

Solid Data Modules

Using Solid as a Platform for Data Portability

Michiel de Jong, software engineer and founder of Ponder Source.



APP DEVS

SOLVE

PROBLEMS

DATA DESIGN

IS PART OF

APP DESIGN

RIGHT?

“Single-app” Solid



RWW

Your webserver
machine read/writable
and linkable to others

HTML & RDF

“data browsers”

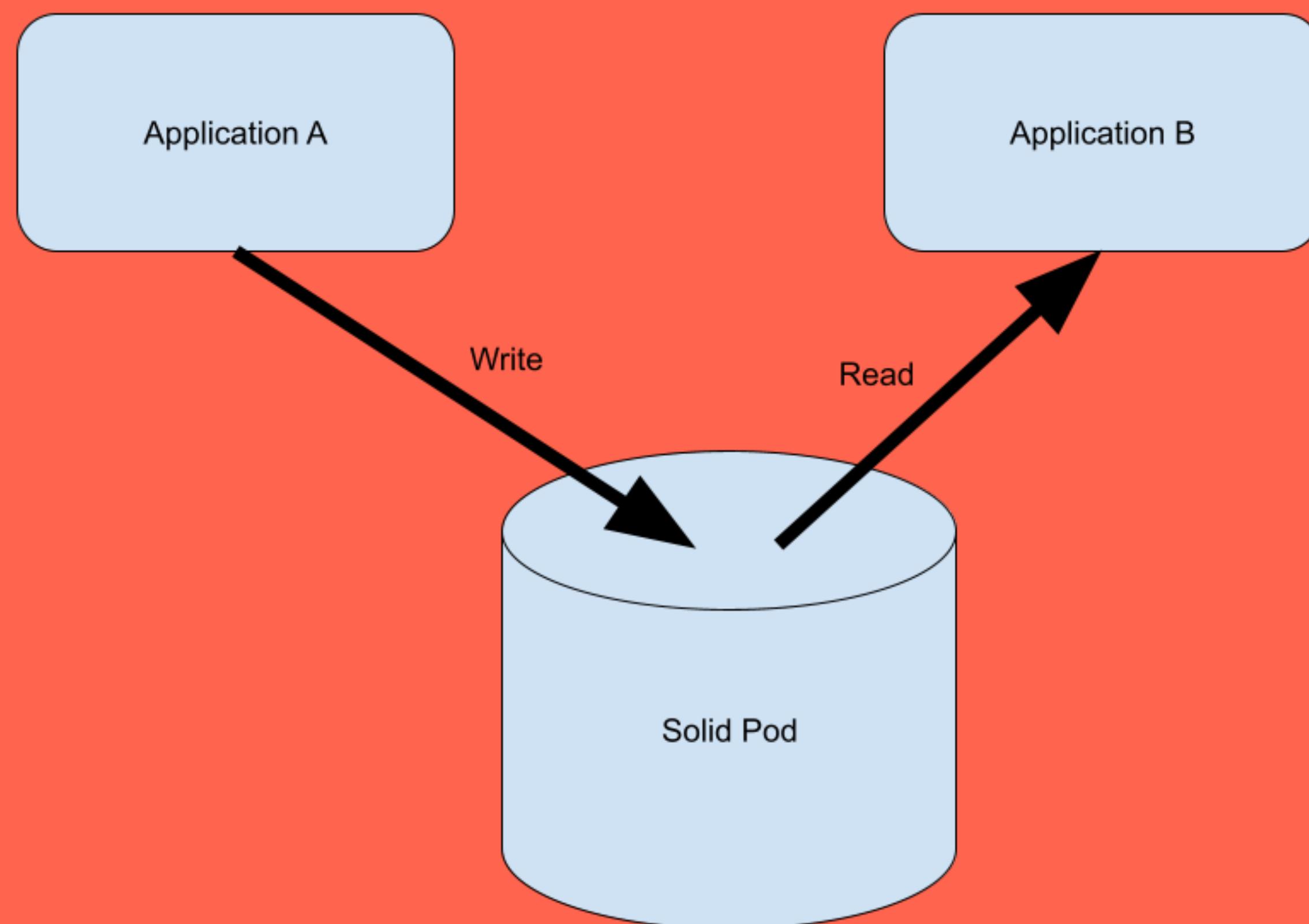
form generators

“Multi-app” Solid



Data pivot

THE POD AS A DATA PIVOT



Interop + exit

=

Data Portability



MY SOLID ROADMAP

AUTH AGENT

BRIDGES

APPS

Indexer

Pivot

DATA SYNC

SERVERS

DATA MODULES

TEST SUITES

CLIENT-CLIENT

SPEC

SPECS

Server Behaviour

DATA CONVENTIONS

Access Control

Document

Translate

Reusable Code

The screenshot shows the homepage of shaperepo.com. At the top, there's a navigation bar with icons for back, forward, search, and refresh, followed by the URL 'shaperepo.com'. Below this is a dark header bar with the 'ShapeRepo' logo on the left, a search input field 'Find shapes' with a magnifying glass icon, and a 'Learn How' button. The main content area features a large heading 'Make your apps interoperable' and a subtext 'Browse a library of data shapes to help you structure your data so that other apps can understand it.' Below this is another search bar. Further down, there's a section titled '217 Schemas' with two examples: 'long chat' and 'solid profile', each with its corresponding schema URL.

shaperepo.com

ShapeRepo

Find shapes

Learn How

Find shapes

217 Schemas

long chat

<https://shaperepo.com/schemas/longChat>

solid profile

<https://shaperepo.com/schemas/solidProfile>

ShapeRepo

Find shapes



long chat

<https://shaperepo.com/schemas/longChat> ↗

Table View

ShexC View

ShexJ View

ChatShape

Allows Extra Properties: type

Each Of:

Property	Type
type	http://www.w3.org/ns/pim/meeting#LongChat
author	iri
Date Created	Date-Time
Title	String
participation	@https://shaperepo.com/schemas/longChat#ChatParticipationShape
Shared Preferences	iri

ChatParticipationShape

Each Of:

TABLE OF CONTENTS

Abstract

Status of This Document

- | | |
|-----|--------------|
| 1 | Introduction |
| 1.1 | Terminology |
| 1.2 | Namespaces |
| 1.3 | Conformance |

2 Chat Channel

3 Chat Message

4 Threads and Replies

5 Deleted and Modified Message

6 Actions

7 Access Control

8 Preferences

9 Discovery

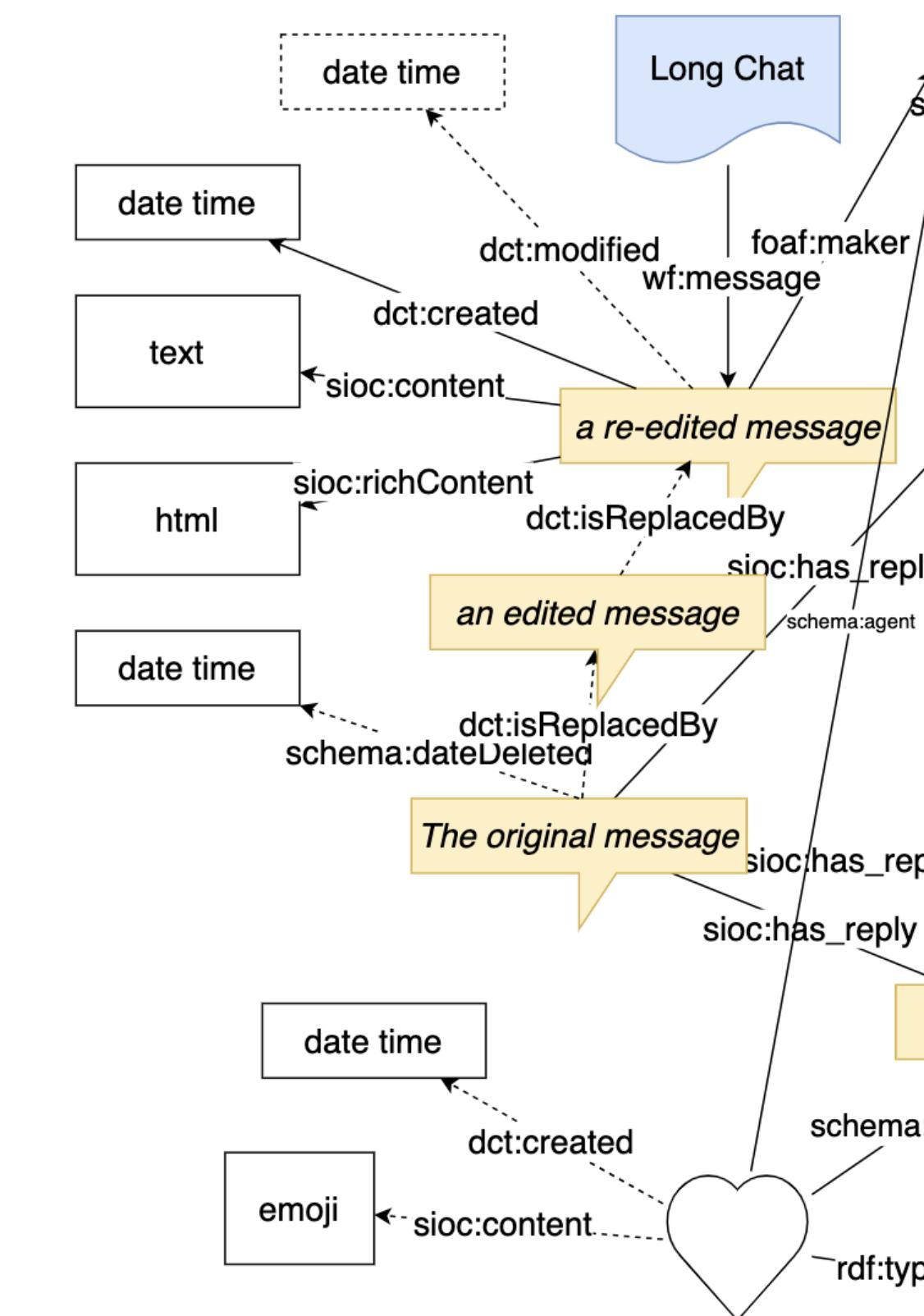
10 Considerations

- 10.1 Security Considerations
 - 10.2 Privacy Considerations
 - 10.3 Internationalization Considerations
 - 10.4 Accessibility Considerations

References

Normative References

Informative References



A simple Long Chat instance

A SIMPLE LONG CHAT INSTANCE

in \$ROOT/index.ttl

```
@prefix : <#>.  
@prefix dc: <http://pu...  
@prefix meeting: <http://
```



Awesome Solid >

Conventions ^

—
Addressbook

Bookmark

Chat

Finance

Food

Images

Meeting

Movies

Notepad

Overview

Profile

To add a message in the LongChat convention, instance, /long-chat/2019/04/17/chat.ttl, following triples to /long-chat/2019/04/17/chat.ttl:

```
</long-chat/2019/04/17/chat.ttl#Msg1555  
</long-chat/2019/04/17/chat.ttl#Msg1555  
</long-chat/2019/04/17/chat.ttl#Msg1555  
</long-chat/index.ttl#this>
```

Note that there is no need to make /long-chat/index.ttl, since it can be discovered via /long-chat/, /long-chat/2019/, /long-chat/2019/04/17/.

Also note that here too, for the chat convention, whereas for the individual chat messages:

One-to-One Chat

One-to-One chat on top of Solid is used, it stores messages on the pod of the sender.

A note about existing implementations:

SOLID
DATA
MODULES



A SOLID BOOKMARK

```
@prefix bk: <http://www.w3.org/2002/01/bookmark#> .  
  
<https://ME.example.org/animals.ttl#Animals>  
a bk:Topic .  
  
<https://YOU.example.org/birds.ttl#Birds>  
a bk:topic ;  
bk:subTopicOf <https://ME.example.org/animals.ttl#Animals> .
```

A SOLID BOOKMARK

```
@prefix boo: <http://www.w3.org/2002/01/bookmark#>.  
@prefix crdt: <https://vocab.noeldemartin.com/crdt/>.
```

bookm:b93d9944-d54d-42f6-a39b-6ea3f9217763

```
a boo:Bookmark;  
rdfs:label "sdf";  
boo:hasTopic "sdfg";  
boo:id "b93d9944-d54d-42f6-a39b-6ea3f9217763";  
boo:recalls <http://example.com>.
```

bookm:b93d9944-d54d-42f6-a39b-6ea3f9217763-metadata

A SOLID BOOKMARK

```
{  
  
  "url": "https://www.google.com",  
  "title": "Google",  
  "description": "alfa",  
  "tags": [  
    "bravo"  
  ],  
  "createdAt": "2023-11-30T15:02:09.711Z",  
  "updatedAt": "2023-11-30T15:08:29.561Z",  
  "id": "a9308a39c5294b0b9269d0c650d8c5f1",  
  "@context": "http://remotestorage.io/spec/modules/bookmarks/archive-bookmark"  
}
```

A SOLID BOOKMARK

```
{  
  "JBXDocumentURL": "https://alsarahthenubatones.bandcamp.com/album/silt",  
  "JBXDocumentName": "Alsarah & the Nubatones: Silt",  
  "JBXDocumentNotes": "",  
  "JBXDocumentTags": [  
    "#listen"  
  ],  
  "JBXDocumentCreationDate": "2023-11-30T15:26:46.811Z",  
  "JBXDocumentModificationDate": "2023-11-30T15:26:46.811Z",  
  "JBXDocumentID": "01HGGDDWGV1A9KFYMNWKFY63ZP",  
  "JBXDocumentEmbedURL": "https://bandcamp.com/EmbeddedPlayer/v=2/album=704487352",  
  "JBXDocumentImageURL": "https://f4.bcbits.com/img/a3732065656_5.jpg",  
  "JBXDocumentDidFetch": true
```

A SOLID BOOKMARK

```
@prefix terms: <http://purl.org/dc/terms/>.
```

```
@prefix bookm: <http://www.w3.org/2002/01/bookmark#>.
```

```
@prefix XML: <http://www.w3.org/2001/XMLSchema#>.
```

```
<> terms:references <#0.716493818605209.ttl>.
```

```
<#0.716493818605209.ttl>
```

```
  a bookm:Bookmark;
```

```
  terms:created "2023-12-13T10:11:12.842Z"^^XML:dateTime;
```

```
  terms:title "alfa";
```

SOLID BOOKMARK MODULE

```
/**  
 * Maps the title from a Solid dataset Thing.  
 *  
 * @param thing - The Thing to map  
 * @returns The title as a string  
 * @internal  
 */  
  
private static mapTitle(thing: ThingPersisted): string {  
    return (  
        getLiteral(thing, DCTERMS.title)?.value ??  
        getLiteral(thing, RDFS.label)?.value ??  
        getLiteral(thing, AS.name)?.value ??  
        ...  
    );  
}
```

 README



Weekly meeting: Fridays 4pm-5pm Amsterdam time in <https://meet.jit.si/solid-data-modules>

Data types covered

(coming soon)

Data types for which we will produce Solid Data Modules:

- bookmarks
- chat
- contacts
- maps
- tasks

Paradigms covered

(coming soon)

The "vanilla" version of each of these data modules will be usable in combination with just an authenticated fetcher, such as [Solid Auth Fetcher](#), or the fetcher from the [Inrupt JS Libs](#). Apart from this "vanilla" version, we will also package each data module for use in combination with the following client-side Solid app paradigms:

- [Soukai-Solid](#)
- [rdflib](#)
- [LDO](#)

DATA MODULES DX

<> **Code** ● Issues 18 Pull requests 2 Discussions Actions Projects Wiki Security Insights

main ▾ [data-modules / contacts / examples / add-contact-to-group.mjs](#)

angelo-v contacts: export ContactsModuleRdfLib as default

Code Blame 19 lines (15 loc) · 814 Bytes

```
1 import ContactsModule from "../dist/index.js";
2 import { Fetcher, graph, UpdateManager } from "rdflib";
3
4 const store = graph();
5 const fetcher = new Fetcher(store);
6 const updater = new UpdateManager(store);
7 const contacts = new ContactsModule({store, fetcher, updater});
8
9 const contactUri = "http://localhost:3000/alice/public-write/ab9694d6-120e-415d-a315-90cd84c2e062/Person/854486c1-504a-4e11-9";
10 const groupUri = "http://localhost:3000/alice/public-write/ab9694d6-120e-415d-a315-90cd84c2e062/Group/41faaf6f-37c3-437a-900d-";
11
12 await contacts.addContactToGroup({
13   contactUri,
14   groupUri,
15 });
16 console.log("added contact", contactUri, "to group", groupUri);
17
```

DATA MODULES DX

The screenshot shows a browser window displaying code examples for a Data Module named 'bookmarks'. The URL in the address bar is `github.com/solid-contrib/data-modules/tree/main/bookmarks/vanilla`. The page title is `data-modules / bookmarks / vanilla /`. The code examples are presented in code blocks with copy icons.

```
import { Bookmark } from "@solid-data-modules/bookmarks-vanilla"
```

use it to work with bookmarks

```
const result: IBookmark[] = await Bookmark.getAll(session.fetch, session.info.webId, "<defaultPrivateBookmarkDocUrl>"
```

the `getAll` method will return an array of `IBookmark`, in case there are no bookmarks it will return an empty array.

```
const result: IBookmark | undefined = await Bookmark.get("url", session.fetch);
```

the `get` method will return an `IBookmark`, in case there is no bookmark it will return `undefined`.

```
const result: boolean = await Bookmark.delete("url", session.fetch);
```

the `delete` method will return a boolean indicating if the bookmark was deleted.

```
const result: boolean = await Bookmark.create(payload: ICreateBookmark, session.fetch, session.info.webId, "<defaultPrivateBookmarkDocUrl>"
```

the `create` method will return a boolean indicating if the bookmark was created.

```
const result: IBookmark | undefined = await Bookmark.update("url", payload: IUpdateBookmark, session.fetch);
```

GITHUB.COM/
SOLID-CONTRIB/
DATA-MODULES



Solid Data Modules

As The Engine Of

Solid App Interop



Shape Trees
As The Engine Of
Data Discovery

Pod Indexer

As The Engine Of

Data Discovery



Shape Trees

As The Engine Of

WAC ACLs

Solid Data Modules

+ Indexer +

App Launcher

SOLIDCOMMUNITY.NET



PIVOT

remoteStorage

SOLID

Backend

Cambria Project

Declarative lenses

+ CRDT sync

THE ULTIMATE BOOKKEEPING SYSTEM

