

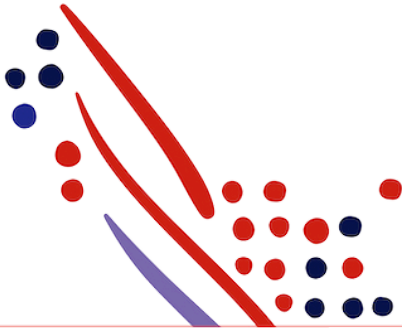
Chapter 6

Step 5: Validate ID Token

from Understanding the End-User App Authorization Process

Published on
Nov 12, 2019, 10:11 PM

Last modified
Jul 17, 2024, 03:40 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.

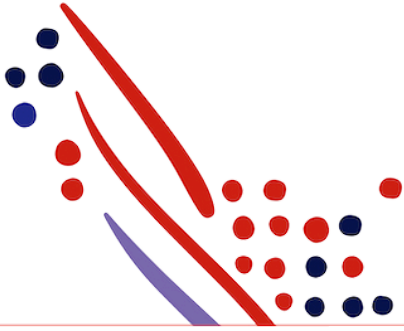
All other trademarks are the property of their respective owners.

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
Nov 12, 2019, 10:11 PM

Published on
Jul 17, 2024, 03:40 PM



Chapter Contents

Chapter 6

Understanding the End-User App Authorization Process

Step 5: Validate ID Token

An identity token is an assertion of the authentication of the end-user by the OpenID Connect identity provider. The assertion is provided as a JSON Web Token (JWT) object in the `id_token` parameter of the JSON object returned on a successful authorization code exchange.

A JWT contains three binhex encoded sections separated by dots ("."): a header, a set of asserted claims, and the signature of the header and the claims. Your consumer application must decode and validate the JWT.

The encoded claims section of the `id_token` decodes to the following JSON object.

```
{ "iss":"https://accounts.adp.com",
  "sub":"https://accounts.adp.com/user/G123123123123213/G456783748392999",
  "exp":1412793735,
  "iat":1412786535,
  "auth_time":1412786535,
  "nonce":"163141324",
  "azp":"c5348fb0-efd6-4632-a359-d6c2f5ab04f6",
  "c_hash":"2oYJkgWbM3jgT5QjuSfNaQ",
  "name":"Anthony Albright",
  "given_name":"Anthony",
  "family_name":"Albright",
  "email":"Anthony.Albright@blah.com"
}
```

Parameter	Description
iss	The issuer of the authentication assertion. In this case, theADP OpenID Connect Provider contains the following value: <code>https://accounts.adp.com</code> .
sub	The subject of the assertion. A unique identifier associated with the end-user authenticating with ADP.
aud	The party to which the <code>id_token</code> was issued. The value is your consumer application's <code>client_id</code> .
exp	The expiration time of the assertion. Your consumer application should not accept expired assertions.
iat	The time the assertion was issued. Your consumer application can use this parameter to determine the age of the assertion.
auth_time	The time when the end-user authentication occurred.
nonce	Returned if your consumer application provided this parameter in the authorize request. This parameter mitigates replay attacks.
azp	The party to which the <code>id_token</code> was issued. The value is your consumer application's <code>client_id</code> .
c_hash	Authorization code hash value. It is the binhex encoding of the left-most half of the authorization. The hash algorithm is the algorithm specified in the header section.

Normally, it is critical that you validate an ID token before you use it, but since you are communicating directly with ADP over an HTTPS channel and using your client credentials and signed certificate to authenticate yourself to ADP, you can be confident that the token you receive really comes from ADP and is valid. If your server passes the ID token to other components of your app, it is extremely important that the other components validate the token before using it.

Your consumer application may decode and minimally validate the `id_token` as follows:

- Validate the signature of the JWT.
- Validate that the value of the `nonce` parameter matches the value provided in the authorize request if your consumer application provided one.
- Validate that the value of the `iss` (Issuer) parameter matches `https://accounts.adp.com`.

- Validate that the "aud" (audience) parameter contains the client_id assigned to your consumer application.
- Validate that the "azp" (authorized party) parameter contains the client_id assigned to your consumer application.
- Validate that the id_token has not expired or that the current time is before the value specified in the "exp" parameter.
- Validate that the c_hash parameter matches the authorization code provided by the result of the authorization request to the ADP Authorization Service.