*Thoughts on*

# Solid Developer Experience

2nd Solid Symposium
May 2nd 2024, Leuven

Noel De Martin

# Hello, my name is Noel

⏳🧠🔥

- Developer & Entrepreneur.
- 4-day workweek @ **Moodle**.
- Making Solid Apps on the side.
- Working in the Open.

noeldemartin.com | noeldemartin.social | @NoelDeMartin

# My Journey with Solid

Solid Focus

August 2019

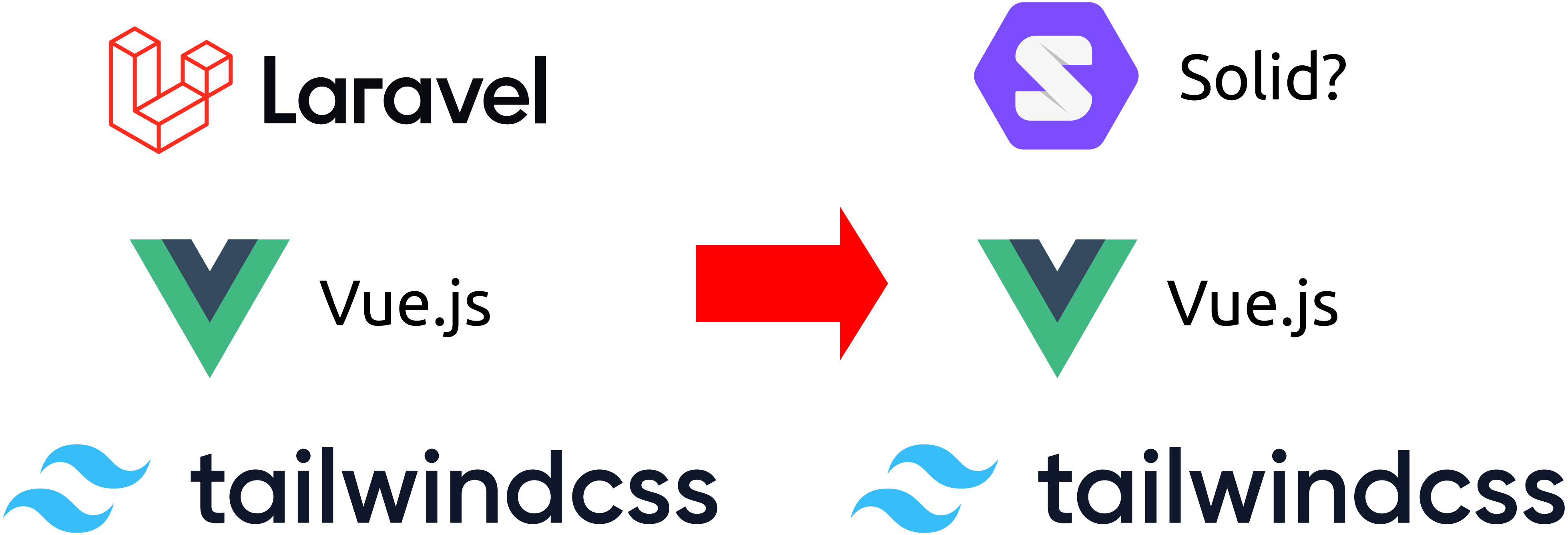
Media Kraken

July 2020

Umai

January 2023

# My Stack



Laravel

Vue.js

tailwindcss

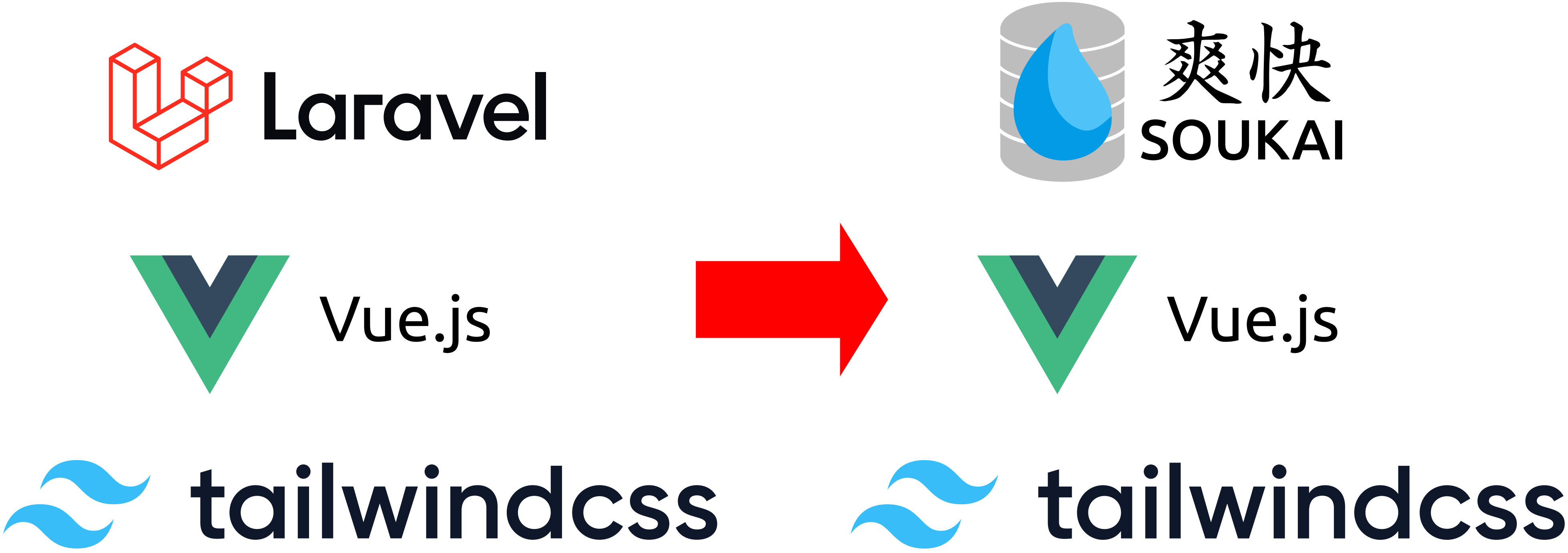
Solid?

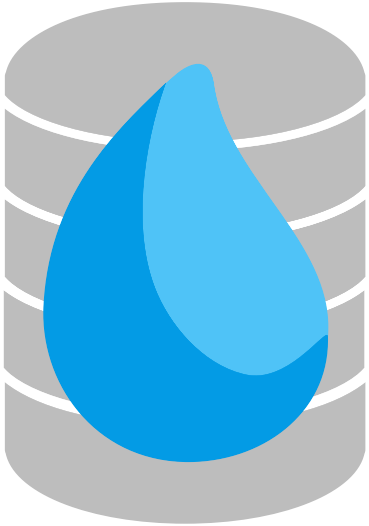
Vue.js

tailwindcss

noeldemartin.com/solid-symposium-dx

# My Stack





# 爽快
# SOUKAI

soukai.js.org

# Active Record

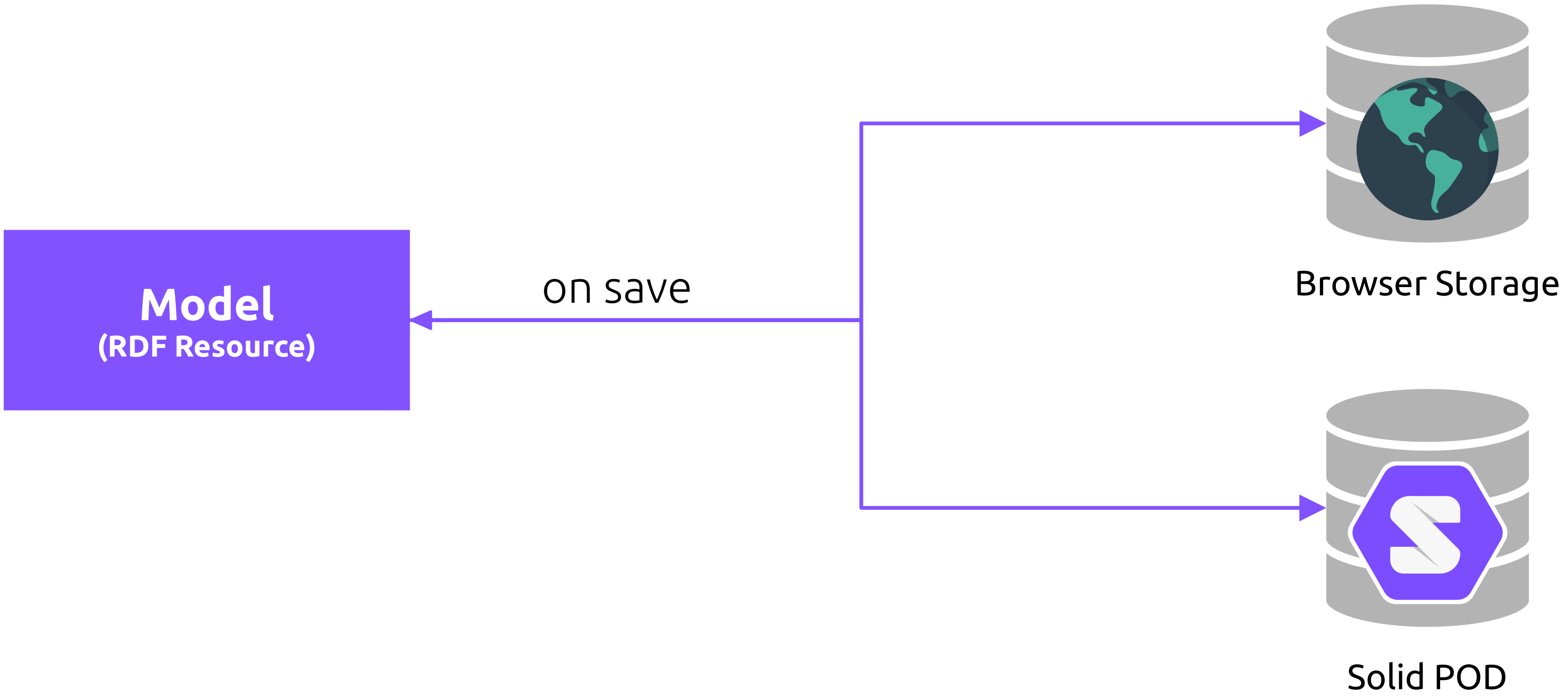
“An object that wraps a row in a database table or view, encapsulates the database access, and adds domain logic on that data.”

Source: Martin Fowler, Patterns of Enterprise Application Architecture

# Active Record

An object that wraps an RDF resource in a container, encapsulates network requests, and adds domain logic on that data.

# Active Record



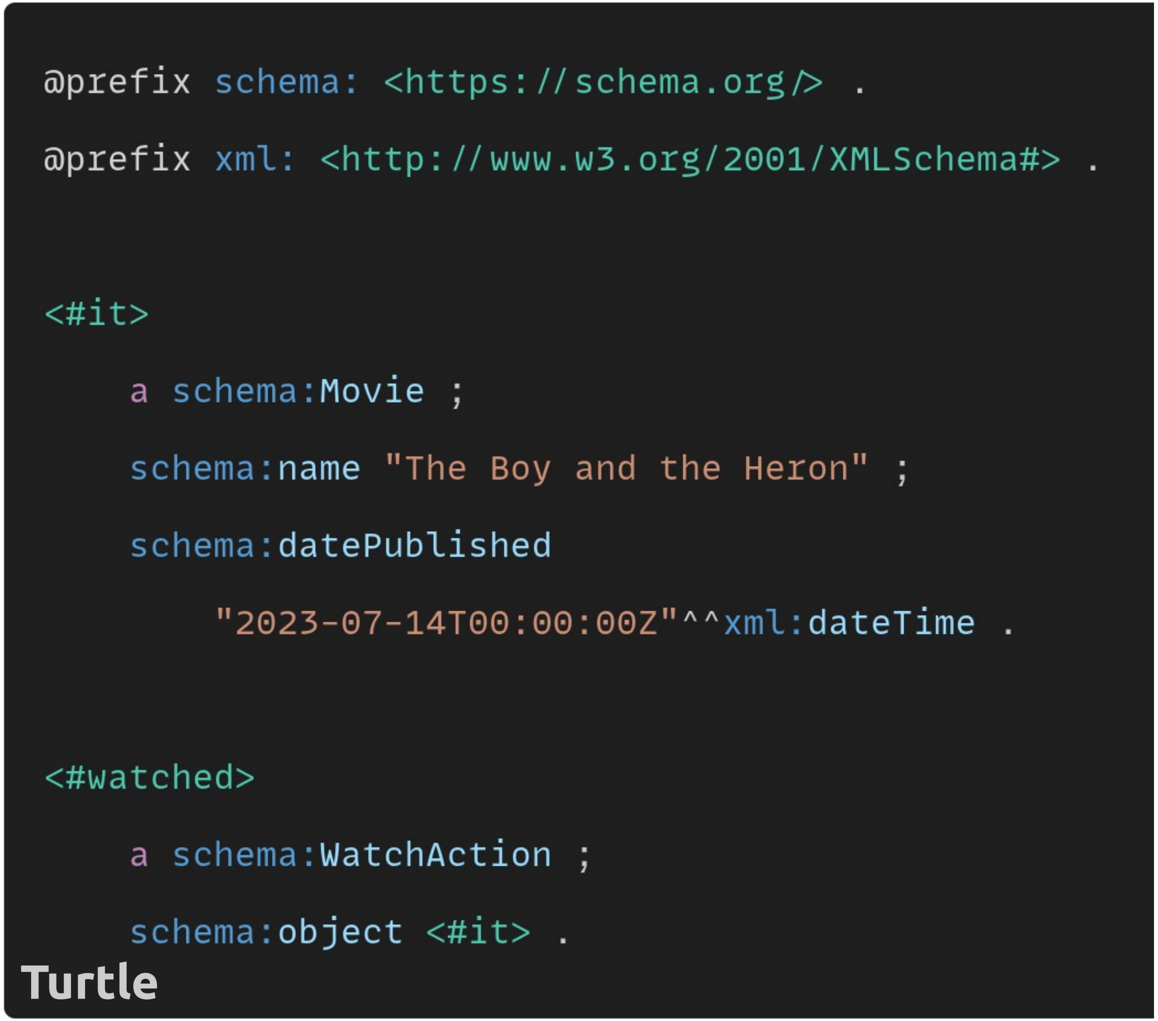
For Solid, see github.com/noeldemartin/soukai-solid

# Is RDF Complicated?

- RDF Primer + RDF Schema === OOP
- Container === Database collection
- RDF Resource === Database document

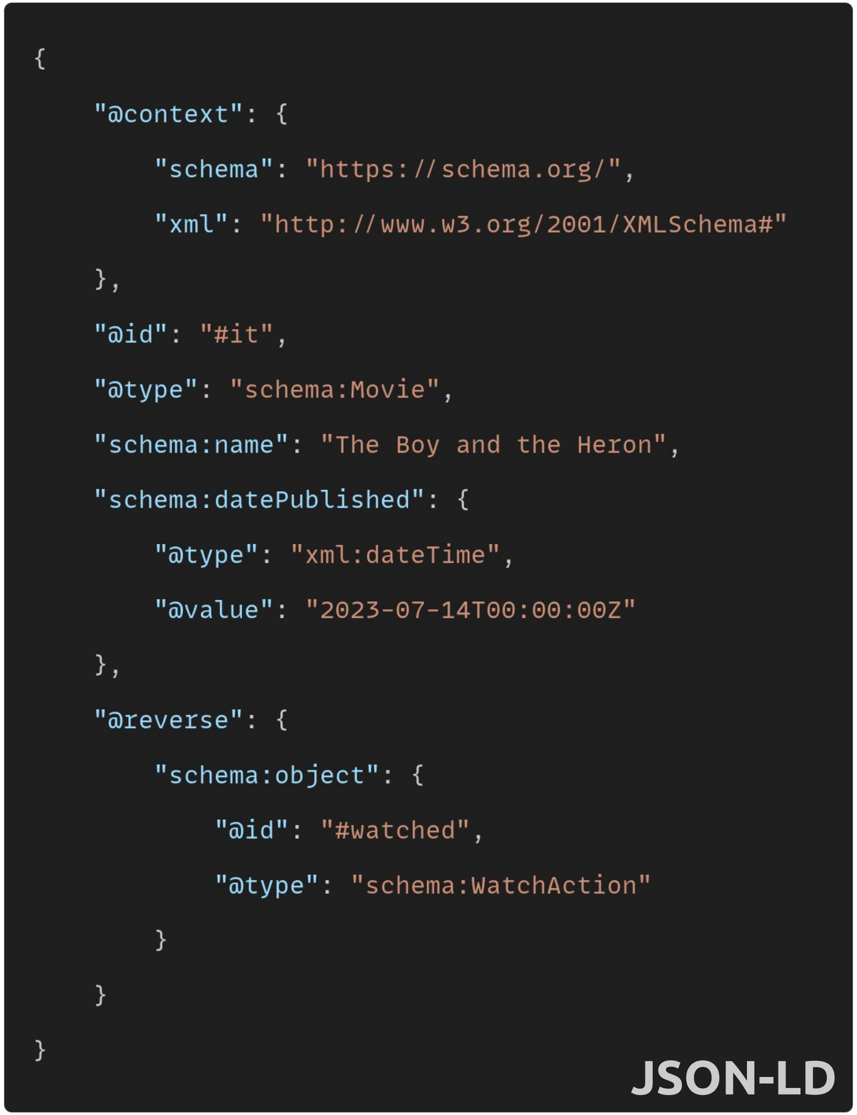
(let’s ignore blank nodes, serialization formats, etc. 😅)

# How I think about RDF



```
@prefix schema: <https://schema.org/> .
@prefix xml: <http://www.w3.org/2001/XMLSchema#> .

<#it>
    a schema:Movie ;
    schema:name "The Boy and the Heron" ;
    schema:datePublished
        "2023-07-14T00:00:00Z"^^xml:dateTime .

<#watched>
    a schema:WatchAction ;
    schema:object <#it> .
```

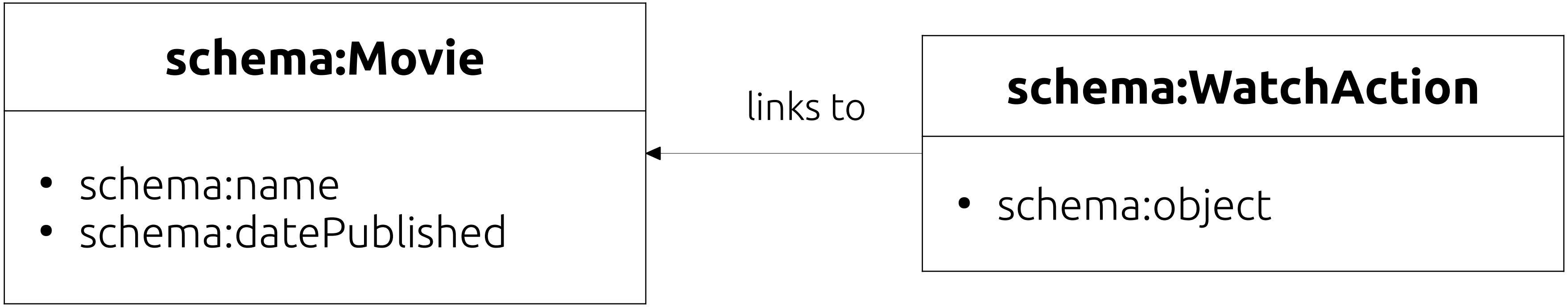
Turtle



```
{
    "@context": {
        "schema": "https://schema.org/",
        "xml": "http://www.w3.org/2001/XMLSchema#"
    },
    "@id": "#it",
    "@type": "schema:Movie",
    "schema:name": "The Boy and the Heron",
    "schema:datePublished": {
        "@type": "xml:dateTime",
        "@value": "2023-07-14T00:00:00Z"
    },
    "@reverse": {
        "schema:object": {
            "@id": "#watched",
            "@type": "schema:WatchAction"
        }
    }
}
```

JSON-LD

# How I think about RDF



**schema:Movie**

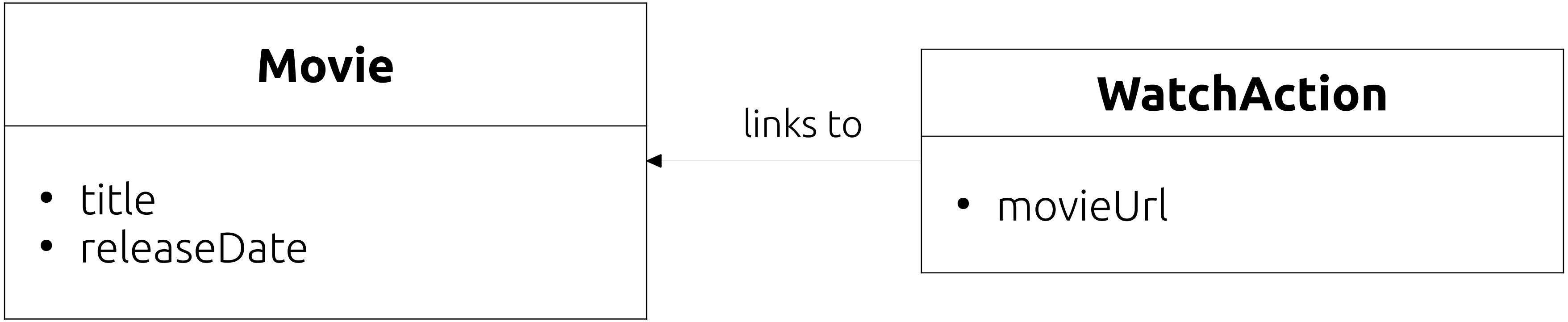
- schema:name
- schema:datePublished

links to

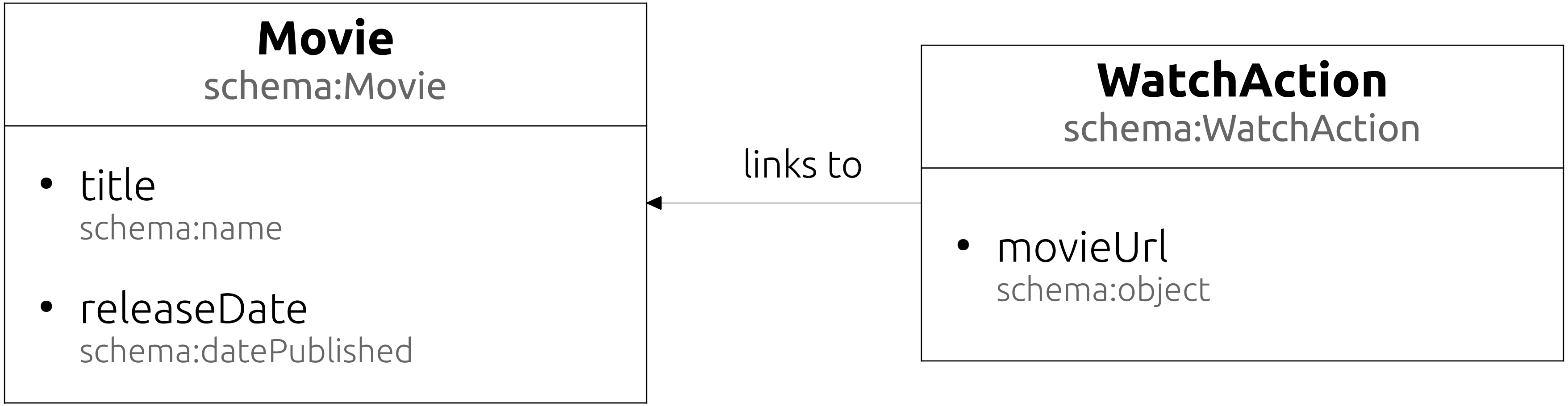
**schema:WatchAction**

- schema:object

# How I think about RDF



# How I think about RDF



**Movie**
schema:Movie

- title
  schema:name
- releaseDate
  schema:datePublished

links to

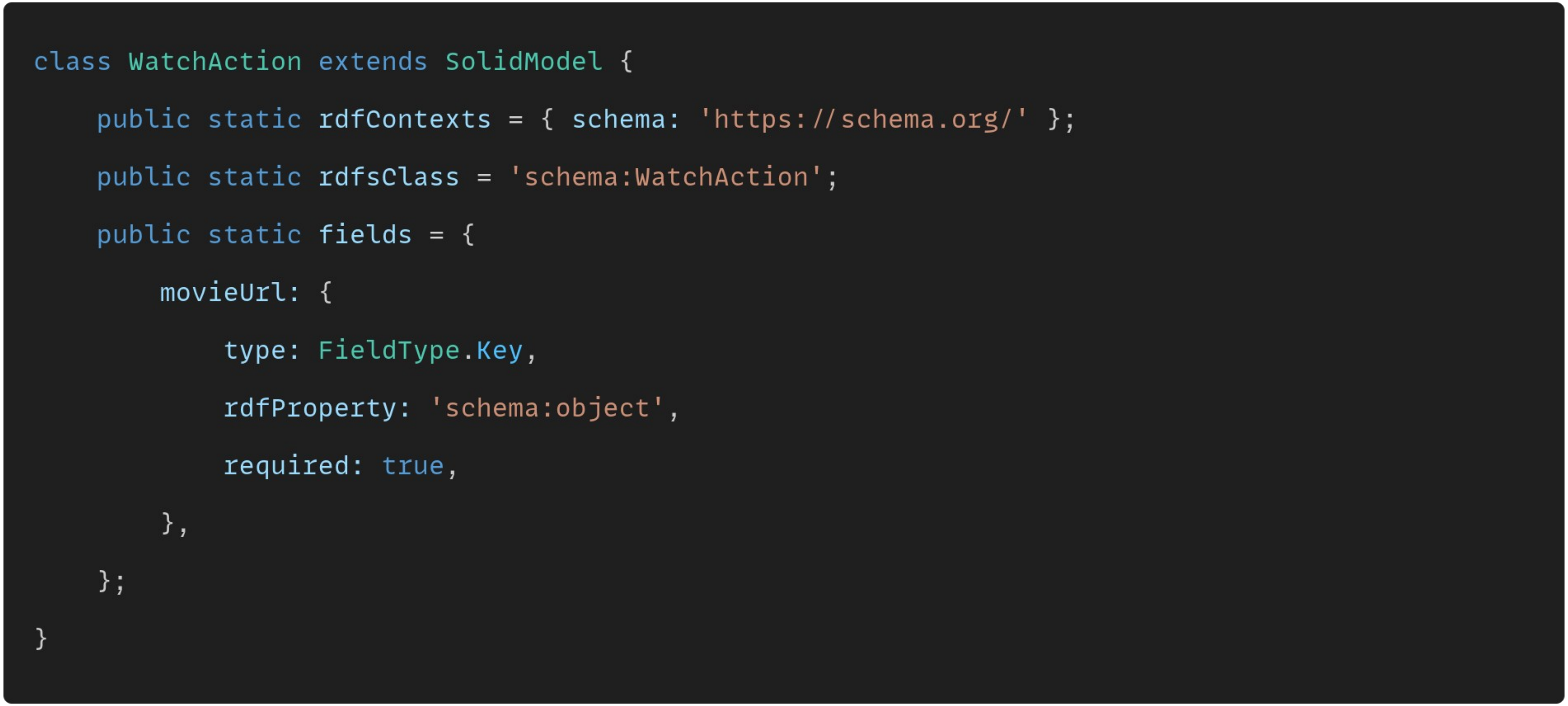
**WatchAction**
schema:WatchAction

- movieUrl
  schema:object

# Model Definitions

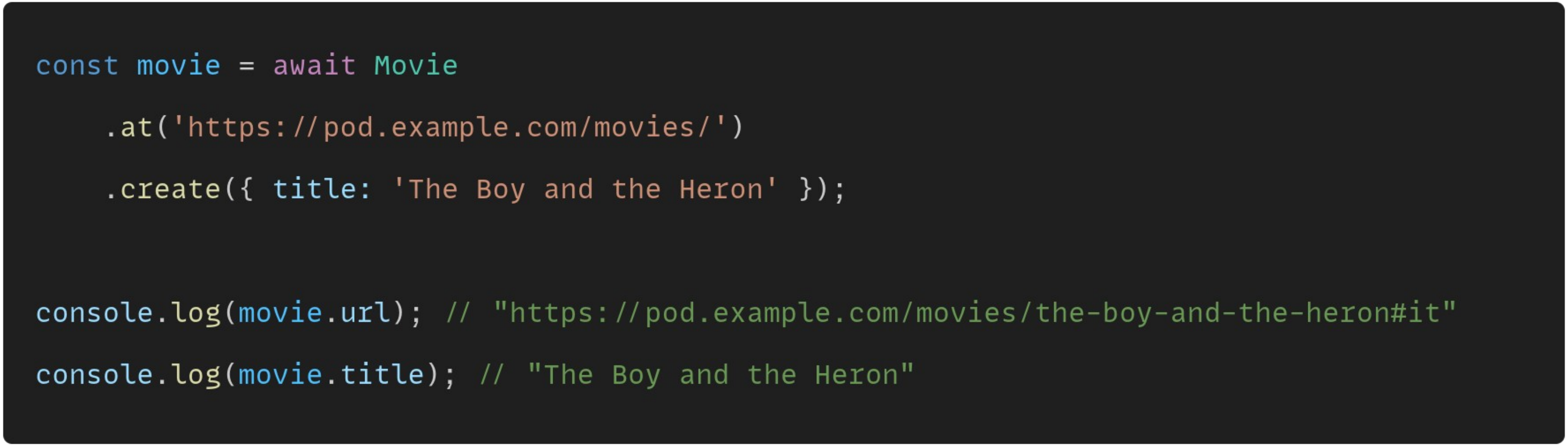
```
class Movie extends SolidModel {
    public static rdfContexts = { schema: 'https://schema.org/' };
    public static rdfsClass = 'schema:Movie';
    public static fields = {
        title: {
            type: FieldType.String,
            rdfProperty: 'schema:name',
            required: true,
        },
        releaseDate: {
            type: FieldType.Date,
            rdfProperty: 'schema:datePublished',
        },
    };
}
```

# Model Definitions



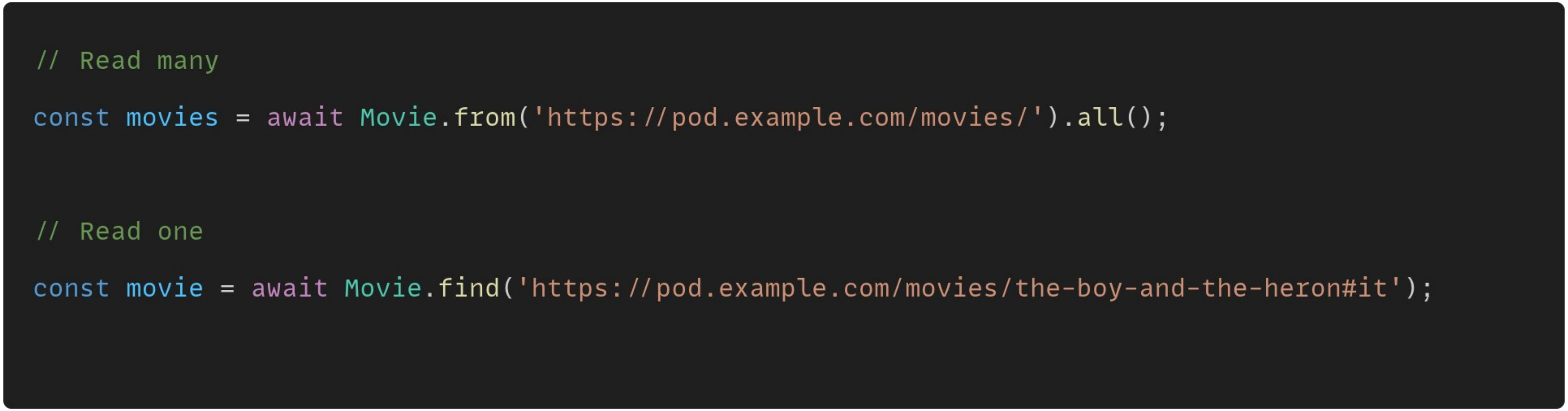
```
class WatchAction extends SolidModel {
    public static rdfContexts = { schema: 'https://schema.org/' };
    public static rdfsClass = 'schema:WatchAction';
    public static fields = {
        movieUrl: {
            type: FieldType.Key,
            rdfProperty: 'schema:object',
            required: true,
        },
    };
}
```

# Basic Operations – Create



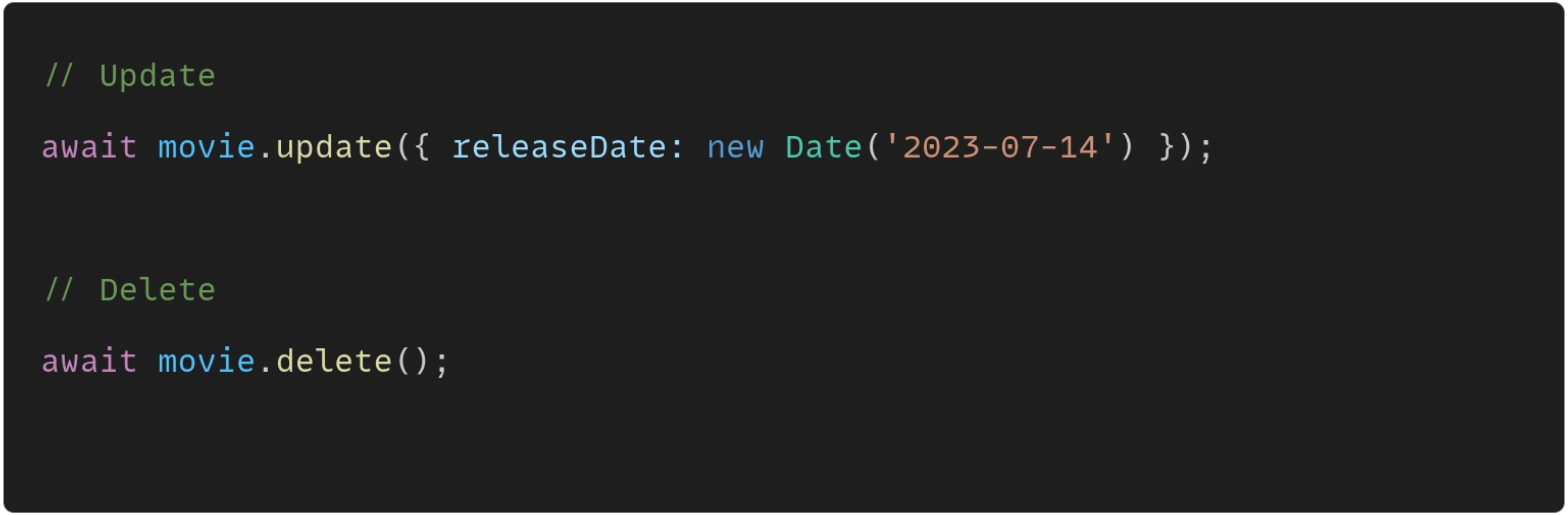
```
const movie = await Movie
    .at('https://pod.example.com/movies/')
    .create({ title: 'The Boy and the Heron' });

console.log(movie.url); // "https://pod.example.com/movies/the-boy-and-the-heron#it"
console.log(movie.title); // "The Boy and the Heron"
```

# Basic Operations – Read



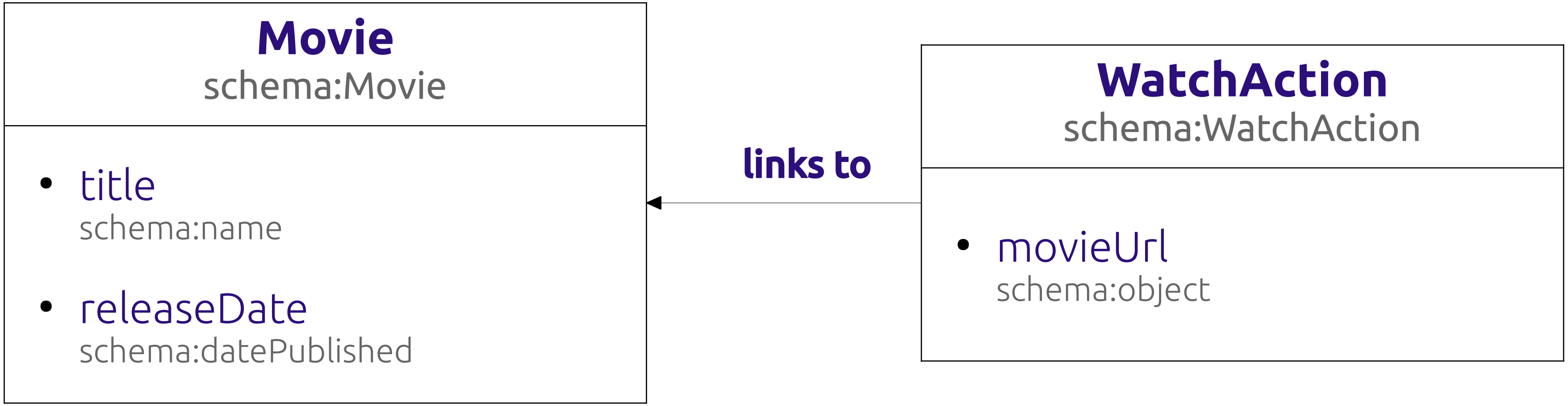
```
// Read many
const movies = await Movie.from('https://pod.example.com/movies/').all();


// Read one
const movie = await Movie.find('https://pod.example.com/movies/the-boy-and-the-heron#it');
```

# Basic Operations – Update & Delete



```
// Update
await movie.update({ releaseDate: new Date('2023-07-14') });


// Delete
await movie.delete();
```

# Relationships



**Movie**
schema:Movie

- title
  schema:name
- releaseDate
  schema:datePublished

**links to**

**WatchAction**
schema:WatchAction

- movieUrl
  schema:object

noeldemartin.com/solid-symposium-dx

# Relationships

```
class Movie extends SolidModel {

    // ...

    public get watched() {
        return !!this.watchAction;
    }

    public async watch() {
        await this.relatedWatchAction.create();
    }

    public watchActionRelationship() {
        return this.hasOne(WatchAction, 'movieUrl');
    }

}
```

# All together!

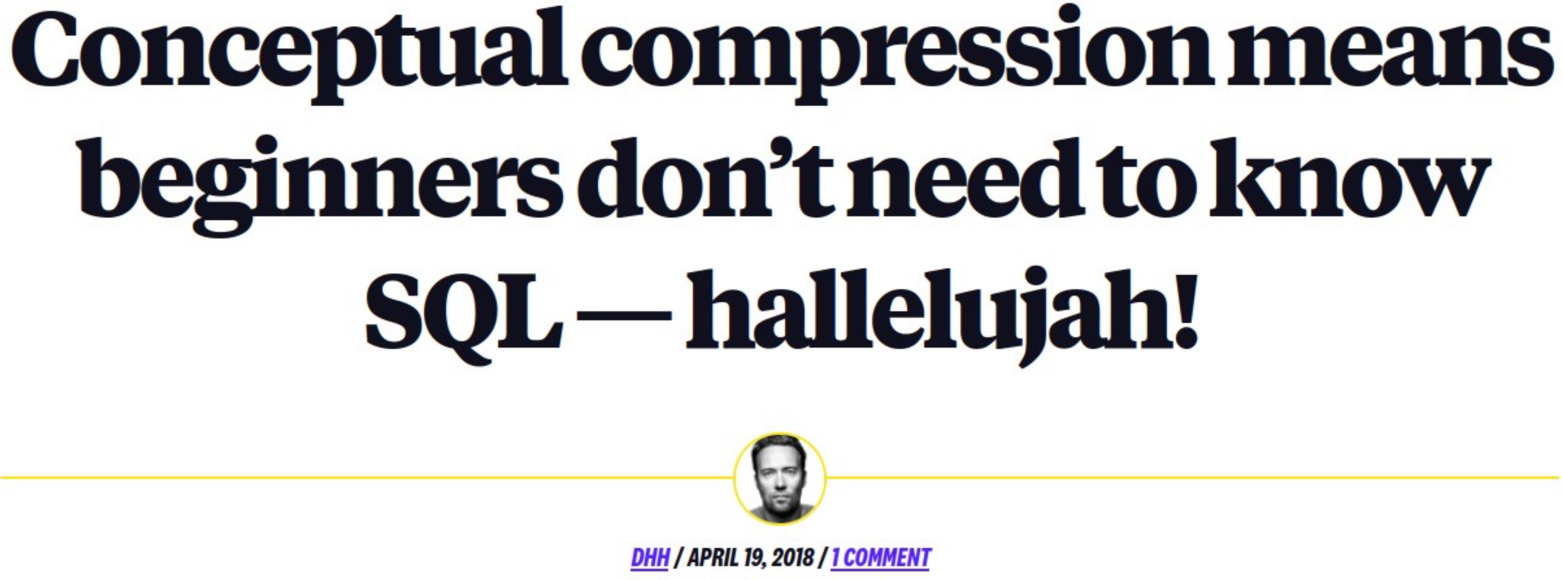
```
// Get all the movies...
const movies = await Movie.from('https://pod.example.com/movies/').all();

// Filter pending...
const pendingMovies = movies.filter(movie => !movie.watched);

// Add a new movie...
const movie = await Movie
    .at('https://pod.example.com/movies/')
    .create({ title: 'The Boy and the Heron' });

// Watch it...
await movie.watch();
```

# And more...

- TypeScript support with *Magic Attributes* ✨
- Url minting
- `toJsonLD()`, `toTurtle()`, ...
- Changes history tracking
  (tune in tomorrow @ Real Time Solid)
- ... Check out the docs!

# Beware of
# The Learning Curve

# Conceptual Compression

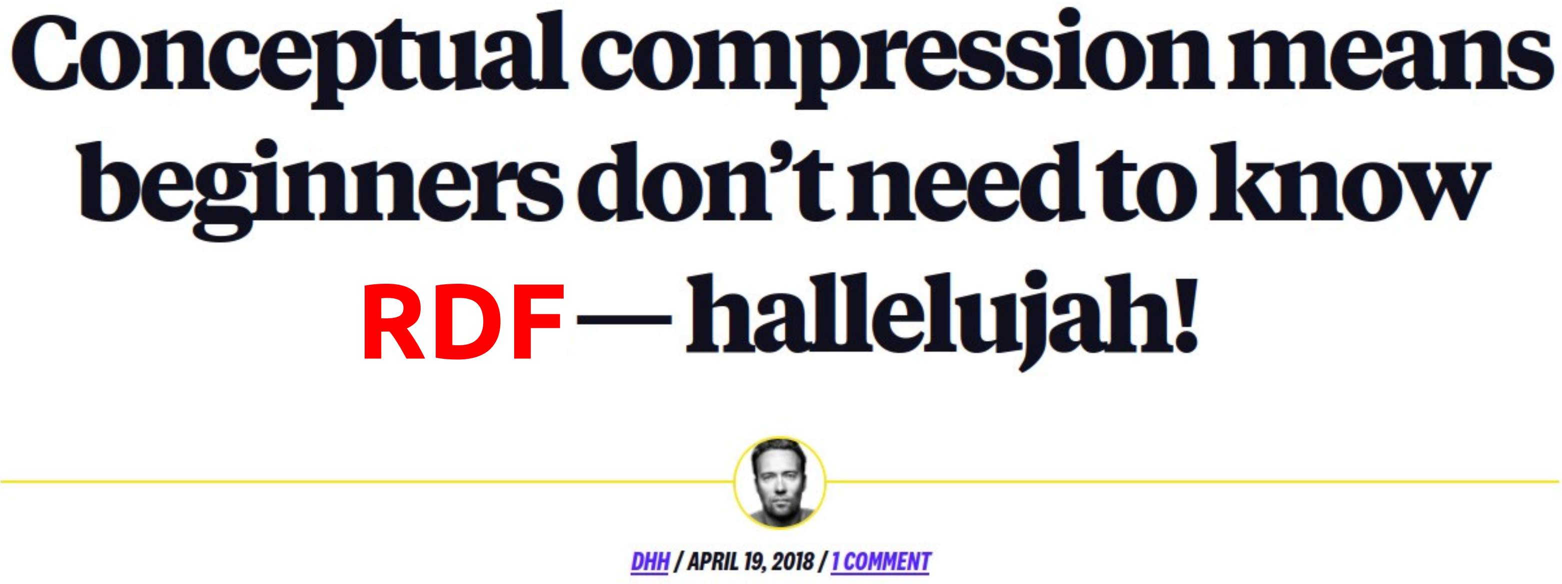


Conceptual compression means beginners don’t need to know SQL — hallelujah!

DHH / APRIL 19, 2018 / 1 COMMENT

signalvnoise.com/svn3/conceptual-compression-means-beginners-dont-need-to-know-sql-hallelujah/

# Conceptual Compression



Conceptual compression means beginners don’t need to know RDF—hallelujah!

DHH / APRIL 19, 2018 / 1 COMMENT

signalvnoise.com/svn3/conceptual-compression-means-beginners-dont-need-to-know-sql-hallelujah/

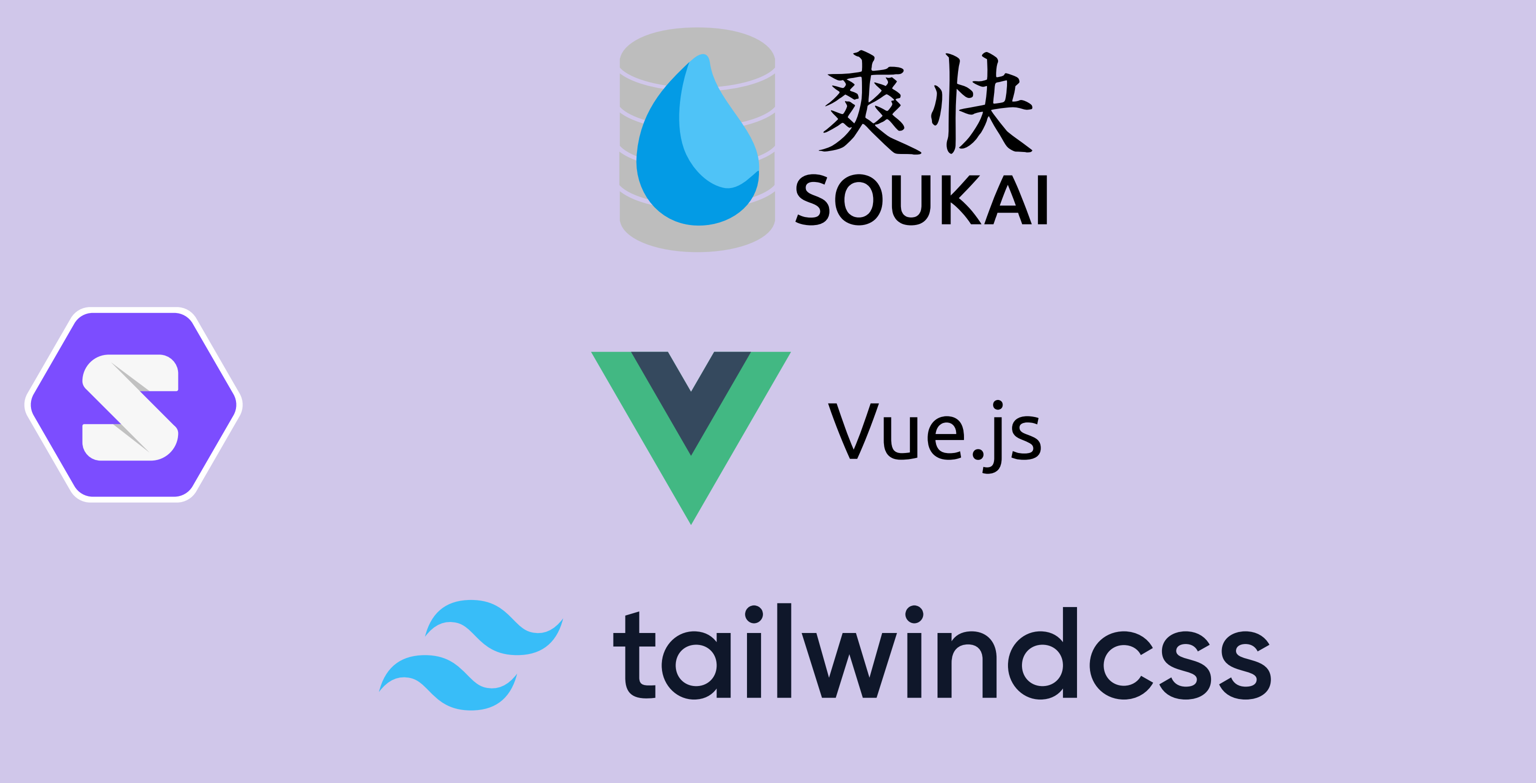
# Is that all?

爽快
SOUKAI

Vue.js

tailwindcss

# Is that all?



Aerogel

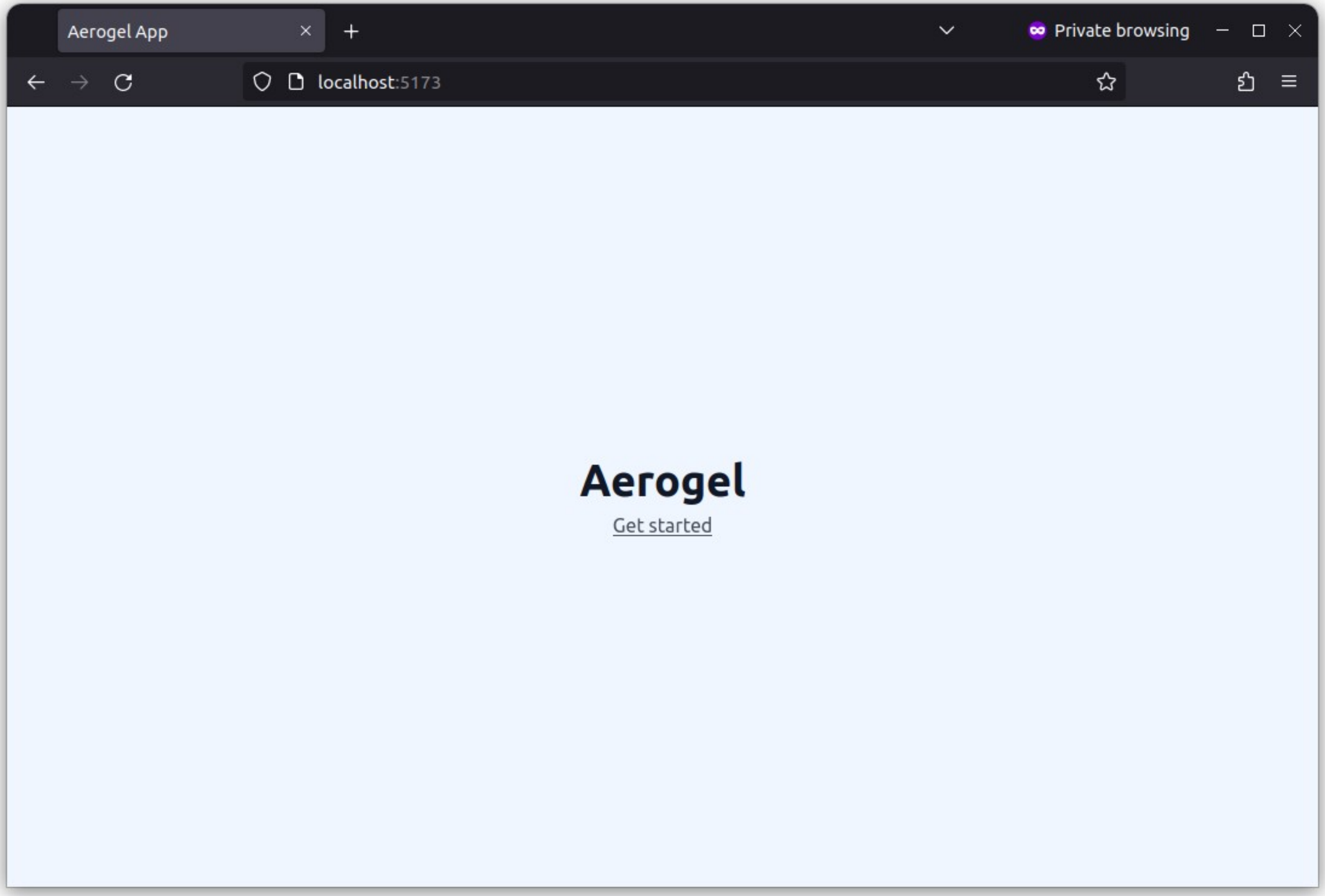
The Lightest Solid

Work in Progress!

aerogel.js.org

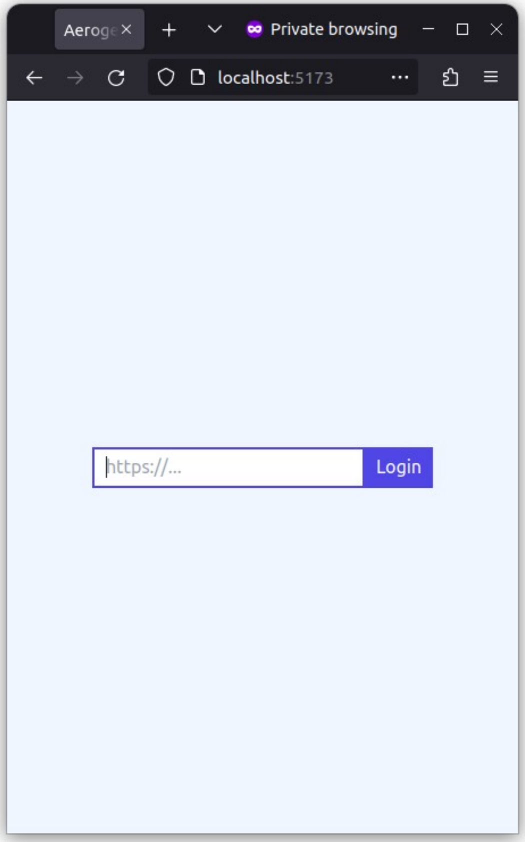
# Quick Start

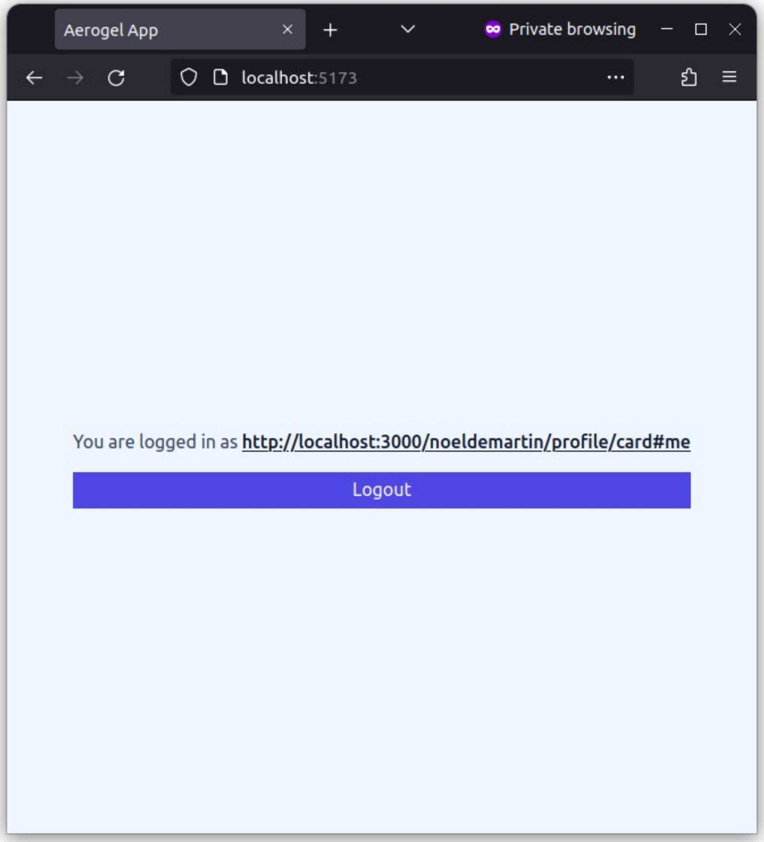
```
npm create aerogel@next app
```



# Components

```
<AGSolidStatus />
```

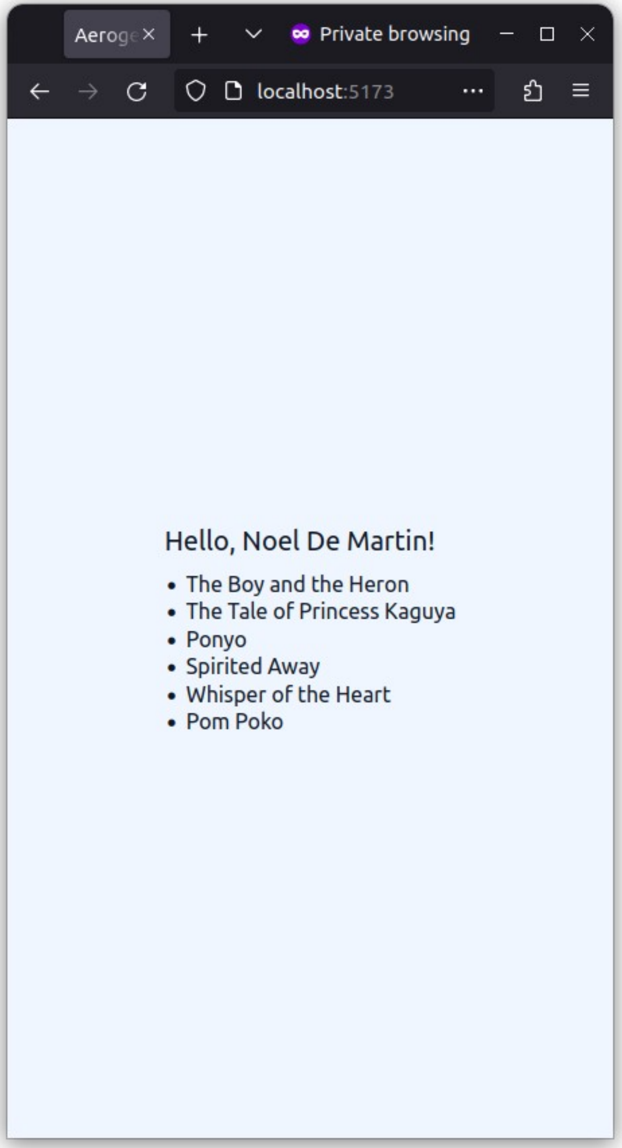




You are logged in as http://localhost:3000/noeldemartin/profile/card#me

# Services

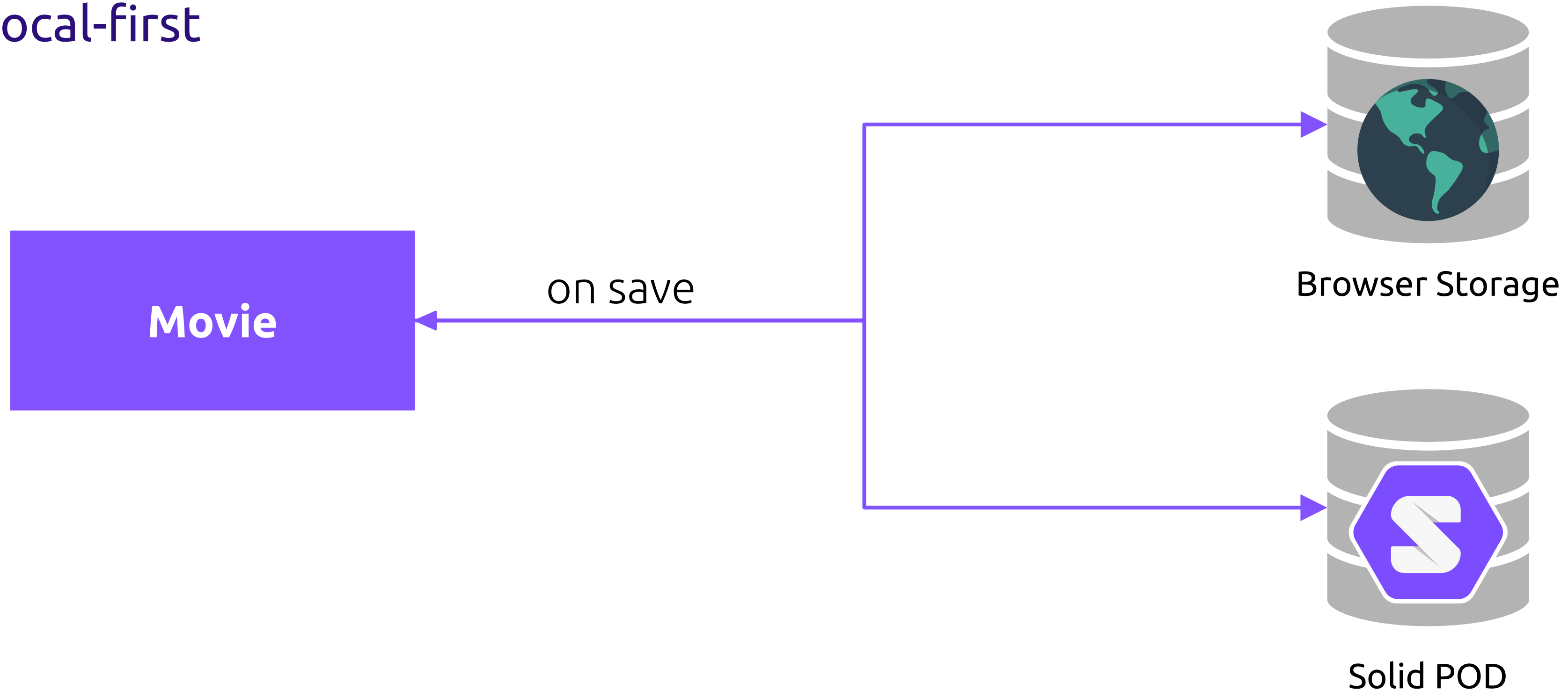
```
<template>
    <div v-if="$solid.isLoggedIn()">
        <p class="text-xl">
            Hello, {{ $solid.user.name }}!
        </p>
        <ul class="mt-2 list-disc">
            <li v-for="movie of movies" :key="movie.url" class="ml-4">
                {{ movie.title }}
            </li>
        </ul>
    </div>

    <AGSolidStatus v-else />
</template>

<script setup lang="ts">
import { useModelCollection } from '@aerogel/plugin-soukai';

import Movie from '@/models/Movie';

const movies = useModelCollection(Movie);
</script>
```



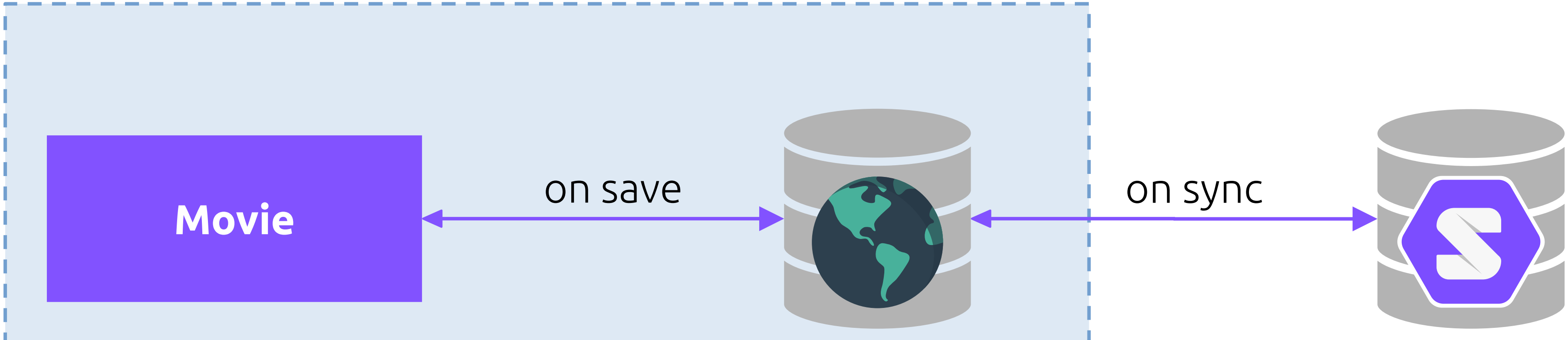
Hello, Noel De Martin!

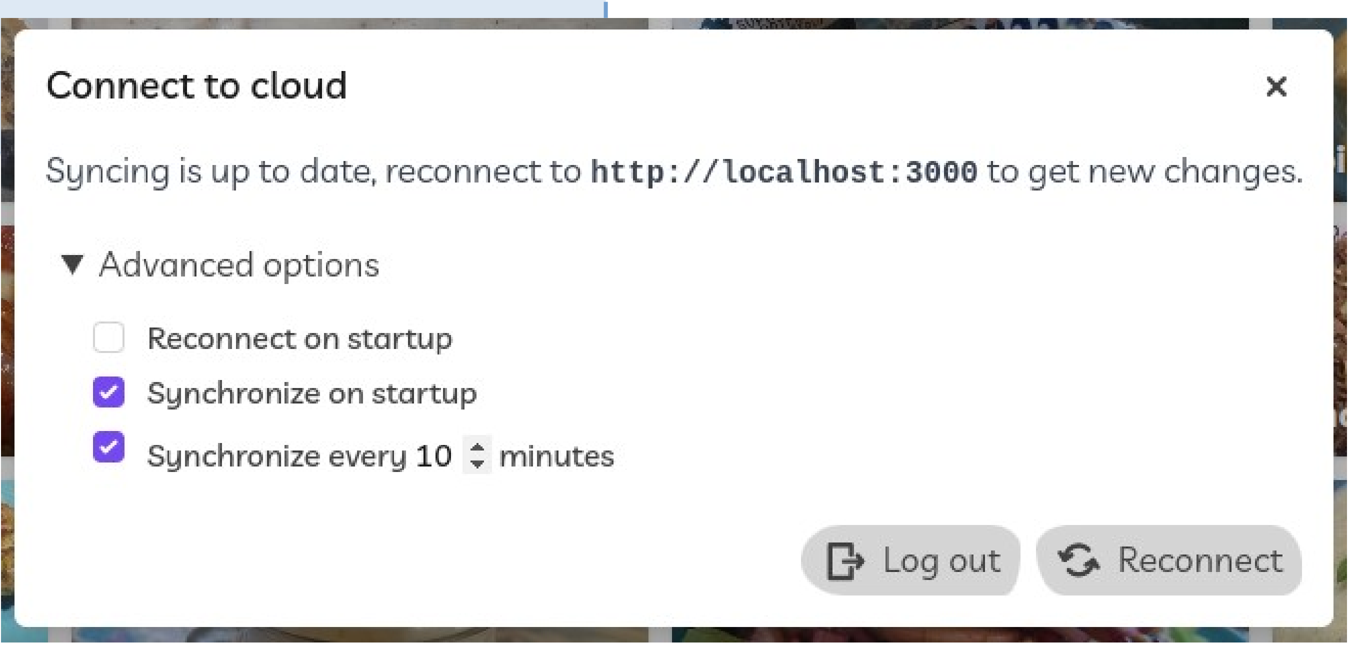
- The Boy and the Heron
- The Tale of Princess Kaguya
- Ponyo
- Spirited Away
- Whisper of the Heart
- Pom Poko

# Local-first



# Local-first





### Connect to cloud

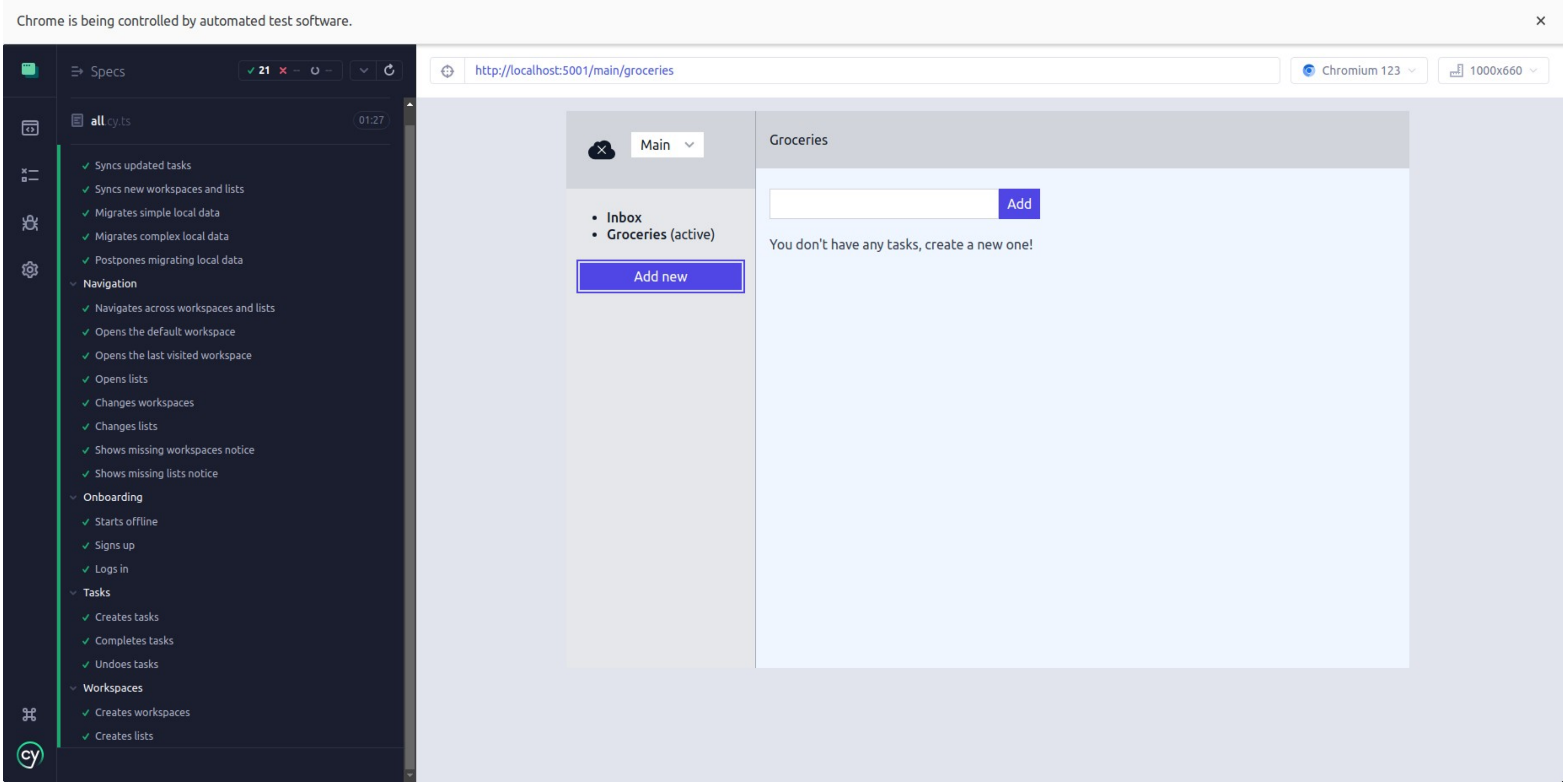
Syncing is up to date, reconnect to `http://localhost:3000` to get new changes.

▼ Advanced options

- ☐ Reconnect on startup
- ☑ Synchronize on startup
- ☑ Synchronize every 10 minutes

# Testing

Cypress + github.com/NoelDeMartin/cypress-solid



Chrome is being controlled by automated test software.

Specs ✓ 21 ✕ -- ○ --

http://localhost:5001/main/groceries

Chromium 123

1000x660

```
all.cy.ts                                   01:27

    ✓ Syncs updated tasks
    ✓ Syncs new workspaces and lists
    ✓ Migrates simple local data
    ✓ Migrates complex local data
    ✓ Postpones migrating local data
  Navigation
    ✓ Navigates across workspaces and lists
    ✓ Opens the default workspace
    ✓ Opens the last visited workspace
    ✓ Opens lists
    ✓ Changes workspaces
    ✓ Changes lists
    ✓ Shows missing workspaces notice
    ✓ Shows missing lists notice
  Onboarding
    ✓ Starts offline
    ✓ Signs up
    ✓ Logs in
  Tasks
    ✓ Creates tasks
    ✓ Completes tasks
    ✓ Undoes tasks
  Workspaces
    ✓ Creates workspaces
    ✓ Creates lists
```

Main

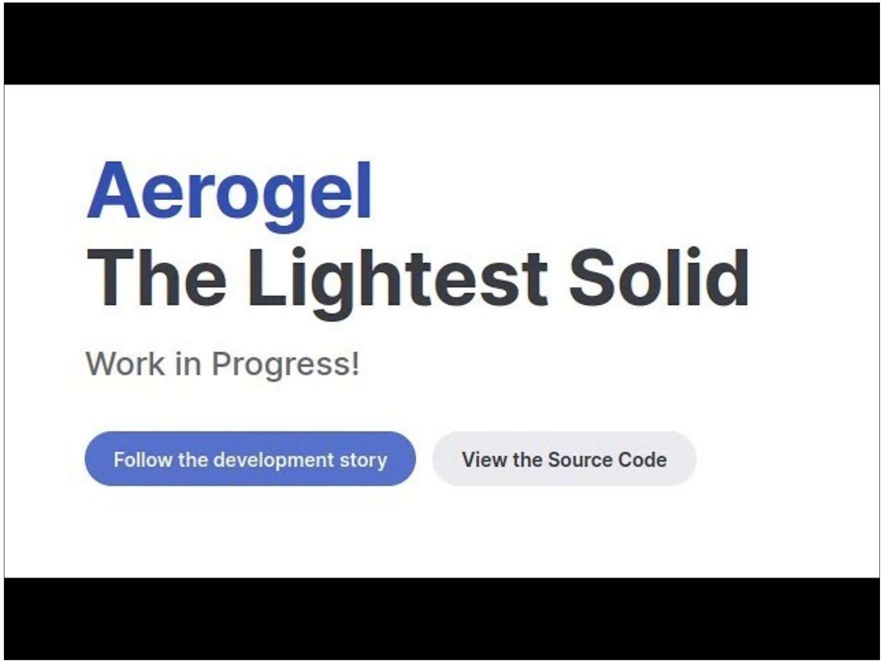
- **Inbox**
- **Groceries** (active)

Groceries

You don't have any tasks, create a new one!

# And more...

- I18n
- Error handling
- Easy publishing
- Forms (+ Validation)
- ...



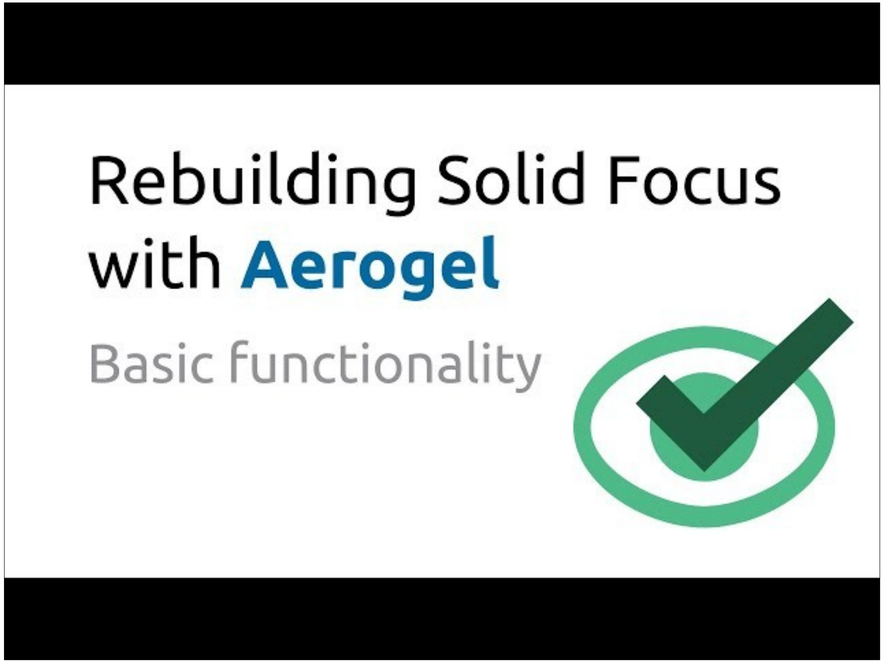
Aerogel
The Lightest Solid

Work in Progress!

Aerogel Sneak Peek (17:52min)

*Scaffolding, Models, Services, CLI, ...*

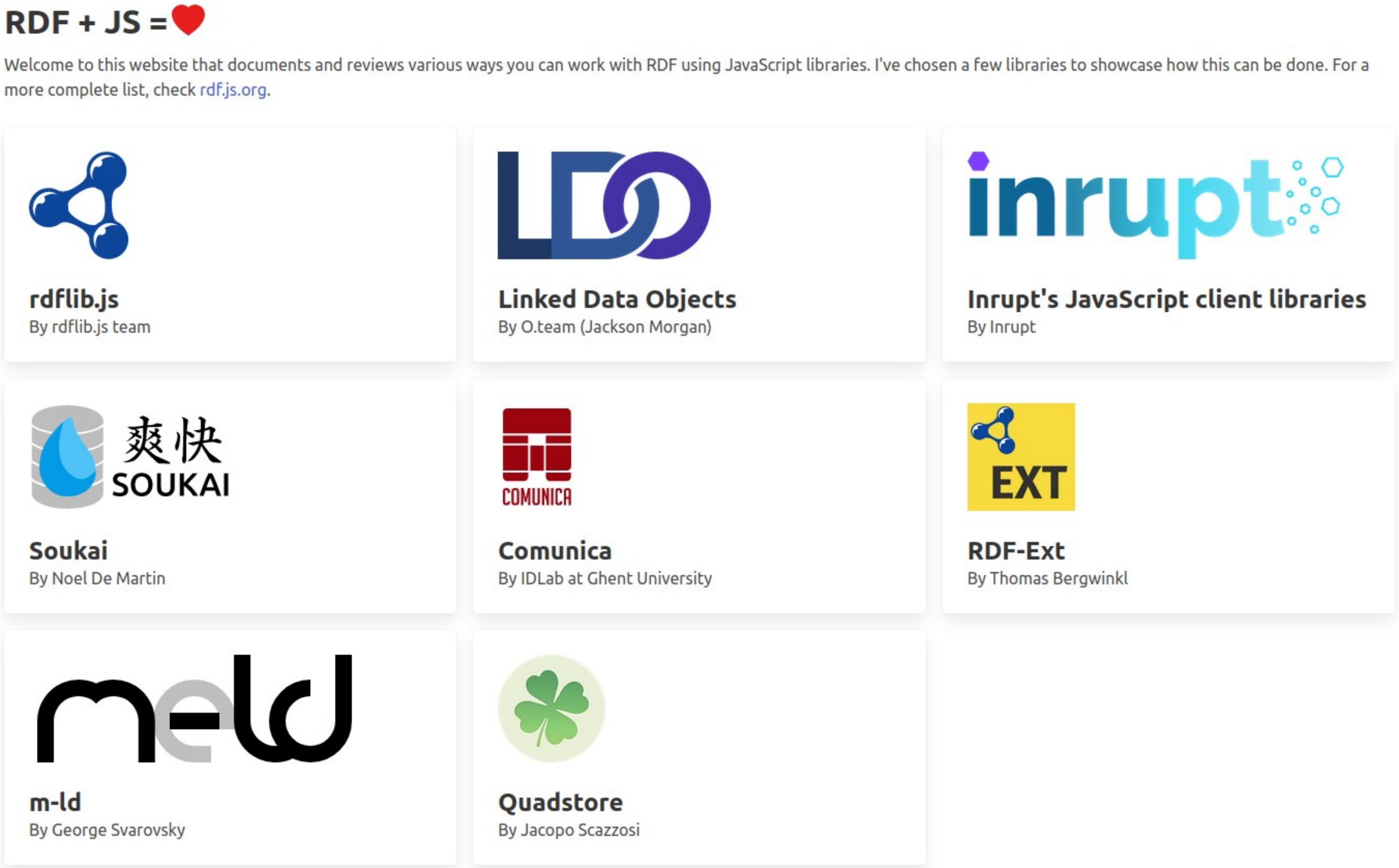
[youtu.be/JXyCH_S9efk](https://youtu.be/JXyCH_S9efk)



Rebuilding Solid Focus with Aerogel
| Basic functionality (10:39min)

*Cloud synchronization, Cypress, ...*

[youtu.be/awaaSorMYhk](https://youtu.be/awaaSorMYhk)



## RDF + JS = ❤

Welcome to this website that documents and reviews various ways you can work with RDF using JavaScript libraries. I've chosen a few libraries to showcase how this can be done. For a more complete list, check rdf.js.org.

**rdflib.js**
By rdflib.js team

**Linked Data Objects**
By O.team (Jackson Morgan)

**Inrupt's JavaScript client libraries**
By Inrupt

**Soukai**
By Noel De Martin

**Comunica**
By IDLab at Ghent University

**RDF-Ext**
By Thomas Bergwinkl

**m-ld**
By George Svarovsky

**Quadstore**
By Jacopo Scazzosi

rdfjs.dev

# Thanks!

Follow my work at [noeldemartin.com/now](noeldemartin.com/now)

## Questions?