

changelog.md for @push.rocks/smartstate

2026-03-27 - 2.3.0 - feat(stateprocess)

add managed state processes with lifecycle controls, scheduled actions, and disposal safety

- introduces StateProcess with start, pause, resume, dispose, status, and auto-pause support
- adds createProcess() and createScheduledAction() on StatePart for polling, streams, and recurring actions
- adds disposal guards and Smartstate.dispose() to clean up state parts and attached processes
- improves selector and computed observables with distinct-until-changed behavior and skipped selector error emissions
- renames npmextra.json to .smartconfig.json and updates package tooling dependencies

2026-03-04 - 2.2.1 - fix(smartstate)

no changes detected; no version bump required

- Git diff shows no changes
- package.json version is 2.2.0
- No files modified — no release needed

2026-03-02 - 2.2.0 - feat(actions)

add action context for safe nested dispatch with depth limit to prevent deadlocks

- Introduce `IActionContext` to allow actions to dispatch sub-actions inline via `context.dispatch`
- Update `IActionDef` signature to accept an optional context parameter for backward compatibility
- Add `StatePart.createActionContext` and `MAX_NESTED_DISPATCH_DEPTH` to track and limit nested dispatch depth (throws on circular dispatchs)
- Pass a created context into `dispatchAction` so `actionDefs` can safely perform nested dispatches without deadlocking the mutation queue
- Add tests covering re-entrancy, deeply nested dispatch, circular dispatch depth detection, backward compatibility with actions that omit context, and concurrent dispatch serialization

2026-02-28 - 2.1.1 - fix(core)

serialize state mutations, fix batch flushing/reentrancy, handle falsy initial values, dispose old `StatePart` on force, and improve notification/error handling

- Serialize `setState()` and `dispatchAction()` using an internal mutation queue to prevent lost updates and race conditions.
- Prevent batch flush deadlocks by introducing `isFlushing` and draining pending notifications iteratively.
- Force `initMode` now disposes the previous `StatePart` so the `Subject` completes and resources are cleaned up.
- Treat falsy but non-null values (0, false) as present: `getStatePart` accepts 0 as initial value and `waitUntilPresent` resolves for false/0.
- Improve `notifyChange`: use a stable snapshot, catch and log hash computation errors, and avoid duplicate notifications; `notifyChangeCumulative` now safely catches async errors.
- Add `StatePart.dispose()` to complete the `Subject` and clear pending timers/middlewares.
- Add/adjust tests for concurrent dispatches, concurrent `setState`, disposal behavior, falsy state handling, batch re-entrancy, force-mode disposal, and zero initialization.
- Documentation and README improvements (examples, clearer descriptions, persistence notes) and minor code cleanup (remove unused import).

2026-02-27 - 2.1.0 - feat(smartstate)

Add middleware, computed, batching, selector memoization, `AbortSignal` support, and `Web Component Context Protocol` provider

- Introduce StatePart middleware API (addMiddleware) — middleware runs sequentially before validation/persistence and can transform or reject a state change.
- Add computed derived observables: standalone computed(sources, fn) and Smartstate.computed to derive values from multiple state parts (lazy subscription).
- Add batching support via Smartstate.batch(fn), isBatching flag, and deferred notifications to batch multiple updates and flush only at the outermost level.
- Enhance select() with selector memoization (WeakMap cache and shareReplay) and optional AbortSignal support (auto-unsubscribe).
- Extend waitUntilPresent() to accept timeout and AbortSignal options and maintain backward-compatible numeric timeout argument.
- Add attachContextProvider(element, options) to bridge state parts to Web Component Context Protocol (context-request events) with subscribe/unsubscribe handling.
- Update StatePart.setState to run middleware, persist processed state atomically, and defer notifications to batching when applicable.
- Tests and README updated to document new features, behaviors, and examples.

2026-02-27 - 2.0.31 - fix(deps)

bump devDependencies and fix README license path

- Bump @git.zone/tsbundle from ^2.8.3 to ^2.9.0
- Bump @types/node from ^25.2.0 to ^25.3.2
- Update documented dependency set/version to v2.0.30 in readme.hints.md
- Fix README license file path from LICENSE to license in readme.md

2026-02-02 - 2.0.30 - fix(config)

update npmextra configuration and improve README: rename package keys, add release registry config, clarify waitUntilPresent timeout and notification/persistence behavior

- Renamed npmextra keys: 'gitzone' → '@git.zone/cli' and 'tsdoc' → '@git.zone/tsdoc'
- Added release configuration for @git.zone/cli including registries (verdaccio and npm) and accessLevel
- Removed top-level 'npmci' section
- Added new '@ship.zone/szci' entry with npmGlobalTools
- README: added waitUntilPresent timeout example with error handling
- README: clarified notifyChangeCumulative is debounced and documented persistence behavior (merge with defaults, atomic writes)
- README: documented concurrency/race-condition safety and timeout support for waitUntilPresent

2026-02-02 - 2.0.29 - fix(smartstate)

prevent duplicate statepart creation and fix persistence/notification race conditions

- Add pendingStatePartCreation map to deduplicate concurrent createStatePart calls
- Adjust init handling so 'force' falls through to creation and concurrent creations are serialized
- Merge persisted state with initial payload in 'persistent' initMode, with persisted values taking precedence
- Persist to WebStore before updating in-memory state to ensure atomicity
- Debounce cumulative notifications via pendingCumulativeNotification to avoid duplicate notifications
- Log selector errors instead of silently swallowing exceptions
- Add optional timeout to waitUntilPresent and ensure subscriptions and timeouts are cleaned up to avoid indefinite waits
- Await setState when performing chained state updates to ensure ordering and avoid race conditions

2026-02-02 - 2.0.28 - fix(deps)

bump devDependencies and dependencies, add tsbundle build config, update docs, and reorganize tests

- Bumped @git.zone/tsbuild to ^4.1.2, @git.zone/tsbundle to ^2.8.3, @git.zone/tsrun to ^2.0.1, @git.zone/tstest to ^3.1.8, and @types/node to ^25.2.0
- Upgraded @push.rocks/smartjson to ^6.0.0
- Added @git.zone/tsbundle bundle configuration to npmextra.json for building a dist bundle
- Removed pnpm-workspace.yaml entries (cleaned workspace constraints)
- Updated readme and readme.hints (docs formatting, version bumped to v2.0.28, issue reporting/security section and dependency list)
- Reorganized tests: removed *.both.ts variants and added consolidated test files under test/ (test.ts, test.initialization.ts)

2025-09-12 - 2.0.27 - fix(StatePart)

Use stable JSON stringify for state hashing; update dependencies and tooling

- Replace `smartjson.stringify` with `smartjson.stableOneWayStringify` when creating SHA256 state hashes to ensure deterministic hashing and avoid duplicate notifications for semantically identical states.
- Bump runtime dependencies: `@push.rocks/smarthash -> ^3.2.6`, `@push.rocks/smartjson -> ^5.2.0`.
- Bump dev tooling versions: `@git.zone/tsbuild -> ^2.6.8`, `@git.zone/tsbundle -> ^2.5.1`, `@git.zone/tstest -> ^2.3.8`.
- Add local `.claude/settings.local.json` configuration for allowed permissions (local tooling/settings file).

2025-08-16 - 2.0.26 - fix(ci)

Add local Claude settings file to allow helper permissions for common local commands

- Added `.claude/settings.local.json` to grant local helper permissions for tooling
- Allowed commands: `Bash(tsx:)`, `Bash(tstest test:)`, `Bash(git add:)`, `Bash(git tag:)`
- No changes to source code or runtime behavior; tooling/config only

2025-07-29 - 2.0.25 - fix(core)

Major state initialization and validation improvements

- Fixed state hash bug: Now properly compares hash values instead of storing state objects
- Fixed state initialization merge order: Initial state now correctly takes precedence over stored state
- Improved type safety: `stateStore` properly typed as potentially undefined
- Simplified init mode logic with clear behavior for 'soft', 'mandatory', 'force', and 'persistent'
- Added state validation with extensible `validateState()` method
- Made `notifyChange()` async to support proper hash comparison
- Enhanced `select()` to filter undefined states automatically
- Added comprehensive test suite for state initialization scenarios
- Updated documentation with clearer examples and improved readme

2025-07-19 - 2.0.24 - fix(core)

Multiple fixes and improvements

- Fixed `StateAction` trigger method to properly return `Promise<TStateType>`

- Updated CI workflows to use new container registry and npmci package name
- Added pnpm workspace configuration for built-only dependencies

2025-07-19 - 2.0.23 - fix(ci)

Update CI workflows to use new container registry and npmci package name

- Changed CI image from 'registry.gitlab.com/hosttoday/ht-docker-node:npmci' to 'code.foss.global/host.today/ht-docker-node:npmci'
- Replaced npmci installation command from '@shipzone/npmci' to '@ship.zone/npmci' in workflow configurations

2025-07-19 - 2.0.22 - fix(smartsstate)

Fix StateAction trigger method to properly return Promise

- Fixed StateAction.trigger() to return Promise<TStateType> as expected
- Updated readme with improved documentation and examples
- Replaced outdated legal information with Task Venture Capital GmbH details
- Added implementation notes in readme.hints.md

2025-06-19 - 2.0.21 - maintenance

General updates and improvements

2025-06-19 - 2.0.20 - fix(smartsstate)

Update build scripts and dependency versions; replace isohash with smarthashWeb for state hash generation

- Adjusted package.json scripts to include verbose testing and modified build command

- Bumped development dependencies (tsbuild, tsbundle, tsrun, tstest, tapbundle) to newer versions
- Updated production dependencies (lik, smarhash, smartpromise, smartrx) with minor version bumps
- Replaced import of isohash with smarhashWeb in state hash generation, ensuring consistency across modules

2024-10-02 - 2.0.19 - fix(dependencies)

Update dependencies to latest versions

- Updated @git.zone/tsbuild to version ^2.1.84
- Updated @git.zone/tsbundle to version ^2.0.15
- Updated @git.zone/tsrun to version ^1.2.49
- Updated @git.zone/tstest to version ^1.0.90
- Updated @push.rocks/tapbundle to version ^5.3.0
- Updated @types/node to version ^22.7.4
- Updated @push.rocks/lik to version ^6.0.15
- Updated @push.rocks/smartjson to version ^5.0.20
- Updated @push.rocks/smartpromise to version ^4.0.4
- Updated @push.rocks/smartrx to version ^3.0.7
- Updated @push.rocks/webstore to version ^2.0.20

2024-10-02 - 2.0.18 - fix(core)

Fix type errors and typos in Smartstate class

- Updated type annotation in Smartstate class to ensure StatePartNameType extends string.
- Fixed a typo in the JSDoc comment: 'existing' instead of 'exiting'.
- Corrected improper type casting in the Smartstate class.

2024-05-29 - 2.0.17 - Maintenance

General updates and improvements.

- Updated project description
- Multiple updates to `tsconfig`
- Updated `npmextra.json` to include `githost`

2023-10-07 - 2.0.16 - Maintenance

General updates and improvements.

- Core update

2023-10-04 - 2.0.15 - Maintenance

General updates and improvements.

- Core update

2023-10-03 - 2.0.14 to 2.0.10 - Maintenance

General updates and improvements.

- Core updates

2023-09-11 - 2.0.9 - Maintenance

General updates and improvements.

- Core update

2023-09-11 - 2.0.8 - Maintenance

General updates and improvements.

- Core update

2023-07-27 - 2.0.7 - Maintenance

General updates and improvements.

- Core update

2023-07-27 - 2.0.6 - Maintenance

General updates and improvements.

- Core update

2023-04-13 - 2.0.5 - Maintenance

General updates and improvements.

- Core update

2023-04-12 - 2.0.4 - Maintenance

General updates and improvements.

- Core update

2023-04-04 - 2.0.3 to 2.0.1 - Maintenance

General updates and improvements.

- Core updates

2023-03-15 - 2.0.0 - Major Update

Core update with significant changes.

2022-03-25 - 1.0.23 - Major Update

Breaking changes and major updates.

- SWITCH TO ESM

2022-01-24 - 1.0.22 - Maintenance

General updates and improvements.

- Core updates

2020-11-30 - 1.0.21 to 1.0.20 - Maintenance

General updates and improvements.

- Core updates

2020-11-30 - 1.0.19 to 1.0.18 - Maintenance

General updates and improvements.

- Core updates

2020-07-27 - 1.0.17 to 1.0.16 - Maintenance

General updates and improvements.

- Core updates

2020-05-27 - 1.0.15 - Maintenance

General updates and improvements.

- Core update

2020-05-27 - 1.0.14 - Maintenance

General updates and improvements.

- Core update

2019-09-25 - 1.0.13 - Maintenance

General updates and improvements.

- Core update

2019-09-25 - 1.0.12 - Maintenance

General updates and improvements.

- Core updates

2019-04-30 - 1.0.11 to 1.0.10 - Maintenance

General updates and improvements.

- Core updates

2019-03-22 - 1.0.9 - Maintenance

General updates and improvements.

- Core update

2019-02-27 - 1.0.8 - Minor Update

Minor updates and improvements.

- Updated action generation
- Core update

2019-02-21 - 1.0.7 - Initial Release

Initial release of the project.

- Initial core implementation

Revision #2

Created 2026-03-28 13:10:36 UTC by foss.global Team

Updated 2026-03-29 16:53:07 UTC by foss.global Team