Package 'itscalledsoccer'

May 8, 2026

Title American Soccer Analysis API Client

Version 0.3.2

Description Provides a wrapper around the same API <https://app.americansocceranalysis.com/api/v1/__docs__/> that powers the American Soccer Analysis app.

License MIT + file LICENSE

URL https://github.com/American-Soccer-Analysis/itscalledsoccer-r, https://american-soccer-analysis.github.io/itscalledsoccer-r/

BugReports https://github.com/American-Soccer-Analysis/itscalledsoccer-r/issues

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 3.2.0)

Imports dplyr (>= 1.1.0), httpcache (>= 1.2.0), glue (>= 1.4.1), httr (>= 1.4.2), jsonlite (>= 1.7.0), magrittr (>= 2.0.0), R6 (>= 2.5.0), tidyr (>= 1.1.1), stringi (>= 1.5.3), rlang (>= 0.4.10), data.table (>= 1.13.0), cli, methods

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Tyler Richardett [aut, cre], Akshay Easwaran [aut], American Soccer Analysis [cph]

Maintainer Tyler Richardett <tyler.richardett@gmail.com>

Repository CRAN

Date/Publication 2025-04-18 21:50:02 UTC

Contents

AmericanSoccerAnalysis

American Soccer Analysis API Client

Description

Class representing an active session connected to the American Soccer Analysis public API.

Details

Does not take any arguments to initialize.

Public fields

`API_VERSION` Latest API version.

`MAX_API_LIMIT` Maximum number of requests returned by the API by default.

`LEAGUES` List of stylized league names.

`base_url` API base URL.

`players` Data frame containing players from all leagues.

`teams` Data frame containing teams from all leagues.

`stadia` Data frame containing stadia from all leagues.

`managers` Data frame containing managers from all leagues.

`referees` Data frame containing referees from all leagues.

`httr_configs` Configs to pass on to all `httr` functions. See documentation.

`latest_update_timestamp` Keeps record of the latest update timestamp for queried data. Used to conditionally clear the client-side cache.

Methods

Public methods:

- `AmericanSoccerAnalysis$new()`
- `AmericanSoccerAnalysis$add_httr_configs()`
- `AmericanSoccerAnalysis$reset_httr_configs()`
- `AmericanSoccerAnalysis$get_players()`
- `AmericanSoccerAnalysis$get_teams()`
- `AmericanSoccerAnalysis$get_stadia()`
- `AmericanSoccerAnalysis$get_managers()`
- `AmericanSoccerAnalysis$get_referees()`
- `AmericanSoccerAnalysis$get_games()`
- `AmericanSoccerAnalysis$get_player_xgoals()`
- `AmericanSoccerAnalysis$get_player_xpass()`
- `AmericanSoccerAnalysis$get_player_goals_added()`

- AmericanSoccerAnalysis$get_player_salaries()
- AmericanSoccerAnalysis$get_goalkeeper_xgoals()
- AmericanSoccerAnalysis$get_goalkeeper_goals_added()
- AmericanSoccerAnalysis$get_team_xgoals()
- AmericanSoccerAnalysis$get_team_xpass()
- AmericanSoccerAnalysis$get_team_goals_added()
- AmericanSoccerAnalysis$get_team_salaries()
- AmericanSoccerAnalysis$get_game_xgoals()
- AmericanSoccerAnalysis$clone()

Method `new()`**:** Creates a new `AmericanSoccerAnalysis` object.

Usage:

```
AmericanSoccerAnalysis$new(...)
```

Arguments:

`...` Configs to pass on to all `httr` functions. See documentation.

Returns: A new `AmericanSoccerAnalysis` object.

Method `add_httr_configs()`**:** Appends new `httr` configs to the existing class.

Usage:

```
AmericanSoccerAnalysis$add_httr_configs(...)
```

Arguments:

`...` Configs to pass on to all `httr` functions. See documentation.

Method `reset_httr_configs()`**:** Removes all `httr` configs from the existing class.

Usage:

```
AmericanSoccerAnalysis$reset_httr_configs()
```

Method `get_players()`**:** Retrieves a data frame containing player names, IDs, and other metadata.

Usage:

```
AmericanSoccerAnalysis$get_players(leagues, ids, names)
```

Arguments:

`leagues` Leagues on which to filter. Accepts a character vector of length >= 1.

`ids` Player IDs on which to filter. Cannot be combined with `names`. Accepts a character vector of length >= 1.

`names` Player names on which to filter. Partial matches are accepted. Cannot be combined with `ids`. Accepts a character vector of length >= 1.

Method `get_teams()`**:** Retrieves a data frame containing team names, abbreviations, and IDs.

Usage:

```
AmericanSoccerAnalysis$get_teams(leagues, ids, names)
```

Arguments:

`leagues` Leagues on which to filter. Accepts a character vector of length >= 1.

`ids` Team IDs on which to filter. Cannot be combined with `names`. Accepts a character vector of length >= 1.

`names` Team names on which to filter. Partial matches and abbreviations are accepted. Cannot be combined with `ids`. Accepts a character vector of length >= 1.

Method `get_stadia()`**:** Retrieves a data frame containing stadium names, IDs, and other metadata.

Usage:

```
AmericanSoccerAnalysis$get_stadia(leagues, ids, names)
```

Arguments:

`leagues` Leagues on which to filter. Accepts a character vector of length >= 1.

`ids` Stadium IDs on which to filter. Cannot be combined with `names`. Accepts a character vector of length >= 1.

`names` Stadium names on which to filter. Partial matches are accepted. Cannot be combined with `ids`. Accepts a character vector of length >= 1.

Method `get_managers()`**:** Retrieves a data frame containing manager names and IDs.

Usage:

```
AmericanSoccerAnalysis$get_managers(leagues, ids, names)
```

Arguments:

`leagues` Leagues on which to filter. Accepts a character vector of length >= 1.

`ids` Manager IDs on which to filter. Cannot be combined with `names`. Accepts a character vector of length >= 1.

`names` Manager names on which to filter. Partial matches are accepted. Cannot be combined with `ids`. Accepts a character vector of length >= 1.

Method `get_referees()`**:** Retrieves a data frame containing referee names and IDs.

Usage:

```
AmericanSoccerAnalysis$get_referees(leagues, ids, names)
```

Arguments:

`leagues` Leagues on which to filter. Accepts a character vector of length >= 1.

`ids` Referee IDs on which to filter. Cannot be combined with `names`. Accepts a character vector of length >= 1.

`names` Referee names on which to filter. Partial matches are accepted. Cannot be combined with `ids`. Accepts a character vector of length >= 1.

Method `get_games()`**:** Retrieves a data frame containing game IDs, dates, opponents, scores, and other metadata.

Usage:

```
AmericanSoccerAnalysis$get_games(
  leagues,
  game_ids,
  team_ids,
  team_names,
  seasons,
  stages
)
```

Arguments:

`leagues` Leagues on which to filter. Accepts a character vector of length >= 1.

`game_ids` Game IDs on which to filter. Accepts a character vector of length >= 1.

`team_ids` Team IDs on which to filter. Cannot be combined with `team_names`. Accepts a character vector of length >= 1.

`team_names` Team names on which to filter. Partial matches and abbreviations are accepted. Cannot be combined with `team_ids`. Accepts a character vector of length >= 1.

`seasons` Name(s)/year(s) of seasons. See the API documentation for possible values. Accepts a character vector of length >= 1.

`stages` Describes the stage of competition in which a game took place. See the API documentation for possible values. Accepts a character vector of length >= 1.

Method `get_player_xgoals()`**:** Retrieves a data frame containing player xG data meeting the specified conditions.

Usage:

```
AmericanSoccerAnalysis$get_player_xgoals(leagues, ...)
```

Arguments:

`leagues` Leagues on which to filter. Accepts a character vector of length >= 1.

`...` The following arguments will be parsed:

- `minimum_minutes`: Minimum threshold for sum of minutes played.
- `minimum_shots`: Minimum threshold for sum of shots taken.
- `minimum_key_passes`: Minimum threshold for sum of key passes.
- `player_ids`: Player IDs on which to filter. Cannot be combined with `player_names`. Accepts a character vector of length >= 1.
- `player_names`: Player names on which to filter. Partial matches are accepted. Cannot be combined with `player_ids`. Accepts a character vector of length >= 1.
- `team_ids`: Team IDs on which to filter. Cannot be combined with `team_names`. Accepts a character vector of length >= 1.
- `team_names`: Team names on which to filter. Partial matches and abbreviations are accepted. Cannot be combined with `team_ids`. Accepts a character vector of length >= 1.
- `season_name`: Name(s)/year(s) of seasons. Cannot be combined with a date range. See the API documentation for possible values. Accepts a character vector of length >= 1.
- `start_date`: Start of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with `season_name`.
- `end_date`: End of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with `season_name`.
- `shot_pattern`: Describes the possessing actions leading to the shot. Valid keywords include: 'Set piece', 'Corner', 'Free kick', 'Penalty', 'Fastbreak', and 'Regular'. Accepts a character vector of length >= 1.
- `split_by_teams`: Logical indicator to group results by team.
- `split_by_seasons`: Logical indicator to group results by season.
- `split_by_games`: Logical indicator to group results by game.
- `stage_name`: Describes the stage of competition in which a game took place. See the API documentation for possible values. Accepts a character vector of length >= 1.

- `general_position`: Describes the most common position played by each player over the specified period of time. Valid keywords include: 'GK', 'CB', 'FB', 'DM', 'CM', 'AM', 'W', and 'ST'. Accepts a character vector of length >= 1.

Method `get_player_xpass()`**:** Retrieves a data frame containing player xPass data meeting the specified conditions.

Usage:

```
AmericanSoccerAnalysis$get_player_xpass(leagues, ...)
```

Arguments:

`leagues` Leagues on which to filter. Accepts a character vector of length >= 1.

`...` The following arguments will be parsed:

- `minimum_minutes`: Minimum threshold for sum of minutes played.
- `minimum_passes`: Minimum threshold for sum of attempted passes.
- `player_ids`: Player IDs on which to filter. Cannot be combined with `player_names`. Accepts a character vector of length >= 1.
- `player_names`: Player names on which to filter. Partial matches are accepted. Cannot be combined with `player_ids`. Accepts a character vector of length >= 1.
- `team_ids`: Team IDs on which to filter. Cannot be combined with `team_names`. Accepts a character vector of length >= 1.
- `team_names`: Team names on which to filter. Partial matches and abbreviations are accepted. Cannot be combined with `team_ids`. Accepts a character vector of length >= 1.
- `season_name`: Name(s)/year(s) of seasons. Cannot be combined with a date range. See the API documentation for possible values. Accepts a character vector of length >= 1.
- `start_date`: Start of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with `season_name`.
- `end_date`: End of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with `season_name`.
- `pass_origin_third`: Describes the third of the field from which the pass originated. Valid keywords include: 'Attacking', 'Middle', and 'Defensive'. Accepts a character vector of length >= 1.
- `split_by_teams`: Logical indicator to group results by team.
- `split_by_seasons`: Logical indicator to group results by season.
- `split_by_games`: Logical indicator to group results by game.
- `stage_name`: Describes the stage of competition in which a game took place. See the API documentation for possible values. Accepts a character vector of length >= 1.
- `general_position`: Describes the most common position played by each player over the specified period of time. Valid keywords include: 'GK', 'CB', 'FB', 'DM', 'CM', 'AM', 'W', and 'ST'. Accepts a character vector of length >= 1.

Method `get_player_goals_added()`**:** Retrieves a data frame containing player goals added (g+) data meeting the specified conditions.

Usage:

```
AmericanSoccerAnalysis$get_player_goals_added(leagues, ...)
```

Arguments:

`leagues` Leagues on which to filter. Accepts a character vector of length >= 1.

`...` The following arguments will be parsed:

- `minimum_minutes`: Minimum threshold for sum of minutes played.
- `player_ids`: Player IDs on which to filter. Cannot be combined with `player_names`. Accepts a character vector of length >= 1.
- `player_names`: Player names on which to filter. Partial matches are accepted. Cannot be combined with `player_ids`. Accepts a character vector of length >= 1.
- `team_ids`: Team IDs on which to filter. Cannot be combined with `team_names`. Accepts a character vector of length >= 1.
- `team_names`: Team names on which to filter. Partial matches and abbreviations are accepted. Cannot be combined with `team_ids`. Accepts a character vector of length >= 1.
- `season_name`: Name(s)/year(s) of seasons. Cannot be combined with a date range. See the API documentation for possible values. Accepts a character vector of length >= 1.
- `start_date`: Start of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with `season_name`.
- `end_date`: End of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with `season_name`.
- `split_by_teams`: Logical indicator to group results by team.
- `split_by_seasons`: Logical indicator to group results by season.
- `split_by_games`: Logical indicator to group results by game.
- `stage_name`: Describes the stage of competition in which a game took place. See the API documentation for possible values. Accepts a character vector of length >= 1.
- `action_type`: Describes the goals added (g+) action type. Valid keywords include: 'Dribbling', 'Fouling', 'Interrupting', 'Passing', 'Receiving', and 'Shooting'. Accepts a character vector of length >= 1.
- `general_position`: Describes the most common position played by each player over the specified period of time. Valid keywords include: 'GK', 'CB', 'FB', 'DM', 'CM', 'AM', 'W', and 'ST'. Accepts a character vector of length >= 1.
- `above_replacement`: Logical indicator to compare players against replacement-level values. This will only return aggregated g+ values, rather than disaggregated g+ values by action type.

Method `get_player_salaries()`**:** Retrieves a data frame containing player salary data meeting the specified conditions.

Usage:

```
AmericanSoccerAnalysis$get_player_salaries(leagues, ...)
```

Arguments:

`leagues` Leagues on which to filter. Currently, only MLS salary data is publicly available.

`...` The following arguments will be parsed:

- `player_ids`: Player IDs on which to filter. Cannot be combined with `player_names`. Accepts a character vector of length >= 1.
- `player_names`: Player names on which to filter. Partial matches are accepted. Cannot be combined with `player_ids`. Accepts a character vector of length >= 1.

- team_ids: Team IDs on which to filter. Cannot be combined with team_names. Accepts a character vector of length >= 1.
- team_names: Team names on which to filter. Partial matches and abbreviations are accepted. Cannot be combined with team_ids. Accepts a character vector of length >= 1.
- position: Describes the general position, as reported by the players' association. Valid keywords include: 'GK', 'D', 'M', and 'F'. Accepts a character vector of length >= 1.
- season_name: Name(s)/year(s) of seasons. Cannot be combined with a date range. See the API documentation for possible values. Accepts a character vector of length >= 1.
- start_date: Start of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with season_name.
- end_date: End of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with season_name.

Method `get_goalkeeper_xgoals()`**:** Retrieves a data frame containing goalkeeper xG data meeting the specified conditions.

Usage:

```
AmericanSoccerAnalysis$get_goalkeeper_xgoals(leagues, ...)
```

Arguments:

`leagues` Leagues on which to filter. Accepts a character vector of length >= 1.

`...` The following arguments will be parsed:

- minimum_minutes: Minimum threshold for sum of minutes played.
- minimum_shots_faced: Minimum threshold for sum of shots faced.
- player_ids: Player IDs on which to filter. Cannot be combined with player_names. Accepts a character vector of length >= 1.
- player_names: Player names on which to filter. Partial matches are accepted. Cannot be combined with player_ids. Accepts a character vector of length >= 1.
- team_ids: Team IDs on which to filter. Cannot be combined with team_names. Accepts a character vector of length >= 1.
- team_names: Team names on which to filter. Partial matches and abbreviations are accepted. Cannot be combined with team_ids. Accepts a character vector of length >= 1.
- season_name: Name(s)/year(s) of seasons. Cannot be combined with a date range. See the API documentation for possible values. Accepts a character vector of length >= 1.
- start_date: Start of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with season_name.
- end_date: End of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with season_name.
- shot_pattern: Describes the possessing actions leading to the shot. Valid keywords include: 'Set piece', 'Corner', 'Free kick', 'Penalty', 'Fastbreak', and 'Regular'. Accepts a character vector of length >= 1.
- split_by_teams: Logical indicator to group results by team.
- split_by_seasons: Logical indicator to group results by season.
- split_by_games: Logical indicator to group results by game.

- `stage_name`: Describes the stage of competition in which a game took place. See the API documentation for possible values. Accepts a character vector of length >= 1.

Method `get_goalkeeper_goals_added()`**:** Retrieves a data frame containing goalkeeper goals added (g+) data meeting the specified conditions.

Usage:

```
AmericanSoccerAnalysis$get_goalkeeper_goals_added(leagues, ...)
```

Arguments:

`leagues` Leagues on which to filter. Accepts a character vector of length >= 1.

`...` The following arguments will be parsed:

- `minimum_minutes`: Minimum threshold for sum of minutes played.
- `player_ids`: Player IDs on which to filter. Cannot be combined with `player_names`. Accepts a character vector of length >= 1.
- `player_names`: Player names on which to filter. Partial matches are accepted. Cannot be combined with `player_ids`. Accepts a character vector of length >= 1.
- `team_ids`: Team IDs on which to filter. Cannot be combined with `team_names`. Accepts a character vector of length >= 1.
- `team_names`: Team names on which to filter. Partial matches and abbreviations are accepted. Cannot be combined with `team_ids`. Accepts a character vector of length >= 1.
- `season_name`: Name(s)/year(s) of seasons. Cannot be combined with a date range. See the API documentation for possible values. Accepts a character vector of length >= 1.
- `start_date`: Start of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with `season_name`.
- `end_date`: End of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with `season_name`.
- `split_by_teams`: Logical indicator to group results by team.
- `split_by_seasons`: Logical indicator to group results by season.
- `split_by_games`: Logical indicator to group results by game.
- `stage_name`: Describes the stage of competition in which a game took place. See the API documentation for possible values. Accepts a character vector of length >= 1.
- `action_type`: Describes the goals added (g+) action type. Valid keywords include: 'Claiming', 'Fielding', 'Handling', 'Passing', 'Shotstopping', and 'Sweeping'. Accepts a character vector of length >= 1.
- `above_replacement`: Logical indicator to compare players against replacement-level values. This will only return aggregated g+ values, rather than disaggregated g+ values by action type.

Method `get_team_xgoals()`**:** Retrieves a data frame containing team xG data meeting the specified conditions.

Usage:

```
AmericanSoccerAnalysis$get_team_xgoals(leagues, ...)
```

Arguments:

`leagues` Leagues on which to filter. Accepts a character vector of length >= 1.

... The following arguments will be parsed:

- team_ids: Team IDs on which to filter. Cannot be combined with team_names. Accepts a character vector of length >= 1.
- team_names: Team names on which to filter. Partial matches and abbreviations are accepted. Cannot be combined with team_ids. Accepts a character vector of length >= 1.
- season_name: Name(s)/year(s) of seasons. Cannot be combined with a date range. See the API documentation for possible values. Accepts a character vector of length >= 1.
- start_date: Start of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with season_name.
- end_date: End of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with season_name.
- shot_pattern: Describes the possessing actions leading to the shot. Valid keywords include: 'Set piece', 'Corner', 'Free kick', 'Penalty', 'Fastbreak', and 'Regular'. Accepts a character vector of length >= 1.
- split_by_seasons: Logical indicator to group results by season.
- split_by_games: Logical indicator to group results by game.
- home_only: Logical indicator to only include results from home games.
- away_only: Logical indicator to only include results from away games.
- home_adjusted: Logical indicator to adjust certain values based on the share of home games a team has played during the specified duration.
- even_game_state: Logical indicator to only include shots taken when the score was level.
- stage_name: Describes the stage of competition in which a game took place. See the API documentation for possible values. Accepts a character vector of length >= 1.

Method `get_team_xpass()`**:** Retrieves a data frame containing team xPass data meeting the specified conditions.

Usage:

```
AmericanSoccerAnalysis$get_team_xpass(leagues, ...)
```

Arguments:

leagues Leagues on which to filter. Accepts a character vector of length >= 1.

... The following arguments will be parsed:

- team_ids: Team IDs on which to filter. Cannot be combined with team_names. Accepts a character vector of length >= 1.
- team_names: Team names on which to filter. Partial matches and abbreviations are accepted. Cannot be combined with team_ids. Accepts a character vector of length >= 1.
- season_name: Name(s)/year(s) of seasons. Cannot be combined with a date range. See the API documentation for possible values. Accepts a character vector of length >= 1.
- start_date: Start of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with season_name.
- end_date: End of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with season_name.

- `pass_origin_third`: Describes the third of the field from which the pass originated. Valid keywords include: 'Attacking', 'Middle', and 'Defensive'. Accepts a character vector of length >= 1.
- `split_by_seasons`: Logical indicator to group results by season.
- `split_by_games`: Logical indicator to group results by game.
- `home_only`: Logical indicator to only include results from home games.
- `away_only`: Logical indicator to only include results from away games.
- `stage_name`: Describes the stage of competition in which a game took place. See the API documentation for possible values. Accepts a character vector of length >= 1.

Method `get_team_goals_added()`**:** Retrieves a data frame containing team goals added (g+) data meeting the specified conditions.

Usage:

```
AmericanSoccerAnalysis$get_team_goals_added(leagues, ...)
```

Arguments:

`leagues` Leagues on which to filter. Accepts a character vector of length >= 1.

`...` The following arguments will be parsed:

- `team_ids`: Team IDs on which to filter. Cannot be combined with `team_names`. Accepts a character vector of length >= 1.
- `team_names`: Team names on which to filter. Partial matches and abbreviations are accepted. Cannot be combined with `team_ids`. Accepts a character vector of length >= 1.
- `season_name`: Name(s)/year(s) of seasons. Cannot be combined with a date range. See the API documentation for possible values. Accepts a character vector of length >= 1.
- `split_by_seasons`: Logical indicator to group results by season.
- `stage_name`: Describes the stage of competition in which a game took place. See the API documentation for possible values. Accepts a character vector of length >= 1.
- `action_type`: Describes the goals added (g+) action type. Valid keywords include: 'Dribbling', 'Fouling', 'Interrupting', 'Passing', 'Receiving', and 'Shooting'. Accepts a character vector of length >= 1.
- `zone`: Zone number on pitch. Zones 1-5 are the defensive-most zones, and zones 26-30 are the attacking-most zones. Accepts a character or integer vector of length >= 1.
- `gamestate_trunc`: Integer (score differential) value between -2 and 2, inclusive. Gamestates more extreme than -2 and 2 have been included with -2 and 2, respectively. Accepts a character or integer vector of length >= 1.

Method `get_team_salaries()`**:** Retrieves a data frame containing team salary data meeting the specified conditions.

Usage:

```
AmericanSoccerAnalysis$get_team_salaries(leagues, ...)
```

Arguments:

`leagues` Leagues on which to filter. Currently, only MLS salary data is publicly available.

`...` The following arguments will be parsed:

- `team_ids`: Team IDs on which to filter. Cannot be combined with `team_names`. Accepts a character vector of length >= 1.

- `team_names`: Team names on which to filter. Partial matches and abbreviations are accepted. Cannot be combined with `team_ids`. Accepts a character vector of length >= 1.
- `season_name`: Name(s)/year(s) of seasons. Cannot be combined with a date range. See the API documentation for possible values. Accepts a character vector of length >= 1.
- `split_by_teams`: Logical indicator to group results by team. Results must be grouped by at least one of teams, positions, or seasons. Value is TRUE by default.
- `split_by_seasons`: Logical indicator to group results by season. Results must be grouped by at least one of teams, positions, or seasons.
- `split_by_positions`: Logical indicator to group results by positions. Results must be grouped by at least one of teams, positions, or seasons.

Method `get_game_xgoals()`**:** Retrieves a data frame containing game xG data meeting the specified conditions.

Usage:

```
AmericanSoccerAnalysis$get_game_xgoals(leagues, ...)
```

Arguments:

`leagues` Leagues on which to filter. Accepts a character vector of length >= 1.

`...` The following arguments will be parsed:

- `game_ids`: Game IDs on which to filter. Accepts a character vector of length >= 1.
- `season_name`: Name(s)/year(s) of seasons. Cannot be combined with a date range. See the API documentation for possible values. Accepts a character vector of length >= 1.
- `start_date`: Start of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with `season_name`.
- `end_date`: End of a date range. Must be a string in YYYY-MM-DD format. Cannot be combined with `season_name`.
- `stage_name`: Describes the stage of competition in which a game took place. See the API documentation for possible values. Accepts a character vector of length >= 1.

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
AmericanSoccerAnalysis$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Index