Providing User Authentication Information to OAuth 2.0 Clients
draft-hunt-oauth-v2-user-a4c-05

Abstract

This specification defines a way for OAuth 2.0 clients to verify the identity of the End-User and obtain consent based upon the authentication performed by an Authorization Server. The interactions defined by this specification are intentionally compatible with the OpenID Connect protocol.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 22, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of
the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction ................................................. 3
   1.1. Requirements Notation and Conventions ................. 3
   1.2. Terminology ........................................... 3
2. Authentication Messages ...................................... 4
   2.1. Authentication Request ................................. 4
   2.2. Authentication Response ............................... 7
   2.2.1. Error Responses .................................... 7
   2.3. Token Request .......................................... 7
   2.4. Token Response ......................................... 7
3. Privacy Considerations ....................................... 11
4. IANA Considerations ........................................... 11
   4.1. Authentication Method Reference Values Registry ..... 11
   4.1.1. Registration Template ............................... 12
   4.1.2. Initial Registry Contents .......................... 13
   4.2. OAuth Authorization Endpoint Response Types
        Registration ............................................ 14
   4.2.1. Registry Contents .................................. 14
   4.3. OAuth Parameters Registration ........................ 15
   4.3.1. Registry Contents .................................. 15
   4.4. OAuth URN Sub-Namespace Registration ................ 15
   4.4.1. Registry Contents .................................. 15
5. Security Considerations ..................................... 15
6. References .................................................. 15
   6.1. Normative References .................................. 15
   6.2. Informative References ................................. 16
Appendix A. Deltas from OpenID Connect ......................... 17
Appendix B. Acknowledgements .................................. 17
Appendix C. Document History .................................. 17
Authors’ Addresses ............................................. 19
1. Introduction

Section 4.1 of the OAuth 2.0 Authorization Framework [RFC6749] defines the Authorization Code Grant flow which defines a redirect flow, typically via a web browser, that enables confidential clients to obtain access and refresh tokens. As part of this flow, resource owners are authenticated via the user agent so that their consent may be obtained.

This document extends the OAuth 2.0 authorization request and response messages for the Authorization Code flow to also request authentication of the End-User and to return information about the authentication performed. The interactions defined by this specification are intentionally compatible with the OpenID Connect [OpenID.Core] protocol. See Appendix A for a description of the features that are present in this specification that are not present in or different from OpenID Connect.

This specification does not define a resource owner profile information API. It is assumed that other APIs such as the SCIM API [I-D.ietf-scim-api] or the OpenID Connect [OpenID.Core] UserInfo Endpoint could be used for this purpose.

1.1. Requirements Notation and Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.2. Terminology


This specification also defines the following terms:

Authentication Request
OAuth 2.0 Authorization Request using extension parameters and scopes defined by this specification to request that the End-User be authenticated by the Authorization Server to the Client.
Authentication Context
Information that the Relying Party can require before it makes an entitlement decision with respect to an authentication response. Such context can include, but is not limited to, the actual authentication method used or level of assurance such as ISO/IEC 29115 [ISO29115] entity authentication assurance level.

Authentication Context Class
Set of authentication methods or procedures that are considered to be equivalent to each other in a particular context.

Authentication Context Class Reference
Identifier for an Authentication Context Class.

Authentication Method
Specific means by which authentication is performed. In some cases, more than one authentication method may be used for a single authentication event.

Authentication Method Reference
Identifier for an Authentication Method.

End-User
Human participant.

ID Token
JSON Web Token (JWT) [I-D.ietf-oauth-json-web-token] that contains Claims about the Authentication event. It MAY contain other Claims.

Issuer
Entity that issues a set of Claims.

2. Authentication Messages

This specification extends the use of the authorization code flow defined in Section 4.1 of RFC 6749 [RFC6749] in ways that enable clients to request authentication as well as to obtain information about the authentication performed.

2.1. Authentication Request

In addition to the parameters defined in Section 4.1.1 of RFC 6749 [RFC6749], the following additional parameters and parameter values are defined:
response_type

REQUIRED. OAuth 2.0 Response Type value that determines the authentication processing flow to be used, including what parameters are returned from the endpoints used. Two "response_type" values are defined for use with this specification:

code
Use of this response type results in both an access token and an ID Token being returned from the token endpoint in exchange for an authorization code.

code_for_id_token
Use of this response type results in an ID Token but no access token being returned from the token endpoint in exchange for an authorization code.

prompt

OPTIONAL. Space delimited, case sensitive list of ASCII string values that specifies whether the Authorization Server prompts the End-User for reauthentication and consent. The defined values are:

none
The authorization server MUST NOT display any authentication or consent user interface pages. An error is returned if the End-User is not already authenticated or the Client does not have pre-configured consent for the requested Claims or does not fulfill other conditions for processing. This can be used as a method to check for existing authentication and/or consent.

login
Regardless of the current user authentication state, the Authorization Server SHOULD prompt the End-User for reauthentication. If it cannot prompt the End-User, it MUST return an error.

select_account
The Authorization Server SHOULD prompt the End-User to select a user account. This allows an End-User who has multiple accounts at the Authorization Server to select amongst the multiple accounts that they might have current sessions for. If it cannot prompt the End-User, it MUST return an error.
acr_values
OPTIONAL. Requested Authentication Context Class Reference values. Space-separated string that specifies the "acr" values that the Authorization Server is being requested to use for processing this Authentication Request, with the values appearing in order of preference. The Authentication Context Class satisfied by the authentication performed is returned as the "acr" Claim Value.

amr_values
OPTIONAL. Requested Authentication Method Reference values. Space-separated string that specifies the "amr" values that the Authorization Server is being requested to use for processing this Authentication Request, with the values appearing in order of preference. The Authentication Methods used for the authentication performed are returned as the "amr" Claim Value.

ui_hint
OPTIONAL. A helpful text message that should be displayed to the End-User during the authentication process. [[[ NOTE: It’s not clear what the use case for this is or how internationalization of the string would be performed. ]]]

id_token_hint
OPTIONAL. ID Token previously issued by the Authorization Server being passed as a hint about the End-User’s current or past authenticated session with the Client. If the End-User identified by the ID Token is logged in or is logged in by the request, then the Authorization Server returns a positive response; otherwise, it SHOULD return an error, such as "login_required". When possible, an "id_token_hint" SHOULD be present when "prompt=none" is used and an "invalid_request" error MAY be returned if it is not; however, the server SHOULD respond successfully when possible, even if it is not present. The Authorization Server need not be listed as an audience of the ID Token when it is used as an "id_token_hint" value.

For example, the client directs the User Agent to make the following HTTP request using TLS (with extra line breaks for display purposes only):

GET /authenticate?
  response_type=code
  &client_id=s6BhdRkqt3
  &redirect_uri=https%3A%2F%2Fclient.example.com%2Fcb
  &state=af0ifjsldkj
  &prompt=login
  Host: server.example.com
The authorization server MUST:

- Perform the normal OAuth 2.0 authorization process,
- MAY elect not to request consent if no access token is to be issued (i.e. this is an authentication only request),
- MUST re-authenticate the End-User if "prompt" contains the parameter "login",
- MUST obtain consent from the End-User if "prompt" contains the parameter "consent", and
- MUST return an error if "prompt" contains "none" and the End-User is not currently authenticated.

2.2. Authentication Response

Both when using "response_type=code" and when using "response_type=code_for_id_token", the response is identical to the one described in Section 4.1.2 of RFC 6749 [RFC6749].

2.2.1. Error Responses

In addition to those defined in Section 4.1.2.1 of RFC 6749 [RFC6749], an additional "error" type is defined. The error value "login_required" MUST be returned after an authentication request parameter "prompt" is provided containing value "none" and the End-User is found to be currently unauthenticated.

2.3. Token Request

When using "response_type=code", the token request is identical to the one described in Section 4.1.3 of RFC 6749 [RFC6749]. When using "response_type=code_for_id_token", the token request is also identical to the one described in Section 4.1.3 of RFC 6749, except that the "grant_type" value used MUST be set to "urn:ietf:params:oauth:grant-type:code-for-id-token" instead of "authorization_code".

2.4. Token Response

When the "authorization_code" "grant_type" is used, the authorization server issues an access token and optional refresh token as described in Sections 4.1.4 and 5.1 of RFC 6749 [RFC6749]. When the "urn:ietf:params:oauth:grant-type:code-for-id-token" grant type is used, the response is the same except that the access token and refresh token are omitted from the response. If the client
authentication failed or the request is invalid, the authorization server returns an error response as described in Section 5.2 of RFC 6749.

In addition to the response parameters described in Section 5 of RFC 6749, a JSON Web Token (JWT) [I-D.ietf-oauth-json-web-token] known as an ID Token is returned for both of these grant types using the "id_token" parameter. The ID Token contains the following claims:

**iss**
*REQUIRED. An identifier representing the issuer of the authentication. This MAY be the authorization endpoint URL.*

**sub**
*REQUIRED. An identifier for the authenticated subject. The same identifier MUST be returned for the same authenticated End-User on the same Client ID. The authenticated End-User’s "sub" value MAY change for different Client ID values.*

**aud**
*REQUIRED. Contains the Client ID of the client receiving the assertion as an audience value. Other audience values MAY also be present.*

**auth_time**
*REQUIRED. The time at which the End-User was authenticated, expressed in number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time. See [RFC3339] for details regarding date/times in general and UTC in particular. "auth_time" MAY be a time earlier than when the ID Token was issued, as defined by "iat".*

**iat**
*REQUIRED. The time at which the ID Token was issued, expressed in number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time. See [RFC3339] for details regarding date/times in general and UTC in particular.*

**exp**
*REQUIRED. The time at which the ID Token expires, expressed in number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time. See [RFC3339] for details regarding date/times in general and UTC in particular. Note that "expires_in" refers to the access token lifespan whereas "exp" refers to the ID Token lifespan.*
acr
OPTIONAL. Authentication Context Class Reference. String specifying an Authentication Context Class Reference value that identifies the Authentication Context Class that the authentication performed satisfied. The value "0" indicates the End-User authentication did not meet the requirements of ISO/IEC 29115 [ISO29115] level 1. Authentication using a long-lived browser cookie, for instance, is one example where the use of "level 0" is appropriate. An absolute URI or an RFC 6711 [RFC6711] registered name SHOULD be used as the "acr" value. Parties using this claim will need to agree upon the meanings of the values used, which may be context-specific. The "acr" value is a case sensitive string.

amr
OPTIONAL. Authentication Methods References. JSON array of strings that are identifiers for Authentication Methods used in the authentication. For instance, values might indicate that both password and OTP authentication methods were used. The definition of particular values to be used in the amr Claim is beyond the scope of this specification. Parties using this claim will need to agree upon the meanings of the values used, which may be context-specific. The "amr" value is an array of case sensitive strings. The following is a list of defined Authentication Method Reference values:

pwd
Password authentication, either by the user or the service if a client secret is used

pop
Proof of possession of a key

otp
One time password

fpt
Fingerprint biometric

eye
Retina scan biometric

vbm
Voice biometric
Confirmation by telephone call

Confirmation by SMS reply

Knowledge based authentication

Windows integrated authentication

Multiple factor authentication. When this is present, the other authentication methods used will also be included.

A non-normative example successful response with an ID Token follows (with line breaks within lines for readability):

HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache

{
  "access_token": "2YotnFZFEjr1zCsicMWpAA",
  "token_type": "example",
  "expires_in": 3600,
  "refresh_token": "tGzv3J0kF0XG5Qx2T1KWIA",
  "id_token": "eyJhbGciOiJub25lIn0.
           eyJpc3MiOiJodHRwczovL3NlcnZlc2Fzc2FkM0FkM1ZmZGQxNjE0MjJkZjNjMDg1N2MxN2NiMzFkY2YxMDIyZjU0ZDA4ZTg0M2U0ZDg4ZjI1MmYxN2IiLCJ0b2tlbiI6IjI6IiwiaWF0
           iV3YXRpdmVFU291cmNlIjoiNzIzM2M1MzY2ODM3OTk4N2UyZTc0NzI3OTQzMzY4MjU1MjUzNzRkMzQxMmQzOTczZGQzN2U1NjAwOTc1ZjEifQ.

As per the JWT specification, the encoded ID Token is separated into parts by the "." character. The first part ("eyJhbGciOiJub25lIn0") contains the signature algorithm and in this case decodes as:

{"alg": "none"}
The claim set is then decoded as:

```
{
  "iss":"https://server.example.com",
  "sub":"5dedcc8b-735c-405f-e029f",
  "aud":"s6BhdRkqt3",
  "auth_time":1367956096,
  "iat":1367956098,
  "exp":1368042496,
  "acr":"2",
  "example_extension_parameter":"example_value"
}
```

If the ID Token contains the claim "acr" and its value represents an authentication level greater than "2", the ID Token MUST be signed (have a signature "alg" value other than "none") and its signature MUST be validated.

All claims defined above MUST be understood before proceeding. Additional claims/parameters that are not understood MAY be ignored.

The client MUST verify that the "auth_time" value is not future dated and "exp" is not a date currently in the past.

3. Privacy Considerations

Profile URL values issued in the ID Token and MAY be directed identifiers. In other words, the identifier/URL returned is valid only for the "aud" indicated. This prevents multiple clients and non-OAuth clients from being able to gather and correlate information about individuals authenticated by the OAuth Authorization Server.

4. IANA Considerations

4.1. Authentication Method Reference Values Registry

This specification establishes the IANA Authentication Method Reference Values registry for ID Token "amr" claim array element values. The registry records the claim array element value and a reference to the specification that defines it. This specification registers the "amr" values defined in Section 2.1.

Values are registered on a Specification Required [RFC5226] basis after a two-week review period on the [TBD]@ietf.org mailing list, on the advice of one or more Designated Experts. However, to allow for the allocation of values prior to publication, the Designated
Expert(s) may approve registration once they are satisfied that such a specification will be published.

Registration requests must be sent to the [TBD]@ietf.org mailing list for review and comment, with an appropriate subject (e.g., "Request for access token type: example"). [Note to the RFC Editor: The name of the mailing list should be determined in consultation with the IESG and IANA. Suggested name: jwt-reg-review.]

Within the review period, the Designated Expert(s) will either approve or deny the registration request, communicating this decision to the review list and IANA. Denials should include an explanation and, if applicable, suggestions as to how to make the request successful. Registration requests that are undetermined for a period longer than 21 days can be brought to the IESG’s attention (using the iesg@iesg.org mailing list) for resolution.

Criteria that should be applied by the Designated Expert(s) includes determining whether the proposed registration duplicates existing functionality, determining whether it is likely to be of general applicability or whether it is useful only for a single application, and whether the registration makes sense.

IANA must only accept registry updates from the Designated Expert(s) and should direct all requests for registration to the review mailing list.

It is suggested that multiple Designated Experts be appointed who are able to represent the perspectives of different applications using this specification, in order to enable broadly-informed review of registration decisions. In cases where a registration decision could be perceived as creating a conflict of interest for a particular Expert, that Expert should defer to the judgment of the other Expert(s).

### 4.1.1. Registration Template

**Authentication Method Reference Name:**

The name requested (e.g., "example"). Because a core goal of this specification is for the resulting representations to be compact, it is RECOMMENDED that the name be short -- not to exceed 8 characters without a compelling reason to do so. This name is case-sensitive. Names may not match other registered names in a case-insensitive manner unless the Designated Expert(s) state that there is a compelling reason to allow an exception in this particular case.
Authentication Method Reference Description:
Brief description of the Authentication Method Reference (e.g., "Example description").

Change Controller:
For Standards Track RFCs, state "IESG". For others, give the name of the responsible party. Other details (e.g., postal address, email address, home page URI) may also be included.

Specification Document(s):
Reference to the document(s) that specify the parameter, preferably including URI(s) that can be used to retrieve copies of the document(s). An indication of the relevant sections may also be included but is not required.

4.1.2. Initial Registry Contents

- Authentication Method Reference Name: "pwd"
  Authentication Method Reference Description: Password authentication, either by the user or the service if a client secret is used
  Change Controller: IESG
  Specification Document(s): Section 2.1 of [[ this document ]]

- Authentication Method Reference Name: "pop"
  Authentication Method Reference Description: Proof of possession of a key
  Change Controller: IESG
  Specification Document(s): Section 2.1 of [[ this document ]]

- Authentication Method Reference Name: "otp"
  Authentication Method Reference Description: One time password
  Change Controller: IESG
  Specification Document(s): Section 2.1 of [[ this document ]]

- Authentication Method Reference Name: "fpt"
  Authentication Method Reference Description: Fingerprint biometric
  Change Controller: IESG
  Specification Document(s): Section 2.1 of [[ this document ]]

- Authentication Method Reference Name: "eye"
  Authentication Method Reference Description: Retina scan biometric
  Change Controller: IESG
  Specification Document(s): Section 2.1 of [[ this document ]]

- Authentication Method Reference Name: "vbm"
4.2. OAuth Authorization Endpoint Response Types Registration

This section registers the "response_type" values defined by this specification in the IANA OAuth Authorization Endpoint Response Types registry defined in RFC 6749 [RFC6749].

4.2.1. Registry Contents

  o Response Type Name: "code_for_id_token"
  o Change Controller: IESG
  o Specification document(s): Section 2.1 of [[ this document ]]
4.3. OAuth Parameters Registration

This section registers the following parameters in the IANA OAuth Parameters registry defined in RFC 6749 [RFC6749].

4.3.1. Registry Contents

- Parameter name: "amr_values"
  - Parameter usage location: Authorization Request
  - Change controller: IESG
  - Specification document(s): Section 2.1 of [[ this document ]]
  - Related information: None

- Parameter name: "ui_hint"
  - Parameter usage location: Authorization Request
  - Change controller: IESG
  - Specification document(s): Section 2.1 of [[ this document ]]
  - Related information: None

4.4. OAuth URN Sub-Namespace Registration

4.4.1. Registry Contents

This specification registers the value "grant-type:code-for-id-token" in the IANA urn:ietf:params:oauth registry established in An IETF URN Sub-Namespace for OAuth [RFC6755], which can be used to indicate that the content is a JWT.

- URN: urn:ietf:params:oauth:grant-type:code-for-id-token
  - Common Name: code-for-id-token grant type
  - Change Controller: IESG
  - Specification Document(s): Section 2.3 of [[this document]]

5. Security Considerations

This draft carries the same risk profiles as those outlined in the Security Considerations for RFC 6749 [RFC6749] and OAuth 2.0 Threat Model [RFC6819].

6. References

6.1. Normative References

[I-D.ietf-oauth-json-web-token]
  Jones, M., Bradley, J., and N. Sakimura, "JSON Web Token (JWT)", draft-ietf-oauth-json-web-token-24 (work in
progress), July 2014.


6.2. Informative References


Appendix A. Deltas from OpenID Connect

This appendix describes the features that are present in this specification that are not present in or different from OpenID Connect [OpenID.Core]. All other features present in both specifications have the same meanings.

New features added by this specification are:

code_for_id_token response type
   This specification defines the new "code_for_id_token" response type value.

urn:ietf:params:oauth:grant-type:code-for-id-token grant type
   This specification defines the new "urn:ietf:params:oauth:grant-type:code-for-id-token" grant type value.

amr claim values
   This specification defines a set of "amr" claim values.

amr_values parameter
   This specification defines the "amr_values" request parameter.

ui_hint parameter
   This specification defines the "ui_hint" request parameter.

auth_time required
   This specification requires that an ID Token always contain an "auth_time" claim.

Appendix B. Acknowledgements

The authors wish to thank the members of the OAuth working group for their contributions and comments.

Appendix C. Document History

[[ to be removed by the RFC editor before publication as an RFC ]]

-05
-04

-03

-02

-01 - PJH 2013-08-15
parameter.

-00 - PJH 2013-04-09

- Initial version

Authors’ Addresses

Phil Hunt (editor)
Oracle

Email: phil.hunt@yahoo.com

Anthony Nadalin
Microsoft

Email: tonynad@microsoft.com

Michael B. Jones
Microsoft

Email: mbj@microsoft.com
URI: http://self-issued.info/