Work Order: Reboot Motion Data Ingestion (Admin-Only) - COMPLETE

Executive Summary

Successfully implemented Phase 1 of Reboot Motion integration for CatchBarrels. This is an **admin/coach-only** feature that ingests swing data from Reboot Motion (MLB, Pro, College, HS, Youth) into a dedicated database table. The data is isolated from the live player experience and available only in the Coach Control Room for future analysis, benchmarking, and comparison.

Key Achievement: Zero impact on player-facing app. All changes are behind feature flags and admin-only routes.

Implementation Overview

1. Database Schema

New Model: `RebootSession`

```
model RebootSession {
  id              String   @id @default(cuid())
  rebootSessionId String   @unique  // Unique ID from Reboot API
  rebootAthleteId String?  // Reboot's athlete identifier
  athleteName     String?  // Name from Reboot
  athleteEmail    String?  // Email from Reboot (if available)
  level           String?  // "MLB", "Pro", "College", "HS", "Youth"
  teamTag         String?  // Team or organization
  swingDate       DateTime? // When the swing was captured
  metrics         Json      // Full payload from Reboot API
  createdAt       DateTime @default(now())
  updatedAt       DateTime @updatedAt

  @@map("reboot_sessions")
  @@index([rebootAthleteId])
  @@index([level])
  @@index([swingDate])
  @@index([createdAt])
}
```

Updated Model: `User`

Added optional field:

```
model User {
  // ...existing fields...
  rebootAthleteId String? // Links this user to a Reboot athlete profile
}
```

Migration Applied: `yarn prisma db push` (no migration file due to drift)

2. Reboot API Client (Placeholder)

File: `lib/reboot/reboot-client.ts`

Status: Placeholder implementation with TODOs

Functions:
- `fetchRebootSessions()`: Fetches all sessions from Reboot API
- `fetchRebootAthleteSessions(athleteId)`: Fetches sessions for a specific athlete
- `fetchRebootAthletes()`: Gets list of all Reboot athletes

Configuration Required (in `.env`):

```
REBOOT_API_BASE_URL=https://api.rebootmotion.com/v1  # Example
REBOOT_API_KEY=your_api_key_here
```

Current Behavior: All functions throw errors with clear messages about missing configuration. This is intentional until Rick provides real API details.

---

3. Feature Flag

File: `lib/config/reboot-flags.ts`

```
export const REBOOT_SYNC_ENABLED = true;
```

Controls:
- `/admin/reboot` route visibility
- Reboot section in player detail page
- Sync API endpoint access

To Disable: Set to `false` and all Reboot UI disappears.

---

4. Admin API Endpoint

Endpoint: `POST /api/admin/reboot/sync`

File: `app/api/admin/reboot/sync/route.ts`

Behavior:
1. Checks `REBOOT_SYNC_ENABLED` flag
2. Verifies admin/coach role
3. Calls `fetchRebootSessions()` from Reboot API
4. Upserts each session into `RebootSession` table by `rebootSessionId`
5. Returns summary:

```
json
   {
     "status": "ok",
     "inserted": 123,
     "updated": 45,
     "total": 500,
     "message": "Successfully synced 168 sessions from Reboot Motion"
   }
```

Error Handling: Continues processing if individual sessions fail, logs errors.

---

5. Admin UI: Reboot Data Page

Route: `/admin/reboot`

Files:
- `app/admin/reboot/page.tsx` (server component)
- `app/admin/reboot/reboot-client.tsx` (client component)

Features:

Summary Stats

- Total sessions
- Breakdown by level: MLB, Pro, College, HS, Youth

Sync Button

- Calls `/api/admin/reboot/sync`
- Shows loading state
- Displays success/error toasts
- Auto-refreshes page on success

Session Table

- Displays most recent 200 sessions
- Shows: athlete name, level (color-coded badge), team, swing date, Reboot IDs
- Search by name/ID/team
- Filter by level dropdown

Detail Drawer

- Click any session to open side drawer
- Shows all session metadata
- Pretty-prints full metrics JSON (expandable)

Access: Admin/coach-only (protected by `/admin` layout)

---

6. Player Linking Feature

Location: `/admin/players/[id]` (Player Detail Page)

File: `app/admin/players/[id]/player-detail-client.tsx`

UI:

When NOT Linked

- Shows gray “Not linked to Reboot” card
- “Link Profile” button (navigates to `/admin/reboot`)
- Explanatory note about future use (comparison, benchmarking)

When Linked

- Shows purple “Linked to Reboot Athlete” card
- Displays `rebootAthleteId` in monospace font
- “Unlink” button (red, destructive action)
- Confirmation dialog before unlinking

API:
- `PATCH /api/admin/players/[id]/reboot-link` - Link user (body: `{rebootAthleteId: string}`)
- `DELETE /api/admin/players/[id]/reboot-link` - Unlink user

Note: Linking functionality UI is simplified for Phase 1. Full search/select modal deferred to Phase 2.

Files Created

Database

- `prisma/schema.prisma` (updated)

Backend

- `lib/reboot/reboot-client.ts` (new)
- `lib/config/reboot-flags.ts` (new)
- `app/api/admin/reboot/sync/route.ts` (new)
- `app/api/admin/players/[id]/reboot-link/route.ts` (new)

Frontend

- `app/admin/reboot/page.tsx` (new)
- `app/admin/reboot/reboot-client.tsx` (new)
- `app/admin/players/[id]/player-detail-client.tsx` (updated)

Documentation

- `docs/WO_REBOOT_IMPORT_V1_COMPLETE.md` (this file)

Safety Verification

✅ No Changes to Player-Facing App

- `/dashboard` - Unchanged
- `/sessions/*` - Unchanged
- `/video/*` - Unchanged
- All player routes - Unchanged

✅ No Changes to Existing Scoring

- `lib/scoring/newScoringEngine.ts` - Unchanged
- `lib/momentum-coaching.ts` - Unchanged
- Video analysis endpoints - Unchanged

✅ No Changes to Session Caps or VIP

- `lib/assessment-vip-config.ts` - Unchanged
- Whop integration - Unchanged

✅ Feature Flag Protection

- All Reboot UI guarded by `REBOOT_SYNC_ENABLED`
- Default: `true` (as requested)
- Can be disabled by setting flag to `false`

Testing Checklist

Build & Compilation

- [x] `yarn tsc --noEmit` - No TypeScript errors
- [x] `yarn build` - Successful Next.js build
- [x] `yarn prisma db push` - Database schema synced

Admin Routes

- [x] `/admin/reboot` page renders
- [x] Summary stats display correctly
- [x] Sync button visible and styled
- [x] Empty state message when no data

Player Detail Page

- [x] Reboot section appears when `REBOOT_SYNC_ENABLED = true`
- [x] “Not linked” state displays correctly
- [x] “Link Profile” button navigates to `/admin/reboot`

Feature Flag

- [x] Setting `REBOOT_SYNC_ENABLED = false` hides all Reboot UI
- [x] API endpoints return 403 when flag is disabled

Configuration Steps for Rick

Step 1: Get Reboot API Credentials

1. Contact Reboot Motion support/sales
2. Request API access for CatchBarrels integration
3. Obtain:
   - Base API URL (e.g., `https://api.rebootmotion.com/v1`)
   - API Key or Bearer Token
   - Confirm endpoint paths (e.g., `/sessions`, `/athletes`)

Step 2: Update Environment Variables

Add to `.env` file:

```
REBOOT_API_BASE_URL=https://api.rebootmotion.com/v1
REBOOT_API_KEY=your_actual_api_key_here
```

Step 3: Update API Client

Edit `lib/reboot/reboot-client.ts` and replace TODOs with real API calls:

```
export async function fetchRebootSessions(): Promise<RebootSessionPayload[]> {
  const response = await fetch(`${REBOOT_API_BASE}/sessions`, {
    headers: {
      'Authorization': `Bearer ${REBOOT_API_KEY}`,
      'Content-Type': 'application/json',
    },
  });

  if (!response.ok) {
    throw new Error(`Reboot API error: ${response.statusText}`);
  }

  const data = await response.json();
  return data.sessions; // Adjust based on actual response structure
}
```

Important: Map Reboot's field names to our `RebootSessionPayload` interface.

Step 4: Test Sync

1. Navigate to `/admin/reboot`
2. Click "Sync from Reboot"
3. Verify sessions appear in table
4. Check server logs for any mapping errors

TODOs for Future Phases

Phase 2: Comparison & Benchmarking

- [ ] Build MLB averages from Reboot data
- [ ] Calculate player percentiles vs level (HS, College, Pro, MLB)
- [ ] Add "Compare to Model" feature in video detail page
- [ ] Show ideal timing ranges from Reboot elite swings

Phase 3: Advanced Linking

- [ ] Add search modal in player detail for selecting Reboot athlete
- [ ] Fetch distinct athletes from `RebootSession` table
- [ ] Auto-suggest matches by name/email

Phase 4: New Timing/Sequence Ideas

- [ ] Extract kinematic sequence from Reboot metrics
- [ ] Compare player sequence to elite benchmarks
- [ ] Visualize energy transfer paths

Deployment Checklist

- [x] Database schema updated (`prisma db push`)
- [x] Prisma client regenerated
- [x] TypeScript compilation passes
- [x] Next.js build succeeds
- [x] Feature flag set to `true`
- [ ] Environment variables configured (requires Rick’s input)
- [ ] Reboot API client implemented (requires Rick’s input)

Summary for Rick

What’s Ready Now

1. **Database**: `RebootSession` table created, `User.rebootAthleteId` field added
2. **Admin UI**: Full Reboot data viewer at `/admin/reboot` with stats, search, filters, detail drawer
3. **Sync Button**: Triggers API sync (will work once you configure real API)
4. **Player Linking**: Manual link/unlink in player detail page
5. **API Endpoints**: Sync and link/unlink endpoints ready

What You Need to Do

1. **Get Reboot API Access**: Contact Reboot, get URL + API key
2. **Configure .env**: Add `REBOOT_API_BASE_URL` and `REBOOT_API_KEY`
3. **Update Client**: Replace TODOs in `lib/reboot/reboot-client.ts` with real API calls
4. **Test**: Click “Sync from Reboot” in `/admin/reboot`

What’s NOT Done (Intentionally)

- Reboot API calls (waiting for your credentials)
- Comparison/benchmarking features (Phase 2)
- Advanced linking UI (Phase 2)
- Using Reboot data in scoring (Phase 3+)

Questions or Issues?

If you encounter errors after configuring the API:

1. Check server logs for detailed error messages
2. Verify Reboot API response structure matches our expectations
3. Update field mappings in `fetchRebootSessions()` as needed

All Reboot code is isolated in `/lib/reboot/` and `/app/api/admin/reboot/` for easy debugging.

Status: ✅ Phase 1 Complete (Admin-Only Data Ingestion)
Next: Configure Reboot API credentials and test sync