ChatGPT

Integrating SportsDataIO into BlazeSportsIntel

The SportsDataIO API provides REST-based endpoints for multiple sports. Endpoints are grouped into **competition feeds** (standings, teams/players/rosters, venues), **event feeds** (schedules, scores, team & player stats, play-by-play) and **player feeds** (depth charts, line-ups & injuries). The API uses HTTP **GET** requests and requires an API key as a query parameter (`?key=YOUR_API_KEY`) or via the `Ocp-Apim-Subscription-Key` header. The SportsDataIO developer portal lists the available feeds – standings, teams/players/rosters, schedules & game-day info, scores/game state, team & player stats, play-by-play, depth charts/injuries and others [1]. The same patterns are followed across NFL, MLB, College Football (CFB) and College Basketball (CBB) APIs.

General Notes

- **Authentication:** attach your API key (e.g., `d34e00995322470fb49b6d02a8a4df33`) either as a `key` query parameter or via the `Ocp-Apim-Subscription-Key` header. The example calls below use the query-parameter method.
- **Season parameter:** most endpoints accept a `season` (e.g., `2025` for current year, `2024` for previous season). Post-season data uses `SEASONPOST` (e.g., `2024POST`).
- **Filtering for SEC teams (college):** SEC teams can be isolated by filtering the returned team or standings JSON by the `Conference` field equal to `SEC`.
- **Data Format:** responses are JSON arrays of objects containing fields like `Team`, `Name`, `Wins`, `Losses`, `Conference`, `Division` and detailed player statistics. For example, the NFL standings endpoint returns an array of team records with wins/losses and ranking fields [2]; the MLB standings endpoint includes runs scored/against and home/away splits [3].

NFL (National Football League)

Purpose	Endpoint	Notes
Standings (current or historical)	`https://api.sportsdata.io/v3/nfl/scores/json/Standings/{season}?key=YOUR_KEY`	Returns regular-season standings for all teams. Each record includes conference/division rank, wins, losses and streak [2]. Example: `/Standings/2025` for the 2025 regular season.
Teams	`https://api.sportsdata.io/v3/nfl/scores/json/Teams?key=YOUR_KEY`	Returns all NFL teams with IDs, keys, city, name and stadium information. Use `TeamID` or `Key` for roster/stats queries.

Purpose	Endpoint	Notes
Rosters / Players	`https://api.sportsdata.io/v3/nfl/scores/json/Players?key=YOUR_KEY` or `.../Players/{teamId}?key=YOUR_KEY`	Returns players with position, status and additional fields. Filter by `TeamID` for team-specific rosters.
Depth Charts / Injuries	`https://api.sportsdata.io/v3/nfl/scores/json/DepthCharts?key=YOUR_KEY`	Provides depth-chart positions and injury status for each team.
Team season stats	`https://api.sportsdata.io/v3/nfl/stats/json/TeamSeasonStats/{season}?key=YOUR_KEY`	Aggregated team statistics (points scored, yards, defensive stats) for the entire season.
Player season stats	`https://api.sportsdata.io/v3/nfl/stats/json/PlayerSeasonStats/{season}?key=YOUR_KEY`	Comprehensive offensive/defensive statistics per player (yards, touchdowns, receptions, tackles, etc.). Useful for historical analysis.
Team game stats (week-by-week)	`https://api.sportsdata.io/v3/nfl/stats/json/TeamGameStats/{season}/{week}?key=YOUR_KEY`	Stats for each team in a specific week; use loops to build full season timeline.
Player game stats	`https://api.sportsdata.io/v3/nfl/stats/json/PlayerGameStatsByWeek/{season}/{week}?key=YOUR_KEY`	Per-player game statistics for a given week.
Schedules & scores	`https://api.sportsdata.io/v3/nfl/scores/json/Games/{season}?key=YOUR_KEY`	Lists all games for a season with dates, home/away teams, final scores and status.

MLB (Major League Baseball)

Purpose	Endpoint	Notes
Standings	`https://api.sportsdata.io/v3/mlb/scores/json/Standings/{season}?key=YOUR_KEY`	Returns standings by league/division with wins, losses, runs scored/against and streaks [3]. Example: `/Standings/2025` for the current MLB season.
Teams	`https://api.sportsdata.io/v3/mlb/scores/json/Teams?key=YOUR_KEY`	Returns all MLB teams with IDs, location, league/division and ballpark details.

Purpose	Endpoint	Notes
Rosters / Players	`https://api.sportsdata.io/v3/mlb/scores/json/Players?key=YOUR_KEY` or `.../Players/{teamId}?key=YOUR_KEY`	Returns player list with positions, batting/throwing hands, height/weight and status. Use `TeamID` for team-specific rosters.
Depth Charts	`https://api.sportsdata.io/v3/mlb/scores/json/DepthCharts?key=YOUR_KEY`	Provides projected lineups, starting pitchers and bullpen roles.
Team season stats	`https://api.sportsdata.io/v3/mlb/stats/json/TeamSeasonStats/{season}?key=YOUR_KEY`	Aggregated hitting, pitching and fielding statistics per team.
Player season stats	`https://api.sportsdata.io/v3/mlb/stats/json/PlayerSeasonStats/{season}?key=YOUR_KEY`	Individual player statistics (AVG, OBP, HR, RBI, ERA, WHIP, etc.) for the chosen season.
Team game stats (by date)	`https://api.sportsdata.io/v3/mlb/stats/json/TeamGameStatsByDate/{date}?key=YOUR_KEY`	Team statistics for all games on a specific date (format `YYYY-MMM-DD`, e.g., `2025-JUL-10`).
Player game stats	`https://api.sportsdata.io/v3/mlb/stats/json/PlayerGameStatsByDate/{date}?key=YOUR_KEY`	Per-player statistics for all games on a given date.
Schedules & scores	`https://api.sportsdata.io/v3/mlb/scores/json/Games/{season}?key=YOUR_KEY`	Full schedule with game status, start times, final scores and venues.

College Football (CFB) – SEC teams only

Purpose	Endpoint	Notes
Teams / rosters	`https://api.sportsdata.io/v3/cfb/scores/json/Teams?key=YOUR_KEY`	Returns all NCAA FBS teams with `TeamID`, `School`, `Conference` and other metadata. Filter the returned JSON to `Conference == 'SEC'` to isolate SEC teams.
Standings / rankings	`https://api.sportsdata.io/v3/cfb/scores/json/Standings/{season}?key=YOUR_KEY`	Provides regular-season standings; filter for SEC teams. (If not available, use the `SeasonTeamStats` endpoint below and compute wins/losses.)

Purpose	Endpoint	Notes
Team season stats	`https://api.sportsdata.io/v3/cfb/stats/json/TeamSeasonStats/{season}?key=YOUR_KEY`	Aggregated team statistics (points for/against, offensive yards, defensive metrics). Filter results for SEC teams using the `Conference` field.
Player season stats	`https://api.sportsdata.io/v3/cfb/stats/json/PlayerSeasonStats/{season}?key=YOUR_KEY`	Comprehensive per-player statistics across the season. Filter players whose `Team` belongs to the SEC.
Team game stats	`https://api.sportsdata.io/v3/cfb/stats/json/TeamGameStats/{season}/{week}?key=YOUR_KEY`	Game-level statistics per team; use loops across weeks. Filter for SEC teams.
Player game stats	`https://api.sportsdata.io/v3/cfb/stats/json/PlayerGameStatsByWeek/{season}/{week}?key=YOUR_KEY`	Per-player stats by week; filter for SEC teams.
Depth charts / injuries	`https://api.sportsdata.io/v3/cfb/scores/json/DepthCharts?key=YOUR_KEY`	Depth charts with starters and injury status.
Schedules & scores	`https://api.sportsdata.io/v3/cfb/scores/json/Games/{season}?key=YOUR_KEY`	Full schedule with results, start times and neutral-site flags. Filter by `HomeTeam` or `AwayTeam` belonging to an SEC team.

College Basketball (CBB) – SEC teams only

Purpose	Endpoint	Notes
Teams / rosters	`https://api.sportsdata.io/v3/cbb/scores/json/Teams?key=YOUR_KEY`	Lists NCAA men's Division I basketball teams. Filter by `Conference == 'SEC'` for SEC teams.
Standings / rankings	`https://api.sportsdata.io/v3/cbb/scores/json/Standings/{season}?key=YOUR_KEY` or `…/Rankings/{season}`	Returns conference standings or national rankings. Filter for SEC teams.
Team season stats	`https://api.sportsdata.io/v3/cbb/stats/json/TeamSeasonStats/{season}?key=YOUR_KEY`	Aggregated team statistics (points per game, rebounds, assists, turnovers, defensive metrics). Filter by SEC.

Purpose	Endpoint	Notes
Player season stats	`https://api.sportsdata.io/v3/cbb/stats/json/PlayerSeasonStats/{season}?key=YOUR_KEY`	Individual statistics per player (points per game, rebounds, assists, shooting percentages). Filter by SEC teams.
Team game stats	`https://api.sportsdata.io/v3/cbb/stats/json/TeamGameStatsByDate/{date}?key=YOUR_KEY`	Game-level statistics for all teams on a date; filter by SEC.
Player game stats	`https://api.sportsdata.io/v3/cbb/stats/json/PlayerGameStatsByDate/{date}?key=YOUR_KEY`	Player statistics for games on a specific date; filter by SEC.
Depth charts / injuries	`https://api.sportsdata.io/v3/cbb/scores/json/DepthCharts?key=YOUR_KEY`	Depth charts and injury status.
Schedules & scores	`https://api.sportsdata.io/v3/cbb/scores/json/Games/{season}?key=YOUR_KEY`	Season schedule with game results and TV information. Filter by SEC teams.

Setting up BlazeSportsIntel.com

1. **Store the API key securely** – do not expose it client-side. Use server-side code (e.g., Node.js, Python or the existing BlazeSportsIntel backend) to make API calls.
2. **Create update scripts** for each league. For example, a cron job can periodically call the endpoints above to fetch updated standings, team stats and player stats. Use the same pattern across leagues.
3. **Historical data ingestion:** iterate over past seasons (e.g., 2010–2024) and call the season-based endpoints (`TeamSeasonStats/{season}`, `PlayerSeasonStats/{season}`, `Standings/{season}`) to build a historical database. For playoff data, use `{season}POST` as the season parameter.
4. **Real-time updates:** for live scores and game stats, poll `Games/{season}` or `TeamGameStats/{season}/{week}`/`…ByDate/{date}` at a suitable frequency (e.g., every few minutes during games). Use event feeds (scores/game state and play-by-play) if detailed real-time data is needed [4].
5. **Filter for SEC teams** when processing NCAA football/basketball data by checking the `Conference` field. Only store or display records where `Conference == 'SEC'`.
6. **Map IDs across leagues** using the `TeamID`, `PlayerID` and `GlobalTeamID` fields returned by the API. These identifiers remain consistent across seasons.
7. **Handle rate limits and errors:** although SportsDataIO offers high call volumes, implement retry logic and respect any rate-limit headers. On error responses (HTTP 4xx/5xx), log and retry or skip.

By following the endpoints and patterns above, BlazeSportsIntel.com can programmatically ingest **current standings**, **historical season stats**, **rosters**, **depth charts**, **schedules** and **player statistics** across NFL, MLB, SEC football and SEC basketball. Use your API key as shown in the sample calls and filter data according to your requirements.

[1] [4] Getting Started with Sports Data APIs | SportsDataIO
https://sportsdata.io/developers

[2] api.sportsdata.io
https://api.sportsdata.io/v3/nfl/scores/json/Standings/2025

[3] api.sportsdata.io
https://api.sportsdata.io/v3/mlb/scores/json/Standings/2025