|  |
| --- |
| [TOC](" \l "toc) |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Draft | B. de Medeiros | |  | N. Agarwal | |  | Google | |  | N. Sakimura, Ed. | |  | NRI | |  | J. Bradley | |  | Ping Identity | |  | M. Jones | |  | Microsoft | |  | May 30, 2013 | |

# OpenID Connect Session Management 1.0 - draft 14

### Abstract

OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 protocol. It allows Clients to verify the identity of the End-User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the End-User in an interoperable and REST-like manner.

This document describes how to manage sessions for OpenID Connect, including when to log out the End-User.

### Table of Contents

[**1.**](#Introduction)  Introduction  
    [**1.1.**](#rnc)  Requirements Notation and Conventions  
    [**1.2.**](#terminology)  Terminology  
[**2.**](#EndpointDiscovery)  Endpoint Discovery  
[**3.**](#CreatingUpdatingSessions)  Creating and Updating Sessions  
[**4.**](#ChangeNotification)  Session Status Change Notification  
    [**4.1.**](#OPiframe)  OP iframe  
    [**4.2.**](#RPiframe)  RP iframe  
[**5.**](#RPLogout)  RP-Initiated Logout  
    [**5.1.**](#RedirectionAfterLogout)  Redirection to RP After Logout  
[**6.**](#Validation)  Validation  
[**7.**](#ImplementationConsiderations)  Implementation Considerations  
[**8.**](#Security)  Security Considerations  
[**9.**](#IANA)  IANA Considerations  
    [**9.1.**](#oauth_params)  OAuth Parameters Registry  
        [**9.1.1.**](#ParametersContents)  Registry Contents  
[**10.**](#rfc.references1)  Normative References  
[**Appendix A.**](#Acknowledgements)  Acknowledgements  
[**Appendix B.**](#Notices)  Notices  
[**Appendix C.**](#History)  Document History  
[**§**](#rfc.authors)  Authors' Addresses

|  |
| --- |
| [TOC](#toc) |

### 1.  Introduction

This specification complements the OpenID Connect Messages 1.0 and OpenID Connect Standard 1.0 specifications by defining how to monitor the End-User's login status at the OpenID Provider on an ongoing basis so that the Relying Party can log out the End-User once he has logged out of the OpenID Provider.

|  |
| --- |
| [TOC](#toc) |

### 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 (**Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.**)](#RFC2119) [RFC2119].

Throughout this document, values are quoted to indicate that they are to be taken literally. When using these values in protocol messages, the quotes MUST NOT be used as part of the value.

|  |
| --- |
| [TOC](#toc) |

### 1.2.  Terminology

This specification uses the terms "Access Token", "Refresh Token", "Authorization Code", "Authorization Grant", "Authorization Server", "Authorization Endpoint", "Client", "Client Identifier", "Client Secret", "Protected Resource", "Resource Owner", "Resource Server", and "Token Endpoint" defined by [OAuth 2.0 (**Hardt, D., “The OAuth 2.0 Authorization Framework,” October 2012.**)](#RFC6749) [RFC6749], and the terms defined by [OpenID Connect Messages 1.0 (**Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., Mortimore, C., and E. Jay, “OpenID Connect Messages 1.0,” May 2013.**)](#OpenID.Messages) [OpenID.Messages].

This specification also defines the following term:

Session

Continuous period of time during which a user accesses a Relying Party relying on the Authentication of the End-User performed by the OpenID Provider

|  |
| --- |
| [TOC](#toc) |

### 2.  Endpoint Discovery

To support OpenID Connect session management, the RP needs to obtain the session management related endpoint URLs. These URLs are normally obtained via the OP's Discovery response, as described in [OpenID Connect Discovery 1.0 (**Sakimura, N., Bradley, J., Jones, M., and E. Jay, “OpenID Connect Discovery 1.0,” May 2013.**)](#OpenID.Discovery) [OpenID.Discovery], or MAY be learned via other mechanisms.

The OP endpoints defined in this specification are as follows:

check\_session\_iframe

URL of an OP iframe that supports cross-origin communications for session state information with the RP Client, using the HTML 5 postMessage API. The page is loaded from an invisible iframe embedded in an RP page so that it can run in the OP's security context. It accepts postMessage requests from the relevant RP iframe and uses postMessage to post back the login status of the End-User at the OP.

end\_session\_endpoint

URL at the OP that an RP can redirect to to request that the End-User be logged out at the OP.

|  |
| --- |
| [TOC](#toc) |

### 3.  Creating and Updating Sessions

In OpenID Connect, the session at the RP typically starts when the RP validates the End-User's ID Token. Refer to the OpenID Connect Messages 1.0 [[OpenID.Messages] (**Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., Mortimore, C., and E. Jay, “OpenID Connect Messages 1.0,” May 2013.**)](#OpenID.Messages) and OpenID Connect Standard 1.0 [[OpenID.Standard] (**Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., Mortimore, C., and E. Jay, “OpenID Connect Standard 1.0,” May 2013.**)](#OpenID.Standard) specifications to find out how to obtain an ID Token and validate it. When the OP supports session management, it MUST also return the Session State as an additional session\_state parameter in the Authorization Response. An OpenID Connect Authorization Response is specified in Section 2.1.2 of OpenID Connect Messages 1.0.

This parameter is:

session\_state

Session State. JSON string that represents the End-User's login state at the OP. This string is opaque to the RP. This is REQUIRED if session management is supported.

|  |
| --- |
| [TOC](#toc) |

### 4.  Session Status Change Notification

ID Token typically comes with the expiry date. The RP MAY rely on it to expire the RP session. However, it is entirely possible that the End-User might have logged out of the OP before the expiry date. Therefore, it is highly desirable to be able to find out the login status of the End-User at the OP.

To do so, it is possible to repeat the authentication request with prompt=none. However, this causes network traffic and this is problematic on the mobile devices that are increasingly becoming popular. Therefore, once the session is established with the authentication request and response, it is desirable to be able to check the login status at the OP without causing network traffic by polling the hidden OP iframe from a RP iframe with origin restricted postMessage as follows.

|  |
| --- |
| [TOC](#toc) |

### 4.1.  OP iframe

The RP typically loads an invisible OP iframe in the page from the OP's check\_session\_iframe.

The RP MUST assign an id attribute to the iframe so that it can address it later.

The OP iframe MUST enforce that the caller has the same origin as its parent frame. It MUST reject postMessage requests from any other source origin.

The postMessage from the RP iframe delivers the concatenation of the following, as the data:

* Client ID + " " + Session State

The OP iframe has access to Browser state at the OP (in a cookie or in HTML5 storage) that it uses to calculate and compare with the OP session state that was passed by the RP.

The OP iframe MUST recalculate it from the previously obtained Client ID, the source origin URL (from the postMessage), and the current OP Browser state. If the received value and the calculated value do not match, then the OP iframe MUST postMessage the string changed back to the source. If it matched, then it MUST postMessage the string unchanged.

Following is non-normative example pseudo-code for the OP iframe:

window.addEventListener("message", receiveMessage, false);

function receiveMessage(e){ // e has client\_id and session\_state

// Validate message origin

var salt;

client\_id = message.split(' ')[0];

session\_state = message.split(' ')[1];

salt = session\_state.split('.')[1];

var opbs = get\_op\_browser\_state();

var ss = CryptoJS.SHA256(client\_id + ' ' + e.origin + ' ' +

opbs + [' ' + salt]) [+ "." + salt];

if (e.session\_state == ss) {

stat = 'unchanged';

} else {

stat = 'changed';

}

e.source.postMessage(stat, e.origin);

};

The OP browser state is typically going to be stored in a cookie or HTML5 local storage. It is origin bound to the Authorization Server. It captures meaningful events such as logins, logouts, change of user, change of authorization status for Clients being used by the End-User, etc. Thus, the OP SHOULD update the value of the browser state in response to such meaningful events. As a result, the next call to check\_session() after such an event will return the value changed. It is RECOMMENDED that the OP not update the browser state too frequently in the absence of meaningful events so as to spare excessive network traffic at the Client in response to spurious changed events.

The computation of the session state returned in response to unsuccessful Authorization Requests SHOULD, in addition to the browser state, incorporate sufficient randomness in the form of a salt so as to prevent identification of an End-User across successive calls to the OP's Authorization Endpoint.

In the case of an authorized Client (successful Authorization Response), the OP SHOULD change the value of the session state returned to the Client under one of the following events:

* The set of users authenticated to the browser changes (login, logout, session add).
* The authorization status of Clients being used by the End-User changes.

In addition, the browser state used to verify the session state SHOULD change with such events. Calls to check\_session() will return changed against earlier versions of session state after such events. It is RECOMMENDED that the browser state SHOULD NOT vary too frequently in the absence of such events to minimize network traffic caused by the Client's response to changed notifications.

In the case of an unsuccessful Authorization Request, the value of the session state returned SHOULD vary with each request. However, the browser session state need not change unless a meaningful event happens. In particular, many values of session state can be simultaneously valid, for instance by the introduction of random salt in the session states issued in response to unsuccessful Authorization Requests.

In some implementations, changed notifications will occur only when changes to the End-User's session occur, whereas in other implementations, they might also occur as a result of changes to other sessions between the User-Agent and the OP. RPs need to be prepared for either eventuality, silently handling any false positives that might occur.

|  |
| --- |
| [TOC](#toc) |

### 4.2.  RP iframe

The RP also loads an invisible iframe from itself in the same page. This iframe MUST know the ID of the OP iframe so that it can postMessage to the OP iframe.

RP iframe polls OP iframe with postMessage with certain interval suitable for the RP application. With each postMessage, it sends the session state defined in Section 4.1. It also has to be able to receive the postMessage back from the OP iframe. The received data would either be changed or unchanged. Upon receipt of changed, the RP MUST perform the re-authentication with prompt=none to find the current session state at the OP.

Following is non-normative example pseudo-code for the RP iframe:

var stat = "unchanged"; var mes = client\_id + " " + session\_state; function check\_session() { var targetOrigin = "http://server.example.com"; var win = window.parent.document.getElementById("op"). contentWindow; win.postMessage( mes, targetOrigin); } function setTimer() { check\_session(); timerID = setInterval("check\_session()",3\*1000); } window.addEventListener("message", receiveMessage, false); function receiveMessage(e) { var targetOrigin = "http://server.example.com"; if (e.origin !== targetOrigin ) {return;} stat = e.data; if stat == "changed" then take the actions below... }

When the RP detects a session state change, it SHOULD first try an immediate mode request within an iframe to obtain a new ID Token and session state, sending the old ID Token as the id\_token\_hint. If the RP receives an ID token for the same End-User, it SHOULD simply update the value of the session state. If it doesn't receive an ID token or receives an ID token for another End-User, then it needs to handle this case as a logout for the original End-User.

Note that the session state is origin bound. Session state SHOULD be returned upon an authorization failure.

|  |
| --- |
| [TOC](#toc) |

### 5.  RP-Initiated Logout

An RP can notify the OP that the End-User has logged out of the site, and might want to log out of the OP as well. In this case, the RP, after having logged the End-User out of the RP, redirects the End-User's User-Agent to the OP's logout endpoint URL. This URL is normally obtained via the end\_session\_endpoint element of the OP's Discovery response, or MAY be learned via other mechanisms.

This specification also defines the following parameters that are passed as query parameters in the logout request:

id\_token\_hint

RECOMMENDED. Previously issued ID Token passed to the logout endpoint as a hint about the End-User's current authenticated session with the Client. This is used as an indication of the identity of the End-User that the RP is requesting be logged out by the OP. The OP need not be listed as an audience of the ID Token when it is used as an id\_token\_hint value.

post\_logout\_redirect\_uri

OPTIONAL. URL to which the RP is requesting that the End-User's User-Agent be redirected after a logout has been performed. The value MUST have been previously registered with the OP, either using the post\_logout\_redirect\_uris Registration parameter or via another mechanism. If supplied, the OP SHOULD honor this request following the logout.

At the logout endpoint, the OP SHOULD ask the End-User whether he wants to log out of the OP as well. If the End-User says "yes", then the OP MUST log out the End-User.

|  |
| --- |
| [TOC](#toc) |

### 5.1.  Redirection to RP After Logout

In some cases, the RP will request that the End-User's User-Agent to be redirected back to the RP after a logout has been performed. This specification defines this dynamic registration parameter for this purpose, per Section 2.1 of [OpenID Connect Dynamic Client Registration 1.0 (**Sakimura, N., Bradley, J., and M. Jones, “OpenID Connect Dynamic Client Registration 1.0,” May 2013.**)](#OpenID.Registration) [OpenID.Registration].

post\_logout\_redirect\_uris

OPTIONAL. Array of URLs supplied by the RP to which it MAY request that the End-User's User-Agent be redirected using the post\_logout\_redirect\_uri parameter after a logout has been performed.

|  |
| --- |
| [TOC](#toc) |

### 6.  Validation

If any of the validation procedures defined in this specification fail, any operations requiring the information that failed to correctly validate MUST be aborted and the information that failed to validate MUST NOT be used.

|  |
| --- |
| [TOC](#toc) |

### 7.  Implementation Considerations

This specification defines features used by both Relying Parties and OpenID Providers that choose to implement Session Management. All of these Relying Parties and OpenID Providers MUST implement the features that are listed in this specification as being "REQUIRED" or are described with a "MUST". No other implementation considerations for implementations of Session Management are defined by this specification.

|  |
| --- |
| [TOC](#toc) |

### 8.  Security Considerations

The OP iframe MUST enforce that the caller has the same origin as its parent frame. It MUST reject postMessage requests from any other source origin, to prevent cross-site scripting attacks.

|  |
| --- |
| [TOC](#toc) |

### 9.  IANA Considerations

|  |
| --- |
| [TOC](#toc) |

### 9.1.  OAuth Parameters Registry

This specification registers the following parameters in the IANA OAuth Parameters registry defined in [RFC 6749 (**Hardt, D., “The OAuth 2.0 Authorization Framework,” October 2012.**)](#RFC6749) [RFC6749].

|  |
| --- |
| [TOC](#toc) |

### 9.1.1.  Registry Contents

* Parameter name: session\_state
* Parameter usage location: Authorization Response, Access Token Response
* Change controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification document(s): [**Section 3 (Creating and Updating Sessions)**](#CreatingUpdatingSessions) of this document
* Related information: None

|  |
| --- |
| [TOC](#toc) |

### 10. Normative References

|  |  |
| --- | --- |
| **[OpenID.Discovery]** | Sakimura, N., Bradley, J., Jones, M., and E. Jay, “[OpenID Connect Discovery 1.0](http://openid.net/specs/openid-connect-discovery-1_0-16.html),” May 2013. |
| **[OpenID.Messages]** | Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., Mortimore, C., and E. Jay, “[OpenID Connect Messages 1.0](http://openid.net/specs/openid-connect-messages-1_0-19.html),” May 2013. |
| **[OpenID.Registration]** | Sakimura, N., Bradley, J., and M. Jones, “[OpenID Connect Dynamic Client Registration 1.0](http://openid.net/specs/openid-connect-registration-1_0-18.html),” May 2013. |
| **[OpenID.Standard]** | Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., Mortimore, C., and E. Jay, “[OpenID Connect Standard 1.0](http://openid.net/specs/openid-connect-standard-1_0-20.html),” May 2013. |
| **[RFC2119]** | [Bradner, S.](mailto:sob@harvard.edu), “[Key words for use in RFCs to Indicate Requirement Levels](http://tools.ietf.org/html/rfc2119),” BCP 14, RFC 2119, March 1997 ([TXT](http://www.rfc-editor.org/rfc/rfc2119.txt), [HTML](http://xml.resource.org/public/rfc/html/rfc2119.html), [XML](http://xml.resource.org/public/rfc/xml/rfc2119.xml)). |
| **[RFC6749]** | Hardt, D., “[The OAuth 2.0 Authorization Framework](http://tools.ietf.org/html/rfc6749),” RFC 6749, October 2012 ([TXT](http://www.rfc-editor.org/rfc/rfc6749.txt)). |

|  |
| --- |
| [TOC](#toc) |

### Appendix A.  Acknowledgements

Naveen Agarwal, Amanda Anganes, John Bradley, Breno de Medeiros, George Fletcher, Edmund Jay, Michael B. Jones, Torsten Lodderstedt, Tony Nadalin, Axel Nennker, Justin Richer, and Nat Sakimura contributed to the design of this specification.

|  |
| --- |
| [TOC](#toc) |

### Appendix B.  Notices

Copyright (c) 2013 The OpenID Foundation.

The OpenID Foundation (OIDF) grants to any Contributor, developer, implementer, or other interested party a non-exclusive, royalty free, worldwide copyright license to reproduce, prepare derivative works from, distribute, perform and display, this Implementers Draft or Final Specification solely for the purposes of (i) developing specifications, and (ii) implementing Implementers Drafts and Final Specifications based on such documents, provided that attribution be made to the OIDF as the source of the material, but that such attribution does not indicate an endorsement by the OIDF.

The technology described in this specification was made available from contributions from various sources, including members of the OpenID Foundation and others. Although the OpenID Foundation has taken steps to help ensure that the technology is available for distribution, it takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this specification or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any independent effort to identify any such rights. The OpenID Foundation and the contributors to this specification make no (and hereby expressly disclaim any) warranties (express, implied, or otherwise), including implied warranties of merchantability, non-infringement, fitness for a particular purpose, or title, related to this specification, and the entire risk as to implementing this specification is assumed by the implementer. The OpenID Intellectual Property Rights policy requires contributors to offer a patent promise not to assert certain patent claims against other contributors and against implementers. The OpenID Foundation invites any interested party to bring to its attention any copyrights, patents, patent applications, or other proprietary rights that may cover technology that may be required to practice this specification.

|  |
| --- |
| [TOC](#toc) |

### Appendix C.  Document History

[[ To be removed from the final specification ]]

-14

* Clarified RP-initiated logout description.
* Added an id\_token\_hint parameter to logout requests.
* Stated that RPs should gracefully any false positive changed notifications that may occur.
* Fixed #842 - Made post\_logout\_redirect\_uri treatment parallel to redirect\_uri.

-13

* Added OpenID Connect Working Group declaration to the document metadata.

-12

* Fixed #364 - Term "Session" not defined.
* State that when any validations fail, any operations requiring the information that failed to correctly validate MUST be aborted and the information that failed to validate MUST NOT be used.
* Fixed #779 - Parameters missing from IANA Considerations.
* Fixed #782 - Changed uses of "\_url" in identifiers to "\_uri".

-11

* Applied changes from October 24, 2012 editing session at the Internet Identity Workshop (IIW). This separates the session state from the ID Token, adding the new session\_state parameter to the authorization response. These identifiers also changed: check\_session\_endpoint to check\_session\_iframe\_url and end\_session\_endpoint to end\_session\_endpoint\_url.
* Fixed #605 - op\_logout\_url description.
* Added Implementation Considerations section.
* Fixed #698 - Inconsistent use of articles.
* Naming consistency changes. Renamed check\_session\_iframe\_url to check\_session\_iframe and end\_session\_endpoint\_url back to end\_session\_endpoint.

-10

* Fixed #689 - Track JWT change that allows JWTs to have multiple audiences.

-09

* Changed user\_hint to id\_token\_hint for consistency with Messages.
* Fixed #666 - JWS signature validation vs. verification.
* Referenced OAuth 2.0 RFC -- RFC 6749.

-08

* Complete rewrite based on the decisions made at the May 5, 2012 face to face working group meeting.

-07

* Added warning about the significant revisions planned to session management to the abstract and introduction.
* Changed client.example.com to client.example.org, per issue #251
* Listed author of ISO29115 as "International Telecommunication Union and International Organization for Standardization", per issue #589
* Use standards track version of JSON Web Token spec (draft-ietf-oauth-json-web-token)

-06

* Updated Notices
* Updated References

-05

* Removed Check Session Endpoint
* Updated ID Token claims to reflect changes in Messages

-04

* Changes associated with renaming "Lite" to "Basic Client" and replacing "Core" and "Framework" with "Messages" and "Standard".
* Numerous cleanups, including updating references.

-03

* Corrected examples.

-02

* Correct issues raised by Johnny Bufu and discussed on the 7-Jul-11 working group call.

-01

* Consistency and cleanup pass, including removing unused references.

-00

* Split from core when all optional features were removed.

|  |
| --- |
| [TOC](#toc) |

### Authors' Addresses

|  |  |
| --- | --- |
|  | Breno de Medeiros |
|  | Google |
| **Email:** | [breno@google.com](mailto:breno@google.com) |
|  |  |
|  | Naveen Agarwal |
|  | Google |
| **Email:** | [naa@google.com](mailto:naa@google.com) |
|  |  |
|  | Nat Sakimura (editor) |
|  | Nomura Research Institute, Ltd. |
| **Email:** | [n-sakimura@nri.co.jp](mailto:n-sakimura@nri.co.jp) |
|  |  |
|  | John Bradley |
|  | Ping Identity |
| **Email:** | [ve7jtb@ve7jtb.com](mailto:ve7jtb@ve7jtb.com) |
|  |  |
|  | Michael B. Jones |
|  | Microsoft |
| **Email:** | [mbj@microsoft.com](mailto:mbj@microsoft.com) |