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

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Draft | B. de Medeiros | |  | N. Agarwal | |  | Google | |  | N. Sakimura | |  | NRI | |  | J. Bradley | |  | Ping Identity | |  | M. Jones | |  | Microsoft | |  | August 3, 2015 | |

# OpenID Connect Session Management 1.0 - draft 25

### Abstract

OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 protocol. It enables 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  
    [**2.1.**](#OPMetadata)  OpenID Provider Discovery Metadata  
[**3.**](#CreatingUpdatingSessions)  Creating and Updating Sessions  
[**4.**](#ChangeNotification)  Session Status Change Notification  
    [**4.1.**](#RPiframe)  RP iframe  
    [**4.2.**](#OPiframe)  OP iframe  
[**5.**](#RPLogout)  RP-Initiated Logout  
    [**5.1.**](#RedirectionAfterLogout)  Redirection to RP After Logout  
        [**5.1.1.**](#ClientMetadata)  Client Registration Metadata  
[**6.**](#Validation)  Validation  
[**7.**](#ImplementationConsiderations)  Implementation Considerations  
[**8.**](#Security)  Security Considerations  
[**9.**](#IANA)  IANA Considerations  
    [**9.1.**](#OAuthParams)  OAuth Parameters Registry  
        [**9.1.1.**](#ParametersContents)  Registry Contents  
    **[9.2.](" \l "DynRegRegistrations)**  OAuth Dynamic Client Registration Metadata Registration  
        **[9.2.1.](" \l "DynRegContents)**  Registry Contents  
[**10.**](#rfc.references1)  References  
    **[10.1.](" \l "rfc.references1)**  Normative References  
    **[10.2.](" \l "rfc.references2)**  Informative References  
[**Appendix A.**](#Acknowledgements)  Acknowledgements  
[**Appendix B.**](#Notices)  Notices  
[**Appendix C.**](#History)  Document History  
[**§**](#rfc.authors)  Authors' Addresses

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

### 1.  Introduction

OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 [[RFC6749] (Hardt, D., Ed., “The OAuth 2.0 Authorization Framework,” October 2012.)](#RFC6749) protocol. It enables 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 specification complements the [OpenID Connect Core 1.0 (Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., and C. Mortimore, “OpenID Connect Core 1.0,” November 2014.)](#OpenID.Core) [OpenID.Core] specification 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 an End-User who 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", "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].

In the .txt version of 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. In the HTML version of this document, values to be taken literally are indicated by the use of this fixed-width font.

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

### 1.2.  Terminology

This specification uses the terms "Access Token", "Authorization Code", "Authorization Endpoint", "Authorization Grant", "Authorization Server", "Client", "Client Identifier", "Client Secret", "Protected Resource", "Redirection URI", "Refresh Token", "Resource Owner", "Resource Server", "Response Type", and "Token Endpoint" defined by [OAuth 2.0 (Hardt, D., Ed., “The OAuth 2.0 Authorization Framework,” October 2012.)](#RFC6749) [RFC6749], the term "User Agent" defined by [RFC 7230 (Fielding, R., Ed. and J. Reschke, Ed., “Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing,” June 2014.)](#RFC7230) [RFC7230], and the terms defined by [OpenID Connect Core 1.0 (Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., and C. Mortimore, “OpenID Connect Core 1.0,” November 2014.)](#OpenID.Core) [OpenID.Core].

This specification also defines the following term:

Session

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

IMPORTANT NOTE TO READERS: The terminology definitions in this section are a normative portion of this specification, imposing requirements upon implementations. All the capitalized words in the text of this specification, such as "Session", reference these defined terms. Whenever the reader encounters them, their definitions found in this section must be followed.

|  |
| --- |
| [**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,” November 2014.)](#OpenID.Discovery) [OpenID.Discovery], or MAY be learned via other mechanisms.

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

### 2.1.  OpenID Provider Discovery Metadata

These OpenID Provider Metadata parameters MUST be included in the Server's discovery responses when Session Management and Discovery are supported:

check\_session\_iframe

REQUIRED. URL of an OP iframe that supports cross-origin communications for session state information with the RP Client, using the HTML5 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

REQUIRED. URL at the OP to which an RP can perform a redirect 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 Core 1.0 [[OpenID.Core] (Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., and C. Mortimore, “OpenID Connect Core 1.0,” November 2014.)](#OpenID.Core) specification 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 Authentication Response. The OpenID Connect Authentication Response is specified in Section 3.1.2.5 of OpenID Connect Core 1.0.

This parameter is:

session\_state

Session State. JSON [[RFC7159] (Bray, T., Ed., “The JavaScript Object Notation (JSON) Data Interchange Format,” March 2014.)](" \l "RFC7159) string that represents the End-User's login state at the OP. It MUST NOT contain the space (" ") character. This value is opaque to the RP. This is REQUIRED if session management is supported.

The Session State value is initially calculated on the server. The same Session State value is also recalculated by the OP iframe in the browser client. The generation of suitable Session State values is specified in [Section 4.2 (OP iframe)](#OPiframe), and is based on a salted cryptographic hash of Client ID, origin URL, and OP browser state. For the origin URL, the server can use the origin URL of the Authentication Response, following the algorithm specified in Section 4 of [RFC 6454 (Barth, A., “The Web Origin Concept,” December 2011.)](#RFC6454) [RFC6454].

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

### 4.  Session Status Change Notification

An ID Token typically comes with an expiration 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 expiration 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 becoming increasingly 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 a hidden OP iframe from an RP iframe with an origin restricted postMessage as follows.

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

### 4.1.  RP iframe

The RP loads an invisible iframe from itself. This iframe MUST know the ID of the OP iframe, as described in [Section 4.2 (OP iframe)](#OPiframe), so that it can postMessage to the OP iframe.

The RP iframe polls the OP iframe with postMessage at an interval suitable for the RP application. With each postMessage, it sends the session state defined in [Section 4.2 (OP iframe)](#OPiframe).

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

* Client ID + " " + Session State

It also has to be able to receive the postMessage back from the OP iframe. The received data will either be changed or unchanged unless the syntax of the message sent was determined by the OP to be malformed, in which case the received data will be error. Upon receipt of changed, the RP MUST perform re-authentication with prompt=none to obtain the current session state at the OP. Upon receipt of error, the RP MUST NOT perform re-authentication with prompt=none, so as to not cause potential infinite loops that generate network traffic to 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 = "https://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 = "https://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 a prompt=none 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 authentication failure.

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

### 4.2.  OP iframe

The RP also loads an invisible OP iframe into itself from the OP's check\_session\_iframe.

The RP MUST assign an id attribute to the iframe so that it can address it, as described above.

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.

As specified in [Section 4.1 (RP iframe)](#RPiframe), the postMessage from the RP iframe delivers the following concatenation 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 postMessage received is syntactically malformed in such a way that the posted Client ID and origin URL cannot be determined or are syntactically invalid, then the OP iframe SHOULD postMessage the string error back to the source. 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.data has client\_id and session\_state

// Validate message origin

var client\_id = e.data.split(' ')[0];

var session\_state = e.data.split(' ')[1];

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

// if message syntactically invalid

// postMessage('error', e.origin) and return

// get\_op\_browser\_state() is an OP defined function

// that returns the browser's login status at the OP.

// How it is done is entirely up to the OP.

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

// Here, the session\_state is calculated in this particular way,

// but it is entirely up to the OP how to do it under the

// requirements defined in this specification.

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

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

var stat = '';

if (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 authentication 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 Authentication 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 Authentication 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 authentication 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 Authentication 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 Authentication Requests.

If a cookie is used to maintain the OP browser state, the HttpOnly flag likely can't be set for this cookie, because it needs to be accessed from JavaScript. Therefore, information that can be used for identifying the user should not be put into the cookie, as it could be read by unrelated JavaScript.

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) |

### 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.

state

OPTIONAL. Opaque value used by the RP to maintain state between the logout request and the callback to the endpoint specified by the post\_logout\_redirect\_uri query parameter. If included in the logout request, the OP passes this value back to the RP using the state query parameter when redirecting the User Agent back to the RP.

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. Post-logout redirection is only done when the logout is RP-initiated, in which case the redirection target is the post\_logout\_redirect\_uri query parameter value used by the initiating RP; otherwise it is not done. 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,” November 2014.)](#OpenID.Registration) [OpenID.Registration].

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

### 5.1.1.  Client Registration Metadata

This Client Metadata parameter MAY be included in the Client's Registration information when Session Management and Dynamic Registration are supported:

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.

The id\_token\_hint parameter to a logout request can be used to determine which RP initiated the logout request. Logout requests without a valid id\_token\_hint value are a potential means of denial of service; therefore, OPs may want to require explicit user confirmation before acting upon them.

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

### 9.  IANA Considerations

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

### 9.1.  OAuth Parameters Registry

This specification registers the following parameter in the IANA "OAuth Parameters" registry [[IANA.OAuth.Parameters] (IANA, “OAuth Parameters,” .)](" \l "IANA.OAuth.Parameters) established by [RFC 6749 (Hardt, D., Ed., “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) |

### 9.2.  OAuth Dynamic Client Registration Metadata Registration

This specification registers the following client metadata definition in the IANA "OAuth Dynamic Client Registration Metadata" registry [[IANA.OAuth.Parameters] (IANA, “OAuth Parameters,” .)](" \l "IANA.OAuth.Parameters) established by [[RFC7591] (Richer, J., Ed., Jones, M., Bradley, J., Machulak, M., and P. Hunt, “OAuth 2.0 Dynamic Client Registration Protocol,” July 2015.)](" \l "RFC7591):

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

### 9.2.1.  Registry Contents

* Client Metadata Name: post\_logout\_redirect\_uris
* Client Metadata Description: 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
* Change Controller: OpenID Foundation Artifact Binding Working Group - openid-specs-ab@lists.openid.net
* Specification Document(s): [Section 5.1.1 (Client Registration Metadata)](" \l "ClientMetadata) of this document

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

### 10.  References

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

### 10.1. Normative References

|  |  |
| --- | --- |
| **[IANA.OAuth.Parameters]** | IANA, “[OAuth Parameters](http://www.iana.org/assignments/oauth-parameters).” |
| **[OpenID.Core]** | Sakimura, N., Bradley, J., Jones, M., de Medeiros, B., and C. Mortimore, “[OpenID Connect Core 1.0](http://openid.net/specs/openid-connect-core-1_0.html),” November 2014. |
| **[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.html),” November 2014. |
| **[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.html),” November 2014. |
| **[RFC2119]** | Bradner, S., “[Key words for use in RFCs to Indicate Requirement Levels](http://www.rfc-editor.org/info/rfc2119),” BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997. |
| **[RFC6454]** | Barth, A., “[The Web Origin Concept](http://www.rfc-editor.org/info/rfc6454),” RFC 6454, DOI 10.17487/RFC6454, December 2011. |
| **[RFC6749]** | Hardt, D., Ed., “[The OAuth 2.0 Authorization Framework](http://www.rfc-editor.org/info/rfc6749),” RFC 6749, DOI 10.17487/RFC6749, October 2012. |
| **[RFC7159]** | Bray, T., Ed., “[The JavaScript Object Notation (JSON) Data Interchange Format](http://www.rfc-editor.org/info/rfc7159),” RFC 7159, DOI 10.17487/RFC7159, March 2014. |
| **[RFC7230]** | Fielding, R., Ed. and J. Reschke, Ed., “[Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing](http://www.rfc-editor.org/info/rfc7230),” RFC 7230, DOI 10.17487/RFC7230, June 2014. |

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

### 10.2. Informative References

|  |  |
| --- | --- |
| **[RFC7591]** | Richer, J., Ed., Jones, M., Bradley, J., Machulak, M., and P. Hunt, “[OAuth 2.0 Dynamic Client Registration Protocol](http://www.rfc-editor.org/info/rfc7591),” RFC 7591, DOI 10.17487/RFC7591, July 2015. |

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

### Appendix A.  Acknowledgements

The OpenID Community would like to thank the following people for their contributions to this specification:

Naveen Agarwal (naa@google.com), Google

Amanda Anganes (aanganes@mitre.org), MITRE

John Bradley (ve7jtb@ve7jtb.com), Ping Identity

Breno de Medeiros (breno@google.com), Google

George Fletcher (george.fletcher@corp.aol.com), AOL

Edmund Jay (ejay@mgi1.com), Illumila

Michael B. Jones (mbj@microsoft.com), Microsoft

Todd Lainhart (lainhart@us.ibm.com), IBM

Torsten Lodderstedt (t.lodderstedt@telekom.de), Deutsche Telekom

Anthony Nadalin (tonynad@microsoft.com), Microsoft

Axel Nennker (axel.nennker@telekom.de), Deutsche Telekom

Justin Richer (jricher@mitre.org), MITRE

Nat Sakimura (n-sakimura@nri.co.jp), Nomura Research Institute, Ltd.

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

### Appendix B.  Notices

Copyright (c) 2015 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 ]]

-25

* Changed instances of http://server.example.com to https://server.example.com.
* Tracked terminology changes made in the referenced IETF specs since errata set 1.
* Registered the post\_logout\_redirect\_uris client metadata value.

-24

* Stated that post-logout redirection is only done when the logout is RP-initiated and that otherwise it is not done.
* Fixed #967 - Errors in the Javascript example for the OP iframe

-23

* Fixed issue #915 - Computation of OP session\_state in the IdP requires origin URI.
* Updated the RFC 2616 reference to RFC 7230.

-22

* Referenced specification versions incorporating errata set 1.

-21

* Updated dates for specs containing errata updates.
* Fixed #930 - Added "error" return from OP iframe to respond to syntactically invalid postMessage values received.
* Fixed #954 - Added "NOT RECOMMENDED" to the list of RFC 2119 terms.

-20

* Fixed #917 - Session state must not contain the space character.
* Fixed #927 - Added "state" parameter to RP-initiated logout messages.

-19

* Updated dates for final OpenID Connect specifications.

-18

* Described that logout requests without a valid id\_token\_hint are a potential means of denial of service.

-17

* Fixed #883 - Moved the RP iframe description ahead of the OP iframe description.
* Fixed #873 - Clarified that the HttpOnly flag can't be set for cookies used to maintain OP browser state.

-16

* Replaced uses of the OpenID Connect Messages and OpenID Connect Standard specifications with OpenID Connect Core.
* Added section headings for Discovery and Registration parameters defined for use when Session Management is supported.

-15

* Fixed #842 - Made post\_logout\_redirect\_uri treatment parallel to redirect\_uri.
* Fixed #859 - Added IMPORTANT NOTE TO READERS about the terminology definitions being a normative part of the 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.

-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) |
| **URI:** | <http://stackoverflow.com/users/311376/breno> |
|  |  |
|  | Naveen Agarwal |
|  | Google |
| **Email:** | [naa@google.com](mailto:naa@google.com) |
| **URI:** | <http://www.linkedin.com/in/nvnagr> |
|  |  |
|  | Nat Sakimura |
|  | Nomura Research Institute, Ltd. |
| **Email:** | [n-sakimura@nri.co.jp](mailto:n-sakimura@nri.co.jp) |
| **URI:** | <http://nat.sakimura.org/> |
|  |  |
|  | John Bradley |
|  | Ping Identity |
| **Email:** | [ve7jtb@ve7jtb.com](mailto:ve7jtb@ve7jtb.com) |
| **URI:** | <http://www.thread-safe.com/> |
|  |  |
|  | Michael B. Jones |
|  | Microsoft |
| **Email:** | [mbj@microsoft.com](mailto:mbj@microsoft.com) |
| **URI:** | <http://self-issued.info/> |