

# changelog.md for @api.global/typedsocket

## 2026-02-26 - 4.1.2 - fix(ci)

update npmextra.json to rename keys, add release registries and add CI entry; remove legacy npmci configuration

- Renamed top-level key "gitzone" to "@git.zone/cli" and "tsdoc" to "@git.zone/tsdoc" to use scoped configuration keys.
- Added release configuration under @git.zone/cli with registries [<https://verdaccio.lossless.digital>, <https://registry.npmjs.org>] and accessLevel set to public.
- Removed legacy "npmci" block and introduced an "@ship.zone/szci" section (npmGlobalTools: []).
- No source code changes; this is a configuration/CI/publishing metadata update. Current package.json version is 4.1.1, recommend a patch bump.

## 2026-02-26 - 4.1.1 - fix(typedsocket)

reject pending requests on disconnect to avoid hanging promises

- Rejects all pending requests with Error('TypedSocket disconnected') when connection is lost
- Clears pendingRequests map to prevent memory leaks
- Preserves existing auto-reconnect scheduling behavior

# 2025-12-04 - 4.1.0 - feat(typedsocket)

Add SmartServe integration, tagging and improved client reconnect/backoff; update deps and tests

- Add SmartServe integration: new `TypedSocket.fromSmartServe(smartServe, typedRouter)` with SmartServe connection wrappers and tag compatibility
- Implement server-side tag handlers and storage (`__typedsocket_setTag` / `__typedsocket_removeTag`) with payloads stored under internal `TAG_PREFIX`
- Add client improvements: `TypedSocket.createClient(...)` with `autoReconnect`, configurable exponential backoff (`initialBackoffMs`, `maxBackoffMs`, `maxRetries`) and connection status observable (`statusSubject`)
- Expose client tag API: `setTag(name, payload)` and `removeTag(name)` to manage server-side tags
- Server-side APIs for connection discovery: `findAllTargetConnections`, `findTargetConnection`, `findAllTargetConnectionsByTag`, `findTargetConnectionByTag`
- Plugins and internal tooling updated: replace `smartsocket` exports with `smartdelay` and `smartpromise`; export `smartpromise` and `smartdelay` for internal use
- Tests updated to use SmartServe for end-to-end scenarios (server/client message flow, tagging and connection discovery); test script changed to run in verbose mode
- Bumped dependency versions and adjusted peer dependency for `@push.rocks/smartsolve` to `>=1.1.0`

# 2025-12-03 - 4.0.0 - BREAKING CHANGE(TypedSocket.createServer)

Remove SmartExpress attachment support from `createServer` and upgrade `smartsocket` to `^3.0.0`

- Bump dependency `@push.rocks/smartsocket` from `^2.1.0` to `^3.0.0`.
- Remove optional `SmartExpress` attachment parameter from `TypedSocket.createServer` and remove the `setExternalServer` call — `createServer` now only accepts a `TypedRouter` and starts its own `Smartsocket` server.
- Update README: remove the `SmartExpress` integration section and adjust the API docs to reflect the new `createServer(router)` signature.

- Adjust TypeScript implementation to match the simplified createServer signature and behavior.

## 2025-12-02 - 3.1.1 - fix(package)

Include pnpm packageManager metadata in package.json

- Added packageManager field to package.json:  
pnpm@10.18.1+sha512.77a884a165cbba2d8d1c19e3b4880eee6d2fcabd0d879121e282196b80042351d5eb3ca0935fa599da1dc51265cc68816ad2bddd2a2de5ea9fdf92adbec7cd34
- No runtime or API changes — metadata-only change

## 2025-12-02 - 3.1.0 - feat(TypedSocket)

Add SmartServe integration to TypedSocket, support SmartServe connections and tagging; update dependencies and docs; remove GitLab CI config

- Add TypedSocket.fromSmartServe(smartServe, typedRouter) to bind TypedSocket to an existing SmartServe instance for native WebSocket handling.
- Introduce ISmartServeConnectionWrapper and wrapSmartServePeer to provide tag compatibility between SmartServe (Set) and TypedSocket (id/payload). Tag payloads are stored in peer.data with '**typedsocket\_tag**' prefix.
- Server-initiated requests in SmartServe mode use TypedRouter.fireEventInterestMap to correlate requests and responses asynchronously.
- Extend createTypedRequest, findAllTargetConnections, findTargetConnection, findAllTargetConnectionsByTag and findTargetConnectionByTag to support both Smartsocket and SmartServe backends (via wrappers).
- eventSubject: Emit a warning and return a new Subject in SmartServe mode (SmartServe lifecycle/events should be consumed via SmartServe hooks).
- stop(): In SmartServe mode TypedSocket clears internal wrappers but does not stop SmartServe (SmartServe lifecycle is managed externally).
- Add optional peer dependency @push.rocks/smartserv and expose type-only imports in ts/typedsocket.plugins.ts.
- Bump several devDependencies and runtime dependencies to newer versions (tsbuild, tsbundle, tsrun, tstest, smartenv, smartserve, smartjson, smartrx, smartsocket, smartstring, smarturl).
- Update README and readme.hints with SmartServe integration docs and examples.

- Remove .gitlab-ci.yml (CI configuration removed).

## 2024-04-18 - 3.0.1 - patch

Small patch release.

- Version bump to 3.0.1.
- No additional details provided in commits.

## 2024-04-18 - 3.0.0 - core

Major 3.0.0 release with core updates and documentation improvements.

- Core fixes and updates applied.
- Documentation updated (commit dated 2024-04-14).
- General maintenance and polishing for the 3.x line.

## 2023-08-06 - 2.0.28 - core (BREAKING CHANGE)

Breaking change in core introduced in 2.0.28; important compatibility considerations.

- BREAKING CHANGE(core): update — review integration points before upgrading.
- This capstone release follows a series of patch releases addressing stability and core behavior.

## 2022-03-24 - 2.0.0 - core

Introduction of the 2.x stable line and subsequent patch series.

- 2.0.0 initial release of the 2.x series.
- Follow-up patch releases (2.0.1 through 2.0.27) contain numerous "fix(core): update" changes — ongoing bug fixes, stability improvements, and small core adjustments across 2022-2023.

- Users upgrading from 1.x should review 2.x changes and the 2.0.28 breaking change above.

# 2020-12-26 - 1.0.2 - core

Initial implemented functionality and the start of 1.x maintenance.

- fix(core): implement basic functionality — initial usable feature set implemented in 1.0.2.
- Subsequent 1.0.3 through 1.0.27 releases (2020–2022) included ongoing fixes and minor updates to stabilize the 1.x line.

---

Revision #4

Created 2026-03-28 12:50:13 UTC by foss.global Team

Updated 2026-03-29 16:48:05 UTC by foss.global Team