

@git.zone/tsview

a unified UI for viewing whats going on with an app.

- [readme.md for @git.zone/tsview](#)
- [changelog.md for @git.zone/tsview](#)

readme.md for @git.zone/tsview

A powerful developer tool for browsing and managing S3-compatible storage and MongoDB databases through a sleek web UI — with real-time change streaming baked in. Built with TypeScript, designed for developers who need quick, visual access to their data stores during development.

Issue Reporting and Security

For reporting bugs, issues, or security vulnerabilities, please visit community.foss.global/. This is the central community hub for all issue reporting. Developers who sign and comply with our contribution agreement and go through identification can also get a code.foss.global/ account to submit Pull Requests directly.

Install

```
# Global installation (recommended for CLI usage)
pnpm add -g @git.zone/tsview

# Local installation (for programmatic usage)
pnpm add @git.zone/tsview
```

📖 Quick Start

1. Configure Your Connection

Create a `.nogit/env.json` file in your project root (auto-generated by `gitzone service`):

```
{
  "S3_ENDPOINT": "localhost",
  "S3_PORT": "9000",
  "S3_ACCESSKEY": "minioadmin",
  "S3_SECRETKEY": "minioadmin",
  "S3_USESSL": false,
  "MONGODB_URL": "mongodb://localhost:27017",
  "MONGODB_NAME": "mydb"
}
```

2. Launch the Viewer

```
tsview
```

That's it! Your browser opens automatically to the viewer interface.

□ Features

□□ S3 Storage Browser

Powered by `dees-storage-browser` from `@design.estate/dees-catalog`:

- **Column View Navigation** — Mac Finder-style interface with resizable columns
- **List View** — Traditional key-based view with hierarchical navigation
- **Real-time Preview** — View images, JSON, text, code, and more directly in the browser
- **Bucket Management** — Create, delete, and switch between buckets
- **File Operations** — Upload, download, delete, move, and copy objects
- **In-place Text Editing** — Edit text files directly with change tracking
- **Smart Content Type Detection** — Automatic recognition for 20+ file types
- **Breadcrumb Navigation** — Clickable path traversal

□□ MongoDB Browser

- **Database Explorer** — Hierarchical navigation through databases and collections
- **Database Overview** — Collection counts, data sizes, index stats at a glance
- **Document Viewer** — Paginated table view with JSON filter support
- **Document Editor** — Full CRUD with syntax-highlighted code editor and change tracking

- **Index Management** — View, create, and drop indexes
- **Aggregation Pipeline** — Run aggregation queries directly
- **Collection Stats** — Document counts, sizes, storage metrics
- **Server Status** — Connection info, version, uptime
- **Show/Hide System Databases** — Toggle visibility of `admin`, `local`, `config`

⚡ Real-Time Change Streaming

- **MongoDB Change Streams** — Live updates via native MongoDB change streams
- **S3 Change Detection** — Polling-based bucket monitoring with ETag comparison (5s intervals)
- **Activity Stream** — Combined timeline of all changes from both sources, filterable by type
- **Live Indicators** — Green dot + change count badges on active views
- **WebSocket Subscriptions** — Per-collection, per-bucket, or global activity feed
- **Auto-Reconnect** — Subscriptions automatically restored after connection loss

☐ Modern Web UI

- Dark theme designed for developer comfort
- Responsive layout with resizable panels
- Context menus for quick actions
- Everything bundled — zero external runtime dependencies in the browser

CLI Usage

```
# Start viewer with auto-detected port (starts from 3010)
tview

# Force a specific port
tview --port 3000

# S3 browser only
tview s3

# MongoDB browser only
tview mongo

# or
```

Programmatic API

Use tsview as a library in your own tools:

```
import { TsView } from '@git.zone/tsview';

const viewer = new TsView();

// Option 1: Load from .nogit/env.json (gitzone service format)
await viewer.loadConfigFromEnv();

// Option 2: Configure programmatically
viewer.setStorageConfig({
  endpoint: 'localhost',
  port: 9000,
  accessKey: 'minioadmin',
  accessSecret: 'minioadmin',
  useSsl: false,
});

viewer.setMongoConfig({
  mongoDbUrl: 'mongodb://localhost:27017',
  mongoDbName: 'mydb',
});

// Option 3: Cloud services
viewer.setStorageConfig({
  endpoint: 's3.amazonaws.com',
  accessKey: 'AKIAXXXXXXX',
  accessSecret: 'your-secret-key',
  useSsl: true,
  region: 'us-east-1',
});

viewer.setMongoConfig({
```

```

mongoDbUrl: 'mongodb+srv://user:pass@cluster.mongodb.net',
mongoDbName: 'production',
});

// Start the server
const port = await viewer.start();
console.log(`Viewer running on http://localhost:${port}`);

// Or specify a port
await viewer.start(3500);

// Graceful shutdown
await viewer.stop();

```

Configuration

Project-level via `.smartconfig.json`

```

{
  "@git.zone/tsview": {
    "port": 3015,
    "killIfBusy": true,
    "openBrowser": false
  }
}

```

Option	Type	Default	Description
<code>port</code>	<code>number</code>	auto	Fixed port (auto-finds from 3010 if not set)
<code>killIfBusy</code>	<code>boolean</code>	<code>false</code>	Kill existing process if port is busy
<code>openBrowser</code>	<code>boolean</code>	<code>true</code>	Automatically open browser on start

Port priority: CLI `--port` flag → `.smartconfig.json` → auto-detect

Environment Variables (`.nokit/env.json`)

S3

Variable	Description
<code>S3_ENDPOINT</code>	S3-compatible server hostname
<code>S3_PORT</code>	Server port (optional)
<code>S3_ACCESSKEY</code>	Access key ID
<code>S3_SECRETKEY</code>	Secret access key
<code>S3_USESSL</code>	Use HTTPS (<code>true</code> / <code>false</code>)

MongoDB

Variable	Description
<code>MONGODB_URL</code>	Full connection string (preferred)
<code>MONGODB_NAME</code>	Default database name

Or use individual variables:

Variable	Description
<code>MONGODB_HOST</code>	Hostname
<code>MONGODB_PORT</code>	Port
<code>MONGODB_USER</code>	Username
<code>MONGODB_PASS</code>	Password
<code>MONGODB_NAME</code>	Database name

Supported S3 Providers

tsview works with any S3-compatible storage:

Provider	Status
MinIO	Perfect for local dev
AWS S3	Amazon's object storage
DigitalOcean Spaces	Simple object storage

Provider	Status
Backblaze B2	S3-compatible API
Cloudflare R2	Zero egress fees
Wasabi	Hot cloud storage
Self-hosted	Any S3-compatible server

Supported File Types for Preview

Category	Extensions
Images	.png, .jpg, .jpeg, .gif, .webp, .svg
Text	.txt, .md, .log, .sh, .env
Code	.json, .js, .ts, .tsx, .jsx, .html, .css
Data	.csv, .xml, .yaml, .yml
Documents	.pdf

📁 Architecture

```

tsview/
├─ ts/                                # Backend (Node.js)
│  └─ api/                             # TypedRequest API handlers
│     └─ handlers.s3.ts                # S3 bucket & object operations
│     └─ handlers.mongodb.ts          # MongoDB CRUD & admin operations
│  └─ config/                          # Configuration management
│  └─ server/                           # Web server (TypedServer + TypedSocket)
│  └─ streaming/                        # Real-time change streaming
│     └─ classes.changestream-manager.ts # MongoDB + S3 watchers
│     └─ interfaces.streaming.ts       # Subscription interfaces
│  └─ interfaces/                       # Shared TypeScript interfaces
└─ tsview.classes.tsview.ts           # Main class
├─ ts_web/                             # Frontend (bundled via esbuild → base64ts)
│  └─ elements/                         # Web components (LitElement)
│     └─ tsview-app.ts                 # App shell + navigation
│     └─ tsview-mongo-*.ts            # MongoDB browser components
│     └─ tsview-activity-stream.ts     # Real-time activity feed

```

```
| └─ adapters/                # Data provider adapters
|   └─ s3-data-provider.ts    # IStorageDataProvider for dees-storage-browser
| └─ services/                # API + WebSocket clients
| └─ styles/                  # Dark theme
|   └─ utilities/             # Formatting helpers
└─ .smartconfig.json         # Build & runtime config
```

How It Works

1. **Backend** — A `TypedServer` serves the bundled web UI and exposes a typed API via `TypedRequest` over HTTP. A `TypedSocket` WebSocket layer handles real-time streaming subscriptions.
2. **Frontend** — LitElement-based web components communicate with the backend via `TypedRequest`. The S3 browser uses `dees-storage-browser` from `@design.estate/dees-catalog` with a custom `IStorageDataProvider` adapter. The `ChangeStreamService` connects over WebSocket and distributes real-time events to active views via RxJS Subjects.
3. **Streaming** — The `ChangeStreamManager` creates MongoDB Change Streams and S3 BucketWatchers on demand (one per subscribed resource). Changes are pushed to subscribed clients and accumulated in a 1000-event ring buffer for the Activity Stream view.

Development

```
# Clone
git clone https://code.foss.global/git.zone/tsview.git
cd tsview

# Install dependencies
pnpm install

# Build (bundles frontend + compiles TypeScript)
pnpm build

# Development mode with hot reload
pnpm run watch

# Run tests
pnpm test
```

License and Legal Information

This repository contains open-source code licensed under the MIT License. A copy of the license can be found in the [license](#) file.

Please note: The MIT License does not grant permission to use the trade names, trademarks, service marks, or product names of the project, except as required for reasonable and customary use in describing the origin of the work and reproducing the content of the NOTICE file.

Trademarks

This project is owned and maintained by Task Venture Capital GmbH. The names and logos associated with Task Venture Capital GmbH and any related products or services are trademarks of Task Venture Capital GmbH or third parties, and are not included within the scope of the MIT license granted herein.

Use of these trademarks must comply with Task Venture Capital GmbH's Trademark Guidelines or the guidelines of the respective third-party owners, and any usage must be approved in writing. Third-party trademarks used herein are the property of their respective owners and used only in a descriptive manner, e.g. for an implementation of an API or similar.

Company Information

Task Venture Capital GmbH Registered at District Court Bremen HRB 35230 HB, Germany

For any legal inquiries or further information, please contact us via email at hello@task.vc.

By using this repository, you acknowledge that you have read this section, agree to comply with its terms, and understand that the licensing of the code does not imply endorsement by Task Venture Capital GmbH of any derivative works.

changelog.md for @git.zone/tsview

2026-03-24 - 1.12.9 - fix(package)

bump package version to 1.12.8

- Updates the package metadata version with no indicated source changes.

2026-03-24 - 1.12.8 - fix(package)

bump package version to 1.12.8

- Updates the project package version from 1.12.7 to 1.12.8.

2026-03-24 - 1.12.7 - fix(deps)

bump aws sdk, smartdata, and tswatch dependencies

- update @aws-sdk/client-s3 from ^3.1015.0 to ^3.1016.0
- update @push.rocks/smartdata from ^7.1.0 to ^7.1.1
- update @git.zone/tswatch from ^3.3.1 to ^3.3.2

2026-03-24 - 1.12.6 - fix(build)

remove skiplibcheck from the build script

- Updates the build command to run tsbuild without the --skiplibcheck flag.
- Tightens TypeScript build validation by including library type checks during builds.

2026-03-24 - 1.12.5 - fix(package)

bump package version to 1.12.5

- Updates the package metadata version from 1.12.4 to 1.12.5.

2026-03-24 - 1.12.4 - fix(config)

migrate project config to .smartconfig.json and refresh build settings

- replaces npmextra.json with .smartconfig.json and updates packaged files accordingly
- adds --skiplibcheck to the build script and simplifies tsconfig settings
- bumps build and runtime dependencies and updates README references to the new config file

2026-03-24 - 1.12.3 - fix(package)

bump package version to 1.12.3

- Updates the package version from 1.12.2 to 1.12.3.

2026-03-24 - 1.12.2 - fix(config)

migrate runtime configuration loading from npmextra to smartconfig

- replace the @push.rocks/npmextra dependency with @push.rocks/smartconfig
- update tsview startup configuration types and loading logic to read port, killIfBusy, and openBrowser from smartconfig
- adjust busy-port error messaging to reference smartconfig.json
- add npmjs.org to the release registries configuration

2026-03-14 - 1.12.1 - fix(storage)

rename S3 configuration and change stream interfaces to storage-oriented types

- Renames public config APIs from setS3Config/getS3Config/hasS3 to setStorageConfig/getStorageConfig/hasStorage.
- Replaces shared S3 interfaces with storage equivalents such as IStorageConfig, IStorageObject, IStorageChangeEvent, and storage subscription request types.
- Updates frontend integration to use dees-storage-browser and the new storage data provider and change stream types.
- Refreshes dependency versions and documentation to match the new storage-oriented naming.

2026-03-12 - 1.12.0 - feat(web)

replace custom S3 browser components with dees-s3-browser integration

- adds an S3 data provider adapter that delegates object operations to the existing ApiService
- wires live bucket change events into dees-s3-browser via changeStreamService
- removes exported custom tsview S3 elements in favor of the shared catalog component

2026-01-29 - 1.11.1 - fix(mongo-browser)

increase default editor width and broaden resize range in Mongo browser pane

- Set default editorWidth from 400 to 700
- Update CSS grid-template default editor width variable to 700
- Expand editor resize bounds: min 300 -> 250, max 700 -> 1000 (resizer calculation adjusted)

2026-01-28 - 1.11.0 - feat(s3)

add rename support for files and folders in S3 UI columns and keys

- Adds 'Rename' option to context menus in tsview-s3-columns and tsview-s3-keys for files and folders
- Implements a rename dialog with validation, auto-focus and smart selection (preserves extensions), and progress/error states

- Performs renames via `apiService.moveObject` / `apiService.movePrefix` and refreshes the UI after success
- Bumps `@design.estate/dees-catalog` dependency to `^3.41.2`

2026-01-28 - 1.10.2 - fix(tsview-s3-columns)

append trailing slash to destination key when moving folders to ensure folder destination path

- `ts_web/elements/tsview-s3-columns.ts`: add `'/'` to `destKey` when `moveSource.isFolder` to preserve folder semantics
- Prevents folder moves from being treated as object moves by ensuring destination key ends with a slash
- Change affects logic that calls `apiService.movePrefix` for folder moves

2026-01-28 - 1.10.1 - fix(playwright-mcp)

remove Playwright-generated snapshot images to avoid committing autogenerated test artifacts and reduce repository size

- Deleted `~22` Playwright PNG snapshot files under `.playwright-mcp` (visual test artifacts).
- No source code logic changed — this is test artifact cleanup only.
- Reduces repository bloat and prevents noisy visual diffs in future Playwright runs.

2026-01-28 - 1.10.0 - feat(s3)

add S3 move (object & prefix) support: server handlers, API client methods, UI dialogs/picker, drag-and-drop and validation

- Server: added typed handlers for `moveObject` (copy+delete for a single object) and `movePrefix` (recursive list, copy all objects to dest then delete source directory). Handlers return success/error and `movedCount` for prefix moves.
- API: added `api.service` methods `moveObject` and `movePrefix` so web client can call new server handlers.

- Interfaces: introduced IReq_MoveObject and IReq_MovePrefix request/response typings in ts/interfaces.
- Web UI: added move dialogs, a move picker, drag-and-drop support for folders/files, UI states and styles in ts_web elements (tsview-s3-columns, tsview-s3-keys). Move dialogs/picker integrated into existing render flows.
- Utilities: added move-validator utility (validateMove, getParentPrefix) and exported it from utilities index to prevent invalid operations (e.g. moving a folder into itself or to the same parent).
- Behavior notes: prefix move implementation performs recursive listing, copies each object to the new prefix, then deletes the source directory (permanent). Errors are caught and surfaced; movedCount is returned for prefix moves.

2026-01-28 - 1.9.0 - feat(s3-columns)

load full prefix path on initial load and add folder upload support

- loadInitialColumn now loads all prefix path segments in parallel and pre-selects child prefixes so multi-column path is restored
- added getPathSegments helper and auto-scroll to show the rightmost column after load
- added separate hidden folder input (webkitdirectory) and folder upload flow; triggerFileUpload now accepts 'files' | 'folder'
- replaced generic 'Upload...' with 'Upload Files...' and added 'Upload Folder...' menu items
- updated updated() to react to currentPrefix changes and cleaned up folder input on disconnectedCallback

2026-01-28 - 1.8.1 - fix(cli)

set executable permission on cli.js

- metadata-only change: updated file mode for cli.js (executable bit set)
- no source code changes; ensures package binary is runnable

2026-01-28 - 1.8.0 - feat(streaming)

add global activity watchers, client-side buffering, and improved real-time streaming UX

- Introduce global MongoDB and S3 watchers in ChangeStreamManager to feed a deployment-level activity stream (start/stop automatically based on subscribers)
- ChangeStreamManager: activity buffering, globalWatchersActive flag, start/stop global watchers, and emitMongoActivityEvent API for handlers
- Mongo API handlers accept an optional ChangeStreamManager and emit activity events for DB/collection/document operations
- Server now initializes ChangeStreamManager earlier and passes it to registerMongoHandlers; request context uses localData.peer.id for WebSocket peer lookup
- Client: ChangeStreamService adds connection promise/ensureConnected, activity buffering across tabs, and more robust connect/subscribe flows (waits for connect in-flight)
- UI updates: activity stream shows relative times with live clock, buffers events from app-level subscription, auto-scroll behavior adjusted, and connection-based re-subscription
- Mongo/S3 browser components ensure RxJS listeners are attached before server-side subscribe and surface connection status
- S3 browser: add drag-and-drop folder upload support (webkitdirectory), recursive folder entry traversal, upload with preserved relative paths, and column refresh/refreshAll logic
- Minor API/behavioral changes use connection peer id via conn.peer?.id when resolving target connections

2026-01-27 - 1.7.0 - feat(s3)

add drag-and-drop and context-menu file uploads, inline text editing in preview, and increase preview width

- Add drag-and-drop uploading into columns and folders with visual drag-over / drag-target states and a delayed folder-hover target
- Add context-menu 'Upload Files' action and hidden file input to trigger multi-file uploads; show upload overlay with progress and refresh column after upload
- Implement upload handling using apiService.putObject and base64 file reads, with upload progress tracking and error logging
- Add inline text editing in the preview (Edit / Save / Discard flow) using dees-input-code for editing and dees-preview for read-only display
- Increase preview default width from 350px to 700px and raise max resize limit from 600px to 1000px
- Bump dependencies: @git.zone/tstest -> ^3.1.8, @design.estate/dees-catalog -> ^3.41.1, @design.estate/dees-element -> ^2.1.6

2026-01-26 - 1.6.1 - fix(ci)

add Gitea CI workflows, documentation updates, and packaging metadata tweaks

- Add .gitea workflow files for tag and non-tag pushes (security, test, release, metadata jobs)
- Add buildDocs script (tsdoc) and add bugs/homepage/pnpm.overrides entries to package.json
- Update README and readme.hints.md with real-time streaming docs, formatting/table fixes, and examples punctuation fixes
- Tidy npmextra.json formatting and array inline style changes
- Update changelog.md entries and correct trailing newline/formatting
- Minor .gitignore whitespace fix

2026-01-25 - 1.6.0 - feat(readme)

document real-time change streaming and expand README with features, architecture, and configuration updates

- Add Real-Time Change Streaming section: MongoDB change streams, S3 polling (ETag), activity stream, WebSocket subscriptions, and auto-reconnect behavior
- Expand S3 and MongoDB feature lists: in-place text editing, enhanced previews (code), show/hide system databases, live indicators, context menus
- Reintroduce project-level npmextra.json example and clarify environment variable names and priority order
- Add Architecture and How It Works sections with tree layout and streaming design details
- Minor wording, formatting, and installation clarifications (prefer pnpm examples, fix LICENSE filename case)

2026-01-25 - 1.5.0 - feat(streaming)

add real-time streaming (MongoDB change streams & S3 bucket watchers) with WebSocket subscriptions and activity stream UI

- Server: add ChangeStreamManager to manage MongoDB change streams and S3 BucketWatcher subscriptions, handle subscription lifecycle, activity ring buffer and push events via TypedSocket.
- API: introduce streaming TypedRequest interfaces and register new handlers (subscribeMongo/unsubscribeMongo, subscribeS3/unsubscribeS3, subscribeActivity/unsubscribeActivity, getRecentActivity, push events).

- Client: add ChangeStreamService (TypedSocket connection, reconnection, RxJS subjects) and integrate into tsview-app, mongo and S3 browsers to show live status, change counters and refresh content on updates.
- UI: add new Activity Stream component (tsview-activity-stream), Activity tab and navigation, plus visual indicators (Live/Offline, change badges, auto-scroll and filters).
- Dependencies & plugins: add @api.global/typedsocket and @push.rocks/smartrx, bump @push.rocks/smartbucket; expose smartrx/typedsocket in plugins for server and web bundles.
- Docs: update readme.hints.md with Real-Time Streaming architecture, interfaces and dependency notes.

2026-01-25 - 1.4.0 - feat(web)

add database overview panel, collection overview and resizable panels; show/hide system databases; use code editor with change-tracking in document view; add getDatabaseStats API and typings; enable overwrite for S3 uploads

- Add backend handler getDatabaseStats + request/response typings (IReq_GetDatabaseStats, IDatabaseStats) and ApiService.getDatabaseStats()
- New UI component tsview-mongo-db-overview to show database statistics (collections, objects, avg size, data/storage size, indexes)
- Collections list: added an "Overview" entry that opens the DB overview when selected
- Sidebar: context menu option to show/hide system databases (admin/config/local) and database collapse-on-click behavior
- Resizable layout improvements: draggable dividers added for sidebar, Mongo editor panel and S3 preview panel (persisted via local state variables)
- Document editor: switch to dees-input-code, track original content and unsaved changes, add discard/save flow and improved save handling
- S3 handlers: fastPut calls now pass overwrite: true to allow replacing existing keys
- Minor dependency bumps: @git.zone/tstest and @design.estate/dees-catalog

2026-01-25 - 1.3.0 - feat(s3)

add S3 create file/folder dialogs and in-place text editor; export mongodb plugin

- Add mongodb dependency and export mongodb in ts/plugins.ts so ObjectId can be reused from plugins.
- Update handlers.mongodb to use plugins.mongodb.ObjectId instead of requiring mongodb directly.
- UI: Add create-file and create-folder dialogs and context-menu entries in tsview-app, tsview-s3-columns, and tsview-s3-keys to create objects (folders use a .keep object).

- Implement client-side helpers to determine content type/default content and call `apiService.putObject` with base64 content when creating files/folders.
- S3 preview: embed dees-input-code editor for text files with language detection, unsaved-changes indicator, Save/Discard flows, and saving via `apiService.putObject`.
- Various styling and UX improvements for dialogs, buttons, and editor states.

2026-01-25 - 1.2.0 - feat(s3,web-ui)

add S3 `deletePrefix` and `getObjectUrl` endpoints and add context menus in UI for S3 and Mongo views

- Add server-side TypedHandlers: `deletePrefix` and `getObjectUrl` (`ts/api/handlers.s3.ts`)
- Add request/response interfaces `IReq_DeletePrefix` and `IReq_GetObjectUrl` (`ts/interfaces/index.ts`)
- Add client API methods `deletePrefix` and `getObjectUrl` (`ts_web/services/api.service.ts`)
- Introduce context menu actions (`DeesContextmenu`) across UI: `bucket/database/collection/document/folder/file` actions including `open`, `copy path`, `delete`, `download` and `duplicate` (`ts_web/elements/tsview-app.ts`, `tsview-mongo-collections.ts`, `tsview-mongo-documents.ts`, `tsview-s3-columns.ts`, `tsview-s3-keys.ts`)
- Switch from inline delete buttons to contextual menus for safer UX; implement downloads via data URLs returned by `getObjectUrl` and deletion of S3 prefixes (folders)

2026-01-25 - 1.1.3 - fix(package)

update package metadata

- metadata-only change; no source code changes
- current version 1.1.2 → recommended patch bump to 1.1.3

2026-01-25 - 1.1.2 - fix(package)

apply minor metadata-only change (one-line edit)

- Change affects 1 file with a +1 -1 (metadata-only) — no behavioral changes
- Recommended bump of patch version from 1.1.1 to 1.1.2

2026-01-25 - 1.1.1 - fix(tsview)

fix bad build commit - remove accidental include

- Removed an accidental include that caused a bad build and unintended files to be part of the commit
- Patch release recommended from 1.1.0 to 1.1.1

2026-01-25 - 1.1.0 - feat(tsview)

add database and S3 handlers, tswatch/watch scripts, web utilities, assets and release config

- Add MongoDB management handlers: createDatabase, dropDatabase, dropCollection (ts/api/handlers.mongodb.ts)
- Enhance S3 handlers: logging, listBuckets improvements, expanded content-type map (ts/api/handlers.s3.ts)
- Add tswatch support and watcher config in npmextra.json and package.json (scripts: watch, startTs); add @git.zone/tswatch dep
- Add web exports for styles and utilities (ts_web/styles/index.ts, ts_web/utilities/index.ts) and minor UI assets (.playwright-mcp images)
- Update test to use toEqual and small test cleanup (test/test.tsview.ts)
- Update package.json: add startTs/watch scripts and bump @push.rocks/smartbucket patch
- Add release/registry and project metadata in npmextra.json for publishing

2026-01-23 - 1.0.0 - initial release: column view UI, S3 integration, and API fixes

Initial public release introducing the new column-based UI with resizable columns and horizontal navigation, plus backend fixes for S3 bucket listing and API endpoint handling.

- feat: Add resizable columns and horizontal scrolling
 - Columns can be resized by dragging the border between them
 - Column widths persist during navigation (150px min, 500px max)
 - Container scrolls horizontally when columns exceed available space
 - Auto-scrolls to show newly opened columns

- Resize handle highlights on hover/active state
- fix: Column view navigation and state preservation
 - Preserve previous columns during navigation; only reset columns when the bucket changes, not when the prefix changes
 - Column navigation now expands horizontally instead of resetting the column stack
 - Remove navigate event dispatch from column folder selection to preserve column state
- fix: Duplicate app instance rendering
 - Prevent duplicate tsview-app rendering by checking if the element already exists in the DOM
- fix / feat: S3 bucket listing
 - Implement listBuckets to actually query S3 (was previously hardcoded to return an empty array)
 - Add @aws-sdk/client-s3 dependency to perform direct S3 operations and return real bucket names
- fix: Resolve API 404s and improve dev caching
 - Update ApiService baseUrl to include the /typedrequest path expected by TypedServer
 - Add noCache option to ViewServer to prevent client caching issues during development
 - Bump @api.global/typedserver to v8.3.0 (includes noCache feature)
- chore: initial project scaffold
 - Initial commit and project scaffolding (summary)