Authentication Architecture

Authentication Architecture

Overview

QODY implements multi-role authentication with separate flows for:

- **Staff** — venue-scoped (email + password + venueSlug)
- **SuperAdmin** — cross-venue global admin (email + password)
- **Guest** — anonymous/session-based (no credential auth required)

Security: Rate-Limiting & Brute-Force Protection

Critical Vulnerability Fixed — 2026-07-03 (MC #104585)

Vulnerability

Impact: CRITICAL — Unlimited brute-force / password-spray against owner/admin/SUPERADMIN accounts.

The login rate limiter trusted the client-supplied `X-Forwarded-For` header (first value) to key rate-limit buckets. An attacker could rotate a forged `X-Forwarded-For` value on each request to obtain a fresh rate-limit bucket every time, bypassing all protections.

Affected Endpoints:

- `POST /staff/auth/login`
- `POST /superadmin/auth/login`
- `POST /auth/forgot-password`

Live Exploitation Verified: 2026-07-01 (Securion security review)

> 7 consecutive requests, same email, different spoofed X-Forwarded-For on every request:
> All attempts → HTTP 401 (unlimited, rate limiter never triggered)
> *Evidence:* `~/system/evidence/104554/qody-ratelimit-securion-review-20260701.md`

Fix (Deployed 2026-07-03)

Strategy: Rate-limit on NON-SPOOFABLE dimensions instead of client IP.

New Design:

- **Email-only bucket:** 5 failures per email per 15 minutes (keyed on `email.trim().lowercase()` — no IP component)
- **Scope bucket:** 20 failures per scope per 15 minutes
  - Staff login: keyed on `venueSlug` (per-venue spray protection)
  - SuperAdmin login: keyed on constant `"superadmin-crossvenue"` (global spray protection)
- **Forgot Password:** Email-only bucket (3 requests per 15 minutes, privacy-preserving — always returns HTTP 200 even when rate-limited)

ClientIpResolver.kt: A helper for trusted-last-hop IP resolution was added (`TRUSTED_PROXY_HOPS=1`) but is used ONLY for non-critical post-auth `ApiRateLimiter` IP dimension + audit logs — NOT the login security key. Login security does NOT depend on IP trust.

Implementation:

- `LoginRateLimiter.kt` — complete rewrite
- `StaffRoutes.kt`, `SuperAdminRoutes.kt` — scope-based rate-limit keys
- `PasswordResetService.kt` — email-only key for forgot-password
- `ClientIpResolver.kt` — NEW (non-critical uses only)

Deployment

- **Commit:** `f176357` (security(api): fix XFF-bypass login rate-limiter — MC #104585)
- **Image:** `qodyprodacr.azurecr.io/qody-api:f176357`
- **Revision:** qody-api-prod--0000012 @ 100% traffic
- **Environment:** rg-qody-prod (swedencentral), api.qody.ba
- **Deploy Date:** 2026-07-03

Verification (Live Production Replay)

Test: Replayed XFF-rotation exploit on production `api.qody.ba` with fresh email + rotating forged `X-Forwarded-For` values (same attack that succeeded unlimited pre-fix).

Result (Staff Login):

Attempt	Forged XFF	Result
1	203.0.113.1	HTTP 401
2	203.0.113.2	HTTP 401
3	203.0.113.3	HTTP 401
4	203.0.113.4	HTTP 401
5	203.0.113.5	HTTP 401
6	203.0.113.6	**HTTP 429** — Rate limited

Outcome: Email-based limiter triggered at threshold despite XFF rotation. Fix confirmed live.

Unit Tests: 261 total, 0 failures, including a regression test replaying the exact exploit.

Evidence: `~/system/evidence/104585/qody-xff-fix-deploy-verify-20260703.md`

Follow-Up (Non-Blocking, L-Priority)

- Empirically verify ACA Envoy XFF hop-count to validate/tune the `ClientIpResolver` `TRUSTED_PROXY_HOPS` default (does not affect login security, which is now email+scope keyed).

JWT & Refresh Tokens

(Planned section — JWT structure, rotation policy, HttpOnly cookies, etc.)

Password Hashing

(Planned section — Argon2id parameters, rotation, etc.)

Session Management

(Planned section — Staff/SuperAdmin session lifecycle, logout, etc.)

Revision #1
Created 2026-07-03 11:50:51 UTC by John
Updated 2026-07-03 11:50:52 UTC by John