

Tracking Services Fix Summary

Date

December 31, 2025

Objective

Fix GA4 and Sentry tracking to achieve 100% tracking functionality (10/10 score).

Initial Status

- **Working (3/5):** Microsoft Clarity ✓, Facebook Pixel ✓, Google Tag Manager ✓
- **Not Working (2/5):** Google Analytics 4 ✗, Sentry ✗

Root Causes Identified

1. Sentry Issues

- **Missing instrumentation.ts:** Sentry requires the `instrumentation.ts` file for proper Next.js App Router integration
- **Production-only initialization:** `sentry.server.ts` was configured to only initialize in production environment
- **Missing instrumentation hook:** `next.config.js` lacked the `instrumentationHook` experimental flag
- **Lack of debugging:** No console logging to verify initialization

2. GA4 Issues

- **Configuration:** GA4 script was present but missing explicit `send_page_view` flag
- **No debugging:** No console logging to verify initialization

Changes Implemented

1. Created `src/instrumentation.ts`

```
/**  
 * Instrumentation for Next.js App Router  
 * This file is automatically loaded by Next.js to initialize monitoring tools  
 */  
export async function register() {  
  // Only run on server-side  
  if (process.env.NEXT_RUNTIME === 'nodejs') {  
    await import('./lib/sentry.server');  
  }  
  
  // Only run on Edge runtime  
  if (process.env.NEXT_RUNTIME === 'edge') {  
    await import('./lib/sentry.server');  
  }  
}
```

Why: Next.js App Router requires this file for proper server-side instrumentation initialization.

2. Updated `next.config.js`

```
experimental: {  
  instrumentationHook: true,  
}
```

Why: Enables Next.js to load and execute the `instrumentation.ts` file.

3. Fixed `src/lib/sentry.client.ts`

Changes:

- Added try-catch block for error handling
- Added debug logging: `console.log('[Sentry] Client-side initialization successful')`
- Added warning messages for missing configuration
- Enabled debug mode in development: `debug: ENVIRONMENT === 'development'`

Why: Better error visibility and debugging capabilities.

4. Fixed `src/lib/sentry.server.ts`

Changes:

- Removed production-only restriction: Changed from `if (SENTRY_DSN && ENVIRONMENT === 'production')` to `if (SENTRY_DSN)`
- Added try-catch block for error handling
- Added debug logging: `console.log('[Sentry] Server-side initialization successful')`
- Added warning message for missing DSN
- Enabled debug mode in development: `debug: ENVIRONMENT === 'development'`

Why: Sentry should work in all environments, not just production. This allows testing in development and staging.

5. Improved `src/app/layout.tsx` GA4 Script

Changes:

```
gtag('config', '${process.env.NEXT_PUBLIC_GA_MEASUREMENT_ID}', {
  page_path: window.location.pathname,
  send_page_view: true // ← Added this
});
console.log('[GA4] Google Analytics 4 initialized with ID: ...'); // ← Added this
```

Why:

- `send_page_view: true` ensures GA4 explicitly sends pageview events
- Console logging helps verify GA4 is loading correctly

Environment Variables Required

These must be set in Render (or your deployment environment):

```
# Google Analytics 4
NEXT_PUBLIC_GA_MEASUREMENT_ID=G-C8L4GJ30Z9

# Google Tag Manager
NEXT_PUBLIC_GTM_ID=GTM-TNZF9G3M

# Microsoft Clarity
NEXT_PUBLIC_CLARITY_PROJECT_ID=uu6rjw7bqo

# Sentry
NEXT_PUBLIC_SENTRY_DSN=https://d649b9c85c145427fcf-
b62cecdeaa2d9e@o4510110703216128.ingest.us.sentry.io/4510154420089472

# Facebook Pixel
NEXT_PUBLIC_FACEBOOK_PIXEL_ID=244432082802031

# Optional: Set environment name
NEXT_PUBLIC_ENVIRONMENT=production
```

Testing Steps

Local Testing (Development)

1. Start the development server: `npm run dev`
2. Open browser console
3. Look for initialization messages:
 - `[GA4] Google Analytics 4 initialized with ID: G-C8L4GJ30Z9`
 - `[Sentry] Client-side initialization successful`
 - `[Sentry] Server-side initialization successful (in terminal)`
4. Check Network tab for:
 - `gtag.js` - Google Analytics script loading
 - `gtm.js` - Google Tag Manager loading
 - `fbevents.js` - Facebook Pixel loading
 - `clarity.ms/tag` - Microsoft Clarity loading
 - Sentry events being sent

Production Testing (Render)

1. After deployment completes, visit your production site
2. Open browser console (F12)

3. Verify the same initialization messages appear
4. Check Network tab for all tracking scripts
5. Test Sentry by triggering an error (optional):

```
javascript
// In browser console
throw new Error("Test Sentry error tracking");
```

6. Check Sentry dashboard for the error event
7. Check GA4 dashboard for pageview events (may take 24-48 hours)

Verification Checklist

- [] GA4 script loads successfully
- [] GA4 initialization message in console
- [] GA4 pageview events in network tab
- [] GTM loads successfully
- [] Facebook Pixel loads successfully
- [] Microsoft Clarity loads successfully
- [] Sentry client initialization message in console
- [] Sentry server initialization in logs
- [] All tracking services return 200 status codes
- [] No console errors related to tracking

Expected Results

Immediate (After Deployment)

- All 5 tracking services should load without errors
- Console messages confirm initialization
- Network tab shows all scripts loading successfully (200 status)

Within 24-48 Hours

- GA4 dashboard shows pageview data
- Sentry dashboard shows any errors that occurred
- Microsoft Clarity shows session recordings
- Facebook Pixel shows event data

Deployment

Git Commit

```
Commit: 01e2869
Message: "fix: GA4 and Sentry tracking implementation"
```

Files Changed

1. `src/instrumentation.ts` (new file)
2. `next.config.js` (added instrumentationHook)
3. `src/lib/sentry.client.ts` (improved initialization)
4. `src/lib/sentry.server.ts` (fixed environment restriction)

5. `src/app/layout.tsx` (improved GA4 config)

Deployment Status

 **Pushed to GitHub** - Auto-deployment to Render will trigger automatically

Monitoring

What to Monitor After Deployment

1. **Render Logs:** Check for Sentry server initialization message
2. **Browser Console:** Verify GA4 and Sentry client messages
3. **Network Tab:** Confirm all tracking scripts load (200 status)
4. **Sentry Dashboard:** Watch for incoming error events
5. **GA4 Real-time:** Check for active users (within minutes)

Troubleshooting

If Sentry Still Not Working

1. Verify `NEXT_PUBLIC_SENTRY_DSN` is set in Render
2. Check Render logs for server-side initialization
3. Check browser console for client-side initialization
4. Ensure no ad blockers are interfering
5. Test by throwing a manual error in console

If GA4 Still Not Working

1. Verify `NEXT_PUBLIC_GA_MEASUREMENT_ID` is set in Render
2. Check Network tab for `gtag/js` script loading
3. Look for GA4 initialization message in console
4. Check for `collect` requests to `google-analytics.com`
5. Wait 24-48 hours for data to appear in dashboard

If Build Fails

1. Check Render build logs for errors
2. Verify all dependencies are installed
3. Run `npm run build` locally to test
4. Check TypeScript errors (though `ignoreBuildErrors` is enabled)

Success Criteria

-  All 5 tracking services load without errors
-  Console shows initialization messages
-  Network tab shows 200 status for all scripts
-  Sentry can capture errors
-  GA4 can track pageviews
-  Build completes successfully
-  No regression in existing functionality

Final Status

Target: 10/10 tracking score (5/5 services working)

Implemented Changes: All fixes deployed and ready for testing.

Next Steps:

1. Wait for Render deployment to complete
2. Test all tracking services in production
3. Monitor dashboards for 24-48 hours
4. Verify data is flowing correctly

Notes

- Build completed successfully locally
- All changes follow Next.js App Router best practices
- Sentry integration uses official `@sentry/nextjs` package
- GA4 uses direct gtag.js integration (not through GTM)
- All tracking services respect user consent (via CookieConsent component)

References

- [Sentry Next.js Documentation](https://docs.sentry.io/platforms/javascript/guides/nextjs/) (<https://docs.sentry.io/platforms/javascript/guides/nextjs/>)
- [Next.js Instrumentation](https://nextjs.org/docs/app/building-your-application/optimizing/instrumentation) (<https://nextjs.org/docs/app/building-your-application/optimizing/instrumentation>)
- [Google Analytics 4 Setup](https://developers.google.com/analytics/devguides/collection/ga4) (<https://developers.google.com/analytics/devguides/collection/ga4>)