Runbook — LumisCare "Unable to load your account" (Entra B2B + JIT/DB mapping) — 2026-06-02

Runbook — LumisCare "Unable to load your account" (Entra B2B + JIT/DB user mapping)

Date: 2026-06-02 · **Evidence:** `/tmp/evidence-session-fe5379ce/` · **Env:** ALAI demo (Azure tenant `3454a03f`, `rg-lumiscare-demo`) · **App:** `app.lumiscare.com`, appId `e422174a-24a6-4889-b66c-f6c483b8631f`

Symptom

After login, the backoffice shows **"Unable to load your account — We could not retrieve your user information."** Rendered by `frontend/packages/shared-ui/src/components/layout/AccountInfoLoader.tsx` when the `GET /users/about-me` query (in `useAccountStore.ts`) errors.

Root cause (NOT a password problem)

A Microsoft **Entra B2B / admin-consent / backend user-mapping** problem. A logged-in guest (`alem@alai.no`, Entra OID `f9275cf4-…`) authenticated, but was **not correctly provisioned/mapped** to a LumisCare DB user + org, so `/users/about-me` could not return a usable account.

What is NOT the cause (verified, do not chase again)

- Backend `GET /api/v1/users/about-me` returns **HTTP 200** with a valid shape for a correct token.
- **CORS** is fine — `CorsConfig` uses `allowedHeaders("*")`, `app.lumiscare.com` allowed, credentials true; `OPTIONS` preflight → 200.
- Frontend **env** is correct — `LUMISCARE_CLIENT_ID` / `LUMISCARE_BACKEND_CUSTOMER_CLIENT_ID` = `e422174a`, scope `api://e422174a/api`.
- **Data exists** — `/service-users` = 15, `/hr/employees` = 10 for org `f714cc2f`.
- Deployed tree = `frontend/packages/backoffice` + `shared-ui` (NOT legacy `frontend/web/`).

Fix (live, in order)

1. **Entra B2B invite** (re)issued for `alem@alai.no` — `lumiscare-b2b-invite.json` / `alem-fresh-invite.txt`.
2. **MSAL config confirmed** correct for `app.lumiscare.com` — `uat-login-diagnosis.md`.
3. **Admin consent grant** repaired for the LumisCare app:
   - scope: `api openid profile offline_access`, `consentType=AllPrincipals`
   - Grant ID: `Tucgg8C0XUKt_DONAFF0Tk7nIIPAtF1CrfwzjQBRdE4`
4. **Backend BFF/JIT + DB user mapping** (the decisive step — frontend deploy/JIT merge alone was NOT enough):
   - `alem@alai.no` promoted directly in Postgres: DB id `eccfeb0c-0b12-4439-9b99-33a12f9110c5`, status `ACTIVE`, org `f714cc2f-a0fc-4e47-9164-baa540fd820f` (Sunshine Home Care LLC), role `system_admin`.
   - JWT linkage: Entra OID `f9275cf4-…` → DB user UUID (via JIT).
   - Backend revs: **identity** `--0000008` (issuer URI + audience GUID match), **web-bff** `--0000032` (audience-permissive + `/users` alias), Hibernate `ddl-auto=update` for schema drift.

Proof

- `bff-about-me-success.txt` → `HTTP=200`, org `f714cc2f`, account `eccfeb0c…`, email mapped from OID.
- `alem-user-record-promoted.txt` → promoted DB record.

Key lesson

For a new Entra guest, **a frontend deploy / JIT merge is not sufficient**. A working login requires the backend chain: **Entra B2B membership + admin consent + identity-service JIT + an ACTIVE DB user mapped to an org**. When "Unable to load account" recurs for a new user → check the **DB user + org mapping** and **admin consent**, not the password or the frontend.

Related security debt

- Demo enablement temporarily relaxed identity-service security; track revert + proper app-roles (MC #102747).

Revision #1
Created 2026-06-02 18:57:28 UTC by John
Updated 2026-06-02 18:57:28 UTC by John