

Work Order 15 - Whop SSO OAuthSignin Error - Debug Implementation

Date: November 27, 2025

Status:  DEBUG MODE ENABLED - AWAITING TEST RESULTS

Priority:  CRITICAL - Blocking User Access

Current Situation

PROGRESS: Past invalid_client!

The redirect URL configuration is now **CORRECT** - Whop is accepting the OAuth request and redirecting to its authorization page.

NEW ISSUE: OAuthSignin Error

Users are now seeing:

```
https://catchbarrels.app/auth/login?callback-
Url=https%3A%2F%2Fcatchbarrels.app%2Fdashboard&error=OAuthSignin
```

This means the OAuth flow **starts** but **fails** somewhere between:

1.  User clicks “Sign in with Whop” → Works
2.  Redirected to Whop authorization page → Works
3.  User clicks “Authorize” → Works
4.  Whop redirects to /api/auth/callback/whop?code=... → Works
5.  **NextAuth processes the callback → FAILS HERE**

What I've Implemented

Enhanced Debug Logging

File: lib/auth-options.ts

Changes:

-  Enabled permanent debug mode: `debug: true`
-  Added comprehensive logging to JWT callback
-  Added detailed logging to token exchange
-  Added detailed logging to userinfo fetch
-  Added profile validation and logging

What you'll see in server logs:

```
[Whop OAuth] Token exchange starting...
[Whop OAuth] Token request context: { hasCode: true, hasClientId: true, hasClientSecret: true }
[Whop OAuth] Token response status: 200
[Whop OAuth] Token exchange successful, keys: ["access_token", "token_type", ...]
[Whop OAuth] Fetching user profile...
[Whop OAuth] Access token present: true
[Whop OAuth] Userinfo response status: 200
[Whop OAuth] User profile fetched, keys: ["id", "email", "name", ...]
[Whop OAuth] Processing profile: {...}
[Whop OAuth] Mapped profile: {...}
[NextAuth JWT Callback] Called with: {...}
```

2 Custom Token Exchange Implementation

Why: The default NextAuth token exchange may not be sending the correct format expected by Whop.

What I added:

```
token: {
  url: 'https://api.whop.com/api/v2/oauth/token',
  async request(context) {
    // Custom token exchange with detailed logging
    // Uses application/x-www-form-urlencoded
    // Includes all required parameters: grant_type, code, redirect_uri, client_id,
    client_secret
  }
}
```

Key points:

- Uses `application/x-www-form-urlencoded` (Whop's expected format)
- Explicitly includes `redirect_uri` in token exchange
- Logs request details and response status
- Catches and logs errors with full context

3 Custom Userinfo Fetch Implementation

Why: The default userinfo fetch may fail if the access token format is incorrect.

What I added:

```
userinfo: {
  url: 'https://api.whop.com/api/v2/me',
  async request(context) {
    // Custom userinfo fetch with Bearer token
    // Logs response status and profile data
  }
}
```

Key points:

- Uses `Authorization: Bearer {access_token}` header

- Logs access token presence (not the value, for security)
 - Logs response status
 - Catches and logs errors with full context
-

4 Enhanced Profile Validation

Why: Missing or malformed profile data can cause OAuthSignin errors.

What I added:

```
profile(profile: any) {
  // Validate required fields
  if (!profile.id) {
    throw new Error('Whop profile missing required "id" field');
  }

  // Map Whop profile to NextAuth user
  // With fallbacks for missing fields
}
```

Key points:

- Validates required `id` field
 - Provides fallbacks for `name` and `username`
 - Logs original and mapped profile data
-

Next Steps: Testing & Debugging

STEP 1: Clear Everything

```
# In incognito/private browser
1. Close all browser tabs
2. Open new incognito window
3. Clear site data if possible
```

STEP 2: Redeploy the App

CRITICAL: The new debug code must be deployed to production for logging to work.

```
cd /home/ubuntu/barrels_pwa/nextjs_space
yarn build
# Then deploy to catchbarrels.app
```

STEP 3: Test the Login Flow

1. Navigate to: `https://catchbarrels.app/auth/login`
2. Click “Sign in with Whop”
3. **Expected:** Redirected to Whop authorization page
4. Click “Authorize”

5. One of two outcomes:

- ✓ **SUCCESS:** Lands on /dashboard
- ✗ **FAILURE:** Back to /auth/login?error=OAuthSignin

STEP 4: Check Server Logs

CRITICAL: This is where we'll see what's failing.

Where to find logs:

- If using Vercel: Dashboard → Deployments → [Latest] → Functions → Runtime Logs
- If using other platform: Check your server console/logs

What to look for:

✓ Successful Flow (All Present):

```
[Whop OAuth] Token exchange starting...
[Whop OAuth] Token response status: 200
[Whop OAuth] Token exchange successful
[Whop OAuth] Fetching user profile...
[Whop OAuth] Userinfo response status: 200
[Whop OAuth] User profile fetched
[Whop OAuth] Processing profile
[Whop OAuth] Mapped profile
[NextAuth JWT Callback] Called with: { hasUser: true, hasAccount: true }
```

✗ Token Exchange Failure:

```
[Whop OAuth] Token exchange starting...
[Whop OAuth] Token response status: 400 (or 401, 403)
[Whop OAuth] Token exchange failed: [error message]
```

This means: Client ID/Secret mismatch, or redirect_uri doesn't match Whop exactly.

✗ Userinfo Fetch Failure:

```
[Whop OAuth] Token exchange successful
[Whop OAuth] Fetching user profile...
[Whop OAuth] Userinfo response status: 401 (or 403)
[Whop OAuth] Userinfo fetch failed: [error message]
```

This means: Access token is invalid or expired, or wrong scope.

✗ Profile Validation Failure:

```
[Whop OAuth] User profile fetched
[Whop OAuth] Processing profile: { ... } (missing "id")
[Whop OAuth] Profile missing required "id" field
```

This means: Whop's API response format is different than expected.

Common OAuthSignin Causes & Fixes

1. Token Exchange Fails

Error: Token response status: 400/401

Causes:

- ✗ WHOP_CLIENT_SECRET is wrong
- ✗ redirect_uri in token exchange doesn't match Whop exactly
- ✗ NEXTAUTH_URL is wrong (should be https://catchbarrels.app)

Fix:

```
# Verify in .env:
NEXTAUTH_URL=https://catchbarrels.app
WHOP_CLIENT_ID=app_WklQSIh1x1uL6d
WHOP_CLIENT_SECRET=apik_JYqngRfc3G5TC_A2019140_ce44952c40b5ccff900a73df7fc239400bb6e9af6d0e8b309ce0a791073f36a6

# Verify in Whop Dashboard:
# Redirect URL: https://catchbarrels.app/api/auth/callback/whop (EXACT)
```

2. Userinfo Fetch Fails

Error: Userinfo response status: 401

Causes:

- ✗ Access token is missing or malformed
- ✗ Whop's /api/v2/me endpoint requires different scope
- ✗ Token was revoked or expired immediately

Fix:

```
// In auth-options.ts, authorization params:
scope: 'openid email profile',

// If this fails, try:
scope: 'openid email profile user:read',
```

Check Whop Docs:

- Verify the correct scope for accessing user profile
- Verify the correct userinfo endpoint URL

3. Profile Missing Required Fields

Error: Profile missing required "id" field

Causes:

- ✗ Whop's /api/v2/me returns a different structure
- ✗ User account on Whop is incomplete

Fix:

Check the logged profile structure:

```
[Whop OAuth] User profile fetched, keys: ["user_id", "whop_id", ...]
```

If `id` is actually `user_id` or `whop_id`, update the profile mapping:

```
profile(profile: any) {
  const mappedProfile = {
    id: profile.id || profile.user_id || profile.whop_id, // Try alternatives
    name: profile.name || profile.username || 'Whop User',
    email: profile.email,
    username: profile.username || profile.email,
    whopUserId: profile.id || profile.user_id || profile.whop_id,
  };
  return mappedProfile;
}
```

4. State/CSRF Validation Fails

Error: Silent failure or `OAuthSignin` with no logs

Causes:

- ✗ checks: ['state'] is enabled but state parameter is missing
- ✗ Cookie issues (SameSite, Secure)

Fix:

Try temporarily removing state check:

```
// In auth-options.ts:
checks: [], // Disable state check temporarily for testing
```

Note: Only disable for testing. Re-enable for production security.



What to Send Me (If Still Failing)

1. Server Logs (CRITICAL)

Copy the **entire** log output from when you click “Sign in with Whop” until it redirects back.

Look for:

- All `[Whop OAuth]` lines
- All `[NextAuth]` lines
- Any error stack traces

2. Browser DevTools → Network Tab

Steps:

1. Open DevTools (F12)
2. Go to Network tab

3. Clear network log
4. Click “Sign in with Whop”
5. Go through the auth flow
6. Take screenshots of:
 - The request to `/api/auth/callback/whop?code=...`
 - Its response (Headers + Preview/Response tabs)

3. Environment Variables (Redacted)

```
NEXTAUTH_URL=https://catchbarrels.app
WHOP_CLIENT_ID=app_WklQ... (redacted)
WHOP_CLIENT_SECRET=apik_... (first 10 chars only)
```

4. Whop Dashboard Settings

Screenshot of:

- OAuth Settings → Redirect URLs
- Scopes (if visible)
- Client ID (redacted)

Success Indicators

Server Logs Should Show:

```
[Whop OAuth] Token exchange starting...
[Whop OAuth] Token response status: 200
[Whop OAuth] Token exchange successful, keys: ["access_token", ...]
[Whop OAuth] Fetching user profile...
[Whop OAuth] Userinfo response status: 200
[Whop OAuth] User profile fetched, keys: ["id", "email", ...]
[Whop OAuth] Processing profile: { "id": "...", "email": "...", ... }
[Whop OAuth] Mapped profile: { "id": "...", "name": "...", ... }
[NextAuth JWT Callback] Called with: { hasUser: true, hasAccount: true, accountProvider: "whop" }
[Whop OAuth] Processing Whop login for user: ...
[NextAuth Redirect] url: https://catchbarrels.app/dashboard
```

User Experience:

-  Redirected to Whop authorization page
-  After clicking “Authorize”, lands on Dashboard
-  No error parameter in URL
-  User data synced (name, email visible in profile)

Next Actions

Immediate (Rick):

1.  Redeploy app with new debug code
2.  Test login flow in incognito

3. Capture server logs
4. Send me the logs + network tab screenshots

Once I have logs:

- I'll pinpoint the exact failure point
 - Implement targeted fix
 - Test again
-



Expected Timeline

- **If token exchange fails:** 10-15 min fix (env var or Whop config)
 - **If userinfo fails:** 20-30 min fix (scope or endpoint adjustment)
 - **If profile validation fails:** 5-10 min fix (field mapping)
-



Current Status Summary

- | | |
|-------------------------------------|--|
| <input checked="" type="checkbox"/> | Redirect URL <code>registration</code> : FIXED |
| <input checked="" type="checkbox"/> | <code>invalid_client error</code> : FIXED |
| <input checked="" type="checkbox"/> | OAuth authorization <code>request</code> : WORKING |
| <input checked="" type="checkbox"/> | Whop authorization <code>page</code> : WORKING |
| <input checked="" type="checkbox"/> | OAuth callback <code>processing</code> : FAILING (investigating with debug logs) |

Blocked on: Server logs to identify failure point.

Next action: Deploy updated code → Test → Send logs.

Ready for testing!