

NLL ChampionData API - Actual Endpoints and Field Mapping

API Structure Overview

The NLL ChampionData API uses a hierarchical structure:
- Leagues → Levels → Seasons → Matches/Squads/Players

Note: "Squads" = Teams, "Matches" = Games, "Persons" = Players/Staff

Complete Endpoint List and Field Mappings

1. LEAGUE ENDPOINTS

GET /v1/leagues

Purpose: Get list of available leagues
Maps to: NLLLeagues entity

Database Field	API Response Field	Example Value
providerLeagueId	leagueId	"nll" or numeric ID
name	name	"National Lacrosse League"
gender	gender	"male"
country	country	"North America"

GET /v1/leagues/{leagueId}

Purpose: Get details for a specific league
Maps to: NLLLeagues entity (detailed info)

GET /v1/leagues/{leagueId}/levels

Purpose: Get available levels (divisions/conferences) for league
Maps to: Additional configuration data

Database Field	API Response Field	Example Value
levelId	levelId	Level identifier
levelName	name	"Professional"

GET /v1/leagues/{leagueId}/levels/{levelId}

Purpose: Get details for a specific level

2. SEASON ENDPOINTS

GET /v1/leagues/{leagueId}/levels/{levelId}/seasons

Purpose: Get list of seasons

Maps to: NLLLeagueYears entity

| Database Field | API Response Field | Example Value |

|-----|-----|-----|

| providerSeasonId | seasonId | "2024" |

| year | year or name | 2024 |

| startDate | startDate | "2024-12-01T00:00:00Z" |

| endDate | endDate | "2025-05-31T23:59:59Z" |

| isActive | isActive | true |

GET /v1/leagues/{leagueId}/levels/{levelId}/seasons/{seasonId}

Purpose: Get specific season details

Maps to: NLLLeagueYears entity (detailed)

3. TEAM (SQUAD) ENDPOINTS

GET /v1/leagues/{leagueId}/levels/{levelId}/seasons/{seasonId}/squads

Purpose: Get all teams in a season

Maps to: NLLTeams entity

| Database Field | API Response Field | Example Value |

|-----|-----|-----|

| providerTeamId | squadId | "buf" or numeric |

| name | name | "Buffalo Bandits" |

| shortName | abbreviation or code | "BUF" |

| city | city or location | "Buffalo" |

| logo | logoUrl | URL to logo |

| conference | conference | "East" |

| division | division | "North" |

GET /v1/leagues/{leagueId}/levels/{levelId}/seasons/{seasonId}/squads/{squadId}

Purpose: Get specific team details

Maps to: NLLTeams entity (detailed)

4. PLAYER ENDPOINTS

GET /v1/leagues/{leagueId}/levels/{levelId}/seasons/{seasonId}/players

Purpose: Get all players in a season

Maps to: NLLAthletes entity

| Database Field | API Response Field | Example Value |

|-----|-----|-----|

| providerId | playerId | Player ID |

| firstName | firstName | "Josh" |

| lastName | lastName | "Byrne" |

| jerseyNumber | jerseyNumber | 22 |

| position | position | "Attack" |

dob	dateOfBirth	"1992-05-15"
nationality	nationality	"Canada"
height	height	183
weight	weight	86

GET

/v1/leagues/{leagueId}/levels/{levelId}/seasons/{seasonId}/players/{playerId}

Purpose: Get specific player details
Maps to: NLLAthletes entity (detailed)

GET

/v1/leagues/{leagueId}/levels/{levelId}/seasons/{seasonId}/squads/{squadId}/persons

Purpose: Get team roster (players and staff)
Maps to: NLLAthletes entity with team association
Database Field	API Response Field	Example Value
teamId	Derived from squadId param	Foreign key
providerId	personId	Person ID
role	role	"player", "coach"

5. MATCH (GAME) ENDPOINTS

GET /v1/leagues/{leagueId}/levels/{levelId}/seasons/{seasonId}/schedule

Purpose: Get season schedule
Maps to: NLLEvents entity
Database Field	API Response Field	Example Value
providerGameId	matchId	Match ID
startDate	startTime or scheduledTime	"2024-10-15T19:00:00Z"
homeTeamId	homeSquadId	Home team reference
awayTeamId	awaySquadId	Away team reference
eventLocation	venueName	"KeyBank Center"
eventStatus	status	"scheduled", "live", "completed"
week	weekNumber	5

GET /v1/matches/{matchId}

Purpose: Get match details
Maps to: NLLEvents entity (detailed)
Database Field	API Response Field	Example Value
providerGameId	matchId	Match ID
attendance	attendance	8543
broadcast	broadcast	"ESPN+"
currentQuarter	currentPeriod	3
elapsedTime	elapsedTime	"08:45"

GET /v1/matches/{matchId}/venue

Purpose: Get venue details
Maps to: NLLEvents.eventLocation (detailed)

GET /v1/matches/{matchId}/score**Purpose:** Get match scores**Maps to:** **NLLScores** entity

| Database Field | API Response Field | Example Value |

|-----|-----|-----|

| **homeScore** | **homeScore** | 12 || **awayScore** | **awayScore** | 10 || **periodScores** | **periodScores**[] | Array of period scores || **roundId** | Period number (1-4, OT) | 1, 2, 3, 4, 5 |**Period Score Mapping:**

```

periodScores.forEach((period, index) => {
  // Home team period score
  {
    teamId: homeTeamId,
    eventId: matchId,
    roundId: index + 1, // Q1=1, Q2=2, etc.
    goals: period.homeScore,
    isHomeTeam: true
  }
  // Away team period score
  {
    teamId: awayTeamId,
    eventId: matchId,
    roundId: index + 1,
    goals: period.awayScore,
    isHomeTeam: false
  }
});

```

6. STATISTICS ENDPOINTS**GET** /v1/matches/{matchId}/statistics/players**Purpose:** Get player statistics for a match**Maps to:** **NLLPlayerStats** entity

| Database Field | API Response Field | Example Value |

|-----|-----|-----|

| **playerId** | **playerId** | Player reference || **eventId** | **matchId** | Match reference || **goals** | **goals** | 2 || **assists** | **assists** | 3 || **points** | **points** | 5 || **shots** | **shots** | 8 || **groundBalls** | **groundBalls** | 4 || **causedTurnovers** | **causedTurnovers** | 2 || **turnovers** | **turnovers** | 1 || **penaltyMinutes** | **penaltyMinutes** | 2 || **faceoffWins** | **faceoffWins** | 8 || **faceoffLosses** | **faceoffLosses** | 5 |**GET** /v1/matches/{matchId}/statistics/squads**Purpose:** Get team statistics for a match**Maps to:** Team statistics

GET

`/v1/leagues/{leagueId}/levels/{levelId}/seasons/{seasonId}/statistics/players`

Purpose: Get season player statistics

Maps to: Season aggregated stats

GET `/v1/leagues/{leagueId}/levels/{levelId}/seasons/{seasonId}/statistics/squads`

Purpose: Get season team statistics

Maps to: Season team stats

7. GAME EVENT ENDPOINTS

GET `/v1/matches/{matchId}/faceoffs`

Purpose: Get faceoff details

Maps to: Game events table

Database Field	API Response Field	Example Value
	<code>winnerId</code>	<code>winningPlayerId</code> Player who won
	<code>loserId</code>	<code>losingPlayerId</code> Player who lost
	<code>period</code>	<code>period</code> 1-4, OT
	<code>time</code>	<code>time</code> "05:23"
	<code>location</code>	<code>location</code> "center"

GET `/v1/matches/{matchId}/penalties`

Purpose: Get penalty details

Maps to: Penalties table

Database Field	API Response Field	Example Value
	<code>playerId</code>	<code>playerId</code> Penalized player
	<code>penaltyType</code>	<code>type</code> "slashing"
	<code>minutes</code>	<code>minutes</code> 2
	<code>period</code>	<code>period</code> Quarter number
	<code>time</code>	<code>time</code> "10:15"

GET `/v1/matches/{matchId}/shots`

Purpose: Get shot details

Maps to: Shots/Goals table

Database Field	API Response Field	Example Value
	<code>shooterId</code>	<code>playerId</code> Shooting player
	<code>result</code>	<code>result</code> "goal", "save", "miss"
	<code>period</code>	<code>period</code> Quarter number
	<code>time</code>	<code>time</code> "12:30"
	<code>shotType</code>	<code>type</code> Shot type

8. STANDINGS ENDPOINT

GET /v1/leagues/{leagueId}/levels/{levelId}/seasons/{seasonId}/standings

Purpose: Get current standings

Maps to: **NLLStandings** entity

| Database Field | API Response Field | Example Value |

|-----|-----|-----|

| **teamId** | **squadId** | Team reference |

| **wins** | **wins** | 8 |

| **losses** | **losses** | 4 |

| **overtimeLosses** | **overtimeLosses** | 1 |

| **points** | **points** | 17 |

| **goalsFor** | **goalsFor** | 145 |

| **goalsAgainst** | **goalsAgainst** | 132 |

| **homeWins** | **homeWins** | 5 |

| **homeLosses** | **homeLosses** | 1 |

| **awayWins** | **awayWins** | 3 |

| **awayLosses** | **awayLosses** | 3 |

| **divisionRank** | **divisionRank** | 2 |

| **conferenceRank** | **conferenceRank** | 4 |

| **streak** | **streak** | "W3" |

GET

/v1/leagues/{leagueId}/levels/{levelId}/seasons/{seasonId}/standings/{weekNumber}

Purpose: Get standings after specific week

Maps to: Historical standings

9. SPECIAL ENDPOINTS

GET /v1/matches/{matchId}/flow/metric

Purpose: Get game flow/timeline of events

Maps to: Play-by-play data

GET /v1/matches/{matchId}/metrics

Purpose: Get available metrics for a match

Maps to: Configuration data

GET /v1/playermapping

Purpose: Get player ID mappings

Maps to: Player mapping table

GET /v1/leagues/{leagueId}/levels/{levelId}/seasons/{seasonId}/conferences

Purpose: Get conference structure

Maps to: Conference configuration

Implementation Guide

1. Initial Data Load Sequence

```
// Step 1: Get league structure
const leagues = await fetch(`/v1/leagues`);
const leagueId = leagues[0].leagueId; // Probably "nll"

// Step 2: Get levels (should be just one for NLL)
const levels = await fetch(`/v1/leagues/${leagueId}/levels`);
const levelId = levels[0].levelId;

// Step 3: Get current season
const seasons = await fetch(`/v1/leagues/${leagueId}/levels/${levelId}/seasons`);
const currentSeason = seasons.find(s => s.isActive);
const seasonId = currentSeason.seasonId;

// Step 4: Get teams
const teams = await fetch(`/v1/leagues/${leagueId}/levels/${levelId}/seasons/${seasonId}/squads`);

// Step 5: Get schedule
const schedule = await fetch(`/v1/leagues/${leagueId}/levels/${levelId}/seasons/${seasonId}/schedule`);

// Step 6: Get players
const players = await fetch(`/v1/leagues/${leagueId}/levels/${levelId}/seasons/${seasonId}/players`);
```

2. Live Game Sync

```
// Get today's games from schedule
const todayGames = schedule.filter(game => isToday(game.startTime));

// For each live game, sync scores
for (const game of todayGames) {
  if (game.status === 'live') {
    const matchId = game.matchId;

    // Get current score
    const score = await fetch(`/v1/matches/${matchId}/score`);

    // Get player statistics
    const playerStats = await fetch(`/v1/matches/${matchId}/statistics/players`);

    // Update database
    await updateScores(matchId, score);
    await updatePlayerStats(matchId, playerStats);
  }
}
```

3. Sync Schedule

Data Type	Endpoint	Sync Frequency
League Structure	/v1/leagues	Once at setup
Current Season	/v1/leagues/{id}/levels/{id}/seasons	Daily
Teams	/v1/leagues/{id}/levels/{id}/seasons/{id}/squads	Daily
Schedule	/v1/leagues/{id}/levels/{id}/seasons/{id}/schedule	Every 6 hours
Live Scores	/v1/matches/{matchId}/score	Every 30 seconds during games
Player Stats	/v1/matches/{matchId}/statistics/players	Every minute during games
Standings	/v1/leagues/{id}/levels/{id}/seasons/{id}/standings	After each game

4. Status Mapping

API Status	Database EventStatus	Value
"scheduled"	UPCOMING	2
"live" or "in_progress"	LIVE	1
"completed" or "final"	COMPLETED	3
"cancelled"	CANCELLED	4
"postponed"	POSTPONED	5

5. Position Mapping

API Position	Database Position
"Attack"	"attack"
"Defense"	"defense"
"Transition"	"transition"
"Goalie"	"goalie"

Critical Notes for Developers

1. API Terminology

- **Squads** = Teams (use `squadId` for `teamId`)
- **Matches** = Games (use `matchId` for `gameId`)
- **Persons** = Players and staff (use `personId` or `playerId`)
- **Levels** = League divisions/tiers (NLL likely has one level)

2. Required Parameters

Most endpoints require the full hierarchy:

- **leagueId** (e.g., "nll")
- **levelId** (e.g., "1" or "professional")
- **seasonId** (e.g., "2024")

3. Data Relationships

```
League
├── Level
│   └── Season
│       ├── Squads (Teams)
│       ├── Players
│       ├── Schedule (Matches)
│       └── Standings
```

4. Missing Endpoints

The API does NOT have direct endpoints for:

- **/api/v1/teams** - Must use **/v1/leagues/{id}/levels/{id}/seasons/{id}/squads**
- **/api/v1/games** - Must use **/v1/leagues/{id}/levels/{id}/seasons/{id}/schedule**
- **/api/v1/rosters** - Must use **/v1/leagues/{id}/levels/{id}/seasons/{id}/squads/{id}/persons**

5. Authentication

All endpoints require authentication. Include Bearer token in headers:

```
headers: {
  'Authorization': `Bearer ${accessToken}`,
  'Content-Type': 'application/json'
}
```

Database Schema Requirements

Core Tables Needed

1. **NLLLeagues** - Store league info
2. **NLLLevels** - Store level info (new table needed)
3. **NLLLeagueYears** - Store seasons
4. **NLLTeams** - Store squads
5. **NLLAthletes** - Store players
6. **NLLEvents** - Store matches
7. **NLLEventTeams** - Store match participants
8. **NLLScores** - Store period and final scores
9. **NLLPlayerStats** - Store player statistics
10. **NLLStandings** - Store standings
11. **NLLPenalties** - Store penalties
12. **NLLFaceoffs** - Store faceoffs

13. **NLLShots** - Store shots

Additional Considerations

- Add **levelId** to season/league relationships
- Use **squadId** as **providerTeamId**
- Use **matchId** as **providerGameId**
- Use **personId** or **playerId** as **providerId**

Next Steps

1. **Obtain OAuth2 Credentials** from ChampionData
2. **Test Authentication** with the API
3. **Verify Response Structures** for each endpoint
4. **Implement Data Sync** following the hierarchy: League → Level → Season → Data
5. **Map Fields** based on actual response structures