Type Indexes

Version 1.0.0, Editor’s Draft, 2023-03-13

▾ **More details about this document**

This version
: https://solid.github.io/type-indexes/

Latest version
: https://solid.github.io/type-indexes/

Latest published version
: https://solid.github.io/type-indexes/

Editors
: Timea Turdean

Contributors
: Jeff Zucker
: Virginia Balseiro
: Sarven Capadisli
: Tim Berners-Lee

Created
: 2022-06-22

Published
: 2023-03-13

Modified
: 2023-03-15

Repository
: GitHub
: Issues

Language
: English

Document Status
Editor’s Draft

Policy
Rule
Offer

Unique Identifier
https://solidproject.org/TR/type-indexs#document-policy-offer

Target
https://solidproject.org/TR/type-indexs

Permission
Assigner
W3C Solid Community Group

Action
Aggregate , Archive , Concurrent Use , Derivative Works , Derive , Digitize , Display , Distribution , Index , Inform , Install , Notice , Present , Print , Read , Reproduce , Reproduction , Stream , Synchronize , Text-to-speech , Transform , Translate

Abstract

Type Indexes is a data discovery mechanism where coarse-grained library types are registered providing a way for applications to discover where an agent keeps data relevant for the application.

Status of This Document

This section describes the status of this document at the time of its publication.

This document was published by the Solid Community Group as an *Editor’s Draft*. The information in this document is still subject to change. You are invited to contribute any feedback, comments, or questions you might have.

Publication as an *Editor’s Draft* does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

This document was produced by a group operating under the W3C Community Contributor License Agreement (CLA). A human-readable summary is available.

Table of Contents

1. Introduction

This section is non-normative.

In the Solid ecosystem, storages and clients are loosely coupled. This means clients are tight to data models. Furthermore, clients and thus storages should be able to be connected in a network of knowledge. The connection is only achievable if the data itself is easy to be interconnected. This specification is written for Solid application developers as a solution that shows how to interconnect clients and data on storages. This specification details the use of Type Indexes which were implemented already in different clients. Type Indexes offer a fine-grained approach to describing the location of specific types of resources on a storage. This specification is accompanied with shape documents, [SHACL] and [ShEx], to help developers improve their implementations and maximize the promise of interconnected data.

1.1 Terminology

This section is non-normative.

This document uses terminology from the *Solid Protocol* [SOLID-PROTOCOL] and *WebID* [WebID] specifications. When terminology is used from these specifications it is linked directly to them.

Storage
A *storage* is a space of URIs that affords agents controlled access to resources.

WebID
A *WebID* is a URI with an HTTP or HTTPS scheme which denotes an Agent (Person, Organization, Group, Device, etc.). For WebIDs with fragment identifiers (e.g. #me), the URI without the fragment denotes the Profile Document. For WebIDs without fragment identifiers an HTTP request on the WebID MUST return a 303 with a Location header URI referring to the Profile Document.

WebID Profile
A *WebID Profile* is an RDF document which uniquely describes the Agent denoted by the WebID in relation to that WebID. The server MUST provide a text/turtle [turtle] representation of the requested profile. This document MAY be available in other RDF serialization formats, such as RDFa [RDFA-CORE], or [RDF-SYNTAX-GRAMMAR] if so requested through content negotiation.

1.2 Namespaces

Prefixes and Namespaces

Prefix	Namespace	Description
`rdf`	http://www.w3.org/1999/02/22-rdf-syntax-ns#	[rdf-schema]
`solid`	http://www.w3.org/ns/solid/terms#	Solid Terms
`vcard`	http://www.w3.org/2006/vcard/ns#	[VCARD-RDF]

1.3 Conformance

All assertions, diagrams, examples, and notes are non-normative, as are all sections explicitly marked non-normative. Everything else is normative.

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Examples in this specification are introduced with the word "for example" or are set apart from the normative text with `class="example"`, like this:

EXAMPLE 1

```
This is an example.
```

2. Type Indexes

A Type Index is a document containing statements that link specific types of data to specific locations. An app might ask a user where they want to store items of type Bookmark and then write that location in the Type Index. Thereafter another app can find the bookmarks by looking for the location in the Type Index. Apps that want to remember user's choices will read from and write to the Type Index.

Since users will undoubtedly have data they want to keep private, there are two Type Indexes - one readable by any app (publicly discoverable) and one only readable by apps operating on behalf of the WebID owner (not publicly discoverable).

To find Type Indexes, an app SHOULD load the WebID Profile Document and if the app has access, also the Preferences Document. In these loaded documents one can find triples where the WebID is the subject and predicates can be: for the Public Type Index document a `solid:publicTypeIndex` predicate, and for the Private Type Index document a `solid:privateTypeIndex` predicate.

This section is non-normative.

Examples:

An example for a Public Type Index document located in settings and called `publicTypeIndex.ttl` is linked from the profile as:

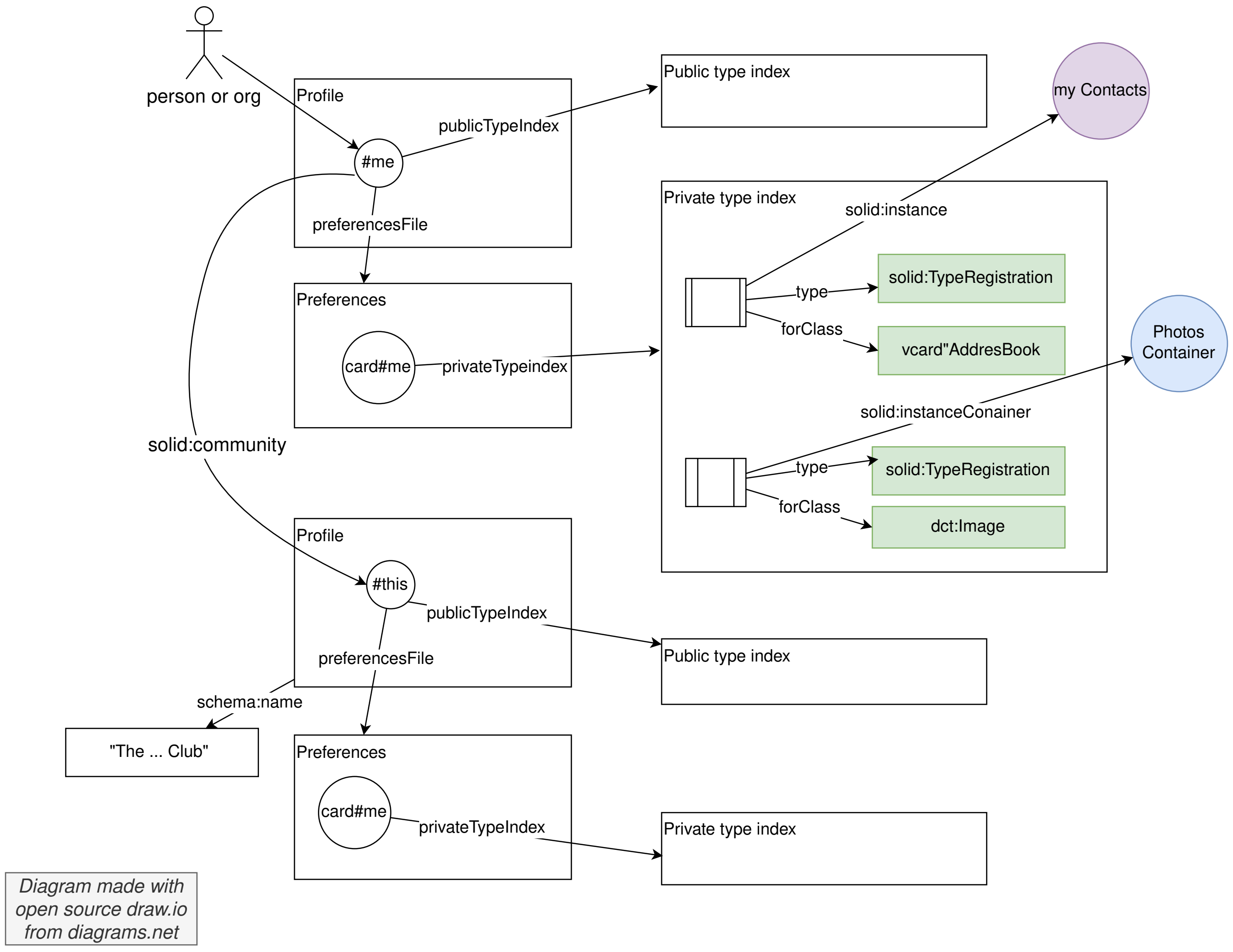
```
<#WebID> <http://www.w3.org/ns/solid/terms#publicTypeIndex> </settings/
```

An example for a Private Type Index document located in settings and called `privateTypeIndex.ttl` is linked from the profile as:

```
<#WebID> <http://www.w3.org/ns/solid/terms#privateTypeIndex> </settings
```

This section is non-normative.

The following diagram shows an example set of Type Indexes:



Type Indexes example setup

2.1 Public Type Index

The Public Type Index document contains registration entries that are discoverable by outside users and applications. These registration entries are not necessarily accessible to all applications or users.

This section is non-normative.

For example, suppose I have several address books - one that is only for myself, one that is for a specific group of colleagues, and one that is for the general public. The one that is only for me, should be listed in the Private Type Index. The other two, should be listed in the Public Type Index. The fully public listing will point to a publicly accessible resource while the one just for my colleagues will point to a resource restricted by access control methods. Both the public and restricted resources are discoverable in the Public Type Index because an app that is not operating as the WebID owner can not see the Private Type Index.

EXAMPLE 1: Public Type Index that is a Listed Document.

```
@prefix solid: <http://www.w3.org/ns/solid/terms#>.
@prefix vcard: <http://www.w3.org/2006/vcard/ns#>.
@prefix bk: <http://www.w3.org/2002/01/bookmark#>.

<>
  a solid:TypeIndex ;
  a solid:ListedDocument.

<#ab09fd> a solid:TypeRegistration;
  solid:forClass vcard:AddressBook;
  solid:instance </public/contacts/myPublicAddressB

<#bq1r5e> a solid:TypeRegistration;
  solid:forClass bk:Bookmark;
  solid:instanceContainer </public/myBookmarks/>.
```

Type registration containing a public resource of type `vcard:AddressBook` located at the resource address `</public/contacts/myPublicAddressBook.ttl>`. And a resources of type `bk:Bookmark` located in the container address `</public/myBookmarks/>`.

2.2 Private Type Index

The Private Type Index document contains registration entries that are private to the user and their apps. It is intended for types of resources meant only to be accessed by who the WebID owner gives access to.

EXAMPLE 2: Private Type Index that is an Unlisted Document.

```
@prefix solid: <http://www.w3.org/ns/solid/terms#
@prefix vcard: <http://www.w3.org/2006/vcard/ns#>
@prefix bk: <http://www.w3.org/2002/01/bookmark#>

<>
  a solid:TypeIndex ;
  a solid:UnlistedDocument.

<#ab09fd> a solid:TypeRegistration;
  solid:forClass vcard:AddressBook;
  solid:instance </private/contacts/myPrivateAddr

<#bq1r5e> a solid:TypeRegistration;
  solid:forClass bk:Bookmark;
  solid:instanceContainer </private/myBookmarks/>
```

Type registration containing a private resource of type `vcard:AddressBook` located at the resource address `</private/contacts/myPrivateAddressBook.ttl>`. And a resources of type `bk:Bookmark` located in the container address `</private/myBookmarks/>`.

2.3 Type Index Registration Entries

The Type Index documents MAY contain any number of statements of type `solid:TypeRegistration` which map RDF classes/types to their locations in a WebID owner's root storage.

The registration entries map a type to a location and MUST use one of two predicates:

`solid:instance` - This predicate maps a type to an individual Solid resource, typically an index or a directory listing resource such as an Address Book.

`solid:instanceContainer` - This predicate maps a type to a Solid container which the client would have to list to get the instances of that type.

3. Supplying missing Type Index documents

If one or both of the Type Indexes are missing, an app needing to write to them that doesn't have Write and Control access to the root storage MAY warn the user that their indexes are missing and that they should use a storage management App to fix their profile.

If one or both of the Type Index documents are missing and the app does have Write and Control access to the root storage, the app MAY create these documents.

The Public Type Index document SHOULD be publicly readable, writable by the user (or a group delegated by the user), and a pointer to it MUST be placed in the WebID Profile Document.

The Private Type Index SHOULD be readable and writeable only by the WebID owner and a pointer to it MUST be placed in the Preferences Document.

> NOTE: Missing pointer to Type Indexes
>
> If the Type Index files are created but the pointer to them are not stored, the risk is that new resource get created, leading to confusion and possible loss of data.

4. Community

Note: This section is "at risk" in that it is more recent and less tested than the rest of this specification.

Type Indexes help find specific types of data in a owner's storage. They connect data to storage locations. Through the use of a community a WebID owner connects to a project or/and an organization. This connection is created to specify the membership of a WebID owner to the project or/and organization.

A WebID owner is part of a project or/and organization if a specific triple is found in either the Public or Private Type Index Document. The triple must have the WebID of the individual as the subject, the `solid:community` as the predicate and the WebId or the project or organization as an object.

EXAMPLE 3: An individual is part of a project or/and organization triple

```
                    <#WebID> <http://www.w3.org/ns/solid/terms#community>
```

5. Considerations

5.1 Privacy considerations

This section is non-normative.

Privacy considerations have been voiced in the past regarding the visibility of different data types in the Public Type Index Document. One can potentially deduce the type of data a WebID owner possesses on their storage.

References

Normative References

[RDF-SYNTAX-GRAMMAR]
RDF-SYNTAX-GRAMMAR. Fabien Gandon; Guus Schreiber. RDF 1.1 XML Syntax. 9 January 2014. W3C Proposed Edited Recommendation. URL: http://www.w3.org/TR/rdf-syntax-grammar/

[RDFA-CORE]
RDFA-CORE. Ben Adida; Mark Birbeck; Shane McCarron; Ivan Herman et al. RDFa Core 1.1 - Second Edition. 22 August 2013. W3C Recommendation. URL: http://www.w3.org/TR/rdfa-core/

[RDF-SCHEMA]
RDF Schema 1.1. Dan Brickley; Ramanathan Guha. W3C. 25 February 2014. W3C Recommendation. URL: https://www.w3.org/TR/rdf-schema/

[RFC2119]
Key words for use in RFCs to Indicate Requirement Levels. S. Bradner. IETF. March 1997. Best Current Practice. URL: https://datatracker.ietf.org/doc/html/rfc2119

[RFC8174]
Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words. B. Leiba. IETF. May 2017. Best Current Practice. URL: https://datatracker.ietf.org/doc/html/rfc8174

[SHACL]
SCHACL. Holger Knublauch; Dimitris Kontokostas. Shapes Constraint Language (SHACL). 20 July 2017. REC. URL: https://www.w3.org/TR/shacl/

[ShEx]
ShEx. Eric Prud'hommeaux; et al. Shape Expressions Language 2.1. URL: http://shex.io/shex-semantics/index.html

[SOLID-PROTOCOL]
Solid Protocol. Sarven Capadisli; Tim Berners-Lee; Ruben Verborgh; Kjetil Kjernsmo. W3C Solid Community Group. 31 December 2022. Version 0.10.0

URL: https://solidproject.org/TR/protocol

[turtle]

turtle. Eric Prud'hommeaux; Gavin Carothers. RDF 1.1 Turtle. 9 January 2014. W3C Proposed Recommendation. URL: http://www.w3.org/TR/turtle/

[VCARD-RDF]

vCard Ontology - for describing People and Organizations. Renato Iannella; James McKinney. W3C. 22 May 2014. W3C Note. URL: https://www.w3.org/TR/vcard-rdf/

[WEBID]

WebID 1.0. Andrei Sambra; Stéphane Corlosquet. W3C WebID Community Group. 5 March 2014. W3C Editor’s Draft. URL: https://www.w3.org/2005/Incubator/webid/spec/identity/

Informative References

[COGA-USABLE]

Making content usable for people with cognitive and learning disabilities. Lisa Seeman-Horwitz; Rachael Bradley Montgomery; Steve Lee; Ruoxi Ran. W3C. 11 December 2020. W3C Working Draft. URL: https://www.w3.org/TR/coga-usable/

[PRIVACY-PRINCIPLES]

Privacy Principles. Robin Berjon; Jeffrey Yasskin. W3C. 12 May 2022. W3C Group Draft Note. URL: https://www.w3.org/TR/privacy-principles/

[SECURITY-PRIVACY-QUESTIONNAIRE]

Self-Review Questionnaire: Security and Privacy. Theresa O'Connor; Peter Snyder. W3C. 23 March 2021. W3C Note. URL: https://www.w3.org/TR/security-privacy-questionnaire/

W3C Accessibility Guidelines (WCAG) 3.0. Jeanne F Spellman; Rachael Bradley Montgomery; Shawn Lauriat; Michael Cooper. W3C. 21 January 2021. W3C Working Draft. URL: https://www.w3.org/TR/wcag-3.0/

↑