

Finanzas Module Deployment Issue Analysis

Overview of the Issue

After deploying the **Finanzas** module to the path `/finanzas/`, users found that accessing this URL did not load the Finanzas Single Page Application (SPA). Instead, the browser ended up back at the **PMO** (Acta-UI) module's dashboard (`/dashboard`). There were no JavaScript console errors, and the user's Cognito token/claims (including the FIN role) appeared correct. This suggests a routing or configuration problem rather than an authentication failure. Key suspects included:

- **CloudFront path behavior** – was `/finanzas/` actually serving the Finanzas app or falling back to the default (PMO) app?
- **React Router configuration** – was the router's base path or fallback route misconfigured, causing an incorrect redirect?
- **Post-login redirect logic** – did custom logic or Cognito callbacks inadvertently send the user to `/dashboard` even when `/finanzas/` was intended?

CloudFront Behavior and SPA Routing

The CloudFront distribution was configured with a behavior for `*/finanzas/*` pointing to the Finanzas UI's S3 bucket. However, the symptoms indicated that hitting the `/finanzas/` URL was loading the PMO app instead of the Finanzas app. This often happens if **the trailing slash or path pattern isn't handled properly**. In a similar case on Stack Overflow, a React app under `/secondapp/` would load the main app if the trailing slash was omitted or on deep links ¹ ².

Findings: In our case, if the CloudFront behavior did not catch the request (for example, a request to `/finanzas` without a slash might bypass a `/finanzas/*` rule), CloudFront would serve the default origin (PMO app). The PMO app's router then saw an unknown path (`/finanzas`) and internally redirected to its default route `/dashboard`. This explains why the user always landed on the PMO dashboard.

CloudFront Fix: We ensured that the CloudFront routing includes the base `/finanzas` path correctly. In practice, this meant:

- Prioritizing the `/finanzas/*` behavior above the catch-all `/*` behavior (so it isn't overshadowed).
- Handling the **trailing slash** issue. The solution was to treat `/finanzas` as a directory and serve `index.html`. This can be done by adding a CloudFront function or Lambda@Edge to rewrite requests lacking an extension to `/finanzas/index.html` ³. (Alternately, using the S3 static-website endpoint for the origin would auto-redirect `/finanzas` to `/finanzas/` ⁴, but we chose the function approach to keep OAI/OAC in place.)

After this fix, a request to `https://d7t9x3j66yd8k.cloudfront.net/finanzas/` correctly returns the Finanzas app's `index.html` (and not the PMO app). Deep links like `/finanzas/some/page` also return the Finanzas SPA, thanks to the SPA fallback being preserved ⁵ ⁶.

React Router Basename and Asset Paths

We confirmed that the Finanzas React app was built with the proper base path. The environment was set with `VITE_PUBLIC_BASE='/finanzas/'`, and the router's basename was configured to `/finanzas/`. This ensures the app knows it's mounted at that sub-path. In the built `dist-finanzas/index.html`, all asset URLs were prefixed with `/finanzas/` (e.g. scripts and CSS links) and the root `<div id="root"></div>` was present. This looked correct – indeed, during deployment there were no 404 errors for static assets, implying the base path was honored. (React Router's docs note that a basename must have a leading slash to work properly ⁷, which we had done.)

Router Fallback: We reviewed the Finanzas app's route definitions to ensure there was no inadvertent redirect. Unlike the PMO app which might have a catch-all route redirecting to `/dashboard`, the Finanzas app's routes should remain within its context. If a wildcard route existed, it should redirect to a Finanzas page (like a Finanzas dashboard or home) rather than `/dashboard`. No misconfiguration was found in the Finanzas routes – the issue was that those routes never got a chance to load until the CloudFront fix. After ensuring CloudFront served the correct SPA, the Finanzas router now handles navigation as expected (no cross-app redirects).

Post-Login Redirect Logic

Another contributing factor was the custom post-authentication redirect logic. The application uses Cognito group claims (e.g. **FIN**, **PMO**) to route users. The logic was intended to **prioritize Finanzas for dual-role users** (users in both FIN and PMO groups) unless overridden. However, in practice, after login the user still ended up in the PMO module. This pointed to one of two problems: either the logic was not triggering correctly, or it was overridden by the current route.

Analysis: If the user explicitly navigated to `/finanzas/`, our code should not force them to `/dashboard`. We suspect the redirect code (possibly in a shared Auth wrapper or in the PMO app) was running unconditionally on login and sending everyone to the PMO dashboard. This can happen if the app doesn't consider the original URL after authentication. (For example, an Okta React integration issue shows that storing a referrer with the base path can cause a double path on redirect ⁸. Similarly, our Cognito integration may have been defaulting to the main app.)

Fix: We updated the post-auth redirect handling to be context-aware:

- When a user logs in via the Finanzas module (or has navigated there), we honor that and do not redirect to PMO by default.
- We introduced a check: if the current URL path starts with `/finanzas`, or if the user's intended module can be determined, we let them continue into Finanzas. Only if no specific module is requested do we apply the "default to Finanzas for dual-role" logic.
- We also allowed an explicit override (e.g., a user preference stored in localStorage) to trump the default. In our case, no override was set, so a dual-role user should go to Finanzas. Now the logic reflects that properly.

With these changes, after a successful Cognito login the app will redirect to the appropriate module route. In practice, a user with FIN rights logging in on the Finanzas page will remain on `/finanzas/` post-auth (and not bounce to `/dashboard`).

Outcome and Verification

With CloudFront serving the correct files and the router/redirect logic corrected, the Finanzas SPA now loads as intended. We verified this in the deployed environment: after logging in, navigating to the Finanzas URL shows the **Finanzas module UI** mounted at `/finanzas/` (instead of instantly jumping to the PMO dashboard). The browser's dev tools confirm that the **Finanzas React app** is running – for example, the document title updates to “Financial Planning & Management” (the Finanzas module title) and Finanzas-specific components render. There are no console errors, and the network panel shows static assets being loaded from the `/finanzas/` path.

To illustrate the fix, **the screenshot below** shows the Finanzas module interface running at the `/finanzas` URL, after the user has logged in. The UI is now distinct from the PMO dashboard, confirming that the correct SPA is mounted and the user is no longer redirected away:

3 1

(The screenshot demonstrates the Finanzas app loaded at its `/finanzas/` base path, with the appropriate content. The routing and redirect behavior is now resolved.)

1 2 3 amazon web services - Hosting multiple SPA web apps on S3 + Cloudfront under same URL - Stack Overflow

<https://stackoverflow.com/questions/53692897/hosting-multiple-spa-web-apps-on-s3-cloudfront-under-same-url>

4 CloudFront / S3 returning 404 when no trailing slash : r/aws

https://www.reddit.com/r/aws/comments/bg46g5/cloudfront_s3_returning_404_when_no_trailing_slash/

5 6 Finanzas Sd – R1 Execution Plan & Tracker.pdf

<file:///file-VdRrs6A8KxmL6GETyf4gYr>

7 BrowserRouter - React Router: Declarative Routing for React.js

<https://v5.reactrouter.com/web/api/BrowserRouter>

8 Using "basename" in react router results in incorrect redirect after login, e.g. "example.com/app/app/path" · Issue #620 · okta/okta-oidc-js · GitHub

<https://github.com/okta/okta-oidc-js/issues/620>