Offline data manipulations for everyone with Plug-Ins and SQLite

04.05.2023

Philipp Hartenfeller, Senior Consultant

APEX Connect 2023, Berlin

Facts and figures.

Your partner for digital transformation.
Individual IT solutions from one single source.

Top 2023 Company
kununu

CHIP Great-JOBS.org
TOP Arbeitgeber für IT-Jobs 2023
Im Test: 2.675 Arbeitgeber
CHIP 02/2023
MT GmbH
Unabhängige Tests seit 1978
Lizenz CCgl0096

Foundation 1994

Headquarter
Ratingen
Branches
Frankfurt am Main,
Köln, München, Hamburg

> 360 Employees

approx. 48 Mio. €
Turnover in 2021

> 125 Customers
Cross-sector

Manufacturer-neutral

Certified partner
leading technology
manufacturer

Training company,
Partner in dual studies

$ whoami

Philipp Hartenfeller

- Düsseldorf, Germany
- Master IT-Management
- Since 2016 @ MT ~~AG~~
- Senior Consultant – Oracle APEX
- Mostly doing WebDev, DBs and APEX Testing (https://lct.software)

Blog: https://hartenfeller.dev/blog/

@phartenfeller @phartenfeller@mastodon.social Oracle ACE Associate

What this talk is about

- Offline APEX is no new topic, there have been a few methods already

- I am interested in **HOW** it should be usable in APEX

What this talk is about

How should it be accessible ?

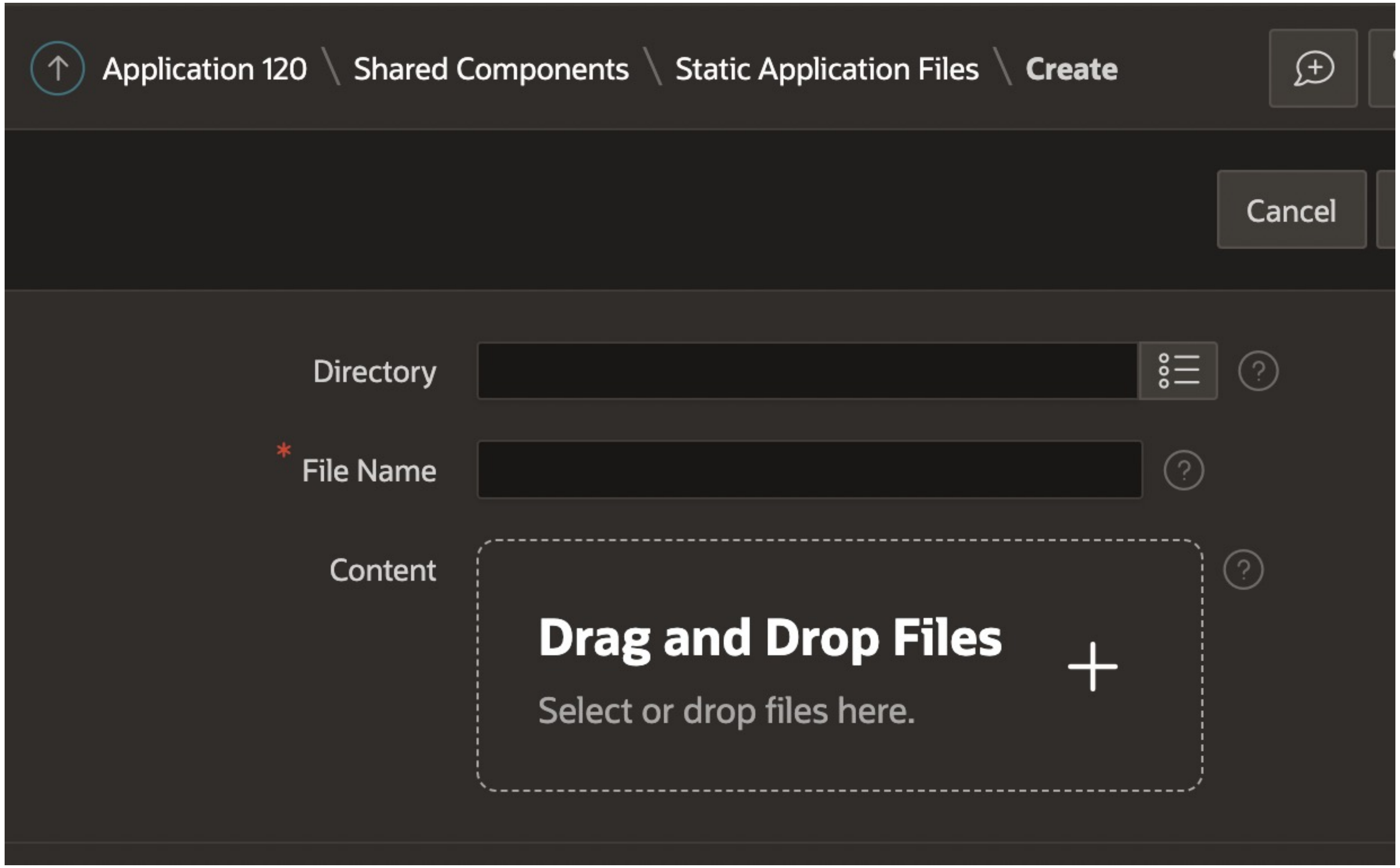
- Low Code Interface -> No JavaScript Code wrangling
  - **APEX Plug-Ins**
- Easy to configure
  - Enter your Query and let’s go!

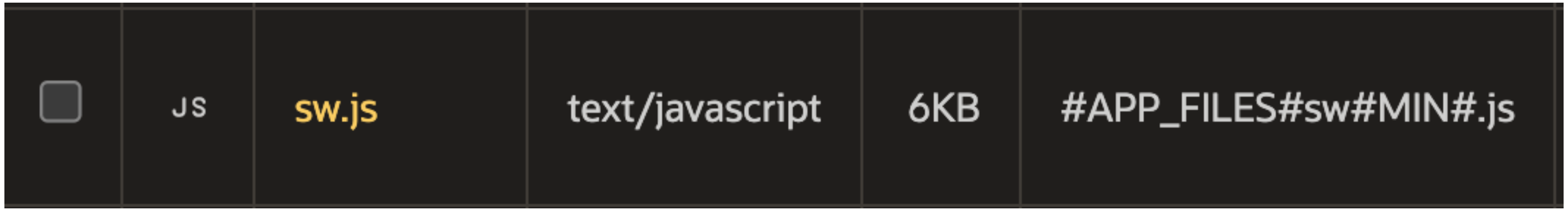
What this talk is about

How should it be accessible ?	How should it work ?
• Low Code Interface -> No JavaScript Code wrangling • **APEX Plug-Ins** • Easy to configure • Enter your Query and let’s go!	• Fast • Handle lots of data • Concept to merge changes back to DB • Somewhat future proof

Proof of Concept: Not production ready!

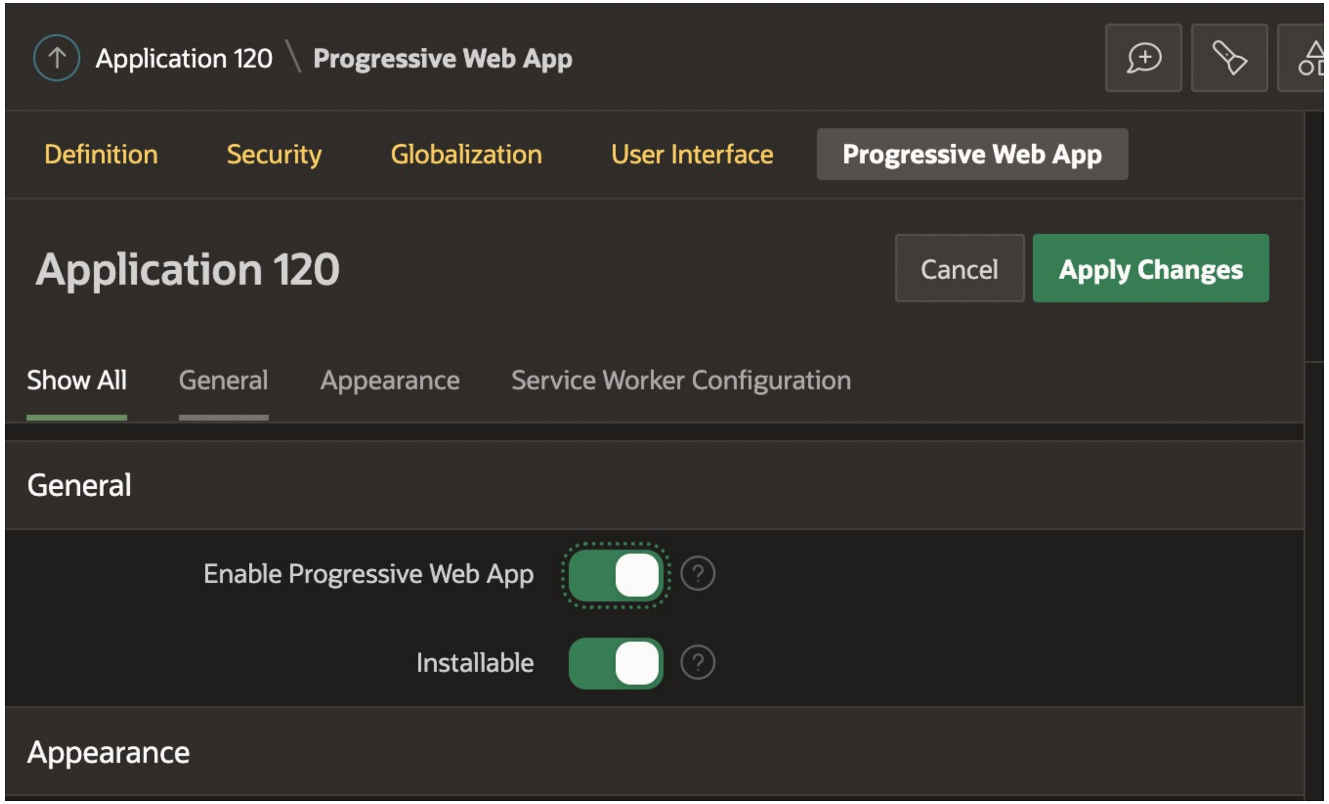
Upload sw.js


Application 120 \ Shared Components \ Static Application Files \ Create
Cancel
Directory
* File Name
Content
Drag and Drop Files
Select or drop files here.



□	JS	sw.js	text/javascript	6KB	#APP_FILES#sw#MIN#.js

Enable PWA features



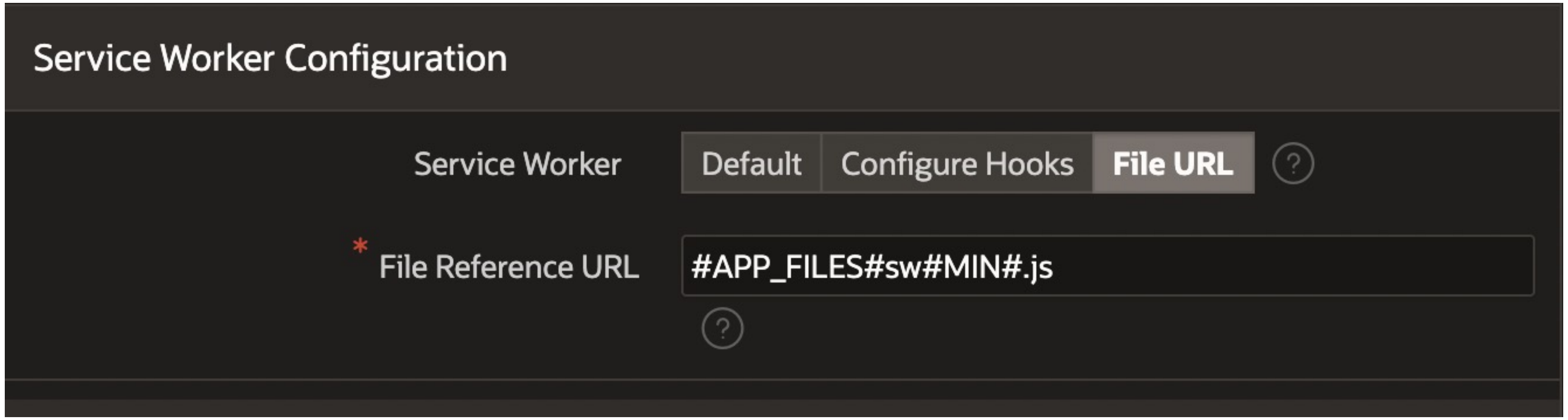
Application 120

General

Enable Progressive Web App ●

Installable ●

Appearance

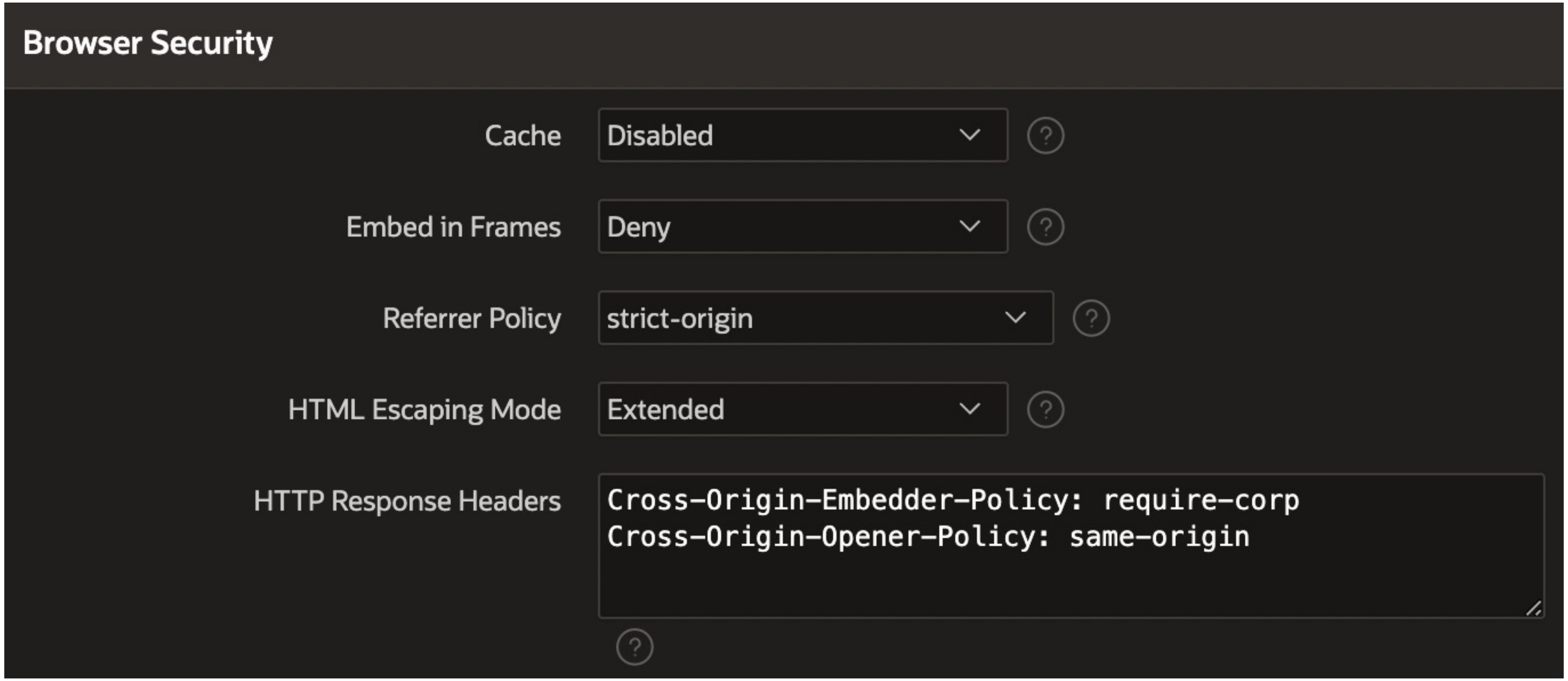


Service Worker Configuration

Service Worker: Default | Configure Hooks | [✓ File URL]

* File Reference URL: #APP_FILES#sw#MIN#.js

Set required Headers



Browser Security

Cache: Disabled

Embed in Frames: Deny

Referrer Policy: strict-origin

HTML Escaping Mode: Extended

HTTP Response Headers:

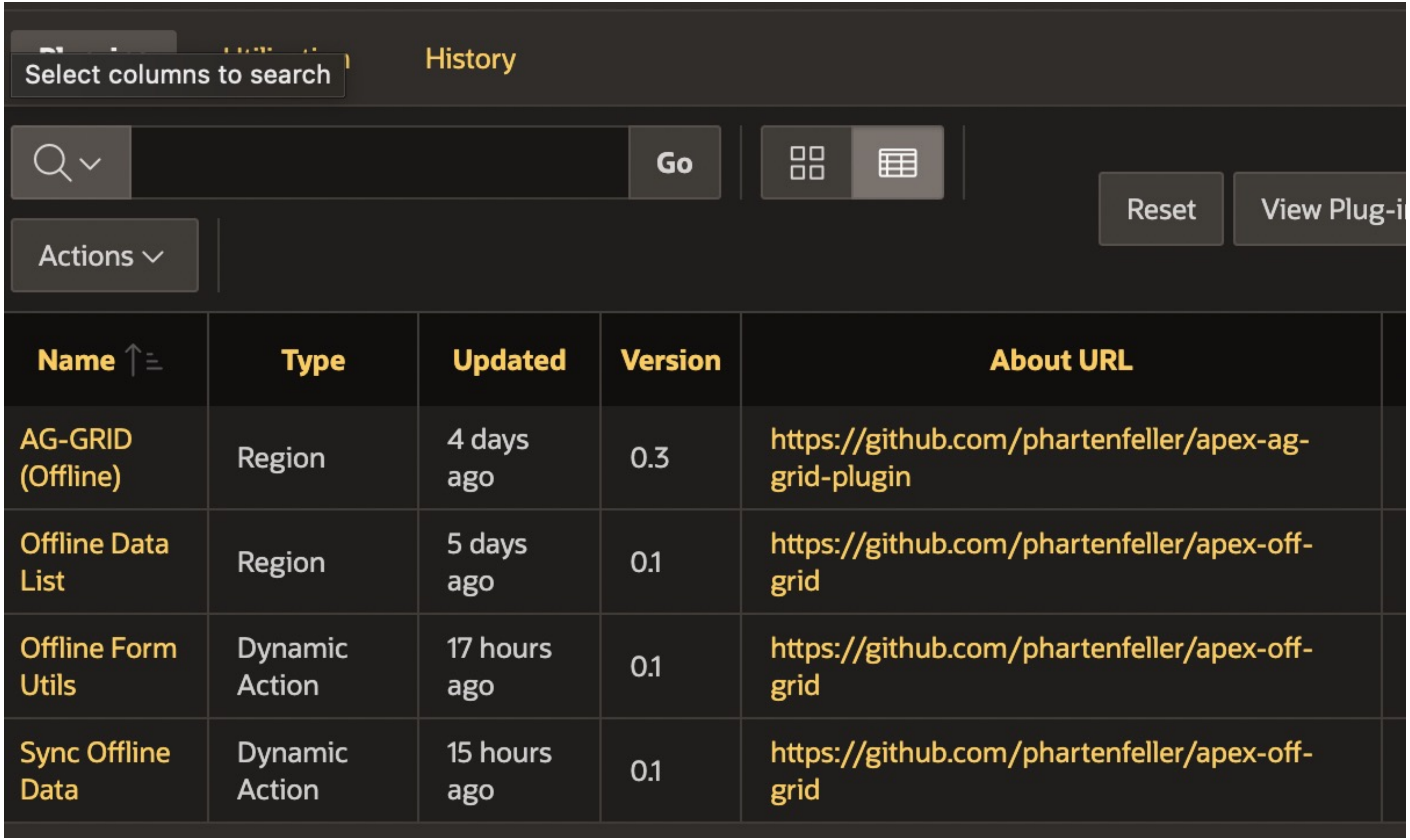
```
Cross-Origin-Embedder-Policy: require-corp
Cross-Origin-Opener-Policy: same-origin
```

Currently a bug in APEX (does not apply to Plug-In files)

A bug is filed

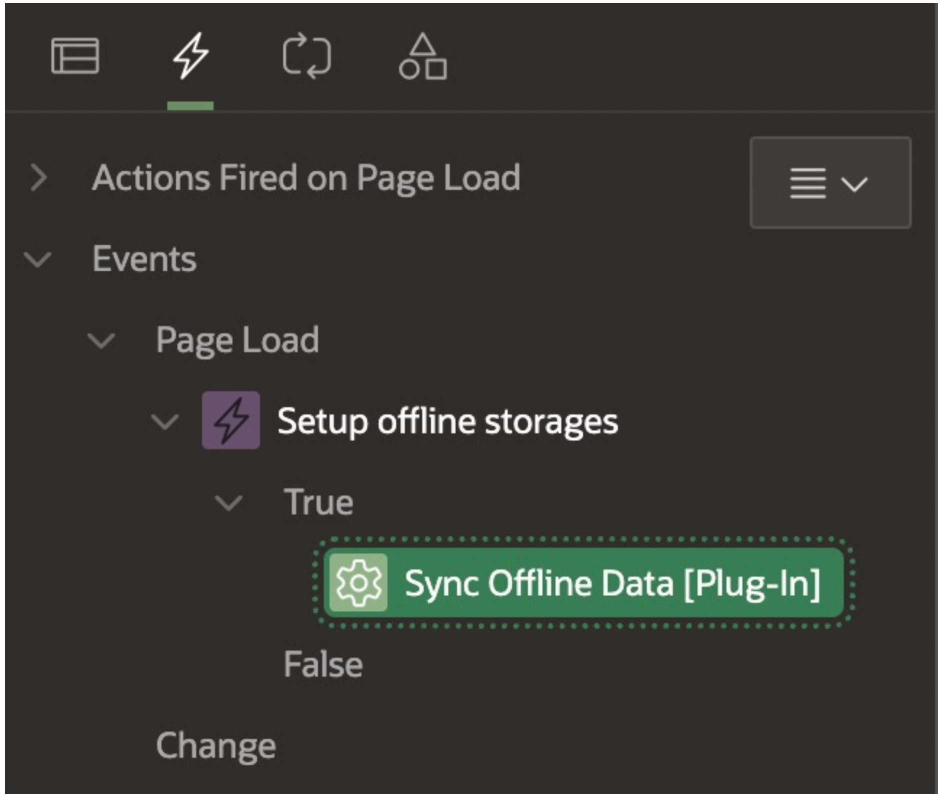
For now, you need to set it on Proxy / Webserver

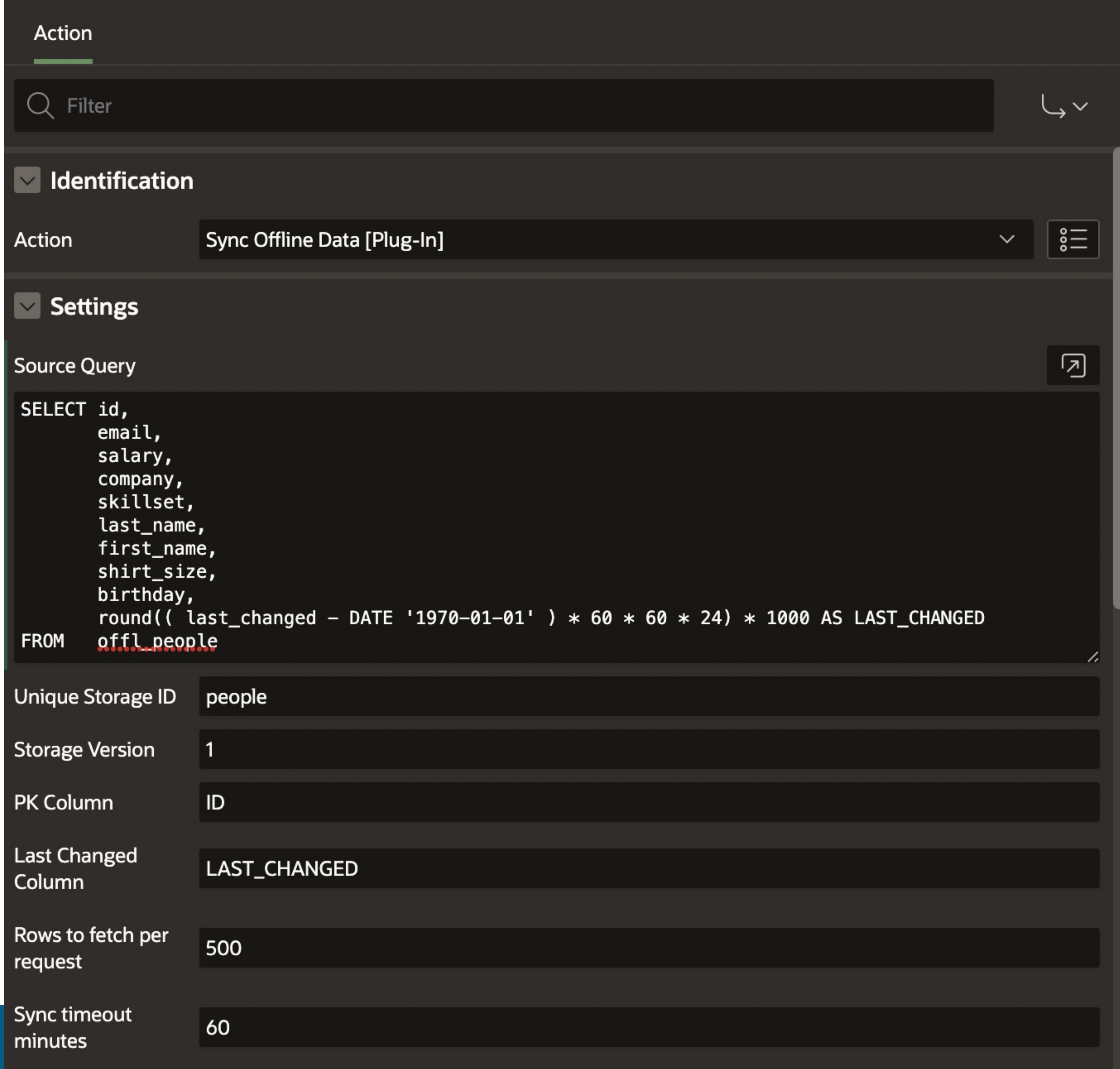
Import Plug-Ins



Name	Type	Updated	Version	About URL
AG-GRID (Offline)	Region	4 days ago	0.3	https://github.com/phartenfeller/apex-ag-grid-plugin
Offline Data List	Region	5 days ago	0.1	https://github.com/phartenfeller/apex-off-grid
Offline Form Utils	Dynamic Action	17 hours ago	0.1	https://github.com/phartenfeller/apex-off-grid
Sync Offline Data	Dynamic Action	15 hours ago	0.1	https://github.com/phartenfeller/apex-off-grid

Setup Sync-Plug-In





Identification

Action: Sync Offline Data [Plug-In]

Settings

Source Query

```
SELECT id,
       email,
       salary,
       company,
       skillset,
       last_name,
       first_name,
       shirt_size,
       birthday,
       round(( last_changed - DATE '1970-01-01' ) * 60 * 60 * 24) * 1000 AS LAST_CHANGED
FROM   offl_people
```

Unique Storage ID: people

Storage Version: 1

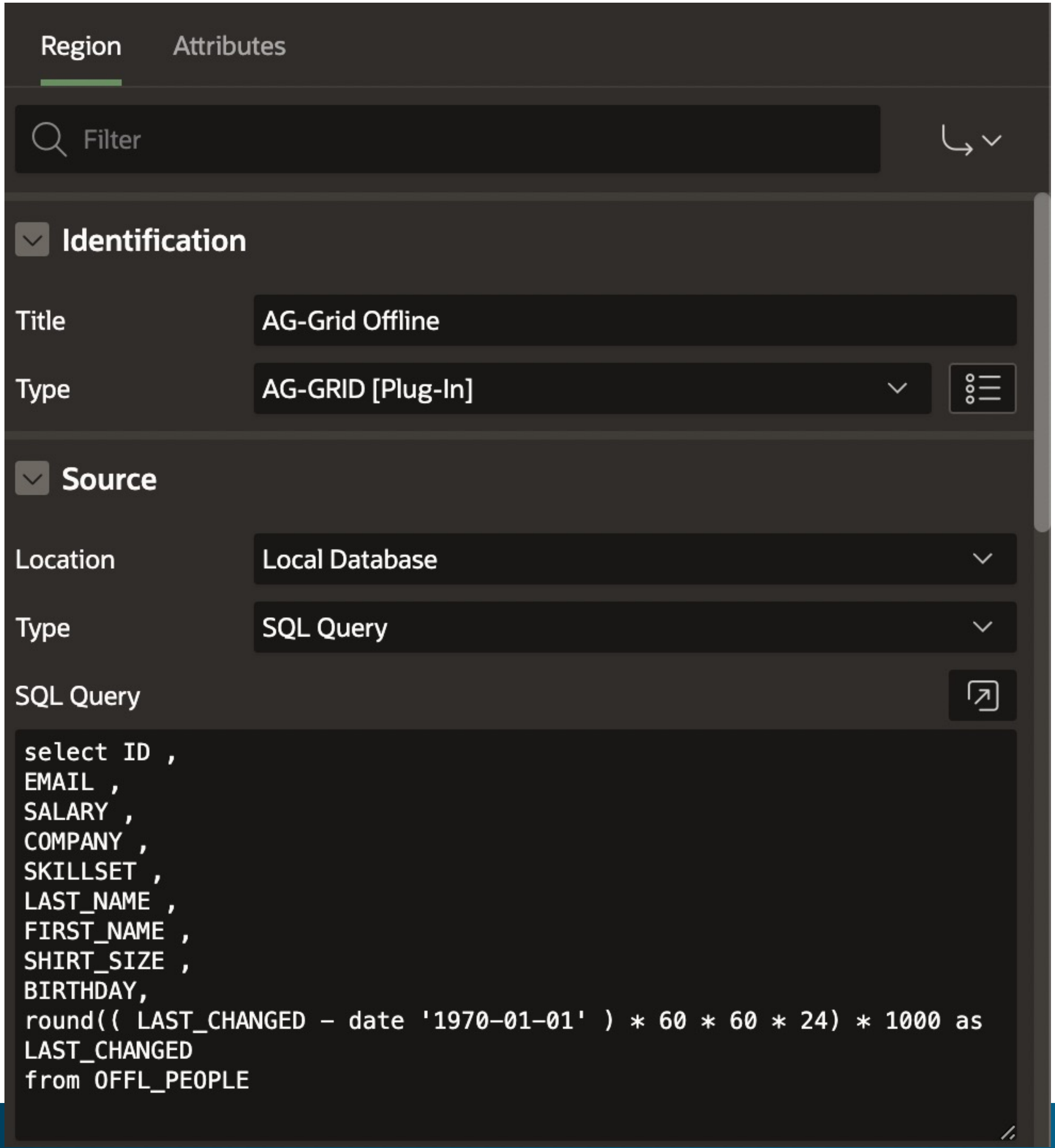
PK Column: ID

Last Changed Column: LAST_CHANGED

Rows to fetch per request: 500

Sync timeout minutes: 60

Add consumer Plug-In



Identification

Title: AG-Grid Offline

Type: AG-GRID [Plug-In]

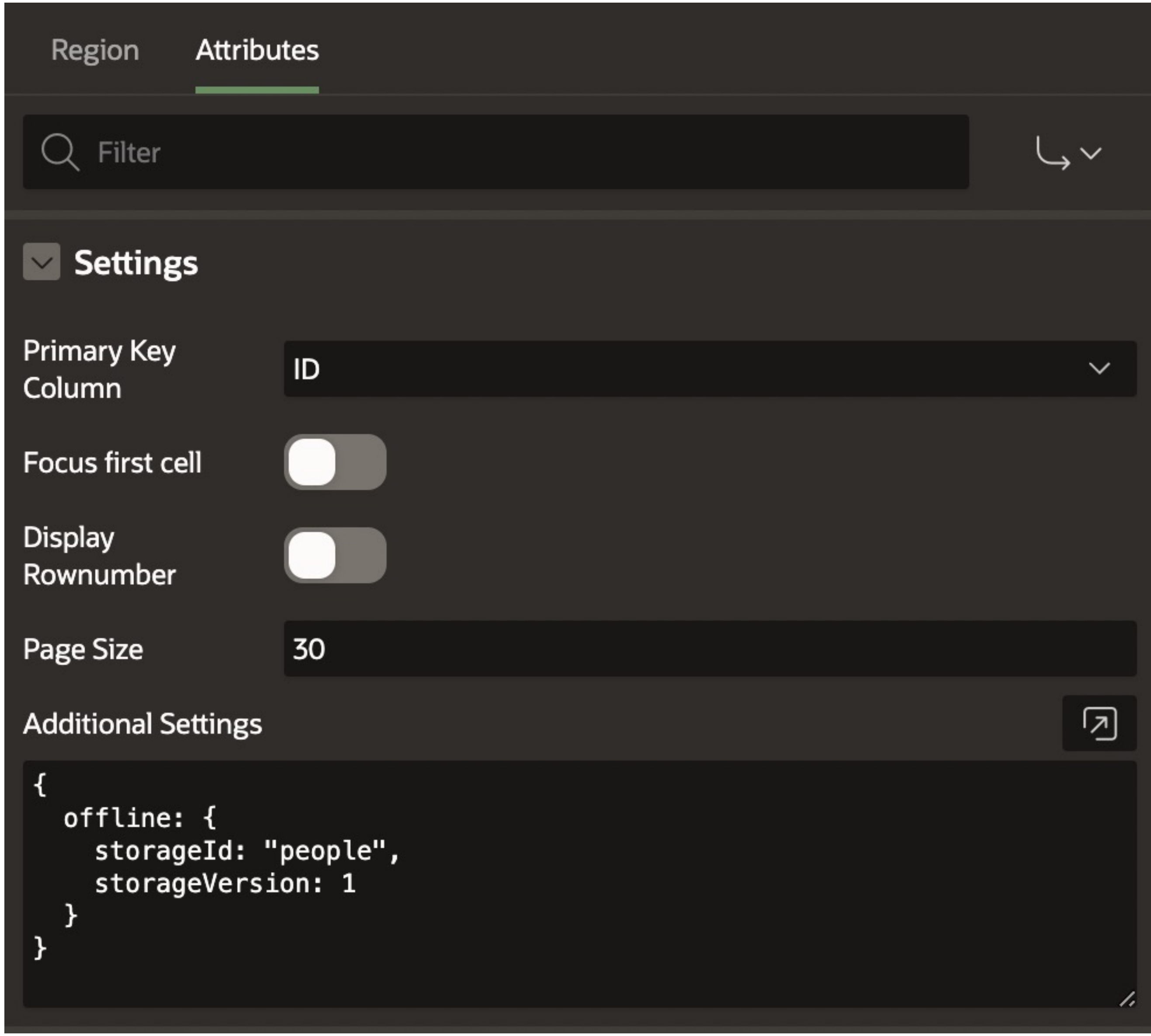
Source

Location: Local Database

Type: SQL Query

SQL Query

```
select ID ,
EMAIL ,
SALARY ,
COMPANY ,
SKILLSET ,
LAST_NAME ,
FIRST_NAME ,
SHIRT_SIZE ,
BIRTHDAY,
round(( LAST_CHANGED - date '1970-01-01' ) * 60 * 60 * 24) * 1000 as
LAST_CHANGED
from OFFL_PEOPLE
```



Settings

Primary Key Column: ID

Focus first cell: ○

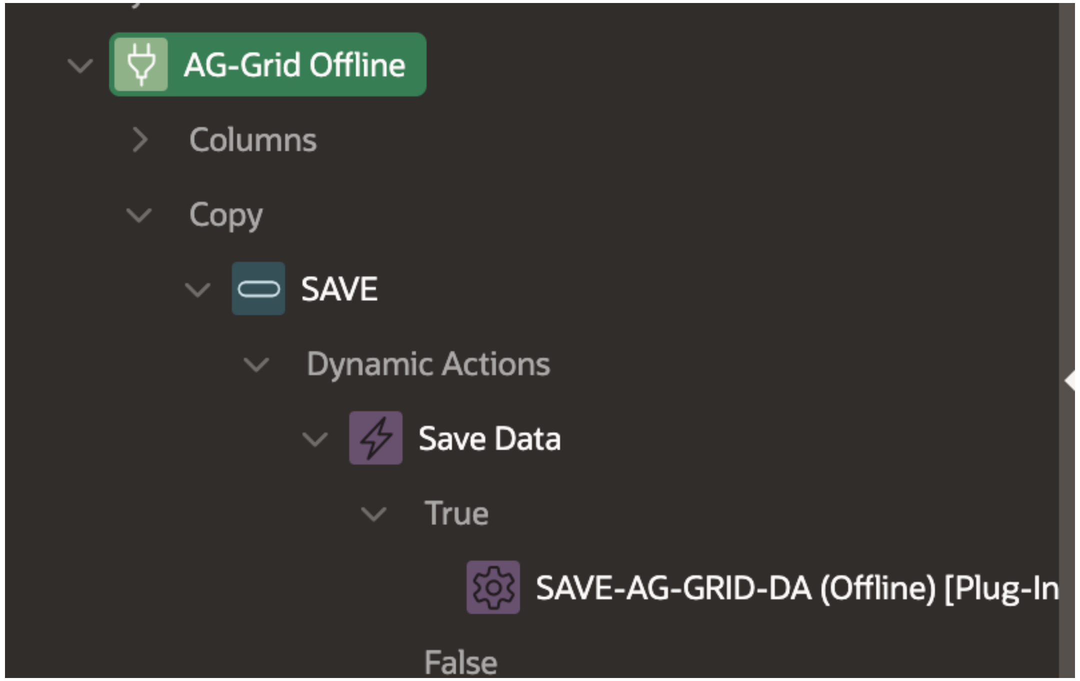
Display Rownumber: ○

Page Size: 30

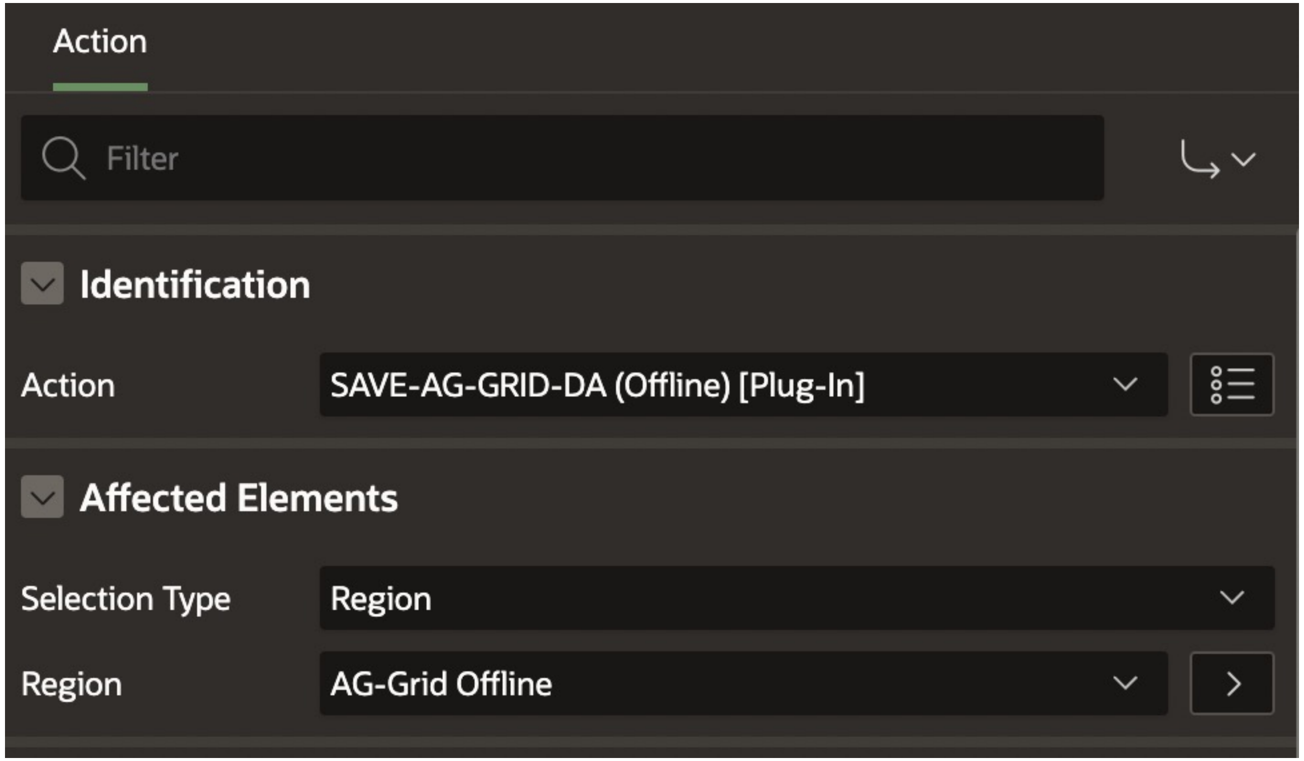
Additional Settings

```
{
  offline: {
    storageId: "people",
    storageVersion: 1
  }
}
```

Add save Action



- AG-Grid Offline
  - Columns
  - Copy
    - SAVE
      - Dynamic Actions
        - Save Data
          - True
            - SAVE-AG-GRID-DA (Offline) [Plug-In
          - False



Action

Filter

Identification

Action: SAVE-AG-GRID-DA (Offline) [Plug-In]

Affected Elements

Selection Type: Region

Region: AG-Grid Offline

Additionally

You need to install packages and a table

You need to write Code to merge your data back to the source (more on that later)

Demo

Photo by Jakob Owens from Unsplash

Data Synchronization

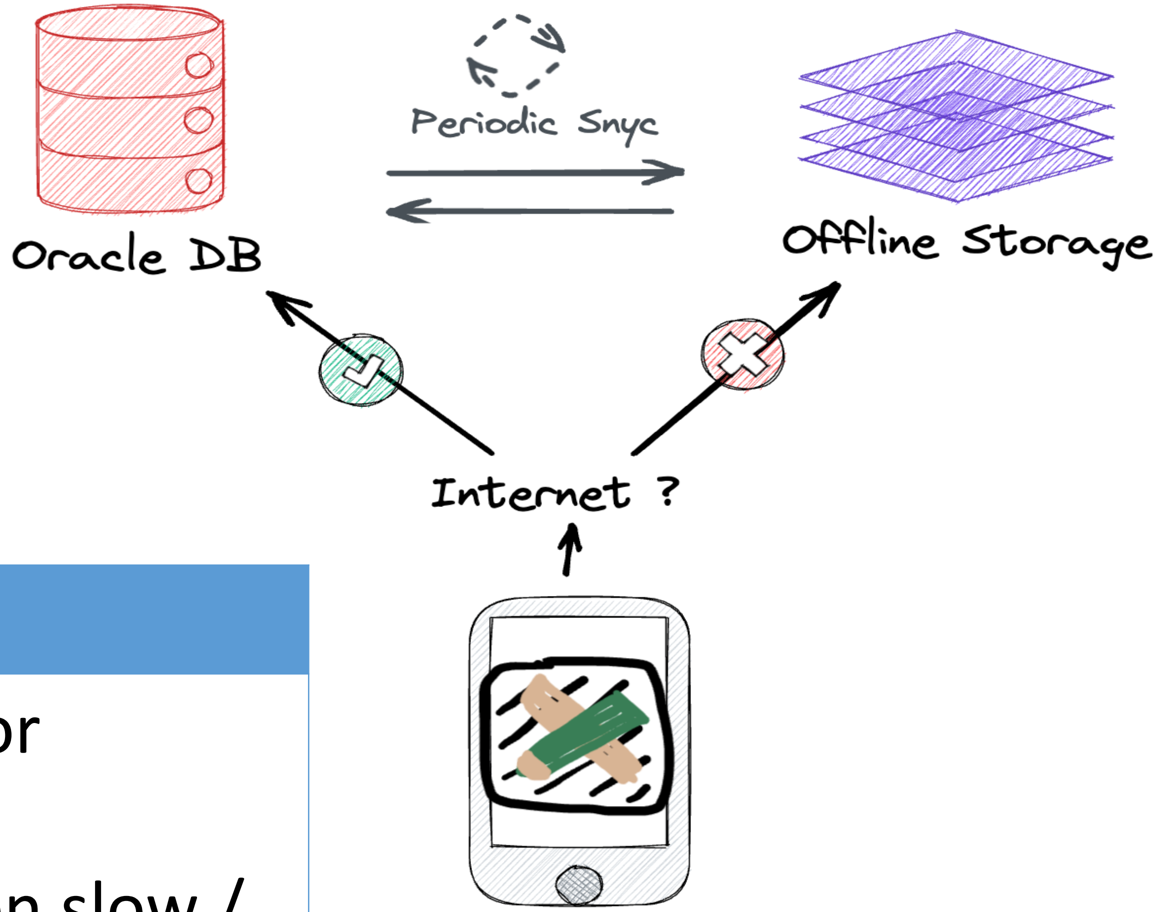
Why this topic is important

- Allowing users to edit data without a connection to the DB has effects of decentralization
- You still want a single source of truth

--> This results in new challenges for data consistency

Data Access Strategies - Online First

- First try to get data from the DB
- Only when no connection use offline storage

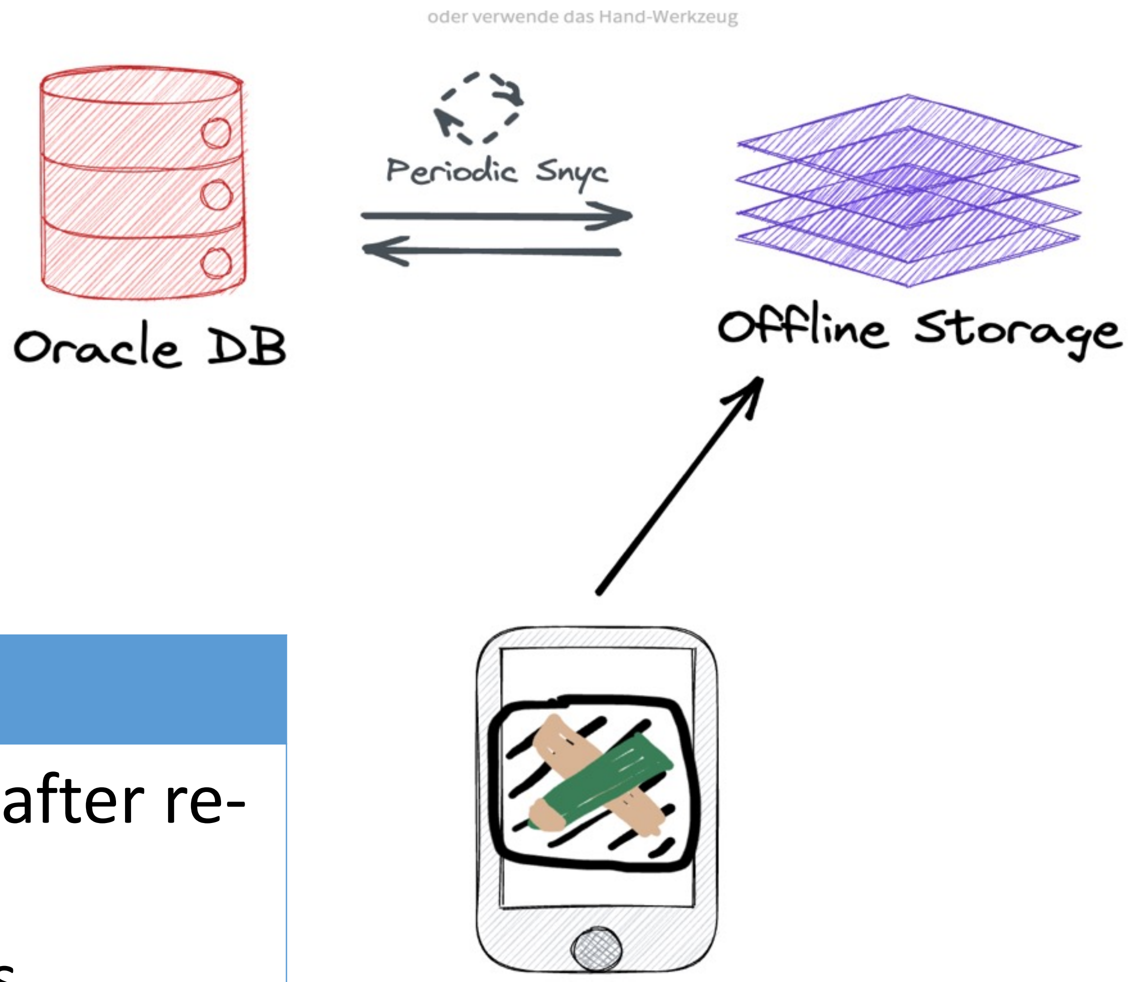


Pros	Cons
• “Normal APEX Mode“ • Less conflicts • Live data	• Much logic for consumers • What to do on slow / unreliable connections

Data Access Strategies - Offline First

Design Decision

- Always use Offline Storage for your components



Pros	Cons
• Reliable Experience • Fast • Less logic for consumers	• Only updates after re-sync • More conflicts

I chose this approach because of this

💡 Build different experiences for different use cases

- Make sure only people who need offline access use it
- Field technician gets offline App / Page
- Office worker gets online App / Page

Photo by MIOPS Trigger from Unsplash

Sync Timeline - 1

1. Cache current HTML page

2. Synchronize Client Changes with DB

   - Send client updates to central DB
   - Delete corresponding rows on the client
   - It is now in the control of the DB what will happen with these rows

Sync Timeline - 2

3. Synchronize DB Changes with Client

- DB sends PK + Timestamp for each row:
  - PK known and Timestamp matches: ✅
  - PK unknown or Timestamp mismatch: ⚠️ request full row -> insert / update
- Delete rows not present in central DB anymore:
  - Store every PK received from server in a temp table
  - After receiving all rows, delete local rows where PK not in temp table

Change Synchronization - Potential outcomes

a: Single Client Row Update (✅)

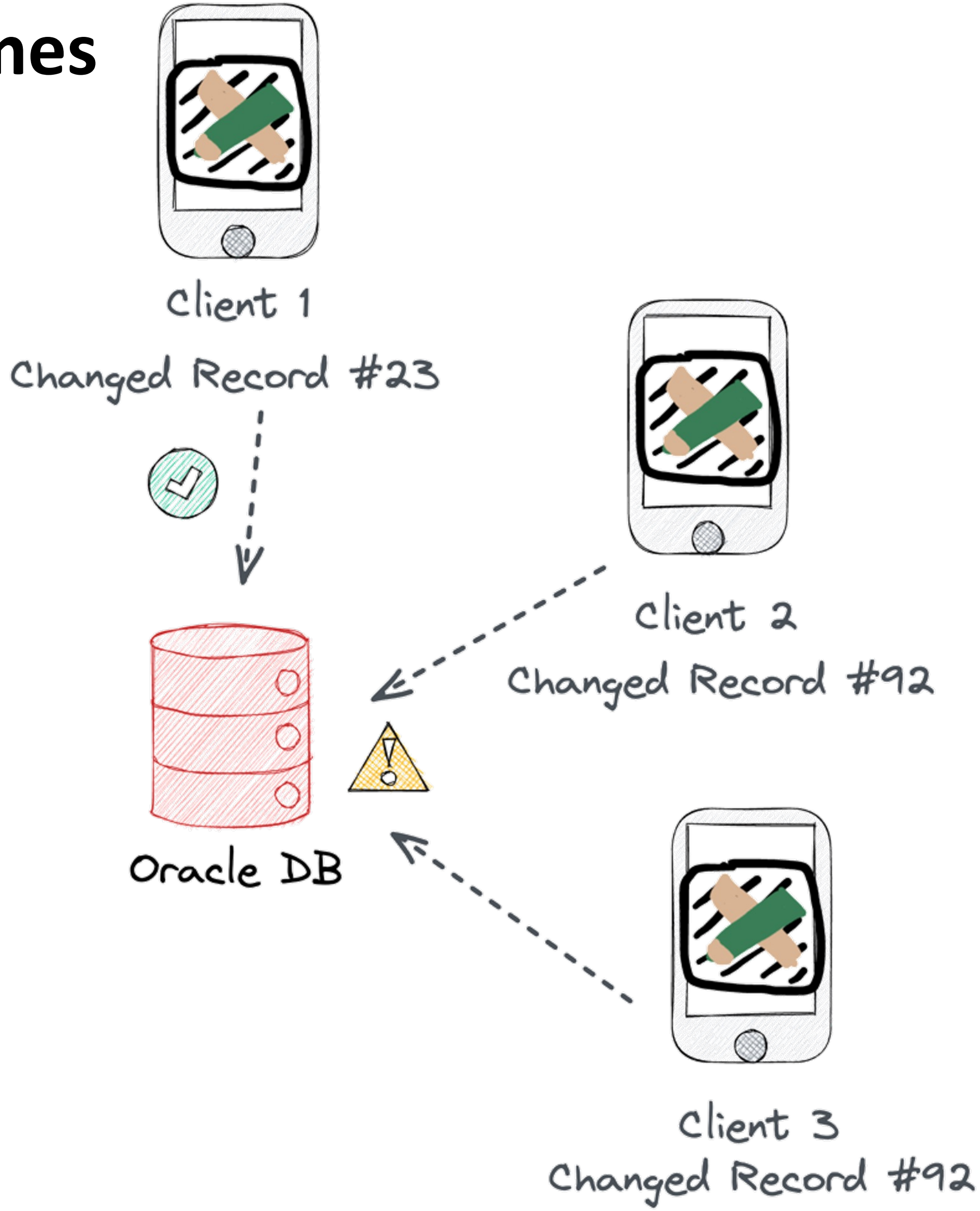
- Only one client modifies the row, straightforward synchronization

b: Sequential Client Row Updates (✅)

- Client 2 updates row after receiving Client 1's changes, preserving data integrity

c: Concurrent Client Row Updates (⚠️)

- Two clients modify the same row version simultaneously, potential data conflict



Conflict Resolution #1 – Last Write Wins

Design Decision

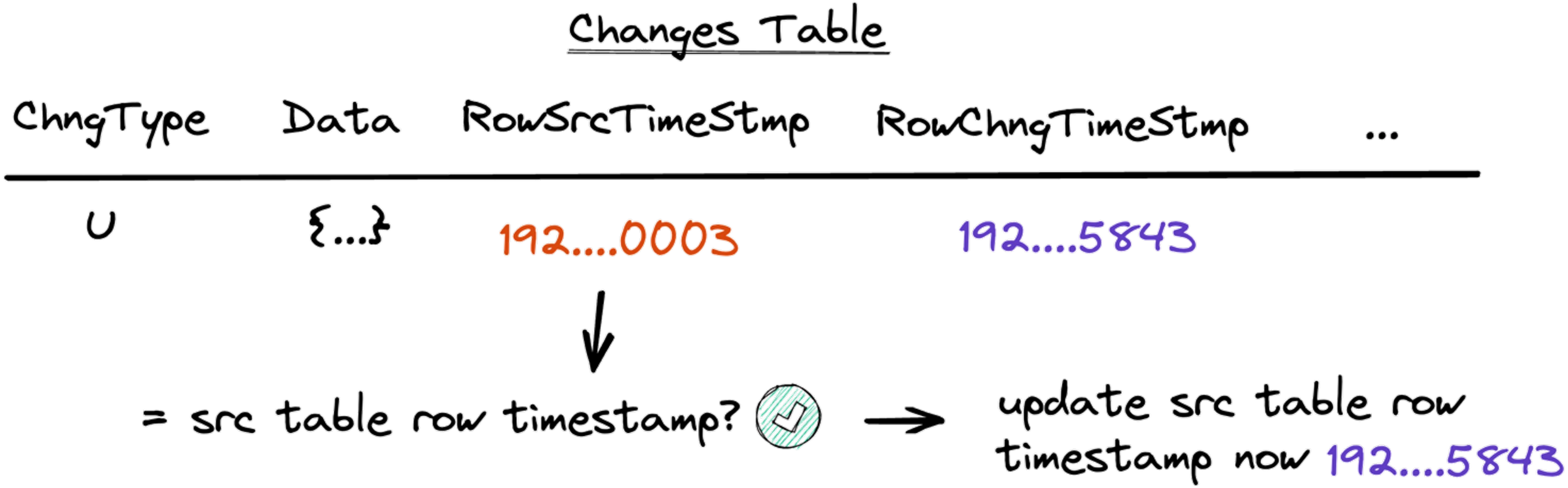
Nothing to consider:
Just write changes

Photo by [Matthias Speicher]() from [Unsplash]()

Photo by [Ryan Moreno]() from [Unsplash]()

Design Decision

Conflict Resolution #2 – Merge Conflict Detection



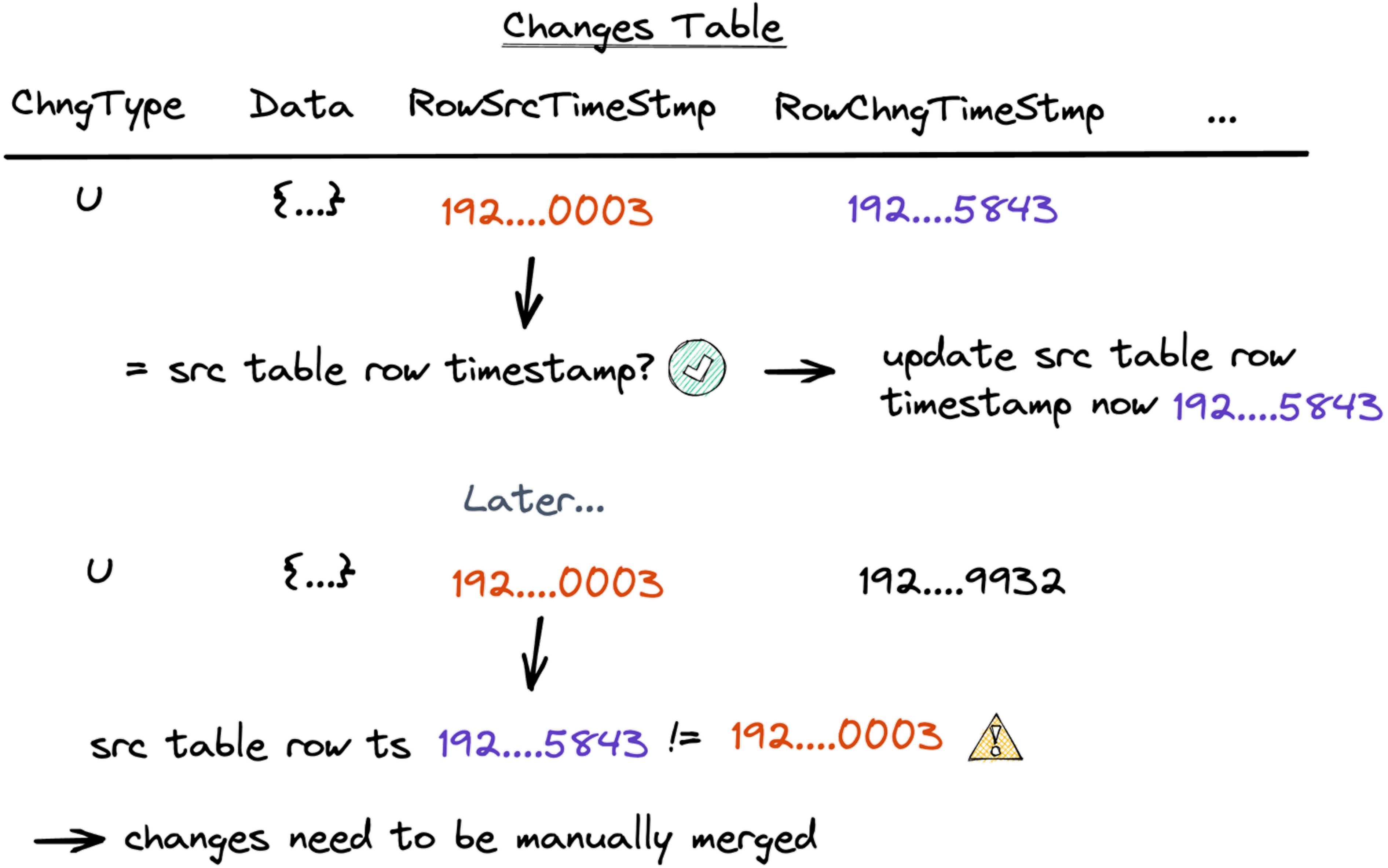
Changes Table

ChngType	Data	RowSrcTimeStmp	RowChngTimeStmp	...
U	{...}	192....0003	192....5843	

= src table row timestamp? ✓ → update src table row timestamp now 192....5843

Design Decision

Conflict Resolution #2 – Merge Conflict Detection



Changes Table

ChngType	Data	RowSrcTimeStmp	RowChngTimeStmp	...
U	{...}	192....0003	192....5843	

↓

= src table row timestamp? ✓ → update src table row timestamp now 192....5843

Later...

ChngType	Data	RowSrcTimeStmp	RowChngTimeStmp	...
U	{...}	192....0003	192....9932	

↓

src table row ts 192....5843 != 192....0003 ⚠

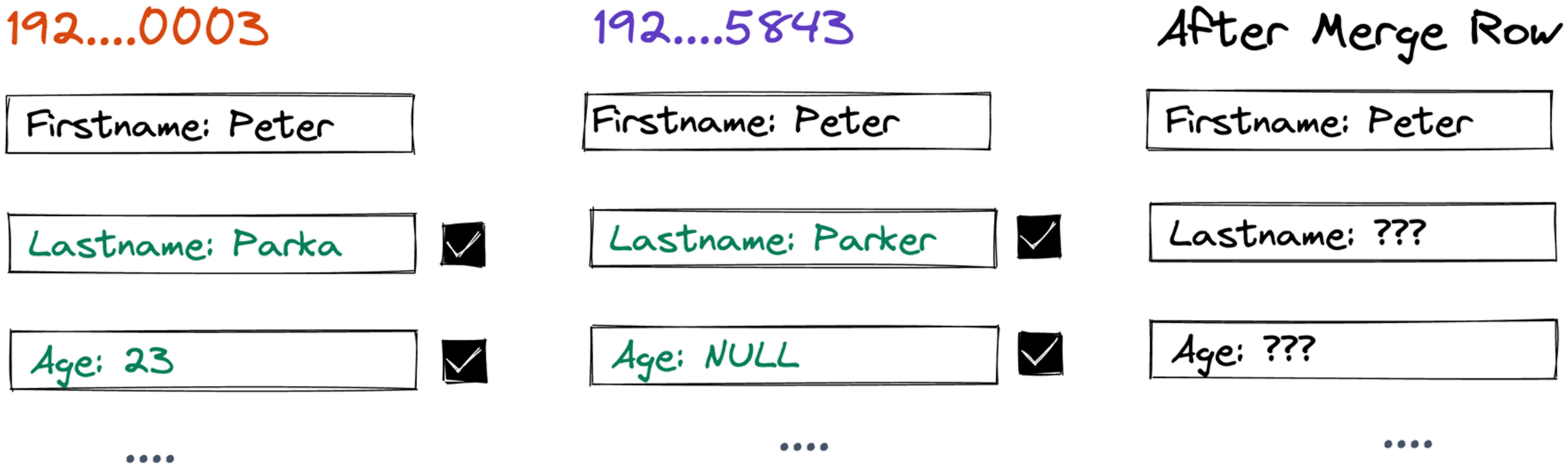
→ changes need to be manually merged

How changes are processed

- Package gets called where you can implement your own logic
  - Resolve the conflicts ← Up to you how to do it
  - Call your existing Table- / Transaction-APIs

How to do a merge

- On conflict: Update failed col = 1
- User must do merge manually
    - Show current row next to changed row (like Git 3-way-merge)



192….0003		192….5843		After Merge Row
Firstname: Peter		Firstname: Peter		Firstname: Peter
Lastname: Parka	☑	Lastname: Parker	☑	Lastname: ???
Age: 23	☑	Age: NULL	☑	Age: ???
….		….		….

Let us create chaos

https://apex.phartenfeller.de/ords/r/plugins/conn23

Currently bug with Safari!

No iOS.

Technical Implementation

Where to store data

Browser Storage Options:

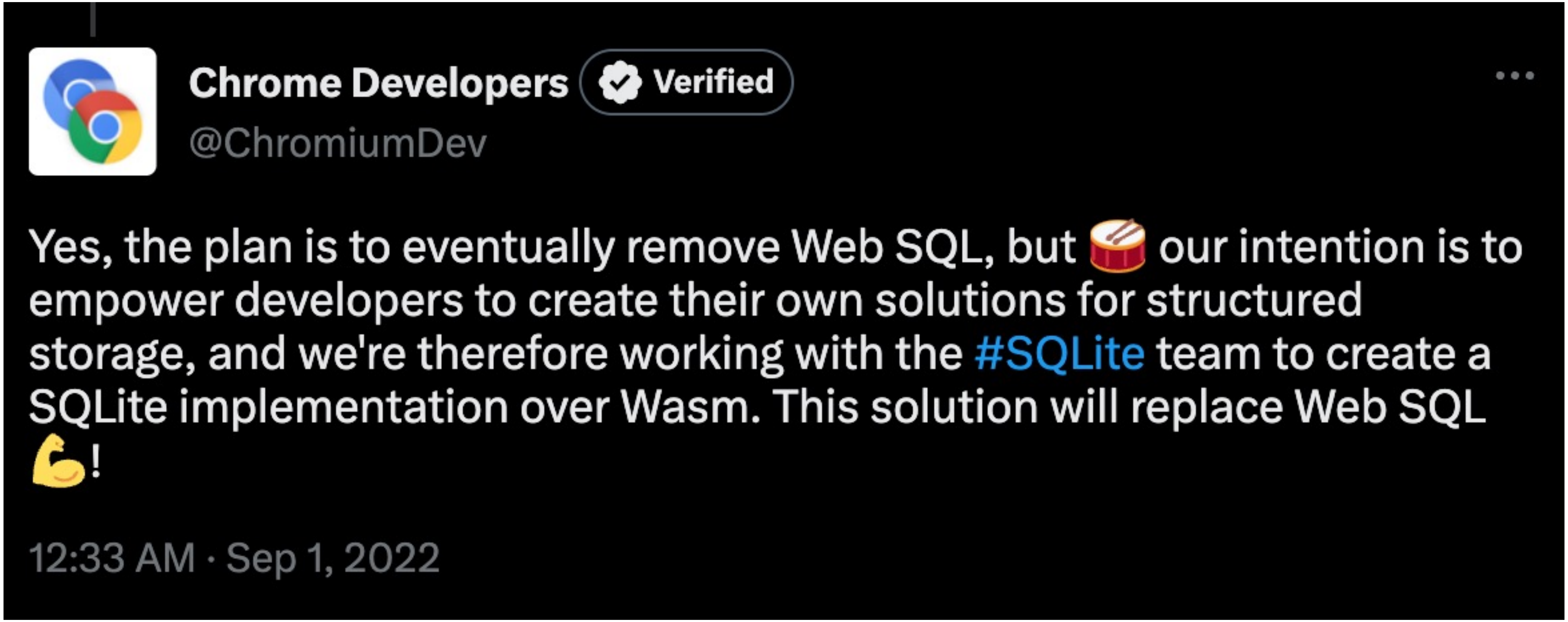
- LocalStorage
  - Key-Value object-store, slow, designed for small amounts
- WebSQL
  - Deprecated
- IndexedDB
  - For large data, transactions, Key-Value object-store

The thing with IndexedDB:

- Complex API
- No support for complex queries
- History of nasty bugs (mostly Safari)
- Not ACID compliant

The pain and anguish of using IndexedDB: problems, bugs and oddities

Not the DB we deserve, but the hero we need



Chrome Developers Verified

@ChromiumDev

Yes, the plan is to eventually remove Web SQL, but 🥁 our intention is to empower developers to create their own solutions for structured storage, and we're therefore working with the #SQLite team to create a SQLite implementation over Wasm. This solution will replace Web SQL 💪!

12:33 AM · Sep 1, 2022

Source

Wait a minute… What is SQLite?

- Most used DB, top 5 most deployed software ever (estimated one trillion active DBs)
- Full-featured SQL
- Serverless (not the cloud thing)
- Transactional
- ACID
- Public Domain

- Really fast
- Stores data in a single file
- Recommended Storage Format by the US Library of Congress

More on SQLite

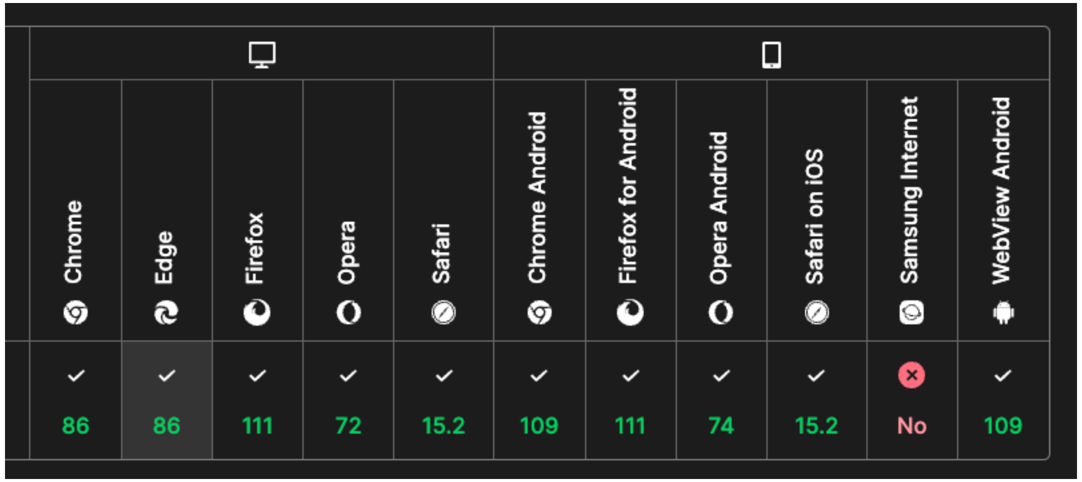
How does it run in the browser

WEBASSEMBLY

Compile existing software to hardware-near browser understood layer

Origin Private File System (OPFS)

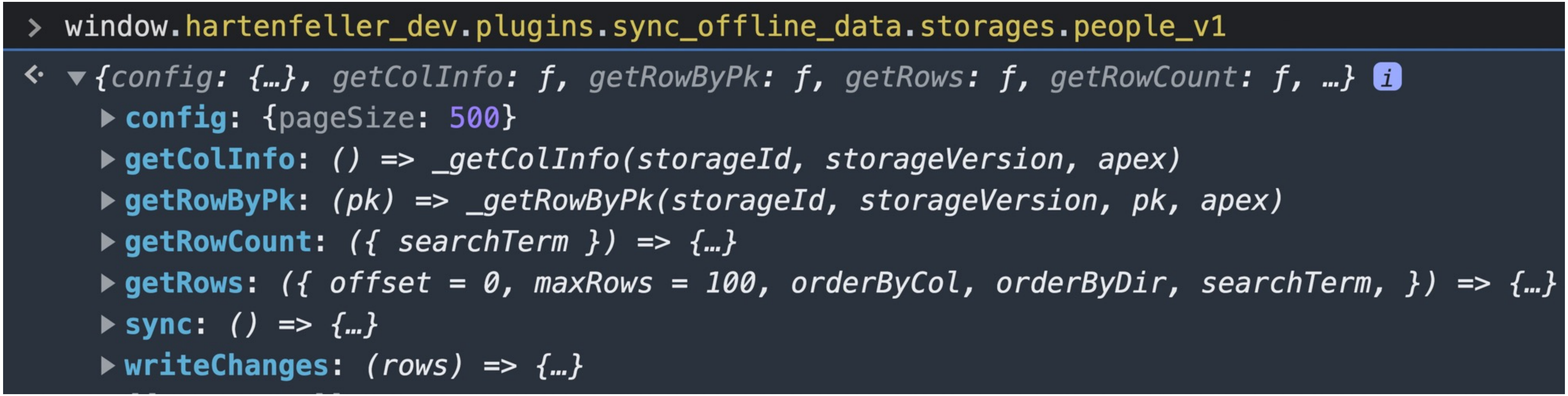
- Sites get a private file system
- SQLite directly writes to here



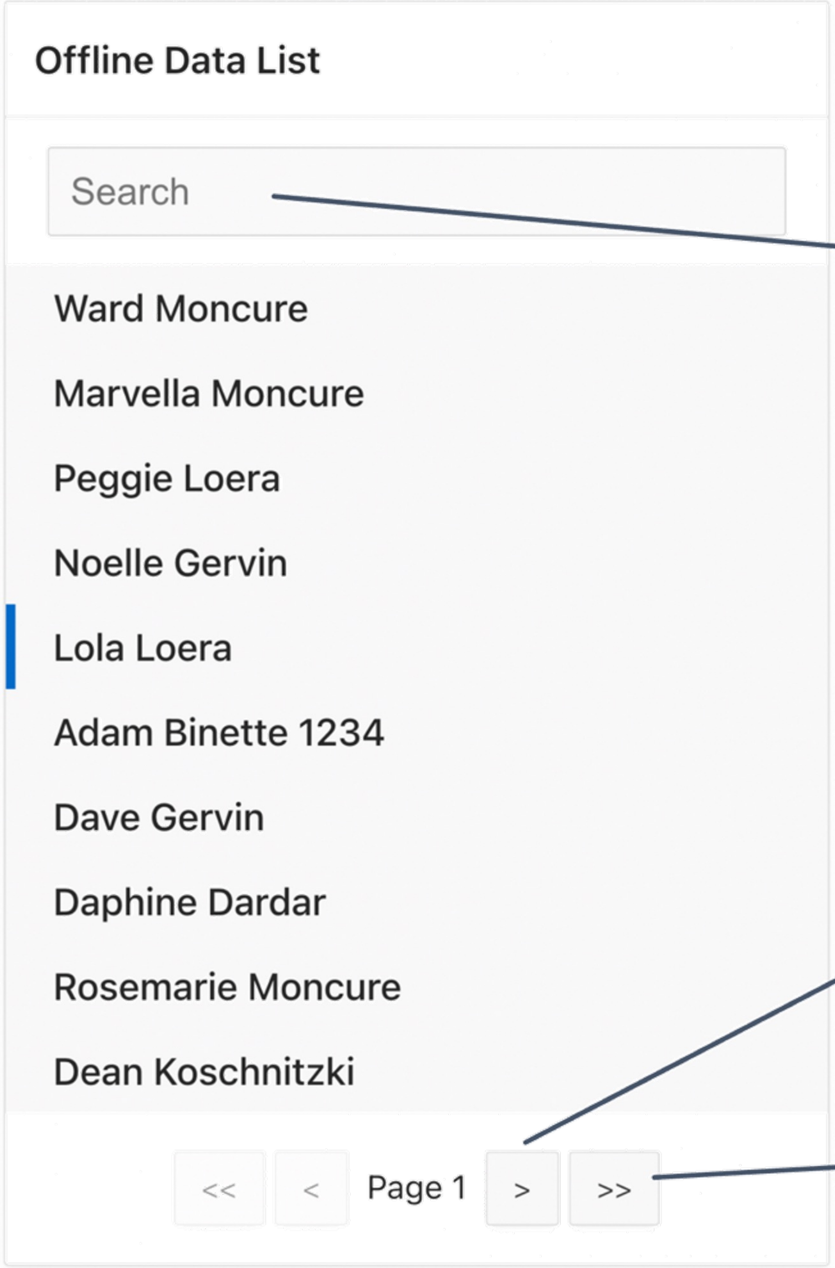
🖥					📱					
Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	Opera Android	Safari on iOS	Samsung Internet	WebView Android
✓ 86	✓ 86	✓ 111	✓ 72	✓ 15.2	✓ 109	✓ 111	✓ 74	✓ 15.2	✗ No	✓ 109

Source

How develop Plug-Ins



```
> window.hartenfeller_dev.plugins.sync_offline_data.storages.people_v1
<· ▼{config: {…}, getColInfo: f, getRowByPk: f, getRows: f, getRowCount: f, …} i
    ▶ config: {pageSize: 500}
    ▶ getColInfo: () => _getColInfo(storageId, storageVersion, apex)
    ▶ getRowByPk: (pk) => _getRowByPk(storageId, storageVersion, pk, apex)
    ▶ getRowCount: ({ searchTerm }) => {…}
    ▶ getRows: ({ offset = 0, maxRows = 100, orderByCol, orderByDir, searchTerm, }) => {…}
    ▶ sync: () => {…}
    ▶ writeChanges: (rows) => {…}
```



Offline Data List

Search

Ward Moncure

Marvella Moncure

Peggie Loera

Noelle Gervin

Lola Loera

Adam Binette 1234

Dave Gervin

Daphine Dardar

Rosemarie Moncure

Dean Koschnitzki

```
getRows({ searchTerm: "xyz" })
```

```
getRows({ offset: 10 })
```

```
getRowCount()
```

Photo by Jakob Owens from Unsplash

Wrapping Up

What still needs to be done

- Data type handling
  - Dates, Timestamps …
  - BLOBs
- LOVs (Select Lists, Popup LOV, Shuttle, Checkboxes…)
- Apply more table rules to client (FKs, check constraints, indexes, defaults…)
- Error message improvements
- Notify when sync completed with data changes
- Expand Plug-In Ecosystem

Why you may not want to use this

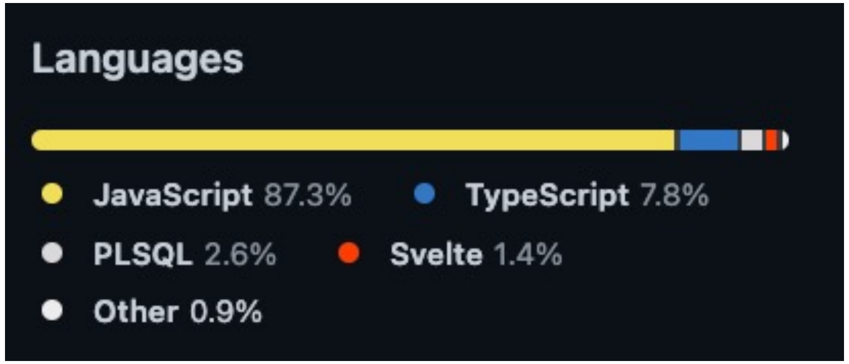
- You lose authority over your data
  - Data on the client can easily be duplicated / shared
- Merge conflicts / Inconsistent data state
- Divergent development approach for offline apps
- Increased complexity
- Scalability: Synchronization processes can demand significant data transfers
- SQLite as fast as the device using it

Photo by JESHOOTS.COM from Unsplash

Where can I get this ?

https://github.com/phartenfeller/apex-off-grid

- Needs Documentation
- Needs Refactoring
- Contributions welcome 😄



Languages

● JavaScript 87.3%	● TypeScript 7.8%
● PLSQL 2.6%	● Svelte 1.4%
● Other 0.9%	

Conclusion

- I am quite happy with the result
- Sync-Plug-In easy to use, hides complexity
- Confident that SQLite is a great choice for small to large data sizes
- In theory great browser support (I will get iOS to work)

Photy by Sebastian Herrmann from Unsplash

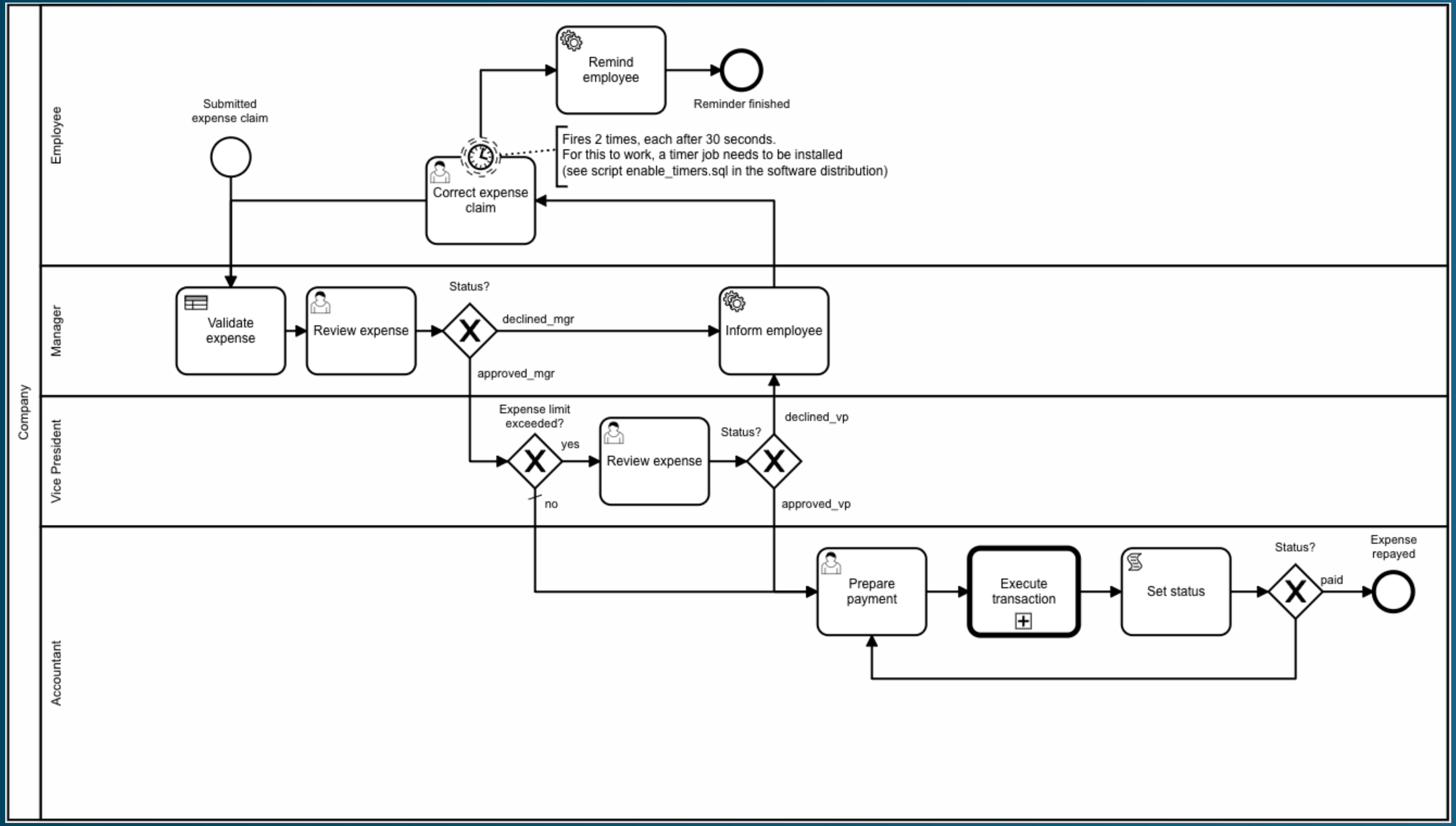
Testing APEX Apps is now as easy as creating them.

- Tailored to APEX
- Save a lot of time on regression tests
- Use our intuitive LCT-App and don‘t write any test code
- Testing on multiple platforms simultaneously

LCT

Flows for APEX

BPMN 2.0 Workflows for APEX



- Open Source
- Community Driven
- Support available

Do you have further questions?

Philipp Hartenfeller
Senior Consultant

philipp.hartenfeller@mt-ag.com
@phartenfeller
https://hartenfeller.dev/blog

MT GmbH
Balcke-Dürr-Allee 9
40882 Ratingen

www.mt-itsolutions.com