

changelog.md for @push.rocks/smartserve

2026-03-23 - 2.0.3 - fix(adapter.node)

unref the node server before closing connections during shutdown

- Calls `server.unref()` before closing all connections so shutdown does not keep the Node.js event loop alive.
- Improves shutdown behavior for the Node adapter when stopping HTTP or HTTPS servers.

2026-03-23 - 2.0.2 - fix(adapter.node)

close websocket server and active connections when stopping the Node adapter

- Store the `WebSocketServer` instance on the adapter so it can be closed during shutdown.
- Close all active HTTP connections before closing the underlying Node server to improve shutdown cleanup.

2025-12-20 - 2.0.1 - fix(readme)

update README: rework features, add OpenAPI/Swagger, compression, request validation, examples, and runtime stats

- Reworked Features section into a markdown table and added items for OpenAPI/Swagger, Request Validation, and Auto Compression.
- Updated Quick Start example (removed Guard import) and added a Table of Contents.
- Expanded examples with a manual route handling snippet and guidance to bypass decorator routing.
- Added runtime example fields: `instance.hostname` and new `stats()` usage (uptime, `requestsTotal`, `requestsActive`).
- Minor legal/formatting updates: fixed LICENSE link casing/path and clarified trademark wording.

2025-12-20 - 2.0.0 - BREAKING CHANGE(request)

introduce lazy request body parsing via `ctx.json()/text()/arrayBuffer()/formData` and remove `IRequestContext.body`

- Add `RequestContext` class implementing lazy, cached body parsing methods: `json()`, `text()`, `arrayBuffer()`, `formData`.
- Remove `IRequestContext.body` property — handlers and interceptors must call `ctx.json()/ctx.text()/...` to access the request body (breaking API change).
- `createContext` now returns a `RequestContext` synchronously and no longer pre-parses or coerces the body.
- OpenAPI validator (`validateRequest`) made async and updated to use `ctx.json()` for request body validation; `createValidationInterceptor` now awaits validation.
- Updated README and tests to use async handlers and `ctx.json()` for body access.
- Updated `npmextra.json`: replaced `gitzone` key with `@git.zone/cli`, replaced `npmci` with `@ship.zone/szci`, and added release registries and `accessLevel`.

2025-12-08 - 1.4.0 - feat(openapi)

Add OpenAPI module: decorators, spec generator, runtime validation and Swagger UI

- Introduce a new OpenAPI module providing decorators, types, spec generator, handlers and validation utilities
- Add decorators: `ApiOperation`, `ApiParam`, `ApiQuery`, `ApiHeader`, `ApiRequestBody`, `ApiResponseBody`, `ApiSecurity`, `ApiTag`
- Add `OpenApiGenerator` to produce OpenAPI 3.1 JSON from registered controllers and route metadata
- Add runtime request validation and coercion using `@cfworker/json-schema` (validate request body, params, query, headers)
- Register OpenAPI endpoints and Swagger UI (and ReDoc) handlers when `SmartServe.openapi` is enabled
- Integrate validation interceptor into controller registry compilation so validation runs before other interceptors
- Expose `openapi` exports from the public API (`ts/index.ts` and `decorators index`)
- Add extensive types (`openapi.types.ts` and `decorator types`) and coercion utilities for query/path params
- Add tests for OpenAPI functionality (`test/test.openapi.ts`)
- Bump dependencies: `@api.global/typedrequest` to `^3.2.5` and add `@cfworker/json-schema` `^4.1.1`

2025-12-05 - 1.3.0 - feat(compression)

Improve compression implementation (buffering and threshold), add Deno brotli support, add compression tests and dynamic route API

- Buffer response bodies before compressing and perform size threshold check after buffering; return uncompressed responses when below threshold.
- Set Content-Length to the compressed size and use provider.compress to produce full compressed payloads instead of streaming compression from the middleware.
- Add Deno-native brotli support via Deno.compress and use CompressionStream for gzip/deflate; brotli streaming is not attempted in web runtime.
- Pass compression threshold from SmartServe configuration into compressResponse so route/global thresholds are honored.
- Expose ControllerRegistry.addRoute and dynamicRoutes to allow adding dynamic routes without controller classes.
- Add comprehensive compression tests (gzip and brotli) using raw HTTP requests to avoid Node fetch auto-decompression; tests cover large/small responses, @Compress/@NoCompress behavior, and global compression disable.
- Change test runner invocation to use verbose mode.

2025-12-05 - 1.2.0 - feat(compression)

Add cross-runtime response compression (Brotli/gzip), per-route decorators, and pre-compressed static file support

- Introduce a cross-runtime compression provider (Node zlib + Web CompressionStream fallback) with create/get provider APIs (ts/compression/compression.runtime.ts).
- Add compression middleware utilities (normalize config, shouldCompressResponse, algorithm selection, streaming/full-body compression) and default configuration (ts/compression/compression.middleware.ts).
- Implement Accept-Encoding parsing, encoding selection, and compressibility checks (ts/utills/utills.encoding.ts) and export types/utilities from utills/index.ts.
- Add @Compress and @NoCompress decorators and route-level compression metadata support (ts/decorators/decorators.compress.ts, decorators.types.ts, registry updates, and exports).
- Integrate compression into SmartServe core: global compression config, applyCompression for custom handlers, WebDAV, static files, and route responses (ts/core/smartserve.classes.smartserve.ts, smartserve.interfaces.ts).
- Enhance FileServer to serve pre-compressed variants (.br/.gz) when available, adjust headers/ETag/Length, and avoid using pre-compressed files for range requests (ts/files/file.server.ts).
- Expose compression APIs from package entry point and export zlib via plugins for Node provider; update readme.hints.md with configuration examples and notes.

2025-12-03 - 1.1.2 - fix(deps)

Bump dependency versions for build and runtime tools

- Update devDependency @git.zone/tsbundle from ^2.0.5 to ^2.6.3
- Update devDependency @types/node from ^20.8.7 to ^24.10.1
- Update dependency @api.global/typedrequest from ^3.0.0 to ^3.1.11

2025-12-03 - 1.1.1 - fix(adapters)

Attach WebSocket peer to TypedRouter request localData and add ws dependency

- When routing incoming WebSocket messages through TypedRouter (node/deno/bun), the connection peer is now attached to requestObj.localData so typed handlers can access the active connection.
- Add runtime dependency on "ws" to enable WebSocket support in the Node adapter (used by dynamic import in the adapter).

2025-12-02 - 1.1.0 - feat(websocket)

Add TypedRouter WebSocket integration, connection registry, peer tagging and broadcast APIs

- Add dependency on @api.global/typedrequest and re-export it via plugins
- Introduce typedRouter support in IWebSocketHooks and adapters (Node, Bun, Deno) to route JSON RPC messages through TypedRouter.routeAndAddResponse
- Add internal IWebSocketConnectionCallbacks to register/unregister peers; adapters receive these via a _connectionCallbacks property on websocket options
- Persist per-peer tags and data (peer.tags: Set) across adapters; Bun adapter stores persistent ws.data so tags survive re-wraps
- Add WebSocketConfigError and validate websocket config to prevent using typedRouter together with onMessage (throws if both are set)
- Expose connection-management APIs on SmartServe: getWebSocketConnections(), getWebSocketConnectionsByTag(tag), broadcastWebSocket(data) and broadcastWebSocketByTag(tag, data)
- Update README/hints to document TypedRouter mode, connection registry, peer tagging, and broadcast methods
- Legacy onMessage mode remains supported; typedRouter mode enables automatic JSON routing and connection registry

2025-11-29 - 1.0.2 - fix(package)

Update package metadata, scripts and dependency pins

- Set package exports entrypoint to ./dist_ts/index.js
- Add/adjust npm scripts: test, build and buildDocs
- Update devDependencies and runtime dependencies
- Add packageManager field with pnpm lock information and update repository/bugs/homepage fields

2025-11-29 - 1.0.1 - initial release

Initial project commit and repository bootstrap.

- Repository initialized (initial commit).
- Added base project scaffold and starter configuration.
- Added initial documentation placeholders (README) and basic metadata.

Revision #2

Created 2026-03-28 13:11:35 UTC by foss.global Team

Updated 2026-03-29 16:54:05 UTC by foss.global Team