

Sentry Test Instructions

Purpose

This document provides step-by-step instructions for testing the Sentry integration after the caching fix and debug enhancements.

What Was Fixed

1. Caching Issue Fixed

- All test endpoints now include Cache-Control headers
- Each test generates a unique test run ID
- No more cached responses - every request is fresh

2. Sampling Increased to 100%

- Server config: `tracesSampleRate: 1.0` , `profilesSampleRate: 1.0`
- Client config: `tracesSampleRate: 1.0` , `profilesSampleRate: 1.0`
- Edge config: `tracesSampleRate: 1.0`
- All events are now captured (not sampled out)

3. Enhanced Debug Logging

- Detailed `beforeSend` logging in all configs
- Shows event ID, type, level, message, timestamp
- Displays exception details, tags, and extra data
- Logs whether event is sent or suppressed

Test Endpoints

1. Server-Side Test

Endpoint: `https://getcarelinkai.com/api/test-sentry-server`

What it tests:

- Server-side Sentry initialization
- Message capture
- Error capture
- Breadcrumb tracking
- Context tagging

How to test:

```
# Using curl (bypasses browser cache)
curl -v https://getcarelinkai.com/api/test-sentry-server

# Or visit in browser (after clearing cache)
https://getcarelinkai.com/api/test-sentry-server
```

Expected response:

```
{
  "success": true,
  "message": "Server-side Sentry tests completed",
  "testRunId": "test-1704329845123-abc123",
  "details": {
    "sentryInitialized": true,
    "messageId": "...",
    "errorId": "...",
    "timestamp": "2026-01-03T...",
    "instructions": [...]
  }
}
```

What to check in Render logs:

```
[Sentry Test] ===== SERVER TEST =====
[Sentry Test] Test Run ID: test-1704329845123-abc123
[Sentry Test] Timestamp: 2026-01-03T...
[Sentry Test] Sentry.isInitialized(): true
[Sentry Test] Message captured with ID: ...
[Sentry Test] Error captured with ID: ...
[Sentry Test] ✅ Events flushed to Sentry

[Sentry beforeSend] ===== EVENT BEING SENT =====
[Sentry beforeSend] Event ID: ...
[Sentry beforeSend] Event Type: ...
[Sentry beforeSend] ✅ Event will be sent to Sentry
```

2. Server-Side Error Test

Endpoint: `https://getcarelinkai.com/api/test-sentry-server-error`**What it tests:**

- Intentional error throwing
- Error capture with context
- Error tracking through beforeSend

How to test:

```
curl -v https://getcarelinkai.com/api/test-sentry-server-error
```

Expected response: 500 error with test run ID

3. Edge Runtime Test

Endpoint: `https://getcarelinkai.com/api/test-sentry-edge`**What it tests:**

- Edge runtime Sentry initialization
- Message and error capture in edge environment

How to test:

```
curl -v https://getcarelinkai.com/api/test-sentry-edge
```

4. Client-Side Test

Endpoint: `https://getcarelinkai.com/api/test-sentry-client-error`

What it tests:

- Client-side error tracking
- Browser-based Sentry integration

How to test:

1. Visit: `https://getcarelinkai.com/api/test-sentry-client-error`
 2. Click "Throw Test Error" button
 3. Check browser console for Sentry logs
 4. Check Sentry dashboard for the error
-



Testing Checklist

Before Testing

- ☐ Verify Render deployment is complete
- ☐ Check Render environment variables include `NEXT_PUBLIC_SENTRY_DSN`
- ☐ Clear browser cache (Ctrl+Shift+Delete)

Test Execution

- ☐ Test 1: Server-side endpoint via curl
- ☐ Verify unique test run ID in response
- ☐ Check Render logs for detailed output
- ☐ Verify no caching (different ID each time)
- ☐ Test 2: Server-side error endpoint
- ☐ Verify error is captured
- ☐ Check test run ID is unique
- ☐ Test 3: Edge runtime endpoint
- ☐ Verify edge environment works
- ☐ Check unique test run ID
- ☐ Test 4: Client-side error page
- ☐ Visit page in browser
- ☐ Click error button
- ☐ Check browser console
- ☐ Verify no caching (hard refresh)

Sentry Dashboard Verification

Wait 2-5 minutes after running tests, then check:

- [] Visit: <https://sentry.io/organizations/carelinkai/issues/>
- [] Look for issues with 🩺 emoji
- [] Verify test run IDs are present
- [] Check event details include:
- [] Correct timestamps
- [] Tags (test_type, test_run_id)
- [] Extra data (testRunId, timestamp)
- [] Breadcrumbs



Debugging Tips

If Events Don't Appear in Sentry

1. Check Render Logs

- Look for "[Sentry Test]" messages
- Verify `Sentry.isInitialized()` is true
- Check for "[Sentry beforeSend]" events
- Verify "✅ Event will be sent to Sentry"

2. Check Tunnel Status

```
bash
```

```
curl -v https://getcarelinkai.com/api/sentry-tunnel
```

Should return: 405 Method Not Allowed (normal - needs POST)

3. Verify DSN Configuration

- Check Render environment variables
- Ensure `NEXT_PUBLIC_SENTRY_DSN` is set
- Verify DSN format: `https://...@...sentry.io/...`

4. Check Sampling

- All configs should have 1.0 (100%) sampling
- Check Render logs for sampling decisions

If Caching Still Occurs

1. Clear Browser Cache

- Hard refresh: `Ctrl+Shift+R` (or `Cmd+Shift+R`)
- Clear cache: `Ctrl+Shift+Delete`

2. Use curl Instead

```
bash
```

```
# Curl bypasses browser cache
```

```
curl -v https://getcarelinkai.com/api/test-sentry-server
```

3. Check Response Headers

Should include:

- `Cache-Control: no-store, no-cache, must-revalidate`
- `Pragma: no-cache`

- Expires: 0
- X-Test-Run-ID: <unique-id>

Success Criteria

✓ Test is successful when:

1. Each test endpoint returns a **unique** test run ID
2. No cached responses (different ID on each request)
3. Render logs show detailed Sentry events
4. Events appear in Sentry dashboard within 2-5 minutes
5. Events include test run ID for easy searching
6. beforeSend logs show event details before sending

Expected Timeline

Step	Time
Deploy to Render	~5-10 minutes
Run all tests	~2 minutes
Wait for Sentry ingestion	2-5 minutes
Verify in Sentry dashboard	~2 minutes
Total	~15-20 minutes

After Testing

If Tests Pass

1. ✓ Document test run IDs from Sentry
2. ✓ Archive test results
3. ✓ Consider reducing sampling rates for production
4. ✓ Monitor Sentry for real errors

If Tests Fail

1. ✗ Review Render logs for errors
2. ✗ Check environment variables
3. ✗ Verify tunnel is working
4. ✗ Test DSN connectivity
5. ✗ Review beforeSend filter logic



Notes

- **Test run IDs** are searchable in Sentry dashboard
 - **Format:** `test-<timestamp>-<random>` or `error-<timestamp>-<random>`
 - **Logs** are prefixed with `[Sentry Test]` or `[Sentry beforeSend]`
 - **Sampling** is temporarily at 100% for debugging
 - **TODO:** Revert sampling to 0.1 (10%) after testing
-



Quick Links

- **Sentry Dashboard:** <https://sentry.io/organizations/carelinkai/issues/>
 - **Render Dashboard:** <https://dashboard.render.com/>
 - **Test Endpoints:**
 - Server: <https://getcarelinkai.com/api/test-sentry-server>
 - Server Error: <https://getcarelinkai.com/api/test-sentry-server-error>
 - Edge: <https://getcarelinkai.com/api/test-sentry-edge>
 - Client: <https://getcarelinkai.com/api/test-sentry-client-error>
-



Support

If you encounter issues:

1. Check Render logs first
 2. Review this document
 3. Test tunnel endpoint
 4. Verify environment variables
 5. Contact development team with test run IDs
-

Document Version: 1.0

Last Updated: January 3, 2026

Git Commit: 93a54e1