

# changelog.md for @push.rocks/smartbucket

## 2026-03-14 - 4.5.1 - fix(readme)

refresh documentation wording and simplify README formatting

- removes emoji-heavy headings and inline log examples for a cleaner presentation
- clarifies S3-compatible storage wording in the project description
- streamlines testing and error handling sections without changing library functionality

## 2026-03-14 - 4.5.0 - feat(storage)

generalize S3 client and watcher interfaces to storage-oriented naming with backward compatibility

- rename SmartBucket internals from s3Client to storageClient and normalize configuration using storage descriptor types
- update watcher and interface types from S3-specific names to storage-oriented equivalents while keeping deprecated aliases
- refresh tests and documentation comments to reflect S3-compatible object storage terminology
- bump build, test, AWS SDK, and related dependency versions

## 2026-01-25 - 4.4.1 - fix(tests)

add explicit 'as string' type assertions to environment variable retrievals in tests

- Add 'as string' assertions for S3\_ACCESSKEY, S3\_SECRETKEY, S3\_ENDPOINT, and S3\_BUCKET to satisfy TypeScript typings in tests
- Cast S3\_PORT to string before parseInt to ensure correct input type and avoid compiler errors

- Changes are localized to test/test.listing.node+deno.ts

## 2026-01-25 - 4.4.0 - feat(watcher)

add polling-based BucketWatcher to detect add/modify/delete events and expose RxJS Observable and EventEmitter APIs

- Add new BucketWatcher implementation (ts/classes.watcher.ts) with polling, buffering and change detection by ETag/Size/LastModified
- Expose createWatcher(...) on Bucket (ts/classes.bucket.ts) and export watcher and interfaces from index (ts/index.ts)
- Add watcher-related interfaces (IS3ChangeEvent, IS3ObjectState, IBucketWatcherOptions) to ts/interfaces.ts
- Comprehensive README/docs additions and examples for BucketWatcher, buffering, options and use cases (readme.md, readme.hints.md)
- Add unit/integration tests for watcher behavior (test/test.watcher.node.ts)
- Non-breaking API additions only — ready for a minor version bump from 4.3.1 to 4.4.0

## 2026-01-24 - 4.3.1 - fix(bucket)

propagate S3 client errors instead of silently logging them; update build script, bump dev/dependencies, and refresh npmextra configuration

- Remove .catch(...) wrappers around s3Client.send so errors are no longer swallowed and will propagate to callers
- Update build script to use 'tsbuild tsfolders --allowimplicitany'
- Bump devDependencies: @git.zone/tsbuild to ^4.1.2, @git.zone/tsrun to ^2.0.1, @git.zone/tstest to ^3.1.6
- Bump dependency @aws-sdk/client-s3 to ^3.975.0
- Adjust npmextra.json structure (@git.zone/cli, @git.zone/tsdoc), add release registries and @ship.zone/szci entry
- Remove pnpm-workspace.yaml onlyBuiltDependencies configuration

## 2025-11-20 - 4.3.0 - feat(listing)

Add memory-efficient listing APIs: async generator, RxJS observable, and cursor pagination; export ListCursor and Minimatch; add minimatch dependency; bump to 4.2.0

- Added memory-efficient listing methods on Bucket: `listAllObjects` (async generator), `listAllObjectsObservable` (RxJS Observable), `createCursor` (returns `ListCursor`) and `listAllObjectsArray` (convenience array collector).
- New `ListCursor` class (`ts/classes.listcursor.ts`) providing page-based iteration: `next()`, `hasMore()`, `reset()`, `getToken()/setToken()`.
- Added glob matching helper `findByGlob(pattern)` using `minimatch` (exported via `plugins.Minimatch`).
- Exported `ListCursor` from `ts/index.ts` and exported `Minimatch` via `ts/plugins.ts`.
- Added `minimatch` dependency in `package.json` and bumped package version to 4.2.0; increased test timeout to 120s.
- Updated tests to read `S3_SECRETKEY`, `S3_PORT` and to assert bucket name from env (`test/test.node+deno.ts`, `test/test.trash.node+deno.ts`).
- No breaking changes: new APIs are additive and existing behavior preserved.

## 2025-11-20 - 4.2.0 - feat(listing)

Add memory-efficient listing with async generators, RxJS observables, and cursor pagination for huge buckets

### New Memory-Efficient Listing Methods:

#### Async Generator (Recommended for most use cases):

- `Bucket.listAllObjects(prefix?)` - Stream object keys one at a time using `for await...of`
- `Bucket.findByGlob(pattern)` - Find objects matching glob patterns (e.g., `**/*.json`, `npm/packages/*/index.json`)
- Memory efficient, supports early termination, composable

#### RxJS Observable (For complex reactive pipelines):

- `Bucket.listAllObjectsObservable(prefix?)` - Emit keys as Observable for use with RxJS operators (`filter`, `map`, `take`, etc.)
- Perfect for complex data transformations and reactive architectures

#### Cursor Pattern (For manual pagination control):

- `Bucket.createCursor(prefix?, options?)` - Create cursor for explicit page-by-page iteration
- `ListCursor.next()` - Fetch next page of results
- `ListCursor.hasMore()` - Check if more results available
- `ListCursor.reset()` - Reset to beginning
- `ListCursor.getToken()` / `ListCursor.setToken()` - Save/restore pagination state
- Ideal for UI pagination and resumable operations

### Convenience Methods:

- `Bucket.listAllObjectsArray(prefix?)` - Collect all keys into array (WARNING: loads all into memory)

### Benefits:

- Memory-efficient streaming for buckets with millions of objects
- Three patterns for different use cases (generators, observables, cursors)
- Support for early termination and incremental processing
- Glob pattern matching with minimatch
- Full TypeScript support with proper types
- Zero breaking changes - all new methods

### Dependencies:

- Added `minimatch` for glob pattern support

### Files Changed:

- `ts/classes.bucket.ts` - Added all listing methods
- `ts/classes.listcursor.ts` - NEW: Cursor implementation
- `ts/plugins.ts` - Export Minimatch
- `ts/index.ts` - Export ListCursor
- `test/test.listing.node+deno.ts` - NEW: Comprehensive listing tests
- `package.json` - Added minimatch dependency

## 2025-11-20 - 4.1.0 - feat(core)

Add S3 endpoint normalization, directory pagination, improved metadata checks, trash support, and related tests

- Add `normalizeS3Descriptor` helper to sanitize and normalize various S3 endpoint formats and emit warnings for mismatches (`helpers.ts`).
- Use normalized endpoint and credentials when constructing S3 client in `SmartBucket` (`classes.smartbucket.ts`).
- Implement paginated listing helper `listObjectsV2AllPages` in `Directory` and use it for `listFiles` and `listDirectories` to aggregate `Contents` and `CommonPrefixes` across pages (`classes.directory.ts`).
- Improve `MetaData.hasMetaData` to catch `NotFound` errors and return `false` instead of throwing (`classes.metadata.ts`).
- Export `metadata` and `trash` modules from `index` (`ts/index.ts`) and add a `Trash` class with utilities for trashed files and key encoding (`classes.trash.ts`).
- Enhance `Bucket` operations: `fastCopy` now preserves or replaces native metadata correctly, `cleanAllContents` supports paginated deletion, and improved `fastExists` error handling (`classes.bucket.ts`).

- Fix `Directory.getSubDirectoryByName` to construct new `Directory` instances with the correct parent directory reference.
- Add tests covering metadata absence and pagination behavior (`test/test.local.node+deno.ts`).

## 2025-11-20 - 4.0.1 - fix(plugings)

Use explicit `node:` imports for native path and stream modules in `ts/plugings.ts`

- Replaced imports of `'path'` and `'stream'` with `'node:path'` and `'node:stream'` in `ts/plugings.ts`.
- Ensures correct ESM resolution of Node built-ins when `package.json` type is `'module'` and avoids accidental conflicts with userland packages.

## 2025-11-20 - 4.0.0 - BREAKING CHANGE(core)

Make API strict-by-default: remove `*Strict` variants, throw on not-found/exists conflicts, add explicit `exists()` methods, update docs/tests and bump deps

- Breaking: Core API methods are strict by default and now throw errors instead of returning null when targets are missing or already exist (e.g. `getBucketByName`, `getFile`, `getSubDirectoryByName`, `fastPut`, `fastPutStream`).
- Removed `*Strict` variants: `fastPutStrict`, `getBucketByNameStrict`, `getFileStrict`, `getSubDirectoryByNameStrict` — use the base methods which are now strict.
- Added explicit existence checks: `bucketExists` (`SmartBucket`), `fileExists` (`Directory/fileExists`), `directoryExists` (`Directory.directoryExists`), and `fastExists` (`Bucket.fastExists`) to allow non-throwing checks before operations.
- Return type updates: `fastPut` now returns `Promise<File>` (no null), `getBucketByName/getFile/getSubDirectoryByName` now return the respective objects or throw.
- Improved error messages to guide callers (e.g. suggest setting `overwrite:true` on `fastPut` when object exists).
- Updated README, changelog and tests to reflect the new strict semantics and usage patterns.
- Developer/runtime dependency bumps: `@git.zone/tsbuild`, `@git.zone/tsrun`, `@git.zone/tstest`, `@aws-sdk/client-s3`, `@push.rocks/smartstring`, `@tsclass/tsclass` (version bumps recorded in `package.json`).
- Major version bump to 4.0.0 to reflect breaking API changes.

# 2025-11-20 - 4.0.0 - BREAKING:

## Strict by default + exists methods

Complete API overhaul: all methods throw by default, removed all \*Strict variants, added dedicated exists methods

### Breaking Changes:

#### Putters (Write Operations):

- `fastPut`: Return type `Promise<File | null>` → `Promise<File>`, throws when file exists and `overwrite` is false
- `fastPutStream`: Now throws when file exists and `overwrite` is false (previously returned silently)
- `fastPutStrict`: **Removed** - use `fastPut` directly

#### Getters (Read Operations):

- `getBucketByName`: Return type `Promise<Bucket | null>` → `Promise<Bucket>`, throws when bucket not found
- `getBucketByNameStrict`: **Removed** - use `getBucketByName` directly
- `getFile`: Return type `Promise<File | null>` → `Promise<File>`, throws when file not found
- `getFileStrict`: **Removed** - use `getFile` directly
- `getSubDirectoryByName`: Return type `Promise<Directory | null>` → `Promise<Directory>`, throws when directory not found
- `getSubDirectoryByNameStrict`: **Removed** - use `getSubDirectoryByName` directly

#### New Methods (Existence Checks):

- `bucket.fastExists({ path })` -  Already existed
- `directory.fileExists({ path })` - **NEW** - Check if file exists
- `directory.directoryExists(name)` - **NEW** - Check if subdirectory exists
- `smartBucket.bucketExists(name)` - **NEW** - Check if bucket exists

#### Benefits:

- **Simpler API**: Removed 4 redundant \*Strict methods
- **Type-safe**: No nullable returns - `Promise<T>` not `Promise<T | null>`
- **Fail-fast**: Errors throw immediately with precise stack traces
- **Consistent**: All methods behave the same way
- **Explicit**: Use `exists()` to check, then `get()` to retrieve
- **Better debugging**: Error location is always precise

## Migration Guide:

```
// =====  
// Pattern 1: Check then Get (Recommended)  
// =====  
  
// Before (v3.x):  
const bucket = await smartBucket.getBucketByName('my-bucket');  
if (bucket) {  
  // use bucket  
}  
  
// After (v4.0):  
if (await smartBucket.bucketExists('my-bucket')) {  
  const bucket = await smartBucket.getBucketByName('my-bucket'); // guaranteed to exist  
  // use bucket  
}  
  
// =====  
// Pattern 2: Try/Catch  
// =====  
  
// Before (v3.x):  
const file = await directory.getFile({ path: 'file.txt' });  
if (!file) {  
  // Handle not found  
}  
  
// After (v4.0):  
try {  
  const file = await directory.getFile({ path: 'file.txt' });  
  // use file  
} catch (error) {  
  // Handle not found  
}  
  
// =====  
// Pattern 3: Remove *Strict calls  
// =====
```

```
// Before (v3.x):
const file = await directory.getFileStrict({ path: 'file.txt' });

// After (v4.0):
const file = await directory.getFile({ path: 'file.txt' }); // already strict

// =====
// Pattern 4: Write Operations
// =====

// Before (v3.x):
const file = await bucket.fastPutStrict({ path: 'file.txt', contents: 'data' });

// After (v4.0):
const file = await bucket.fastPut({ path: 'file.txt', contents: 'data' }); // already strict
```

## 2025-08-18 - 3.3.10 - fix(helpers)

Normalize and robustly parse S3 endpoint configuration; use normalized descriptor in SmartBucket and update dev tooling

- Add normalizeS3Descriptor to ts/helpers.ts: robust endpoint parsing, coercion of useSsl/port, sanitization, warnings for dropped URL parts, and canonical endpoint URL output.
- Update SmartBucket (ts/classes.smartbucket.ts) to use the normalized endpoint, region, credentials and forcePathStyle from normalizeS3Descriptor.
- Adjust dev tooling: bump @git.zone/tsbuild -> ^2.6.7, @git.zone/tstest -> ^2.3.4, @push.rocks/qenv -> ^6.1.3 and update test script to run tstest with --verbose --logfile --timeout 60.
- Add .claude/settings.local.json containing local assistant/CI permission settings (local config only).

## 2025-08-15 - 3.3.9 - fix(docs)

Revise README with detailed usage examples and add local Claude settings

- Revamped README: reorganized content, added emojis and clearer headings for install, getting started, bucket/file/directory operations, streaming, metadata, trash/recovery, locking, and advanced configuration.
- Added many concrete code examples for SmartBucket, Bucket, Directory, File, streaming (node/web), RxJS replay subjects, metadata handling, trash workflow, file locking, magic-bytes detection, JSON operations, and cleaning bucket contents.
- Included testing instructions (pnpm test) and a Best Practices section with recommendations for strict mode, streaming, metadata, trash usage, and locking.
- Added `.claude/settings.local.json` to include local Claude configuration and tool permissions.
- No source code or public API changes; documentation and local tooling config only.

## 2025-08-15 - 3.3.8 - fix(tests)

Update tests to use `@git.zone/tstest`, upgrade dependencies, remove GitLab CI and add local CI/workspace config

- Tests: replace imports from `@push.rocks/tapbundle` with `@git.zone/tstest/tapbundle` and switch `tap.start()` to export default `tap.start()`
- Dependencies: bump `@aws-sdk/client-s3` and several `@push.rocks` packages; upgrade `@tsclass/tsclass` to a newer major
- DevDependencies: upgrade `@git.zone/tsbuild`, `@git.zone/tstest`, `@push.rocks/qenv`, and `@push.rocks/tapbundle`
- CI/config: remove `.gitlab-ci.yml`, add `.claude/settings.local.json`
- Workspace: add `pnpm-workspace.yaml` and `packageManager` field in `package.json`

## 2024-12-02 - 3.3.7 - fix(package)

Update author field in `package.json`

- Corrected the author field from 'Lossless GmbH' to 'Task Venture Capital GmbH' in the `package.json` file.

## 2024-12-02 - 3.3.6 - fix(package)

Fix license field in `package.json` to reflect MIT licensing

# 2024-11-25 - 3.3.5 - fix(test)

Refactor trash test to improve metadata validation

- Added new checks in trash tests to ensure metadata files are correctly moved to trash.
- Validated the presence and integrity of metadata within trashed files.

# 2024-11-25 - 3.3.4 - fix(core)

Minor refactoring and cleanup of TypeScript source files for improved readability and maintainability.

# 2024-11-24 - 3.3.3 - fix(documentation)

Improved documentation accuracy and consistency

- Updated the project description to reflect the cloud-agnostic nature and advanced capabilities
- Enhanced the README with detailed explanations and code examples for advanced features like trash management
- Clarified the handling and importance of metadata using the MetaData utility

# 2024-11-24 - 3.3.2 - fix(documentation)

Updated keywords and description for clarity and consistency.

- Modified keywords and description in package.json and npmextra.json.
- Enhanced readme.md file structure and examples

# 2024-11-24 - 3.3.1 - fix(File)

Fixed issue with file restore metadata operations.

- Corrected the order of operations in the file restore function to ensure custom metadata is appropriately deleted after moving the file.

## 2024-11-24 - 3.3.0 - feat(core)

Enhanced directory handling and file restoration from trash

- Refined getSubDirectoryByName to handle file paths treated as directories.
- Introduced file restoration function from trash to original or specified paths.

## 2024-11-24 - 3.2.2 - fix(core)

Refactor Bucket class for improved error handling

- Ensured safe access using non-null assertions when finding a bucket.
- Enhanced fastPut method by adding fastPutStrict for safer operations.
- Added explicit error handling and type checking in fastExists method.

## 2024-11-24 - 3.2.1 - fix(metadata)

Fix metadata handling for deleted files

- Ensured metadata is correctly stored and managed when files are deleted into the trash.

## 2024-11-24 - 3.2.0 - feat(bucket)

Enhanced SmartBucket with trash management and metadata handling

- Added functionality to move files to a trash directory.
- Introduced methods to handle file metadata more robustly.
- Implemented a method to clean all contents from a bucket.
- Enhanced directory retrieval to handle non-existent directories with options.
- Improved handling of file paths and metadata within the storage system.

# 2024-11-18 - 3.1.0 - feat(file)

Added functionality to retrieve magic bytes from files and detect file types using magic bytes.

- Introduced method `getMagicBytes` in `File` and `Bucket` classes to retrieve a specific number of bytes from a file.
- Enhanced file type detection by utilizing magic bytes in `MetaData` class.
- Updated dependencies for better performance and compatibility.

# 2024-11-18 - 3.0.24 - fix(metadata)

Fix metadata handling to address type assertion and data retrieval.

- Fixed type assertion issues in `MetaData` class properties with type non-null assertions.
- Corrected the handling of JSON data retrieval in `MetaData.storeCustomMetaData` function.

# 2024-10-16 - 3.0.23 - fix(dependencies)

Update package dependencies for improved functionality and security.

- Updated `@aws-sdk/client-s3` to version `^3.670.0` for enhanced S3 client capabilities.
- Updated `@push.rocks/smartstream` to version `^3.2.4`.
- Updated the dev dependency `@push.rocks/tapbundle` to version `^5.3.0`.

# 2024-07-28 - 3.0.22 - fix(dependencies)

Update dependencies and improve bucket retrieval logging

- Updated `@aws-sdk/client-s3` to `^3.620.0`

- Updated @git.zone/tsbuild to ^2.1.84
- Updated @git.zone/tsrun to ^1.2.49
- Updated @push.rocks/smartpromise to ^4.0.4
- Updated @tsclass/tsclass to ^4.1.2
- Added a log for when a bucket is not found by name in getBucketByName method

## 2024-07-04 - 3.0.21 - fix(test)

Update endpoint configuration in tests to use environment variable

- Modified `qenv.yml` to include `S3_ENDPOINT` as a required environment variable.
- Updated test files to fetch `S3_ENDPOINT` from environment instead of hardcoding.

## 2024-06-19 - 3.0.20 - Fix and Stability Updates

Improved overall stability and consistency.

## 2024-06-18 - 3.0.18 - Delete Functions Consistency

Ensured more consistency between delete methods and trash behavior.

## 2024-06-17 - 3.0.17 to 3.0.16 - Fix and Update

Routine updates and fixes performed.

# 2024-06-11 - 3.0.15 to 3.0.14 - Fix and Update

Routine updates and fixes performed.

# 2024-06-10 - 3.0.13 - Trash Feature Completion

Finished work on trash feature.

# 2024-06-09 - 3.0.12 - Fix and Update

Routine updates and fixes performed.

# 2024-06-08 - 3.0.11 to 3.0.10 - Fix and Update

Routine updates and fixes performed.

# 2024-06-03 - 3.0.10 - Fix and Update

Routine updates and fixes performed.

# 2024-05-29 - 3.0.9 - Update

## Description

Updated project description.

# 2024-05-27 - 3.0.8 to 3.0.6 - Pathing and Core Updates

Routine updates and fixes performed.

- S3 paths' pathing differences now correctly handled with a reducePath method.

# 2024-05-21 - 3.0.5 to 3.0.4 - Fix and Update

Routine updates and fixes performed.

# 2024-05-17 - 3.0.3 to 3.0.2 - Fix and Update

Routine updates and fixes performed.

# 2024-05-17 - 3.0.0 - Major Release

Introduced breaking changes in core and significant improvements.

# 2024-05-05 - 2.0.5 - Breaking Changes

Introduced breaking changes in core functionality.

# 2024-04-14 - 2.0.4 - TSCConfig Update

Updated TypeScript configuration.

# 2024-01-01 - 2.0.2 - Organization Scheme Update

Switched to the new organizational scheme.

---

Revision #2

Created 2026-03-28 13:08:33 UTC by foss.global Team

Updated 2026-03-29 16:51:04 UTC by foss.global Team