

Sentry Debug Guide - Comprehensive Troubleshooting

Overview

This guide documents the comprehensive Sentry debugging implementation for the CareLinkAI project on Render.





Date Created: January 2, 2026

Status: Debugging Phase

Environment: Production (Render)

Problem Analysis

Current Situation

-  Server-side Sentry shows: "Server-side initialization successful"
-  Edge Sentry shows: "Edge initialization successful"
-  Client-side shows: "Not running in browser environment"
-  No events appearing in Sentry dashboard

Root Cause Hypothesis

Most Likely Issue: Missing `NEXT_PUBLIC_SENTRY_DSN`

The client-side message "Not running in browser environment" suggests that the client-side Sentry is not initializing in the browser. This happens when:

1. `NEXT_PUBLIC_SENTRY_DSN` is not set in Render environment variables
2. The environment variable is set but not being read correctly
3. The client-side code is running during SSR instead of in the browser

Required Environment Variables

Critical Variables for Render

Add these to your Render dashboard under "Environment" section:

```
# Option 1: Use NEXT_PUBLIC_SENTRY_DSN (Recommended for Next.js)
NEXT_PUBLIC_SENTRY_DSN=https://your-sentry-dsn@sentry.io/project-id

# Option 2: Use SENTRY_DSN (Server/Edge only)
SENTRY_DSN=https://your-sentry-dsn@sentry.io/project-id

# Best Practice: Set BOTH for maximum compatibility
NEXT_PUBLIC_SENTRY_DSN=https://your-sentry-dsn@sentry.io/project-id
SENTRY_DSN=https://your-sentry-dsn@sentry.io/project-id
```

Key Differences

Variable	Available In	Usage
<code>SENTRY_DSN</code>	Server & Edge Runtime Only	Not available in browser
<code>NEXT_PUBLIC_SENTRY_DSN</code>	Server, Edge, AND Browser	Available everywhere (Next.js bundles it)

Why `NEXT_PUBLIC_` Prefix?

In Next.js:

- Variables WITHOUT `NEXT_PUBLIC_` are server-side only
- Variables WITH `NEXT_PUBLIC_` are embedded in the browser bundle
- Client-side Sentry MUST use `NEXT_PUBLIC_SENTRY_DSN`



Debug Logging Added

What We Changed

We added comprehensive debug logging to all three Sentry config files:

1. `sentry.client.config.ts` - Client-side initialization
2. `sentry.server.config.ts` - Server-side initialization
3. `sentry.edge.config.ts` - Edge runtime initialization

What to Look For in Logs


Good Client-Side Logs (When Working)

```
[Sentry Debug] ===== CLIENT CONFIG =====
[Sentry Debug] Browser Environment: true
[Sentry Debug] NEXT_PUBLIC_SENTRY_DSN exists: true
[Sentry Debug] NEXT_PUBLIC_SENTRY_DSN length: 89
[Sentry Debug] NODE_ENV: production
[Sentry Debug] Sentry already initialized: false
[Sentry Debug] Window object exists: true
[Sentry Debug] Document object exists: true
[Sentry Debug] =====
[Sentry Debug] Attempting to initialize client-side Sentry...
[Sentry] ✅ Client-side initialization successful
[Sentry Debug] Sentry.isInitialized(): true
```

Bad Client-Side Logs (Current Issue)

```
[Sentry Debug] ===== CLIENT CONFIG =====
[Sentry Debug] Browser Environment: false
[Sentry Debug] NEXT_PUBLIC_SENTRY_DSN exists: false ❌ PROBLEM!
[Sentry Debug] NEXT_PUBLIC_SENTRY_DSN length: 0
[Sentry Debug] NODE_ENV: production
[Sentry Debug] Sentry already initialized: N/A (not in browser)
[Sentry] ⚠️ Not running in browser environment (this is normal during SSR)
[Sentry] ❌ CRITICAL: NEXT_PUBLIC_SENTRY_DSN is not set
```

Good Server-Side Logs

```
[Sentry Debug] ===== SERVER CONFIG =====
[Sentry Debug] SENTRY_DSN exists: true
[Sentry Debug] NEXT_PUBLIC_SENTRY_DSN exists: true
[Sentry Debug] Using DSN: FOUND
[Sentry Debug] DSN length: 89
[Sentry Debug] NODE_ENV: production
[Sentry Debug] Sentry already initialized: false
[Sentry Debug] =====
[Sentry Debug] Attempting to initialize server-side Sentry...
[Sentry]  Server-side initialization successful
[Sentry Debug] Sentry.isInitialized(): true
```



Testing Tools Created

1. Server-Side Test Endpoint

URL: `/api/test-sentry-server`

What it tests:

- Server-side Sentry initialization
- Error capture
- Message capture
- Breadcrumbs
- Context/tags

How to use:

```
# From terminal
curl https://your-app.onrender.com/api/test-sentry-server

# Or visit in browser
https://your-app.onrender.com/api/test-sentry-server
```

Expected response:

```
{
  "success": true,
  "message": "Server-side Sentry tests completed",
  "details": {
    "sentryInitialized": true,
    "messageId": "abc123...",
    "errorId": "def456...",
    "timestamp": "2026-01-02T..."
  }
}
```

2. Edge Runtime Test Endpoint

URL: `/api/test-sentry-edge`

What it tests:

- Edge runtime Sentry initialization

- Error capture in edge environment
- Message capture in edge environment

How to use:

Same as server-side endpoint, just different URL.





3. Client-Side Test Page

URL: `/test-sentry-client`

What it tests:

- Client-side Sentry initialization
- Browser error capture
- Multiple error types (sync, async, throw)
- Context and breadcrumbs

Features:

-  Visual status indicators
-  Real-time logging
-  Multiple test scenarios
-  Clear warnings if Sentry not initialized


How to use:

1. Visit `https://your-app.onrender.com/test-sentry-client`
2. Click test buttons
3. Check browser console (F12)
4. Verify events in Sentry dashboard

4. Floating Debug Button

Location: Operator dashboard (bottom-right corner)

Features:

-  Purple floating button
- Quick access to all test functions
- Only shown in development or when `NEXT_PUBLIC_SHOW_DEBUG_TOOLS=true`

How to enable in production:

```
# Add to Render environment variables
NEXT_PUBLIC_SHOW_DEBUG_TOOLS=true
```



Step-by-Step Debugging Process

Step 1: Check Environment Variables in Render

1. Go to Render dashboard
2. Select your service
3. Go to "Environment" tab
4. Verify these variables exist:
 - `[] NEXT_PUBLIC_SENTRY_DSN`
 - `[] SENTRY_DSN` (optional but recommended)

Step 2: Check Render Logs

1. Go to Render dashboard → Logs
2. Look for Sentry Debug messages during startup
3. Check for three sections:

```
- [ ] ===== CLIENT CONFIG =====
- [ ] ===== SERVER CONFIG =====
- [ ] ===== EDGE CONFIG =====
```

Step 3: Verify Client-Side Initialization

What to check:

```
[Sentry Debug] NEXT_PUBLIC_SENTRY_DSN exists: true/false
```

- If `false` : Variable not set in Render
- If `true` : Check if `Sentry.isInitialized()` is true

Step 4: Test Each Environment

Test Server-Side

```
curl https://your-app.onrender.com/api/test-sentry-server
```

Check Render logs for:

- [Sentry Test] All server-side tests completed!
- [Sentry Debug] Server beforeSend called

Test Edge Runtime

```
curl https://your-app.onrender.com/api/test-sentry-edge
```

Check Render logs for similar messages.

Test Client-Side

1. Visit `/test-sentry-client` in browser
2. Open DevTools console (F12)
3. Click “Test Message” button
4. Look for console logs:
 - [Sentry Test Client] Message ID: ...
 - Browser network tab shows requests to Sentry

Step 5: Verify Events in Sentry Dashboard

1. Log into Sentry
2. Go to Issues → All Issues
3. Look for events with 🛠 emoji
4. Should see:
 - Server-side test error
 - Edge runtime test error
 - Client-side test error
 - Test messages

Timeline: Events should appear within 1-2 minutes

Troubleshooting Common Issues

Issue 1: Client-Side Not Initializing

Symptoms:

```
[Sentry Debug] NEXT_PUBLIC_SENTRY_DSN exists: false
[Sentry] ❌ CRITICAL: NEXT_PUBLIC_SENTRY_DSN is not set
```

Solution:

1. Add `NEXT_PUBLIC_SENTRY_DSN` to Render environment variables
2. **Important:** Rebuild the application (not just restart)
3. Next.js bundles environment variables at build time
4. Restart alone won't pick up new `NEXT_PUBLIC_` variables

Steps:

```
# In Render:
1. Environment → Add Variable
2. Name: NEXT_PUBLIC_SENTRY_DSN
3. Value: https://your-sentry-dsn@sentry.io/project-id
4. Save
5. Manual Deploy → Clear build cache & deploy
```

Issue 2: Events Sent But Not Appearing

Symptoms:

- Logs show `beforeSend` called
- Logs show event IDs
- No events in Sentry dashboard

Possible Causes:

1. **Wrong DSN:** Check DSN matches your Sentry project
2. **Sentry project disabled:** Check project status in Sentry
3. **Rate limiting:** Check Sentry quotas
4. **Filtered out:** Check `beforeSend` filters

Solution:

```
// Temporarily disable all filters
beforeSend(event, hint) {
  console.log('[Sentry Debug] beforeSend called - event will be sent');
  return event; // Send everything
}
```

Issue 3: Debug Mode Not Working

Symptoms:

- No detailed Sentry logs
- Can't see SDK messages

Solution:

We already set `debug: true` in all configs. If still not seeing logs:

1. Check browser console (F12) for client-side
2. Check Render logs for server-side
3. Ensure you're looking at the right environment

Issue 4: Source Maps Not Uploading**Symptoms:**

- Events appear but stack traces are minified
- Can't read error locations

Note: This doesn't prevent events from appearing. If events aren't appearing at all, source maps are NOT the issue.




Solution (if needed later):

```
# Add to Render environment variables
SENTRY_AUTH_TOKEN=your-auth-token
SENTRY_ORG=your-org
SENTRY_PROJECT=your-project
```


✓ Success Criteria Checklist**Environment Setup**

- [] `NEXT_PUBLIC_SENTRY_DSN` set in Render
- [] `SENTRY_DSN` set in Render (optional)
- [] Application rebuilt (not just restarted)
- [] Environment variables verified in Render logs




Initialization

- [] Client-side: `NEXT_PUBLIC_SENTRY_DSN exists: true`
- [] Client-side:  Client-side initialization successful
- [] Server-side:  Server-side initialization successful
- [] Edge:  Edge initialization successful

Testing

- [] `/api/test-sentry-server` returns success
- [] `/api/test-sentry-edge` returns success
- [] `/test-sentry-client` shows "Initialized: Yes 
- [] Test buttons work without errors

Sentry Dashboard

- [] Server test event appears (with )
- [] Edge test event appears (with )
- [] Client test event appears (with )
- [] Events include breadcrumbs and context

- [] Real errors (if any) are being captured

Deployment Instructions

1. Commit Changes

```
cd /home/ubuntu/carelinkai-project
git add .
git commit -m "debug: Add comprehensive Sentry debugging and test endpoints"
```

- Added detailed environment variable logging to all Sentry configs
- Enabled debug mode in all configs (temporary for troubleshooting)
- Created test endpoints: /api/test-sentry-server, /api/test-sentry-edge
- Created client test page: /test-sentry-client
- Added floating debug button to operator dashboard
- Comprehensive documentation in SENTRY_DEBUG_GUIDE.md

This will help identify why Sentry events aren't appearing in dashboard.
Focus on client-side initialization issue."

2. Push to GitHub

```
git push origin main
```

3. Configure Render Environment

Go to Render → Your Service → Environment:

```
# Add these variables:
NEXT_PUBLIC_SENTRY_DSN=<your-sentry-dsn>
SENTRY_DSN=<your-sentry-dsn>
```

Get your DSN from:

1. Log into Sentry
2. Settings → Projects → Your Project
3. Client Keys (DSN)
4. Copy the DSN (looks like `https://abc123@o123456.ingest.sentry.io/123456`)

4. Deploy with Build Cache Clear

Important: Must clear build cache because `NEXT_PUBLIC_` variables are bundled at build time!

```
# In Render:
1. Go to Manual Deploy
2. Check "Clear build cache"
3. Click "Deploy latest commit"
```

5. Monitor Deployment

Watch Render logs during deployment for:


```
[Sentry Debug] ===== CLIENT CONFIG =====
[Sentry Debug] NEXT_PUBLIC_SENTRY_DSN exists: true ❌ Should be TRUE!
```

6. Test After Deployment

```
# Test server
curl https://your-app.onrender.com/api/test-sentry-server

# Test edge
curl https://your-app.onrender.com/api/test-sentry-edge

# Test client (in browser)
https://your-app.onrender.com/test-sentry-client
```

7. Verify in Sentry

Within 1-2 minutes, you should see 9 test events in Sentry:

- 3 from server endpoint
- 3 from edge endpoint
- 3 from client page (when you click buttons)

Expected Outcome

After following this guide, you should see:

In Render Logs

```
[Sentry Debug] ===== CLIENT CONFIG =====
[Sentry Debug] Browser Environment: true
[Sentry Debug] NEXT_PUBLIC_SENTRY_DSN exists: true ✅
[Sentry] ✅ Client-side initialization successful

[Sentry Debug] ===== SERVER CONFIG =====
[Sentry Debug] SENTRY_DSN exists: true ✅
[Sentry] ✅ Server-side initialization successful

[Sentry Debug] ===== EDGE CONFIG =====
[Sentry Debug] SENTRY_DSN exists: true ✅
[Sentry] ✅ Edge initialization successful
```

In Sentry Dashboard

- Test events appearing within 1-2 minutes
- Real production errors being captured
- Full stack traces with context
- Breadcrumbs attached to events

Next Steps After Debugging

Once Confirmed Working

1. Disable Debug Mode:

```
typescript
// Change in all three config files:
debug: ENVIRONMENT === 'development', // Back to conditional
```

2. Remove Test Button (Optional):

```
typescript
// Or just remove NEXT_PUBLIC_SHOW_DEBUG_TOOLS from Render
```

3. Keep Test Endpoints:

- They're useful for monitoring
- Can be restricted to admin users if needed

4. Monitor Production:

- Check Sentry regularly for real errors
- Set up alerts for critical issues
- Review error trends weekly

Support & Resources

Internal Resources

- `/test-sentry-client` - Full test page
- `/api/test-sentry-server` - Server test endpoint
- `/api/test-sentry-edge` - Edge test endpoint
- Floating debug button - Quick tests

External Resources

- [Sentry Next.js Docs](https://docs.sentry.io/platforms/javascript/guides/nextjs/) (https://docs.sentry.io/platforms/javascript/guides/nextjs/)
- [Next.js Environment Variables](https://nextjs.org/docs/basic-features/environment-variables) (https://nextjs.org/docs/basic-features/environment-variables)
- [Render Environment Variables](https://render.com/docs/environment-variables) (https://render.com/docs/environment-variables)

Common Commands

```
# Check current Sentry status
curl https://your-app.onrender.com/api/test-sentry-server | jq

# View Render logs
render logs --tail

# Check environment variables
render env list
```

Notes

- **Debug mode is TEMPORARY:** Remember to disable after confirming it works

- **Test endpoints are safe:** They don't expose sensitive data
 - **Client-side is the issue:** Focus on `NEXT_PUBLIC_SENTRY_DSN`
 - **Rebuild is critical:** Restart alone won't work for client-side variables
-

Last Updated: January 2, 2026

Version: 1.0

Status: Ready for deployment and testing