

Strapi Migration Transforms Digital Experience for Luxury Brand

CMS migration from Strapi v4 to v5 to modernize content workflows, improve performance, and ensure long-term support.

Overview

- QBurst migrated the client's content platform to v5, introducing improved editorial capabilities, faster API responses, and long-term support alignment.
- API response times improved by 60% and the backend became easier to maintain.
- Scalability was achieved by migrating to Strapi v5, adopting the new Document Service architecture, and aligning with Strapi's long-term roadmap.



Client Profile

A globally recognized luxury retail brand operating across EMEA, the client maintains a premium digital presence to support its omnichannel strategy. The platform plays a key role in driving personalized customer engagement across diverse markets.

Challenges in Future Proofing a Global CMS

- The client was on Strapi v4, which was stable but nearing end-of-life.
- Upcoming feature requirements like live preview, versioning, and better i18n support weren't available in v4.
- Legacy APIs and deprecated plugins added technical debt and integration complexity.
- Staying aligned with Strapi's roadmap was essential to maintain security and compatibility.

QBurst Solution: Seamless Strapi v5 Migration

QBurst executed a structured migration from Strapi v4 to v5, enabling the client to modernize their content platform without affecting day-to-day operations.

The migration induced a more powerful editorial experience and created opportunities for future enhancements.

Key Innovations

- **Live Preview & Unified Document Model:** Editors can preview content with drafts and translations managed in a single document for simpler workflows.
- **Document Service API:** Replaced deprecated EntityService for modern content handling and normalized responses.
- **Flatter API Responses:** Simplified nested data for improved frontend performance and integration.
- **Built-in i18n Support:** Native internationalization eliminates plugin dependency and streamlines translation.
- **Official Strapi Client Library:** Enables reliable and simplified API consumption from frontend applications.
- **Enhanced Developer Experience:** Stronger TypeScript support, Plugin SDK usage, and faster admin UI bundling via Vite.
- **Plugin Modernization:** Replaced unsupported plugins (e.g., SEO, Content Versioning) with compatible alternatives.

Implementation Approach: Continuity & Growth

- Full backup of database, assets, and code before migration
- Compatibility checks on all custom code, plugins, and models
- Official CLI upgrade tool used, followed by manual refactoring
- Adopted incremental migration with compatibility headers, allowing frontends to update progressively without breaking changes
- Refactored custom controllers (e.g., wishlist/account) and CSV import/bulk upsert logic to support documentId-based relations, normalized responses, and updated i18n handling via the Document Service API
- Modernized BFF layer to handle normalized responses and new relation models
- Rigorous testing with real-world CSVs and multi-locale scenarios

Impact: Streamlined and Hassle-free Upgrade

- 60% improvement in API response time for search and listing endpoints
- Zero downtime during migration ensured uninterrupted editorial operations
- Editorial teams gained faster workflows through preview and versioning features, reducing publishing time by up to 30%
- Cleaner backend architecture reduced technical debt and enabled faster deployments
- CMS is now aligned with Strapi's long-term roadmap, ensuring sustainability