®. An event was sent, but was not consumed by your endpoint, so you will need to fix the issue with your endpoint. See event details below.

**What should you do?**

Please check and fix your endpoints to make sure they are up and running, then confirm by running an Integration Report to show that all endpoints are now accessible. If you do not fix the endpoint(s), we will continue to notify you to take action, and your sales could be impacted if clients are unable to successfully complete their purchase or make updates.

**Supporting documentation**

Review [this documentation](#) with articles on editing and testing your endpoints. If you need additional assistance, please file a ticket in our [help center](#).

**Event details:**

**Event type:** SUBSCRIPTION\_ORDER  
**Marketplace:** US Marketplace  
**Product name:** Employee Contact Database for: ADP Workforce Now®  
**Product Edition:** Classic Plan  
**Account identifier:** 123-abc-456-def

Thanks!  
The ADP Marketplace team



Build your HCM ecosystem.  
Choice, flexibility and security.  
All on ADP Marketplace.





[apps.adp.com](https://apps.adp.com)



## Info

The partner facing email will be triggered when:

1. You send a 200 success = false message(like the one shown above)
2. If your endpoint is down

It will contain links to our documentation on how to test your endpoint using the "**Integration Report**" as well as a link to put a ticket in if you need additional assistance.

**In the event of an error, partner application should return only the following predefined error codes listed [here](#).**

8. The return status information from the above step is used by ADP Marketplace to provide feedback to the customer via the ADP Marketplace UI.

- **Subscription cancel notification:** When a buyer cancels the subscription, your application will receive a notification (JSON message) through the registered URL. This means the client no longer wants to use the application. You should develop code to take the necessary actions as per your policy when you receive the cancel notification.
- You are required to implement a **subscription change** notification and **subscription status** notification as well. The technology and flow are the same as the implementation of the subscription create and subscription cancellation.