

Redirect Loop Fix - Documentation

Issue

The application was experiencing an infinite redirect loop after login, with repeated calls to:

- `/api/auth/session`
- `/` (home page)
- `/api/auth/error`

Root Cause Analysis

1. Double Redirect Logic

- **Middleware** (`middleware.ts`): Redirected authenticated users from `/` to `/dashboard`
- **Server Component** (`app/page.tsx`): Also redirected authenticated users from `/` to `/dashboard`
- This created a race condition and conflicting redirects

2. Edge Runtime Session Issues

- The `withAuth` middleware runs on the edge runtime
- Session handling in edge contexts can be unreliable
- Token synchronization between edge and server-side creates race conditions

3. NEXTAUTH_URL Configuration

- The `.env` file had `NEXTAUTH_URL="http://localhost:3000"`
- This is incorrect for production deployments
- Must be set to the actual deployment URL in Vercel

4. Unnecessary Complexity

- NextAuth doesn't require custom middleware for basic authentication
- Server-side session checks are more reliable and sufficient

Solution Implemented



Removed Middleware

Action: Deleted `middleware.ts` entirely

Reasoning:

- Server-side authentication checks are already in place in layouts
- The dashboard layout (`app/dashboard/layout.tsx`) properly checks for session and redirects
- The home page (`app/page.tsx`) properly redirects authenticated users
- No middleware is needed for this authentication pattern



Kept Server-Side Authentication

The existing server-side pattern is correct and sufficient:

```
// In app/dashboard/layout.tsx
const session = await getSession(authOptions)
if (!session) {
  redirect("/auth/login")
}

// In app/page.tsx
const session = await getSession(authOptions)
if (session) {
  redirect("/dashboard")
}
```

✓ Environment Variable Configuration

Critical: The `NEXTAUTH_URL` must be set correctly in Vercel:

1. Go to Vercel Project Settings → Environment Variables
2. Set `NEXTAUTH_URL` to your production URL (e.g., `https://your-app.vercel.app`)
3. Redeploy the application

Why This Fix Works

1. Single Source of Truth

- Only server-side components check authentication
- No conflicting redirects between middleware and server components

2. Reliable Session Handling

- Server-side `getSession()` is more reliable than edge middleware
- No token synchronization issues

3. Simplified Flow

User visits "/" → Server checks session → Redirects accordingly
 User visits "/dashboard" → Layout checks session → Allows or redirects to login

4. No Race Conditions

- Single check per route
- No competing redirect logic

Testing the Fix

Expected Behavior After Deploy:

1. Unauthenticated User:

- Visits `/` → Sees home page
- Visits `/dashboard` → Redirected to `/auth/login`
- Logs in → Redirected to `/dashboard`

2. Authenticated User:

- Visits `/` → Redirected to `/dashboard`
- Visits `/dashboard` → Sees dashboard
- Visits `/auth/login` → Can access (to logout/switch accounts)

3. No More Redirect Loops:

- Single redirect per authentication check
- No calls to `/api/auth/error`
- Stable session handling

Deployment Steps

1. **This code is already fixed** - middleware removed

2. **Set NEXTAUTH_URL in Vercel:**

```
NEXTAUTH_URL=https://your-actual-deployment-url.vercel.app
```

3. **Commit and push changes**

4. **Verify the deployment**

5. **Test authentication flow**

Key Learnings

❌ What NOT to Do:

- Don't use `withAuth` middleware if you have server-side session checks
- Don't create competing redirect logic
- Don't run complex session operations on edge runtime
- Don't forget to set `NEXTAUTH_URL` to production URL

✅ Best Practices:

- Use server-side `getSession()` for authentication checks
- Keep authentication logic simple and in one place (layouts)
- Set environment variables correctly for each environment
- Trust NextAuth's built-in flow without custom middleware

Additional Notes

- The `SessionProvider` wrapper in the root layout handles client-side session state
- The server-side checks in layouts handle route protection
- This combination is the recommended NextAuth pattern
- No custom middleware is needed for basic authentication flows

Monitoring

After deployment, monitor for:

- No more redirect loops
- Successful login flows
- Proper dashboard access
- No `/api/auth/error` calls in logs

Fix Applied: October 23, 2025

Status: Ready for deployment