3.7 Release Notes

Releases · https://namazustudios.com/docs/releases/3-7-release-notes/

Overview

Elements 3.7 is the biggest release yet. Elements can now be packaged, distributed, and hot-loaded at runtime — no server restart required. The new `.elm` archive format gives developers a single self-contained artifact to ship, and the new Deployments UI makes managing the full lifecycle a first-class experience in the admin console.
Highlights

Highlights

- **Hot-deploy Elements at runtime** — load, update, and unload Elements dynamically via the new deployment API and runtime/container services.
- **ELM package format** — distribute Elements as self-contained `.elm` archives, loadable from the filesystem or MongoDB GridFS, with nested JAR and SPI support.
- **Deployments UI** — brand-new admin UI with a guided wizard for creating and managing deployments, runtimes, and containers.
- **Maven archetypes** — new standard Java and Kotlin archetypes (`sdk-element-standard`, `sdk-element-kotlin`) to scaffold a production-ready Element in seconds.
- **SDK Bill of Materials** — `sdk-bom` makes dependency management across Element projects consistent and effortless.
- **Security hardening** — fixed OAuth2 account linking vulnerabilities, anonymous user collision bugs, and tightened classloader isolation.
- **Windows support** — full build and test compatibility on Windows, including path-length mitigations.

Coming Soon

The deployment and packaging work in 3.7 is laying the foundation for something bigger. Namazu Cloud will be a fully managed hosting solution for Namazu Elements; spin up and manage your own instances with one-click deploys, automated backups, and self-service scaling, all without touching infrastructure. Stay tuned.

New Features

Dynamic Deployment

Introduced a full hot-deploy system for Elements, enabling Elements to be loaded, updated, and unloaded at runtime without restarting the server.

- New `sdk-deployment` module extracts deployment services as a first-class SDK concern.
- `ElementDeployment` model supports multiple Elements per deployment, full CRUD, status tracking, and lifecycle events.
- New REST API endpoints for managing deployments (`/api/element/deployment`).
- `ElementRuntimeService` and `ElementContainerService` now support lifecycle events and handler cleanup on unmount.
- `LoadConfiguration` with `AttributesLoader` for customizable Element initialization.
- `ElementDeployment` can now inject the hosting `Application` into the Element's `Attributes` .
- SPI directory support and flat element loading architecture.
- New `AttributesLoader` and `SpiLoader` interfaces for external configuration.

ELM Package Format and Local SDK

Full support for distributing and loading Elements as self-contained `.elm` archive files.

- Load Elements from `.elm` packages stored in MongoDB GridFS or from the local filesystem.
- Nested JAR classloading within `.elm` packages.
- New `sdk-element-standard` Maven archetype for scaffolding standard Elements.
- New `sdk-bom` Bill of Materials for consistent SDK dependency management across Element projects.

- ELM Inspector: new REST endpoint and UI for introspecting a deployed `.elm` package.
- `PermittedTypesClassLoader` with `TypeRequest` / `PackageRequest` (literal, regex, wildcard) for fine-grained classloader isolation.
- `ElementDependencyMetadata` DTO for reporting deployed element dependencies.
- OpenAPI spec integration test suite.
- Local SDK improvements: simplified `ElementsLocal` API, Maven-based local builder, abstract base class for local tests.
- `sdk-bom` fleshed out as a proper BOM encompassing all SDK modules.
- Features endpoint added to expose server capabilities.

Signup Creates Session

The signup API now creates an authenticated session immediately after account creation, matching the behavior users expect from a signup flow.

- New endpoint path added to avoid breaking existing login integrations.
- Legacy endpoint preserved and deprecated.

Deployments UI

New web UI for managing Element deployments.

- Deployment wizard with Runtimes and Containers pages.
- Features dialog on the Deployments page.
- File upload support to pre-fill deployment fields.
- Edit element flow aligned with the create wizard.
- Search filter presets to hide large `.elm` objects from general object lists.
- Updated Container and Runtime detail views for better screen fit.

Kotlin Archetype

Added a Kotlin Maven archetype (`sdk-element-kotlin`) for developers who prefer Kotlin when building Elements.

Bug Fixes

Obsolete Fields

Removed obsolete application fields from the admin UI. Updated tests to reflect that metadata name changes are now permitted.

Codegen Creating Duplicate Methods

Fixed code generation producing duplicate method names in OpenAPI-generated clients. Resources without an explicit tag are now automatically tagged with `ElementsCore`.

Elements API Cleanup

Normalized REST API path conventions across applications, leaderboards, progress, and mission endpoints. Added `@Deprecated` annotations to all renamed methods. Renamed methods referencing "active" or "inactive" applications for consistency.

ShrinkWrap Module Loader

Added a ShrinkWrap-based module loader for test harnesses, enabling more reliable module assembly in integration tests.

Windows Build and Path Length Issues

Fixed failures building and running tests on Windows caused by path lengths exceeding the Windows filesystem limit. Added cleanup on exception to avoid leaving temporary state behind.

ClassLoader Memory Leak

Fixed a resource leak in `ElementImplementationClassLoader` where native resources were not released when a deployment was unloaded.

Error Hiding in DirectoryElementPathLoader

Improved error handling in `DirectoryElementPathLoader` so that load failures are surfaced rather than silently swallowed.

Elements Not Loading After Deployment (Regression)

Fixed a regression where Elements failed to load after deployment due to incorrect attribute loading hierarchy. Reworked `PropertiesAttributes` to fix a `NullPointerException` and corrected the attribute merge order so system attributes take proper precedence.

OAuth2 Account Linking

Fixed a bug where the JWK cache was considered out of date on the first authentication attempt, causing the first OIDC login to fail. Fixed related issues in the account linking flow.

Anonymous User Collision and OAuth2 Security

- Fixed a soft-deleted anonymous user being returned as a live user under certain conditions.
- Fixed a security issue where an OAuth2 identity could be linked to a second account under the same scheme.
- Prevented duplicate linking when the same identity provider scheme is used more than once.
- Added additional guard rails and expanded test coverage.

Missing SLF4J Dependency on Container B

Added a missing `slf4j` dependency that caused startup failures in certain deployment configurations.

Codegen / OAS3 Integration Tests

Added integration tests validating OpenAPI 3 code generation output. Hardened related codegen logic to prevent regressions.

Other Changes

- **Application configuration hotfix**: Fixed an error when creating an application configuration with no product bundles; `description` field is no longer required to be non-null.
- **Dependency updates**: Updated Jackson and Swagger to their latest versions. Migrated Jetty, Jersey, and Swagger to BOM-managed versions.

- **Branding cleanup**: Replaced outdated references to "Elemental-Computing" with current project naming.
- **CI improvements**: Reduced double-builds in Bitbucket Pipelines; added Maven version as a Surefire system property; fixed Makefile syntax.
- **Javadoc**: Added Javadoc generation to all builds; fixed misplaced Javadoc tags across multiple modules.