

changelog.md for @apiclient.xyz/bunq

2025-07-29 - 4.4.0 - feat(export)

Added buffer download methods to ExportBuilder for in-memory statement handling

- Added `download()` method to get statements as Buffer without saving to disk
- Added `downloadAsArrayBuffer()` method for web API compatibility
- Enhanced documentation for account's `getAccountStatement()` method with month-based selection
- Updated README with comprehensive examples for all statement export options

2025-07-29 - 4.3.0 - feat(http)

Enhanced HTTP client with automatic rate limit handling

- Added `@push.rocks/smartrequest` dependency for robust HTTP handling
- Implemented automatic retry with exponential backoff for rate-limited requests
- Built-in handling of HTTP 429 responses with intelligent waiting
- Respects `Retry-After` headers when provided by the server
- Maximum of 3 retry attempts with configurable backoff (1s, 2s, 4s)
- Improved error handling and network resilience
- Updated readme documentation with automatic rate limit handling examples

2025-07-27 - 4.2.1 - fix(tests)

Fix test compatibility with breaking changes from v4.0.0

- Updated all tests to handle new API structure where methods return objects
- Fixed destructuring for `getAccounts()` which now returns `{ accounts, sessionData? }`
- Ensures all 83 tests pass successfully with the stateless architecture

2025-07-27 - 4.2.0 - feat(core)

Switch to native fetch API for all HTTP requests

- Replaced @push.rocks/smartrequest with native fetch API throughout the codebase
- Updated HTTP client to use fetch with proper error handling
- Updated attachment upload/download methods to use fetch
- Updated export download method to use fetch
- Updated sandbox user creation to use fetch
- Removed smartrequest dependency from package.json and plugins
- Improved error messages with HTTP status codes

2025-07-26 - 4.1.3 - fix(export)

Fix PDF statement download to use direct content endpoint

- Changed `downloadContent()` method to use the `/content` endpoint directly for PDF statements
- Removed unnecessary attachment lookup step that was causing issues
- Simplified the download process for customer statement exports

2025-07-25 - 4.1.1 - fix(httpclient)

Fix query parameter handling for smartrequest compatibility

- Changed query parameter handling to pass objects directly instead of URLSearchParams
- Removed URLSearchParams usage and string conversion
- Now passes queryParams as an object to smartrequest which handles URL encoding internally

2025-07-25 - 4.1.0 - feat(transactions)

Enhanced transaction pagination support with full control over historical data retrieval

- Added full `IBunqPaginationOptions` support to `getTransactions()` method

- Now supports `older_id` for paginating backwards through historical transactions
- Supports custom `count` parameter (defaults to 200)
- Maintains backward compatibility - passing a number is still treated as `newer_id`
- Only includes pagination parameters that are explicitly set (not false/undefined)
- Added `example.pagination.ts` demonstrating various pagination patterns

This enhancement allows banking applications to properly fetch and paginate through historical transaction data using both `newer_id` and `older_id` parameters.

2025-07-25 - 4.0.0 - BREAKING CHANGE(core)

Complete stateless architecture - consumers now have full control over session persistence

- **BREAKING:** Removed all file-based persistence - no more automatic saving to `.nokit/` directory
- **BREAKING:** `init()` now returns `ISessionData` that must be persisted by the consumer
- **BREAKING:** API methods like `getAccounts()` now return `{ data, sessionData? }` objects
- Added `ISessionData` interface exposing complete session state including `sessionId`
- Added `initWithSession(sessionData)` to initialize with previously saved sessions
- Added `exportSession()` and `getSessionData()` methods for session access
- Added `isSessionValid()` to check session validity
- Fixed session destruction to use actual session ID instead of hardcoded '0'
- Added `initOAuthWithExistingInstallation()` for explicit OAuth session handling
- Session refresh now returns updated session data for consumer persistence
- Added `example.stateless.ts` showing session management patterns

This change gives consumers full control over session persistence strategy (database, Redis, files, etc.) and makes the library suitable for serverless/microservices architectures.

2025-07-22 - 3.1.2 - fix(oauth)

Remove OAuth session caching to prevent authentication issues

- Removed static OAuth session cache that was causing incomplete session issues
- Each OAuth token now creates a fresh session without caching
- Removed cache management methods (`clearOAuthCache`, `clearOAuthCacheForToken`, `getOAuthCacheSize`)
- Simplified `init()` method to treat OAuth tokens the same as regular API keys

- OAuth tokens still handle "Superfluous authentication" errors with `initWithExistingInstallation`

2025-07-22 - 3.1.1 - fix(oauth)

Fix OAuth token authentication flow for existing installations

- Fixed `initWithExistingInstallation` to properly create new sessions with existing installation/device
- OAuth tokens now correctly skip installation/device steps when they already exist
- Session creation still uses OAuth token as the secret parameter
- Properly handles "Superfluous authentication" errors by reusing existing installation
- Renamed `initWithExistingSession` to `initWithExistingInstallation` for clarity

2025-07-22 - 3.1.0 - feat(oauth)

Add OAuth session caching to prevent multiple authentication attempts

- Implemented static OAuth session cache in `BunqAccount` class
- Added automatic session reuse for OAuth tokens across multiple instances
- Added handling for "Superfluous authentication" and "Authentication token already has a user session" errors
- Added `initWithExistingSession()` method to reuse OAuth tokens as session tokens
- Added cache management methods: `clearOAuthCache()`, `clearOAuthCacheForToken()`, `getOAuthCacheSize()`
- Added `hasValidSession()` method to check session validity
- OAuth tokens now properly cache and reuse sessions to prevent authentication conflicts

2025-07-22 - 3.0.8 - fix(oauth)

Correct OAuth implementation to match bunq documentation

- Removed OAuth mode from HTTP client - OAuth tokens use normal request signing
- OAuth tokens now work exactly like regular API keys (per bunq docs)
- Fixed test comments to reflect correct OAuth behavior
- Simplified OAuth handling by removing unnecessary special cases
- OAuth tokens properly go through full auth flow with request signing

2025-07-22 - 3.0.7 - fix(oauth)

Fix OAuth token authentication flow

- OAuth tokens now go through full initialization (installation → device → session)
- Fixed "Insufficient authentication" errors by treating OAuth tokens as API keys
- OAuth tokens are used as the 'secret' parameter, not as session tokens
- Follows bunq documentation: "Just use the OAuth Token as a normal bunq API key"
- Removed incorrect session skip logic for OAuth tokens

2025-07-22 - 3.0.6 - fix(oauth)

Fix OAuth token private key error

- Fixed "Private key not generated yet" error for OAuth tokens
- Added OAuth mode to HTTP client to skip request signing
- Skip response signature verification for OAuth tokens
- Properly handle missing private keys in OAuth mode

2025-07-22 - 3.0.5 - feat(oauth)

Add OAuth token support

- Added support for OAuth access tokens with isOAuthToken flag
- OAuth tokens skip session creation since they already have an associated session
- Fixed "Authentication token already has a user session" error for OAuth tokens
- Added OAuth documentation to readme with usage examples
- Created test cases for OAuth token flow

2025-07-22 - 3.0.4 - fix(tests,security)

Improve test reliability and remove sensitive file

- Added error handling for "Superfluous authentication" errors in session tests

- Improved retry mechanism with rate limiting delays in error tests
- Skipped tests that require access to private properties
- Removed genv.yml from repository for security reasons

2025-07-22 - 3.0.3 - fix(tests)

Fix test failures and draft payment API compatibility

- Fixed draft payment test by removing unsupported cancel operation in sandbox
- Added error handling for "Insufficient authentication" errors in transaction tests
- Fixed draft payment API payload formatting to use snake_case properly
- Removed problematic draft update operations that are limited in sandbox

2025-07-22 - 3.0.2 - fix(tests,webhooks)

Fix test assertions and webhook API structure

- Updated test assertions from `.toBe()` to `.toEqual()` for better compatibility
- Made error message assertions more flexible to handle varying error messages
- Fixed webhook API payload structure by removing unnecessary wrapper object
- Added `--logfile` flag to test script for better debugging

2025-07-18 - 3.0.1 - fix(docs)

docs: update readme examples for card management, export statements and error handling; add local settings for CLI permissions

- Replaced outdated card management examples with a note emphasizing that activation, PIN updates, and ordering should be handled via the bunq app or API.
- Updated export examples to use methods like `.lastDays(90)` and `.includeAttachments` for clearer instructions.
- Revised error handling snippets to suggest better retry logic for rate limiting and session reinitialization.
- Added a new `.claude/settings.local.json` file to configure allowed CLI commands and permissions.

2025-07-18 - 3.0.0 - BREAKING CHANGE(core)

Major restructuring and feature enhancements: added batch payments and scheduled payments with builder patterns, improved webhook management, migrated package naming to @apiclient.xyz/bunq, and updated documentation and tests.

- Introduced BunqPaymentBatch for creating multiple payments in a single API call.
- Implemented BunqSchedulePayment builder for scheduled and recurring payments.
- Enhanced webhook support with integrated webhook server and improved signature verification.
- Migrated package from @bunq-community/bunq to @apiclient.xyz/bunq with complete module restructure.
- Updated README and changelog to reflect breaking changes and provide a migration guide.
- Improved ESM compatibility and full TypeScript support.

2025-07-18 - 3.0.0 - BREAKING CHANGE(core)

Major update: Introduced batch payments, scheduled payment builder, and comprehensive webhook improvements with a complete migration from bunq-js-client to the new package structure. This release brings breaking changes in API signatures, module exports, and session management for enhanced ESM and TypeScript support.

- Added BunqPaymentBatch for creating multiple payments in a single API call
- Introduced BunqSchedulePayment with builder pattern for scheduled and recurring payments
- Enhanced webhook management with BunqWebhook and integrated webhook server support
- Migrated package naming from @bunq-community/bunq to @apiclient.xyz/bunq with a complete module restructure
- Improved ESM compatibility with proper .js extensions and TypeScript verbatimModuleSyntax support
- Updated documentation, changelog, and tests to reflect breaking changes and migration updates

2025-07-18 - 3.0.0 - BREAKING CHANGE(core)

Release 2.0.0: Major updates including batch payment support, scheduled payments with a builder pattern, comprehensive webhook enhancements, migration from bunq-js-client to the new package structure, and improved ESM/TypeScript compatibility.

- Added BunqPaymentBatch for creating multiple payments in a single API call.
- Introduced BunqSchedulePayment with builder pattern for scheduled and recurring payments.
- Implemented comprehensive webhook management with BunqWebhook and built-in webhook server.
- Migrated package naming from @bunq-community/bunq to @apiclient.xyz/bunq and restructured module exports.
- Improved ESM compatibility with proper .js extension usage and TypeScript verbatimModuleSyntax support.
- Updated documentation, changelog, and tests to reflect the new API and migration changes.

2019-10-02 to 2025-07-18 - Various - Minor updates

These releases did not include any feature or bug-fix changes beyond routine updates. The following versions are summarized here: 2.0.0, 1.0.22, 1.0.7, and 1.0.6.

2020-08-25 - 1.0.21 - core

Main change: fix(core): update

- Fixed issues in the core module

2020-08-21 - 1.0.20 - core

Main change: fix(core): update

- Fixed issues in the core module

2020-08-21 - 1.0.19 - core

Main change: fix(core): update

- Fixed issues in the core module

2020-08-21 - 1.0.18 - core

Main change: fix(core): update

- Fixed issues in the core module

2020-08-20 - 1.0.17 - core

Main change: fix(core): update

- Fixed issues in the core module

2020-08-20 - 1.0.16 - core

Main change: fix(core): update

- Fixed issues in the core module

2020-06-20 - 1.0.15 - core

Main change: fix(core): update

- Fixed issues in the core module

2019-12-15 - 1.0.14 - transactions

Main change: fix(transactions): enter a starting transaction

- Entered a starting transaction in the transactions module

2019-12-15 - 1.0.13 - core

Main change: fix(core): update

- Fixed issues in the core module

2019-12-15 - 1.0.12 - core

Main change: fix(core): update

- Fixed issues in the core module

2019-10-03 - 1.0.11 - core

Main change: fix(core): update

- Fixed issues in the core module

2019-10-03 - 1.0.10 - core

Main change: fix(core): update

- Fixed issues in the core module

2019-10-02 - 1.0.9 - core

Main change: fix(core): update

- Fixed issues in the core module

2019-10-02 - 1.0.8 - core

Main change: fix(core): update

- Fixed issues in the core module

2019-10-02 - 1.0.5 - core

Main change: fix(core): update

- Fixed issues in the core module

2019-10-02 - 1.0.4 - core

Main change: fix(core): update

- Fixed issues in the core module

2019-09-26 - 1.0.3 - core

Main change: fix(core): update

- Fixed issues in the core module

2019-09-26 - 1.0.2 - core

Main change: fix(core): update

- Fixed issues in the core module

2019-09-26 - 1.0.1 - core

Main change: fix(core): update

- Fixed issues in the core module

Revision #4

Created 2026-03-28 12:50:16 UTC by foss.global Team

Updated 2026-03-29 16:48:08 UTC by foss.global Team