

# Comprehensive Authentication Diagnostic Report

**Date:** November 4, 2025

**Repository:** <https://github.com/Damadruda/servicephere-mvp.git>

**Branch:** main

**Latest Commit:** 0bce90a - "fix: resolve NextAuth session 404 error with proper handler exports and trustHost config"



## Executive Summary

This report provides a comprehensive analysis of authentication and API configuration issues affecting the servicephere-mvp application deployed on Vercel at [www.servicephere.com](http://www.servicephere.com).

### Critical Issues Identified:

1. **RESOLVED:** Duplicate `export const dynamic` declarations (fixed in commit e780aa7)
2. **ACTIVE:** NextAuth /api/auth/session endpoint returning 404 in production
3. **ACTIVE:** CLIENT\_FETCH\_ERROR - JSON parsing issues (receiving HTML instead of JSON)
4. **ACTIVE:** 405 Method Not Allowed on /api/auth/\_log endpoint



## Technical Analysis

### 1. NextAuth Route Configuration

**Location:** `app/api/auth/[...nextauth]/route.ts`

#### Current Implementation:

- Correct file path structure
- Proper Next.js 14 App Router pattern
- Named exports: `export { handler as GET, handler as POST }`
- Runtime configuration: `runtime = 'nodejs'`, `dynamic = 'force-dynamic'`
- TrustHost enabled for custom domains

#### Code Structure:

```
const handler = NextAuth(authOptions as NextAuthOptions)
export { handler as GET, handler as POST }
export const runtime = 'nodejs'
export const dynamic = 'force-dynamic'
```

### 2. Auth Configuration

**Location:** `lib/auth.ts`

#### Configuration Status:

- Credentials provider properly configured

- JWT strategy enabled
- Session callbacks implemented
- trustHost: true set (CRITICAL for Vercel custom domains)
- Cookie configuration appropriate for production
- Secure cookies enabled for production
- Error handling and logging implemented

### 3. AuthProvider Setup

**Location:** components/auth-provider.tsx

**Implementation:**

- SessionProvider properly wrapped
- basePath="/api/auth" correctly set
- Client-side only rendering
- Session refetch configured (5 minutes)

### 4. Root Layout

**Location:** app/layout.tsx

**Configuration:**

- AuthProvider properly integrated
- No conflicting providers
- Proper component hierarchy

### 5. Build Errors - Duplicate Exports

**Status:**  RESOLVED (commit e780aa7)

Previously affected files (now fixed):

- app/api/chat/analytics/route.ts
- app/api/contracts/[id]/route.ts
- app/api/contracts/[id]/sign/route.ts
- app/api/contracts/create/route.ts
- app/api/contracts/my-contracts/route.ts

### 6. Environment Variables

**Configured in Vercel:**

- NEXTAUTH\_URL = https://www.servicephere.com
- NEXTAUTH\_SECRET = [32-character secret]
- DATABASE\_URL = [Neon PostgreSQL connection]
- DIRECT\_URL = [Supabase PostgreSQL connection]

### 7. Conflicting Auth Files

**Check Results:**

- No pages/api/auth/ directory found
- No conflicting Pages Router authentication
- Pure App Router implementation

## 8. Middleware Configuration

### Status:

- No middleware.ts file present
- No route-blocking middleware

## 9. Next.js Configuration

File: `next.config.js`

### Configuration:

- Standard configuration
- ESLint ignored during builds
- TypeScript errors NOT ignored (proper validation)
- Images unoptimized

## 10. Vercel Configuration

File: `vercel.json`

### Configuration:

- Framework set to "nextjs"
- Standard build command
- No custom routing rules
- No conflicting configurations



## Root Cause Analysis

### Primary Issue: 404 on /api/auth/session

#### Symptoms:

1. GET request to `/api/auth/session` returns 404
2. Response is HTML (site's 404 page) instead of JSON
3. Client receives `CLIENT_FETCH_ERROR: Unexpected token '<', "<!DOCTYPE ... is not valid JSON`

#### Potential Causes:

##### A. Route Not Being Built

The catch-all route `[...nextauth]` might not be properly registered during build time.

#### Evidence:

- Route file exists and is correctly structured
- Build succeeds without route-related errors
- Other API routes work correctly

##### B. Vercel Routing Issue

Vercel might not be correctly routing requests to the catch-all route.

#### Evidence:

- Custom domain ([www.servicephere.com](http://www.servicephere.com)) is being used
- `trustHost: true` is set in authOptions

## C. Build Output Issue

The route might not be included in the production build output.

### Evidence:

- Need to verify build output includes the route
- Need to check if route is prerendered or dynamic

## D. Next.js Version Compatibility

NextAuth 4.24.11 with Next.js 14.2.28 might have compatibility issues with the current export pattern.

### Evidence:

- Using NextAuth 4.24.11 (latest v4)
- Using Next.js 14.2.28
- App Router is relatively new feature

## 🎯 Recommended Fixes

### Fix 1: Ensure Route is Properly Exported (CRITICAL)

The current route structure might have an issue with how Next.js 14.2+ handles route exports. We should ensure the route is using the most compatible pattern.

#### Current:

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

#### Recommended Alternative 1 - Direct Initialization:

```
import { handlers } from '@/auth'
export const { GET, POST } = handlers
```

#### Recommended Alternative 2 - Direct Export:

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

### Fix 2: Add Route Segment Config at Top

Ensure route segment configuration is at the very top of exports:

```
export const runtime = 'nodejs'
export const dynamic = 'force-dynamic'
export const revalidate = 0

// Then handler exports
```

### Fix 3: Create Standalone Auth Module

Create a separate `auth.ts` file that exports the handlers:

**File:** auth.ts (root directory)

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

const { handlers, auth, signIn, signOut } = NextAuth(authOptions)

export { handlers, auth, signIn, signOut }
```

**File:** app/api/auth/[...nextauth]/route.ts

```
export { GET, POST } from '@/auth'
export const runtime = 'nodejs'
export const dynamic = 'force-dynamic'
```

## Fix 4: Add Explicit Route Configuration

Add a `route.ts` in the auth directory to ensure proper registration:

```
// app/api/auth/route.ts
import { NextResponse } from 'next/server'

export async function GET() {
  return NextResponse.redirect('/api/auth/signin')
}
```

## Fix 5: Verify Vercel Build Settings

Ensure Vercel is building with correct settings:

- Node.js version: 20.x
- Build command: `next build`
- Output directory: `.next`
- Install command: `npm install`

## Fix 6: Add Debugging to Route File

Add comprehensive logging to verify route is being loaded:

```
// Log at module level (runs once during build/start)
console.log('[ROUTE INIT] NextAuth route handler initializing...')
console.log('[ROUTE INIT] File: app/api/auth/[...nextauth]/route.ts')
console.log('[ROUTE INIT] NEXTAUTH_URL:', process.env.NEXTAUTH_URL)

// Add response logging in handler
```

## Implementation Plan

### Phase 1: Critical Fixes (Immediate)

1.  Refactor NextAuth route to use most compatible export pattern
2.  Create standalone auth module for better separation
3.  Add comprehensive logging and debugging

4.  Ensure route segment config is properly ordered

## Phase 2: Verification (Deploy & Test)

1. Deploy to Vercel
2. Test /api/auth/session endpoint
3. Test /api/auth/diagnostic endpoint
4. Verify authentication flow works end-to-end

## Phase 3: Monitoring (Post-Deploy)

1. Monitor Vercel logs for errors
  2. Check Next.js build output
  3. Verify all NextAuth endpoints respond correctly
  4. Monitor CLIENT\_FETCH\_ERROR occurrences
- 

## Testing Checklist

After implementing fixes, verify:

- [ ] /api/auth/session returns JSON (not HTML 404)
  - [ ] /api/auth/providers responds correctly
  - [ ] /api/auth/csrf returns CSRF token
  - [ ] /api/auth/signin page loads
  - [ ] /api/auth/signout works correctly
  - [ ] /api/auth/diagnostic shows all config as valid
  - [ ] No CLIENT\_FETCH\_ERROR in browser console
  - [ ] No 404 errors in Vercel logs
  - [ ] Authentication flow completes successfully
  - [ ] Session persists across page reloads
- 

## Dependencies Analysis

### Current Versions:

- next: 14.2.28
- next-auth: 4.24.11
- @prisma/client: 6.7.0
- @next-auth/prisma-adapter: 1.0.7

### Compatibility:

- Next.js 14.2.28 supports App Router
- NextAuth 4.24.11 supports Next.js 14
- NextAuth v5 (Auth.js) has better Next.js 14+ support

### Recommendation:

- Current stack should work
  - Consider upgrading to NextAuth v5 (Auth.js) in future for better App Router integration
-

## Security Considerations

---

-  NEXTAUTH\_SECRET is properly configured (32+ characters)
  -  Secure cookies enabled in production
  -  httpOnly flag set on session tokens
  -  SameSite policy configured
  -  trustHost enabled for custom domain
  -  CSRF protection enabled by default
  -  Password hashing with bcrypt
- 

## References

---

- [NextAuth.js Documentation](https://next-auth.js.org/) (<https://next-auth.js.org/>)
  - [Next.js 14 App Router Documentation](https://nextjs.org/docs/app) (<https://nextjs.org/docs/app>)
  - [NextAuth with Next.js 14 App Router](https://next-auth.js.org/configuration/initialization#route-handlers-app) (<https://next-auth.js.org/configuration/initialization#route-handlers-app>)
  - [Vercel Next.js Deployment](https://vercel.com/docs/frameworks/nextjs) (<https://vercel.com/docs/frameworks/nextjs>)
- 

## Notes

---

1. The duplicate export errors have already been fixed in the codebase
  2. Main issue is runtime 404 on NextAuth endpoints
  3. All configuration appears correct on paper
  4. Issue is likely related to how the route is being built/deployed
  5. The recommended fixes focus on ensuring maximum compatibility with Next.js 14.2+ and Vercel deployment
- 

**Report Generated:** November 4, 2025

**Analyst:** DeepAgent AI

**Status:** Ready for Implementation