

NextAuth Configuration Fixes

Date: November 3, 2025

Issues Fixed

1. 404 Error on `/api/auth/session`

Problem: The NextAuth session endpoint was returning 404 errors, preventing authentication from working.

Root Causes:

- NextAuth v4.24.11 with Next.js 14.2.28 App Router requires explicit export patterns
- Missing `basePath` configuration in both server and client components
- Missing `dynamic = 'force-dynamic'` export to prevent static generation of auth routes
- SessionProvider was not explicitly configured with the auth `basePath`

Solutions Applied:

a. Updated `app/api/auth/[...nextauth]/route.ts` :

- Changed from `export { handler as GET, handler as POST }` to explicit `export const GET = handler` and `export const POST = handler`
- Added `export const dynamic = 'force-dynamic'` to prevent static optimization
- Added `NextRequest` type import for better type safety

b. Enhanced `lib/auth.ts` :

- Added validation to ensure `NEXTAUTH_SECRET` is set (throws error on startup if missing)
- Added explicit `basePath: '/api/auth'` configuration
- Added `useSecureCookies: process.env.NODE_ENV === 'production'` for proper cookie handling in production (Vercel)

c. Improved `components/auth-provider.tsx` :

- Added explicit `basePath="/api/auth"` prop to SessionProvider
- Added `refetchInterval={5 * 60}` to refresh session every 5 minutes
- Added `refetchOnWindowFocus={true}` to revalidate session when user returns to tab

2. `CLIENT_FETCH_ERROR` with “String did not match expected pattern”

Problem: NextAuth client was receiving malformed responses or incorrect URLs.

Solution:

- Explicit `basePath` configuration ensures NextAuth knows exactly where its endpoints are located
- `useSecureCookies` configuration prevents cookie-related issues in production

3. 405 Error on `_log` Endpoint

Problem: NextAuth’s internal logging endpoint was not properly handling HTTP methods.

Solution:

- The explicit GET and POST exports now ensure all NextAuth endpoints (including internal ones like `_log`) handle the correct HTTP methods

Files Modified

1. `app/api/auth/[...nextauth]/route.ts`

- Explicit handler exports
- Added dynamic export
- Better TypeScript imports

2. `lib/auth.ts`

- Added `NEXTAUTH_SECRET` validation
- Added `basePath` configuration
- Added `useSecureCookies` configuration

3. `components/auth-provider.tsx`

- Added `basePath` prop to `SessionProvider`
- Added session refetch configuration

Environment Variables Required

Ensure these are set in Vercel (or your `.env` file for local development):

```
# Required - Generate with: openssl rand -base64 32
NEXTAUTH_SECRET="your-secret-here"

# Required for production
NEXTAUTH_URL="https://www.servicehere.com"

# Required - Your Neon PostgreSQL connection string
DATABASE_URL="postgresql://..."
```

Verification

To verify the fixes are working:

1. Check Auth Config Endpoint:

`https://www.servicehere.com/api/check-auth-config`

Should return all green checkmarks for environment variables.

2. Check Session Endpoint (after login):

`https://www.servicehere.com/api/auth/session`

Should return session data (if logged in) or null (if not logged in), NOT a 404 error.

3. Test Login Flow:

- Navigate to `/login`
- Enter credentials
- Should successfully authenticate without being redirected back to login

Next Steps

1. **Deploy to Vercel:** Push these changes to trigger a new deployment
2. **Verify Environment Variables:** Ensure all required variables are set in Vercel dashboard
3. **Test Authentication:** Try registering a new user and logging in
4. **Monitor Logs:** Check Vercel logs for any remaining NextAuth errors

Technical Details

NextAuth v4 + App Router Compatibility

Next.js 13+ App Router changed how API routes work. The key differences:

Pages Router (old):

```
// pages/api/auth/[...nextauth].ts
export default NextAuth(authOptions)
```

App Router (new):

```
// app/api/auth/[...nextauth]/route.ts
const handler = NextAuth(authOptions)
export const GET = handler
export const POST = handler
export const dynamic = 'force-dynamic'
```

The App Router requires:

- Explicit named exports for HTTP methods (GET, POST, etc.)
- Route handlers in `route.ts` files
- Manual configuration of dynamic/static behavior
- Explicit `basePath` configuration in both server and client

Why `basePath` is Critical

Without an explicit `basePath`:

- SessionProvider may try to call wrong endpoints
- NextAuth may not correctly generate URLs for callbacks
- CSRF tokens may fail validation
- Session refresh may fail silently


With explicit `basePath="/api/auth"`:

- All auth endpoints are correctly resolved
- Client and server agree on endpoint locations
- Works consistently across development and production

References

- NextAuth.js Documentation: <https://next-auth.js.org/configuration/initialization#route-handlers-app>
- Next.js App Router: <https://nextjs.org/docs/app/building-your-application/routing/route-handlers>
- NextAuth with App Router: <https://next-auth.js.org/configuration/initialization#app-router>

Author: DeepAgent AI

Status:  Fixes Applied - Ready for Deployment