Expected Data Format from Scrapers

This document shows the expected data structure when scrapers successfully retrieve player data.

NBA.com Scraper (`commonallplayers` endpoint)

Expected Response Structure:

```
{
  "resultSets": [
    {
      "name": "CommonAllPlayers",
      "headers": [
        "PERSON_ID",
        "DISPLAY_LAST_COMMA_FIRST",
        "DISPLAY_FIRST_LAST",
        "ROSTERSTATUS",
        "FROM_YEAR",
        "TO_YEAR",
        "PLAYERCODE",
        "TEAM_ID",
        "TEAM_CITY",
        "TEAM_NAME",
        "TEAM_ABBREVIATION",
        "TEAM_CODE",
        "GAMES_PLAYED_FLAG"
      ],
      "rowSet": [
        [
          203081,
          "Antetokounmpo, Giannis",
          "Giannis Antetokounmpo",
          1,
          "2013",
          "2024",
          "giannis_antetokounmpo",
          1610612749,
          "Milwaukee",
          "Bucks",
          "MIL",
          "bucks",
          "Y"
        ],
        ...
      ]
    }
  ]
}
```

Scraped Player Data Fields:

```
{
    "name": "Giannis Antetokounmpo",
    "position": "PF",
    "height_inches": 83,  # 6'11"
    "weight_lbs": 242,
    "wingspan_inches": None,  # Not available from API
    "arm_length_inches": None,  # Not available from API
    "body_type": None,  # Inferred later
    "dominant_hand": None,  # Not available from API
    "career_fg_percentage": 54.2,
    "career_3pt_percentage": 29.1,
    "career_ft_percentage": 71.4,
    "shooting_style": None,  # Analyzed from video
    "era": "Modern",
    "skill_level": "Professional",
    "profile_image_url": "https://cdn.nba.com/headshots/nba/latest/
1040x760/203081.png",
    "team": "Milwaukee Bucks",
    "player_id_nba": 203081,
    "scraped_at": "2025-12-13T07:49:18.123456"
}
```

Sample CSV Output (if successful):

name	position	height_inches	weight_lbs	career_3pt_percentage	career_fg_percentage	career_ft_percentage	team
Stephen Curry	PG	75	185	42.8	47.3	90.8	Golden State Warriors
Klay Thompson	SG	78	220	41.9	45.3	85.6	Dallas Mavericks
Damian Lillard	PG	75	195	37.3	43.7	89.2	Milwaukee Bucks
Kevin Durant	SF	82	240	38.6	50.3	88.5	Phoenix Suns
James Harden	SG	77	220	36.4	44.3	85.7	LA Clippers

Basketball-Reference Scraper

Expected HTML Table Structure:

```
<table id="per_game_stats">
  <thead>
    <tr>
      <th>Rank</th>
      <th>Player</th>
      <th>Pos</th>
      <th>Age</th>
      <th>Tm</th>
      <th>G</th>
      <th>FG%</th>
      <th>3P%</th>
      <th>FT%</th>
      ...
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td><a href="/players/c/curryst01.html">Stephen Curry</a></td>
      <td>PG</td>
      <td>36</td>
      <td>GSW</td>
      <td>74</td>
      <td>.450</td>
      <td>.408</td>
      <td>.921</td>
      ...
    </tr>
    ...
  </tbody>
</table>
```

Scraped Player Data Fields:

```
{
    "name": "Stephen Curry",
    "position": "PG",
    "height_inches": 75,  # 6'3"
    "weight_lbs": 185,
    "wingspan_inches": None,  # Not available
    "career_fg_percentage": 47.3,
    "career_3pt_percentage": 42.8,
    "career_ft_percentage": 90.8,
    "career_games": 941,
    "career_points": 23668,
    "career_assists": 5897,
    "seasons_played": 15,
    "era": "Modern",
    "skill_level": "Professional",
    "player_url": "https://www.basketball-reference.com/players/c/curryst01.html",
    "scraped_at": "2025-12-13T07:53:06.123456"
}
```

Sample CSV Output (if successful):

name	position	height_inches	weight_lbs	career_3pt_percentage	career_fg_percentage	career_ft_percentage	career_games
Steve Nash	PG	75	178	42.8	49.0	90.4	1217
Stephen Curry	PG	75	185	42.8	47.3	90.8	941
Ray Allen	SG	77	205	40.0	45.2	88.1	1300
Reggie Miller	SG	78	185	39.5	47.1	88.8	1389
Larry Bird	SF	81	220	37.6	49.6	88.6	897

Database Schema

Shooter Table Structure:

```
CREATE TABLE shooters (
    id SERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    position VARCHAR(10),
    height_inches INTEGER,
    weight_lbs INTEGER,
    wingspan_inches INTEGER,
    arm_length_inches INTEGER,
    body_type VARCHAR(50),
    dominant_hand VARCHAR(10),

    -- Career Statistics
    career_fg_percentage DECIMAL(5,2),
    career_3pt_percentage DECIMAL(5,2),
    career_ft_percentage DECIMAL(5,2),
    career_games INTEGER,
    career_points INTEGER,
    career_assists INTEGER,
    seasons_played INTEGER,

    -- Metadata
    shooting_style TEXT,
    era VARCHAR(50),
    skill_level VARCHAR(50),
    profile_image_url TEXT,
    team VARCHAR(100),
    player_id_nba INTEGER,
    player_url TEXT,

    -- Tracking
    scraped_at TIMESTAMP,
    created_at TIMESTAMP DEFAULT NOW(),
    updated_at TIMESTAMP DEFAULT NOW()
);
```

Example INSERT Statements:

```
INSERT INTO shooters (
    name, position, height_inches, weight_lbs,
    career_fg_percentage, career_3pt_percentage, career_ft_percentage,
    era, skill_level, team, player_id_nba, scraped_at
) VALUES
(
    'Stephen Curry', 'PG', 75, 185,
    47.3, 42.8, 90.8,
    'Modern', 'Professional', 'Golden State Warriors', 201939,
    NOW()
),
(
    'Klay Thompson', 'SG', 78, 220,
    45.3, 41.9, 85.6,
    'Modern', 'Professional', 'Dallas Mavericks', 202691,
    NOW()
);
```

Data Processing Pipeline (When Scrapers Work)

Step 1: Scrape Data

```
python scrapers/nba_scraper.py --limit 100
```

- Fetches 100 players from NBA API
- Retrieves detailed stats for each player
- Saves to `nba_players_2024-25.csv`

Step 2: Download Images

```
python storage/batch_download_images.py --csv nba_players_2024-25.csv
```

- Downloads player headshots from CDN
- Uploads to S3 bucket (if configured)
- Updates CSV with image URLs

Step 3: Insert to Database

```
python main.py --scrape nba --limit 100
```

- Reads scraped data
- Validates and cleans records
- Inserts into PostgreSQL database
- Updates existing records if already present

Step 4: Verify Data

```
python -c "
from database import get_db_session
from sqlalchemy import text

session = get_db_session()
result = session.execute(text('SELECT COUNT(*) FROM shooters'))
count = result.scalar()
print(f'Total shooters in database: {count}')

result = session.execute(text('SELECT name, career_3pt_percentage FROM shooters ORDER
BY career_3pt_percentage DESC LIMIT 5'))
print('\\nTop 5 shooters by 3PT%:')
for row in result:
    print(f'  {row[0]}: {row[1]}%')
"
```

API Response Times (Expected)

When working properly:
- **NBA API**: 0.5-2 seconds per request

- **Basketball-Reference**: 1-3 seconds per page
- **Player Details**: 2-5 seconds per player (multiple API calls)
- **Image Download**: 0.5-1 second per image

With current blocking:
- **NBA API**: Timeout after 20-30 seconds ❌
- **Basketball-Reference**: 403 Forbidden immediately ❌

Success Metrics

Good Scraping Session:

- ✅ Response time < 3 seconds per request
- ✅ 0% timeout errors
- ✅ 0% 403/429 errors
- ✅ 90%+ data completeness (all fields populated)
- ✅ All images successfully downloaded

Current Session (Blocked):

- ❌ Response time: Timeout
- ❌ Timeout errors: 100% (NBA)
- ❌ 403 errors: 100% (Basketball-Reference)
- ❌ Data completeness: 0%
- ❌ Images downloaded: 0

Sample Data for Testing

If you need to test the database insertion without scraping, use this sample data:

```
sample_shooters = [
    {
        "name": "Stephen Curry",
        "position": "PG",
        "height_inches": 75,
        "weight_lbs": 185,
        "career_fg_percentage": 47.3,
        "career_3pt_percentage": 42.8,
        "career_ft_percentage": 90.8,
        "era": "Modern",
        "skill_level": "Professional",
        "team": "Golden State Warriors",
        "player_id_nba": 201939
    },
    {
        "name": "Klay Thompson",
        "position": "SG",
        "height_inches": 78,
        "weight_lbs": 220,
        "career_fg_percentage": 45.3,
        "career_3pt_percentage": 41.9,
        "career_ft_percentage": 85.6,
        "era": "Modern",
        "skill_level": "Professional",
        "team": "Dallas Mavericks",
        "player_id_nba": 202691
    },
    {
        "name": "Damian Lillard",
        "position": "PG",
        "height_inches": 75,
        "weight_lbs": 195,
        "career_fg_percentage": 43.7,
        "career_3pt_percentage": 37.3,
        "career_ft_percentage": 89.2,
        "era": "Modern",
        "skill_level": "Professional",
        "team": "Milwaukee Bucks",
        "player_id_nba": 203081
    }
]
```

Save this to a JSON file and use it for testing:

```python
import json
from database import get_db_session
from models import Shooter

# Load sample data
with open('sample_shooters.json', 'r') as f:
    shooters = json.load(f)

# Insert to database
session = get_db_session()
for shooter_data in shooters:
    shooter = Shooter(**shooter_data)
    session.add(shooter)

session.commit()
print(f"Inserted {len(shooters)} sample shooters")
```

Conclusion

This document shows the expected data format and structure when the scrapers successfully retrieve data from NBA.com and Basketball-Reference. Currently, both sites are blocking automated requests, but the scraper code is ready to process data in these formats once the blocking issues are resolved.