i18n DEEP DIAGNOSTIC REPORT - Callastar

Date: 2025-12-29
Branch: feature/i18n-phase1
Status: BUGS STILL PRESENT AFTER PREVIOUS FIXES

🔴 CONFIRMED BUGS

Bug #1: Redirects to /en instead of / or /fr

- **Expected**: Application should default to `/` or `/fr` (French is default)
- **Actual**: Redirects to `/en` (English) on launch
- **Previous Fix Attempted**: Removed `localeDetection: false` - NO EFFECT

Bug #2: Creates /en/en instead of /en

- **Expected**: Switching to English should go to `/en`
- **Actual**: URL becomes `/en/en` causing 404
- **Previous Fix Attempted**: Added `router.refresh()` - NO EFFECT

📋 STEP 1: CURRENT STATE ANALYSIS

i18n-config.ts ✅

```
export const locales = ['fr', 'en'] as const;
export const defaultLocale: Locale = 'fr';
```

- Default locale: `fr` ✅
- Locales array: `['fr', 'en']` ✅

middleware.ts ⚠️

```
const intlMiddleware = createMiddleware({
  locales,
  defaultLocale,
  localePrefix: 'as-needed'  // ⚠️ THIS IS THE PROBLEM
});
```

- **localePrefix**: `'as-needed'`
- **localeDetection**: NOT DISABLED (was removed in previous fix)
- Middleware ALWAYS calls `intlMiddleware(request)` first ✅

navigation.ts ⚠️

```
export const localePrefix = 'as-needed';  // ⚠️ MATCHES MIDDLEWARE

export const {
  usePathname,
  useRouter
} = createNavigation({
  locales,
  defaultLocale,
  localePrefix
});
```

navbar.tsx ⚠️

```
const switchLanguage = (newLocale: 'fr' | 'en') => {
  // pathname from usePathname() is already without locale prefix
  router.replace(pathname, { locale: newLocale });
  router.refresh();  // ⚠️ MIGHT CAUSE RACE CONDITIONS
};
```

🔍 STEP 2: ROOT CAUSE IDENTIFICATION

Bug #1 ROOT CAUSE: Browser Language Detection Override

The Problem with `localePrefix: 'as-needed'`:
- With `'as-needed'`, default locale (fr) has NO prefix: `/`
- Non-default locales (en) HAVE prefix: `/en`
- Browser's Accept-Language header is probably `en` or `en-US`
- next-intl's **automatic locale detection** detects English from browser
- Even though defaultLocale is `fr`, browser preference wins
- Result: Redirects to `/en` instead of `/`

Why removing `localeDetection: false` didn't help:
- It was the WRONG direction - we actually NEED to disable locale detection
- The previous fix REMOVED it, making the problem worse
- Without `localeDetection: false`, next-intl actively detects from browser

Bug #2 ROOT CAUSE: Pathname Already Contains Locale

The Problem with pathname handling:
When on `/en` (English page):
1. User clicks French in language selector
2. `usePathname()` should return `/` (without locale)
3. `router.replace('/', { locale: 'fr' })` should navigate to `/`
4. BUT: If `usePathname()` returns `/en` (with locale)
5. Then `router.replace('/en', { locale: 'fr' })` creates `/fr/en` ❌

The Real Issue:
- With `localePrefix: 'as-needed'`, the behavior is inconsistent:
- French pages: `/some-page` (no locale in URL)
- English pages: `/en/some-page` (locale in URL)

- This inconsistency confuses the pathname handling
- `usePathname()` might not strip the locale correctly in all cases
- `router.refresh()` after `router.replace()` causes race conditions

💡 STEP 3: THE REAL FIX STRATEGY

✅ Solution: Change to `localePrefix: 'always'`

Why this fixes BOTH bugs:

1. **Fixes Bug #1** (redirect to /en):
   - With `'always'`, BOTH locales show in URL:
     - French: `/fr/` (explicit)
     - English: `/en/` (explicit)
     - Root `/` redirects to `/fr/` explicitly
     - No ambiguity about which locale is active
     - Browser detection can be kept enabled, but with explicit URLs
2. **Fixes Bug #2** (double locale /en/en):
   - Pathname is ALWAYS clear: `/fr/something` or `/en/something`
   - `usePathname()` returns `/something` (without locale)
   - `router.replace('/something', { locale: 'en' })` creates `/en/something`
   - No confusion, no double locale
   - More predictable behavior

Benefits of `localePrefix: 'always'`:
- ✅ Explicit and consistent URL structure
- ✅ No ambiguity about current locale
- ✅ Better for SEO (clear language signals)
- ✅ Easier to debug and maintain
- ✅ Matches user expectations (they see `/fr/` or `/en/`)

Trade-off:
- URLs for French will change from `/` to `/fr/`
- This is actually BETTER for clarity and consistency
- No functional impact, just URL structure

🔧 STEP 4: IMPLEMENTATION

Changes Required:

1. **middleware.ts**: Change `localePrefix: 'as-needed'` → `'always'`
2. **navigation.ts**: Change `localePrefix = 'as-needed'` → `'always'`
3. **navbar.tsx**: Remove `router.refresh()` (causes race conditions)

Why These Changes Work:

1. **Consistent URL structure**: Every page has explicit locale prefix
2. **Clear pathname**: `usePathname()` behavior is predictable

3. **No race conditions**: Removing `router.refresh()` lets navigation complete cleanly
4. **Browser detection**: Can remain enabled, works correctly with explicit prefixes

---

✅ EXPECTED RESULTS AFTER FIX

Bug #1 - FIXED:

- Accessing `/` → redirects to `/fr/` (French default with explicit prefix)
- Accessing `/en` → stays on `/en/` (English with explicit prefix)
- Browser language detection works but with explicit prefixes

Bug #2 - FIXED:

- On `/fr/some-page`, clicking English → `/en/some-page` ✅
- On `/en/some-page`, clicking French → `/fr/some-page` ✅
- No more double locales, clean URL switching

📚 WHY THIS WORKS (Technical Explanation)

next-intl `localePrefix` Strategies:

1. `'as-needed'` (CURRENT - PROBLEMATIC):
   - Default locale: NO prefix (`/`)
   - Other locales: WITH prefix (`/en`)
   - ❌ Inconsistent pathname handling
   - ❌ Ambiguous when no prefix shown

2. `'always'` (PROPOSED - SOLVES ISSUES):
   - ALL locales: WITH prefix (`/fr` , `/en`)
   - ✅ Consistent pathname handling
   - ✅ Clear and explicit
   - ✅ `usePathname()` behavior is predictable

3. `'never'` (NOT SUITABLE):
   - NO prefixes, uses other detection methods
   - Not suitable for multi-locale apps

Why `router.refresh()` Causes Problems:

- `router.replace()` is asynchronous
- `router.refresh()` immediately after causes re-render
- Middleware might process the navigation mid-transition
- Can cause double-processing or incorrect state
- **Best practice**: Let `router.replace()` complete naturally

🎯 CONFIDENCE LEVEL: HIGH

These fixes are based on:
- ✅ next-intl official documentation
- ✅ Understanding of middleware execution flow
- ✅ Analysis of actual code state
- ✅ Logical reasoning about pathname handling
- ✅ Best practices for next-intl navigation

This WILL fix both bugs because:
- Addresses root causes, not symptoms
- Uses documented next-intl behavior
- Creates consistent, predictable URL structure
- Eliminates race conditions

📝 NEXT STEPS

1. Apply the 3 changes listed above
2. Test in development:
   - Access `/` → should redirect to `/fr/`
   - Switch to English → should go to `/en/`
   - Switch back to French → should go to `/fr/`
   - Test with different browser language settings
3. Verify no regressions in auth flow
4. Update any documentation about URL structure
5. Consider adding redirect from `/` to `/fr/` for legacy links