OAuth Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 1, 2014

J. Richer
The MITRE Corporation
M. Jones
Microsoft
J. Bradley
Ping Identity
M. Machulak
Newcastle University
January 28, 2014

OAuth 2.0 Dynamic Client Registration Metadata
draft-jones-oauth-dyn-reg-metadata-00

Abstract

This specification defines client metadata values used to describe attributes of dynamically registered OAuth 2.0 clients.

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 August 1, 2014.

Copyright Notice

Table of Contents

1. Introduction

In order for an OAuth 2.0 client to utilize an OAuth 2.0 authorization server, the client needs specific information to interact with the server, including an OAuth 2.0 Client ID to use at that server. The OAuth 2.0 Dynamic Client Registration Core Protocol [OAuth.Registration] specification describes how an OAuth 2.0 client can be dynamically registered with an authorization server to obtain this information and how metadata about the client can be registered with the server.

This specification extends the core registration specification by defining a specific set of client metadata values that can be used to describe additional attributes of dynamically registered OAuth 2.0 clients beyond those defined in the core registration specification.

1.1. Notational 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 [RFC2119].

Unless otherwise noted, all the protocol parameter names and values are case sensitive.

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", "Response Type", and "Token Endpoint" defined by OAuth 2.0 [RFC6749] and the terms defined by the OAuth 2.0 Client Dynamic Registration Core Protocol [OAuth.Registration].

2. Client Metadata

Registering client metadata values with an authorization server may be necessary or useful to facilitate usage of the authorization server by the client. This specification extends the list of client metadata values defined in OAuth 2.0 Client Dynamic Registration Core Protocol [OAuth.Registration] with the following fields:

client_name Human-readable name of the client to be presented to the user. If omitted, the authorization server MAY display the raw "client_id" value to the user instead. It is RECOMMENDED that clients always send this field. The value of this field MAY be internationalized, as described in Section 2.1.

client_uri URL of the homepage of the client. If present, the server SHOULD display this URL to the end user in a clickable fashion. It is RECOMMENDED that clients always send this field. The value of this field MUST point to a valid web page. The value of this field MAY be internationalized, as described in Section 2.1.

logo_uri URL that references a logo for the client. If present, the server SHOULD display this image to the end user during approval. The value of this field MUST point to a valid image file. The value of this field MAY be internationalized, as described in Section 2.1.

scope Space separated list of scope values (as described in OAuth 2.0 Section 3.3 [RFC6749]) that the client can use when requesting access tokens. The semantics of values in this list is service specific. If omitted, an authorization server MAY register a Client with a default set of scopes.

contacts Array of email addresses for people responsible for this client. The authorization server MAY make these addresses available to end users for support requests for the client. An authorization server MAY use these email addresses as identifiers

for an administrative page for this client.

tos_uri URL that points to a human-readable Terms of Service document for the client. The Authorization Server SHOULD display this URL to the end-user if it is given. The Terms of Service usually describe a contractual relationship between the end-user and the client that the end-user accepts when authorizing the client. The value of this field MUST point to a valid web page. The value of this field MAY be internationalized, as described in Section 2.1.

policy_uri URL that points to a human-readable Policy document for the client. The authorization server SHOULD display this URL to the end-user if it is given. The policy usually describes how an end-user's data will be used by the client. The value of this field MUST point to a valid web page. The value of this field MAY be internationalized, as described in Section 2.1.

jwks_uri URL for the Client's JSON Web Key Set [JWK] document representing the client's public keys. The value of this field MUST point to a valid JWK Set. These keys MAY be used for higher level protocols that require signing or encryption.

software_id Identifier for the software that comprises a client. Unlike "client_id", which is issued by the authorization server and generally varies between instances, the "software_id" is asserted by the client software and is intended to be shared between all copies of the client software. The value for this field MAY be a UUID [RFC4122]. The identifier SHOULD NOT change when software version changes or when a new installation instance is detected. Authorization servers MUST treat this field as self-asserted by the client and MUST NOT make any trusted decisions on the value of this field alone.

software_version Version identifier for the software that comprises a client. The value of this field is a string that is intended to be compared using string equality matching. The value of the "software_version" SHOULD change on any update to the client software. Authorization servers MUST treat this field as self-asserted by the client and MUST NOT make any trusted decisions on

the value of this field alone.

2.1. Human Readable Client Metadata

Human-readable client metadata values and client metadata values that reference human-readable values MAY be represented in multiple languages and scripts. For example, the values of fields such as "client_name", "tos_uri", "policy_uri", "logo_uri", and "client_uri" might have multiple locale-specific values in some client registrations.

To specify the languages and scripts, BCP47 [RFC5646] language tags are added to client metadata member names, delimited by a # character. Since JSON member names are case sensitive, it is RECOMMENDED that language tag values used in Claim Names be spelled using the character case with which they are registered in the IANA Language Subtag Registry [IANA.Language]. In particular, normally language names are spelled with lowercase characters, region names are spelled with uppercase characters, and languages are spelled with mixed case characters. However, since BCP47 language tag values are case insensitive, implementations SHOULD interpret the language tag values supplied in a case insensitive manner. Per the recommendations in BCP47, language tag values used in metadata member names should only be as specific as necessary. For instance, using "fr" might be sufficient in many contexts, rather than "fr-CA" or "fr-FR".

For example, a client could represent its name in English as ""client_name#en": "My Client"" and its name in Japanese as ""client_name#ja-Jpan-JP": "\u30AF\u30E9\u30A4\u30A2\u30F3\u30C8\u540D"" within the same registration request. The authorization server MAY display any or all of these names to the resource owner during the authorization step, choosing which name to display based on system configuration, user preferences or other factors.

If any human-readable field is sent without a language tag, parties using it MUST NOT make any assumptions about the language, character set, or script of the string value, and the string value MUST be used as-is wherever it is presented in a user interface. To facilitate interoperability, it is RECOMMENDED that clients and servers use a human-readable field without any language tags in addition to any

language-specific fields, and it is RECOMMENDED that any human-readable fields sent without language tags contain values suitable for display on a wide variety of systems.

Implementer's Note: Many JSON libraries make it possible to reference members of a JSON object as members of an object construct in the native programming environment of the library. However, while the "#" character is a valid character inside of a JSON object's member names, it is not a valid character for use in an object member name in many programming environments. Therefore, implementations will need to use alternative access forms for these claims. For instance, in JavaScript, if one parses the JSON as follows, "var j = JSON.parse(json);", then the member "client_name#en-us" can be accessed using the JavaScript syntax "j["client_name#en-us"]".

3. IANA Considerations

3.1. OAuth Registration Client Metadata Registration

This specification registers the Client Metadata values defined in Section 2 in the IANA OAuth Registration Client Metadata registry defined in [OAuth.Registration].

3.1.1. Registry Contents

- Client Metadata Name: "client_name"
- Client Metadata Description: Human-readable name of the client to be presented to the user
- Change Controller: IESG
- Specification Document(s): [[this document]]

- Client Metadata Name: "client_uri"
- Client Metadata Description: URL of the homepage of the client
- Change Controller: IESG
- Specification Document(s): [[this document]]

- Client Metadata Name: "logo_uri"
- Client Metadata Description: URL that references a logo for the client

- Change Controller: IESG
- Specification Document(s): [[this document]]

- Client Metadata Name: "scope"
- Client Metadata Description: Space separated list of scope values
- Change Controller: IESG
- Specification Document(s): [[this document]]

- Client Metadata Name: "contacts"
- Client Metadata Description: Array of email addresses for people responsible for this client
- Change Controller: IESG
- Specification document(s): [[this document]]

- Client Metadata Name: "tos_uri"
- Client Metadata Description: URL that points to a human-readable Terms of Service document for the client
- Change Controller: IESG
- Specification Document(s): [[this document]]

- Client Metadata Name: "policy_uri"
- Client Metadata Description: URL that points to a human-readable Policy document for the client
- Change Controller: IESG
- Specification Document(s): [[this document]]

- Client Metadata Name: "jwks_uri"
- Client Metadata Description: URL for the Client's JSON Web Key Set [JWK] document representing the client's public keys
- Change Controller: IESG
- Specification Document(s): [[this document]]

- Client Metadata Name: "software_id"
- Client Metadata Description: Identifier for the software that comprises a client
- Change Controller: IESG
- Specification Document(s): [[this document]]

- Client Metadata Name: "software_version"

- Client Metadata Description: Version identifier for the software that comprises a client
- Change Controller: IESG
- Specification Document(s): [[this document]]

4. Security Considerations

The authorization server MUST treat all client metadata as self-asserted. For instance, a rogue client might use the name and logo for the legitimate client which it is trying to impersonate. Additionally, a rogue client might try to use the software identifier or software version of a legitimate client to attempt to associate itself on the authorization server instances of the legitimate client. To counteract this, an authorization server needs to take steps to mitigate this phishing risk by looking at the entire registration request and client configuration. For instance, an authorization server could warn if the domain/site of the logo doesn't match the domain/site of redirect URIs. An authorization server could also refuse registration from a known software identifier that is requesting different redirect URIs or a different client homepage uri. An authorization server can also present warning messages to end users about dynamically registered clients in all cases, especially if such clients have been recently registered or have not been trusted by any users at the authorization server before.

In a situation where the authorization server is supporting open client registration, it must be extremely careful with any URL provided by the client that will be displayed to the user (e.g. "logo_uri", "tos_uri", "client_uri", and "policy_uri"). For instance, a rogue client could specify a registration request with a reference to a drive-by download in the "policy_uri". The authorization server SHOULD check to see if the "logo_uri", "tos_uri", "client_uri", and "policy_uri" have the same host and scheme as the those defined in the array of "redirect_uris" and that all of these resolve to valid web pages.

5. Normative References

[IANA.Language]
Internet Assigned Numbers Authority (IANA), "Language Subtag Registry", 2005.

[JWK] Jones, M., "JSON Web Key (JWK)", draft-ietf-jose-json-web-key (work in progress),

January 2014.

[OAuth.Registration]
Richer, J., Jones, M., Bradley, J., and M. Machulak, "OAuth 2.0 Dynamic Client Registration Core Protocol", draft-ietf-oauth-dyn-reg (work in progress), January 2014.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, July 2005.

[RFC5646] Phillips, A. and M. Davis, "Tags for Identifying Languages", BCP 47, RFC 5646, September 2009.

[RFC6749] Hardt, D., "The OAuth 2.0 Authorization Framework", RFC 6749, October 2012.

Appendix A. Acknowledgments

The authors thank the OAuth Working Group, the User-Managed Access Working Group, and the OpenID Connect Working Group participants for their input to this document. In particular, the following individuals have been instrumental in their review and contribution to various versions of this document: Amanda Anganes, Derek Atkins, Tim Bray, Domenico Catalano, Donald Coffin, Vladimir Dzhuvinov, George Fletcher, Thomas Hardjono, Phil Hunt, William Kim, Torsten Lodderstedt, Eve Maler, Josh Mandel, Nov Matake, Tony Nadalin, Nat Sakimura, Christian Scholz, and Hannes Tschofenig.

Appendix B. Open Issues

- Should this specification become a working group document so that the functionality defined in this document that was previously defined in draft-ietf-oauth-dyn-reg-14 is retained in working group drafts?

- Should this specification become a working group document so that the functionality that was previously defined in draft-ietf-oauth-dyn-reg-14 is retained in working group drafts?

Appendix C. Document History

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

-00

- Partitioned the Dynamic Client Registration specification into core, metadata, and management specifications. This built on work first published as draft-richer-oauth-dyn-reg-core-00 and draft-richer-oauth-dyn-reg-management-00.

- Registered the Client Metadata values defined by this specification in the IANA OAuth Registration Client Metadata registry.

- Rewrote the introduction.

Authors' Addresses

Justin Richer
The MITRE Corporation

Email: jricher@mitre.org

Michael B. Jones
Microsoft

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

John Bradley
Ping Identity

Email: ve7jtb@ve7jtb.com

Maciej Machulak
Newcastle University

Email: m.p.machulak@ncl.ac.uk
URI: http://ncl.ac.uk/