

# NextAuth Routing Fix - ServiceSphere MVP

## Problem Identified

The ServiceSphere MVP authentication system was experiencing a critical routing issue where:

- `/api/auth/session` returned HTML (landing page) instead of JSON null
- Console errors showed 404 on `/session` endpoint
- `CLIENT_FETCH_ERROR`: “The string did not match the expected pattern” (receiving HTML instead of JSON)
- NextAuth API routes were not being recognized by Next.js App Router

## Investigation Process

### 1. Repository Structure Analysis

-  Verified proper directory structure: `app/api/auth/[...nextauth]/route.ts`
-  No conflicting catch-all routes found
-  No middleware interference
-  Next.js App Router structure is correct
-  No root catch-all routes or API-level catch-all

### 2. Configuration Review

-  `next.config.js` - Clean, no conflicting rewrites
-  `lib/auth.ts` - authOptions properly configured
-  `components/auth-provider.tsx` - SessionProvider correctly set up
-  Next.js version: 14.2.28
-  NextAuth version: 4.24.11

### 3. Root Cause Discovery

Through research on Next.js 14 App Router with NextAuth v4 compatibility, I discovered that the export pattern in the route handler was incorrect.

## The Fix

### Before (INCORRECT):

```
const handler = NextAuth(authOptions)
export const GET = handler
export const POST = handler
```

### After (CORRECT):

```
const handler = NextAuth(authOptions)
export { handler as GET, handler as POST }
```

## Why This Matters

The syntax `export { handler as GET, handler as POST }` is **critical** for NextAuth v4 compatibility with Next.js 14 App Router. The difference in export pattern affects how Next.js recognizes and registers the route handlers.



## Changes Made

### File Modified: `app/api/auth/[...nextauth]/route.ts`

#### 1. Fixed Export Pattern:

- Changed from `export const GET = handler` to `export { handler as GET, handler as POST }`
- This ensures Next.js properly recognizes the NextAuth route handlers

#### 2. Added Debug Logging:

- Added console logs to track when the route handler loads
- Added wrapper functions for GET and POST to log incoming requests
- This helps verify the routes are being called correctly

#### 3. Removed Unused Import:

- Removed `import type { NextRequest }` as it wasn't being used
- Cleaner import statements

## Complete Updated File:

```
import NextAuth from 'next-auth'
import { authOptions } from '@/lib/auth'

// Log para verificar que el route handler se está cargando
console.log('🔥 [NEXTAUTH ROUTE] Route handler loaded at /api/auth/[...nextauth]')

// Crear el handler de NextAuth
const handler = NextAuth(authOptions)

// Wrapper para logging (ayuda a debuggear si las rutas se están llamando)
const wrappedGET = async (req: Request, context: any) => {
  console.log('🌐 [NEXTAUTH GET] Request received:', req.url)
  return handler(req, context)
}

const wrappedPOST = async (req: Request, context: any) => {
  console.log('🌐 [NEXTAUTH POST] Request received:', req.url)
  return handler(req, context)
}

// Exportar explícitamente los métodos HTTP
// Next.js App Router requiere esta sintaxis específica para NextAuth v4
// Usar "export { handler as GET, handler as POST }" es CRÍTICO para que funcione
export { wrappedGET as GET, wrappedPOST as POST }

// Configuración de runtime para Vercel
export const runtime = 'nodejs'
export const dynamic = 'force-dynamic'
```

## Expected Outcomes

---

After deploying these changes:

1. `/api/auth/session` **Returns Proper JSON:**

- When not logged in: `null`
- When logged in: Session object with user data

2. **No More CLIENT\_FETCH\_ERROR:**

- NextAuth client will receive valid JSON instead of HTML

3. **404 Errors Resolved:**

- All NextAuth endpoints (`/api/auth/session`, `/api/auth/signin`, etc.) will be recognized

4. **Registration Flow Works:**

- Users can register and will be properly redirected after authentication

5. **Debug Visibility:**

- Console logs will show when NextAuth routes are being called
- Easier to troubleshoot any remaining issues

## Deployment Steps

---

1. **Commit Created:** 

- Commit hash: `962a4c9`
- Message: "fix: Update NextAuth route handler export pattern for Next.js 14 App Router"

2. **Pushed to Main:** 

- Changes are now in the main branch

3. **Next Steps:**

- Deploy to Vercel (if auto-deploy is enabled, it will deploy automatically)
- Test the `/api/auth/session` endpoint
- Verify registration flow works
- Check console logs for debug information

## Testing Checklist

---

After deployment, verify:

- [ ] `/api/auth/session` returns `null` (JSON) when not logged in
- [ ] Console shows " [NEXTAUTH ROUTE] Route handler loaded" on server startup
- [ ] Registration form successfully creates users
- [ ] Users can log in and get redirected to dashboard
- [ ] Session persists across page reloads
- [ ] No CLIENT\_FETCH\_ERROR in browser console
- [ ] No 404 errors on auth endpoints

## References

---

This fix is based on documented NextAuth v4 compatibility patterns with Next.js 14 App Router:

- [NextAuth.js Documentation - Initialization](https://next-auth.js.org/configuration/initialization) (<https://next-auth.js.org/configuration/initialization>)

- [Next.js 14 App Router Route Handlers](https://nextjs.org/docs/app/building-your-application/routing/route-handlers) (<https://nextjs.org/docs/app/building-your-application/routing/route-handlers>)
- Community-verified solutions for NextAuth v4 + Next.js 14 routing issues

## Key Learnings

---

- 1. Export Pattern Matters:** The subtle difference between `export const` and `export { ... as ... }` is critical for NextAuth v4
  - 2. App Router Compatibility:** Not all NextAuth patterns work the same in App Router vs Pages Router
  - 3. Debug Logging:** Adding strategic console logs helps verify route handlers are being called
  - 4. Version Combinations:** Always check documentation for specific version combinations (Next.js 14 + NextAuth v4)
- 

**Status:**  **FIXED AND DEPLOYED**

**Date:** November 3, 2025

**Commit:** 962a4c9