Finanzas SD runtime validation (2025-11-29)

Finanzas SD – Architecture, Flows & SOPs

Arquitectura, Flujos y Procedimientos

December 14, 2025

1 Finanzas SD runtime validation (2025-11-29)

This note captures an end-to-end check of the live Finanzas stack using the same OIDC-assumed role as the deployment workflow. The goal was to explain why the UI reports HTTP 500s even though deployments are green.

1.1 Identity and scope

- Assumed `arn:aws:iam::703671891952:role/ProjectplaceLambdaRole` via STS; the caller shows as the assumed role in account `703671891952`.94a621†L1-L4

1.2 CloudFront routing

- Distribution **EPQU7PVDLQXUA** contains explicit behaviors for `/finanzas`, `/finanzas/`, and `/finanzas/*`, all pointing to the **finanzas-ui-s3** origin and associating the `finanzas-path-rewrite` viewer-request function.0741af†L1-L9
- Live CloudFront Function code rewrites SPA paths exactly as expected (root, callback passthrough, and extensionless `/finanzas/*` requests route to `index.html`).ab3df6†L1-L22

1.3 S3 deployment contents

- Bucket `ukusi-ui-finanzas-prod` under the `/finanzas/` prefix holds `index.html`, `auth/`, `assets/`, and `docs/latest/` as expected.e5976d†L1-L5,7f9d4f†L1-L9
- Only the latest bundle `index-BADbQQHf.js` and matching CSS exist under `assets/`; no stale JS artifacts remain.6365c0†L1-L3
- The live `index.html` references only `index-BADbQQHf.js`, matching the current asset set.1998f6†L1-L2

1.4 API Gateway + Lambda wiring

- HTTP API `finanzas-sd-api-dev` (ID `pyorjw6lbe`) is active with routes for all Finanzas SD endpoints (projects, changes, catalog, line-items, etc.). Each target is wired to the corresponding Lambda proxy integration (e.g., `/projects` → `ProjectsFn`, `/projects/{projectId}/changes` → `ChangesFn`, `/catalog/rubros` → `CatalogFn`).9d2874†L1-L34,db4296†L1-L11,7d7111†L1-L5,2c6384†L1-L9,1db3b4†L1-L5
- JWT authorizer `CognitoJwt` enforces tokens from pool `us-east-2_FyHLtOhiY` and client `dshos5iou44tuach7ta3ici5m`. A tokenless request returns `401 Unauthorized`, confirming auth is required and enforced.92ee79†L1-L16,e5eac5†L1-L13

1.5 Lambda environment + logs

- Finanzas Lambdas share consistent environment variables: Dynamo tables (`finz_projects`, `finz_changes`, etc.), Cognito pool/client IDs, policy store ID, and allowed origin `https://d7t9x3j66yd8k.cloudfront.net`.【1af284†L1-L66】
- Recent CloudWatch streams for `ProjectsFn` and `ChangesFn` show clean invocations with no error stack traces (only table resolution INFO logs).【fd4aa9†L1-L12】【40e5e1†L1-L13】
- `LineItemsFn`'s latest stream (Nov 21) also contains no errors, suggesting minimal traffic rather than runtime faults.【5e2de4†L1-L8】

1.6 DynamoDB data shape

- All expected tables exist in `us-east-2`, including projects, changes, rubros, allocations, prefacturas, etc.【28b371†L1-L19】
- `finz_projects` items contain both Spanish and English aliases (e.g., `id`/`projectId`, `nombre`/`name`, `cliente`/`client`, `fecha_inicio`/`start_date`), matching the frontend normalization logic in `src/lib/api.ts` and indicating data parity with the UI contract.【4ec48d†L1-L107】
- Querying `finz_changes` for a sample project returned no rows, implying the UI could legitimately show empty change histories rather than throwing server errors.【8beb10†L1-L3】

1.7 Likely cause of "HTTP 500" symptom

- Backend wiring, content deployment, and data stores look healthy with no CloudWatch errors. Direct API calls without an Authorization header return `401`, and the Lambdas log normal traffic when invoked. This points to **missing/expired Cognito tokens from the UI** (or CORS/auth header issues) as the probable reason the UI surfaces generic "500" messages. The frontend's error handling in `src/lib/api.ts` converts fetch failures into generic errors, which could be presented as 500s even when the API is returning 401s.

1.8 Recommendations

1. Reproduce in the browser with the network tab to confirm responses are 401/403 rather than 500 and ensure the Cognito session is present. If tokens are missing, verify Hosted UI callback configuration and local storage handling in the UI auth flow.
2. Improve frontend error messaging to distinguish auth failures from server errors (e.g., map 401/403 to a login prompt).

3. Add CloudWatch logging for authorization failures at the API Gateway stage or enable access logs to capture real client responses for UI troubleshooting.