

Whop Player Sync - Complete Implementation

Overview

The Whop player sync has been completely rebuilt to properly fetch **ALL** paying Whop customers and create/update their accounts in CatchBarrels, regardless of whether they purchased before the app was ready.

What Was Fixed

Previous Issues

1. **Only synced existing users:** The old sync only updated users who already had a `whopUserId` in the database
2. **Didn't fetch all Whop members:** It wasn't calling the Whop API to get all customers
3. **No account creation:** It couldn't create new accounts for customers who paid before the app launched
4. **Poor visibility:** Limited logging made it hard to see what was happening

New Implementation

1. **✓ Fetches ALL Whop memberships:** Calls `client.memberships.list()` without filters to get every customer
2. **✓ Creates new accounts:** Automatically creates user accounts for Whop customers who don't have one yet
3. **✓ Updates existing accounts:** Syncs membership tiers and status for users who already exist
4. **✓ Proper upsert logic:** Checks for users by email OR `whopUserId`, creates if not found, updates if found
5. **✓ Comprehensive logging:** Detailed console logs show exactly what's happening with each user
6. **✓ Error handling:** Gracefully handles errors and reports them without breaking the entire sync
7. **✓ Pagination support:** Handles large customer lists by fetching all pages from Whop API

How It Works

1. Admin Triggers Sync

- Admin goes to `/admin/players`
- Clicks "Sync from Whop" button
- Button calls `POST /api/admin/whop/sync-players`

2. Fetch ALL Whop Memberships

```
const allMemberships = await getAllWhopMemberships();  
// This fetches EVERY customer from Whop, regardless of whether  
// they have an account in CatchBarrels
```

3. Group by User

Memberships are grouped by Whop user ID, so users with multiple products are handled correctly:

```
const membershipsByUser = new Map<string, Membership[]>();
// user_abc123 -> [Athlete membership, Pro membership]
```

4. Process Each User

For each Whop customer:

1. **Extract user data:** Email, name from membership
2. **Determine highest tier:** If they have multiple memberships, use the highest tier
3. **Check for assessment purchase:** Detect VIP offer eligibility
4. **Find or create user:**
 - Search database by email OR whopUserId
 - If exists: UPDATE with latest data
 - If new: CREATE account with Whop data
5. **Set membership fields:**
 - whopUserId : Link to Whop account
 - membershipTier : athlete/pro/elite
 - membershipStatus : active/inactive
 - membershipExpiresAt : Subscription expiry
 - VIP offer fields if applicable

5. Return Results

```
{
  "success": true,
  "syncedCount": 15,
  "createdCount": 10,
  "updatedCount": 5,
  "skippedCount": 0,
  "totalWhopUsers": 15,
  "errors": []
}
```

User Login Flow

How Synced Users Log In

Users created by the sync **do NOT have passwords**. They log in using Whop OAuth:

1. **User goes to login page:** <https://catchbarrels.app/auth/login>
2. **Clicks “Log in with Whop”:** Triggers OAuth flow
3. **Whop authenticates:** User logs in on Whop’s site
4. **OAuth callback:** Whop redirects back to CatchBarrels
5. **User matched:** JWT callback finds user by whopUserId or email
6. **Redirected to dashboard:** User lands on /dashboard

Why This Works

- Users created by sync have whopUserId and email set

- Whop OAuth returns the same `whopUserId`
- JWT callback matches: `prisma.user.findUnique({ where: { whopUserId } })`
- User is authenticated without needing a password

Testing the Sync

Step 1: Verify Environment

Make sure these are set in `.env`:

```
WHOP_API_KEY=apik_...
WHOP_APP_ID=app_...
WHOP_CLIENT_ID=app_...
WHOP_CLIENT_SECRET=apik_...
```

Step 2: Run the Sync

1. Log in as admin: `coach@catchbarrels.app` / `CoachBarrels2024!`
2. Go to `/admin/players`
3. Click "Sync from Whop"
4. Watch the console logs (server-side):

```
[Whop Sync] =====
[Whop Sync] Starting FULL player sync from Whop
[Whop Sync] =====
[Whop Client] Initializing Whop client...
[Whop Client] [✓] Client initialized, fetching all memberships...
[Whop Client] [✓] Fetched 15 memberships from first page
[Whop Sync] [✓] Found 15 total memberships in Whop
[Whop Sync] [✓] Found 12 unique Whop users with memberships

[Whop Sync] --- Processing Whop User: user_abc123 ---
[Whop Sync] Email: john@example.com
[Whop Sync] Name: John Smith
[Whop Sync] Active memberships: 1/1
[Whop Sync]   - Product prod_04CB6y0IzNJLe [→] pro tier
[Whop Sync] [→] Determined tier: pro
[Whop Sync] [→] User does not exist, creating new account...
[Whop Sync] [✓] Created new user john@example.com (user_xyz789)

...

[Whop Sync] =====
[Whop Sync] SYNC COMPLETE
[Whop Sync] [✓] Created: 10 new users
[Whop Sync] [✓] Updated: 2 existing users
[Whop Sync] [⚠] Skipped: 0 users (no email)
[Whop Sync] [✗] Errors: 0
[Whop Sync] [→] Total processed: 12/12 users
[Whop Sync] =====
```

Step 3: Verify in Database

```
SELECT
  email,
  name,
  whopUserId,
  membershipTier,
  membershipStatus,
  password
FROM "User"
WHERE whopUserId IS NOT NULL;
```

You should see:






- All Whop customers listed
- `password` field is `NULL` (they don't have passwords)
- `whopUserId` is set
- `membershipTier` matches their Whop product
- `membershipStatus` is "active" for current subscribers

Step 4: Test Login

1. **Open incognito window:** `https://catchbarrels.app/auth/login`
2. **Click "Log in with Whop"**
3. **Authenticate with Whop:** Use a test customer's credentials
4. **Should redirect to dashboard:** You should land on `/dashboard` as an authenticated user
5. **Check session:** User info should match the synced account

Admin UI

The admin players page now shows:

-  **Created count:** How many new accounts were created
-  **Updated count:** How many existing accounts were updated
-  **Skipped count:** Users without emails (rare)
-  **Total Whop users:** Total customers in Whop
-  **Error details:** Any sync failures

Toast notification example:

```
Successfully synced 12 Whop customers!
✓ Created 10 new player accounts
✓ Updated 2 existing players
```

Troubleshooting

No memberships found

Error: No memberships found in Whop. Check `WHOP_API_KEY` configuration.

Cause: Either:

1. `WHOP_API_KEY` is incorrect

2. No customers have purchased yet
3. Whop API is down

Fix:

1. Verify `WHOP_API_KEY` in `.env` matches Whop dashboard
2. Check Whop dashboard to confirm customers exist
3. Test with `curl` :

```
curl -H "Authorization: Bearer apik_..." \
https://api.whop.com/api/v2/memberships
```

Users created but can't log in

Symptom: Sync shows “Created 10 new users” but login fails

Cause: Whop OAuth not configured correctly

Fix:

1. Verify redirect URL in Whop: `https://catchbarrels.app/api/auth/callback/whop`
2. Verify `WHOP_CLIENT_ID` and `WHOP_CLIENT_SECRET` match Whop dashboard
3. Check server logs for OAuth errors

Some users skipped

Message: ⚠ Skipped 2 users (no email)

Cause: Whop membership doesn't have user email

Fix: These users need to complete their Whop profile. They can:

1. Log into Whop
2. Update their account email
3. Re-sync will pick them up

Duplicate users

Symptom: Multiple accounts for same person

Cause: User has different emails in Whop vs. CatchBarrels

Fix: The sync uses this logic to prevent duplicates:

```
const existingUser = await prisma.user.findFirst({
  where: {
    OR: [
      { email: userEmail },
      { whopUserId: whopUserId },
    ],
  },
});
```

If duplicates still occur:

1. Manually merge accounts in database
2. Keep the one with `whopUserId` set
3. Delete the duplicate

Product Tier Mapping

Make sure `lib/whop-client.ts` has the correct product IDs:

```
export function getWhopProductTier(productId: string): string {
  const productMapping: Record<string, string> = {
    // BARRELS Athlete - $49/mo or $417/yr
    "prod_kNyobCww4tc2p": "athlete",

    // BARRELS Pro - $99/mo or $839/yr
    "prod_04CB6y0IzNJLe": "pro",

    // BARRELS Elite - $199/mo or $1,699/yr
    "prod_vCV6UQH3K18QZ": "elite",

    // The 90-Day Transformation - $997 one-time
    "prod_zH1wnZs0JKKfd": "elite",
  };



  return productMapping[productId] || "free";
}
```

To verify product IDs:

1. Go to Whop Dashboard
2. Products → Select a product
3. Copy the product ID (starts with `prod_`)
4. Update the mapping if needed
5. Re-sync to apply changes

Files Changed

Core Sync Logic

-  `/app/api/admin/whop/sync-players/route.ts` - Complete rewrite
-  `/lib/whop-client.ts` - Added `getAllWhopMemberships()`

UI Updates

-  `/app/admin/players/players-client.tsx` - Enhanced sync feedback

Documentation

-  `/docs/WHOP_SYNC_COMPLETE.md` - This file

Next Steps

1. Run the Sync

```
# 1. Deploy the updated code
yarn build

# 2. Log in as admin
open https://catchbarrels.app/admin/players

# 3. Click "Sync from Whop"

# 4. Watch server logs to verify
```

2. Test with a Real Customer

1. Have a test customer purchase a membership on Whop
2. Run sync
3. Verify they appear in `/admin/players`
4. Have them log in via “Log in with Whop”
5. Verify they land on dashboard

3. Monitor

- Check sync logs regularly
- Verify all customers are synced
- Watch for any errors in console

4. Backfill Existing Customers

If you have customers who purchased before the app was ready:

1. Run sync NOW to create their accounts
2. Send them an email:

Subject: Your CatchBarrels PWA **is** Ready! 🎉

Hey [Name],

Your CatchBarrels account **is** now active! 🎉 Since you're an early customer, we've already **set** up your account.

To access the **app**:

1. **Go to** `https://catchbarrels.app/auth/login`
2. Click **"Log in with Whop"**
3. **Use** the same email you used **to purchase**: [email]








You'll be redirected **to** your dashboard **and** can **start** uploading swing videos! 🎉

Questions? Hit reply **or** **check out** our help docs.

- The CatchBarrels Team








Success Metrics


After running the sync, you should see:

-  All Whop customers appear in `/admin/players`
-  Users can log in via Whop OAuth
-  Membership tiers match Whop products
-  VIP offers detected for assessment purchasers
-  Session limits enforced based on tier
-  No duplicate accounts
-  No sync errors in console

Summary

The Whop sync is now fully functional:

- **Backfills all existing customers** 
- **Creates accounts automatically** 
- **Syncs membership tiers** 
- **Handles VIP offers** 
- **Comprehensive logging** 
- **Proper error handling** 
- **Passwordless login via Whop OAuth** 

Status:  COMPLETE - Ready for production use