getCompetitionAheadBehind

Get competition ahead-behind standings

Description
Get competition ahead-behind standings

Usage
getCompetitionAheadBehind(season_code, round)

Arguments

season_code One or more season codes as obtained from getCompetitionHistory(). Examples are E2023 for Euroleague or U2023 for Eurocup 2023.

round One or more round codes as obtained from getCompetitionRounds().

Value
Returns a summary tibble of ahead-behind for chosen competition and round

Reference webpage: Ahead-behind standings

Examples

## Not run:
getCompetitionAheadBehind(season_code = c("E2023", "E2022"), round = 1)

## End(Not run)
getCompetitionCalendar

Get competition calendar standings

Description
Get competition calendar standings

Usage
getCompetitionCalendar(season_code, round)

Arguments
- season_code: One or more season codes as obtained from `getCompetitionHistory()`. Examples are E2023 for Euroleague or U2023 for Eurocup 2023.
- round: One or more round codes as obtained from `getCompetitionRounds()`.

Value
Returns a summary tibble of calendar standings for chosen competitions and rounds

Reference webpage: Calendar standings

Examples
if(interactive()) {
  getCompetitionCalendar(season_code = c("E2023", "E2022"), round = 1)
}

getCompetitionHistory

Competition metadata

Description
[Experimental]
Retrieve values of arguments for specific data collection functions across all package.
Usage

getCompetitionHistory(competition_code)

getCompetitionRounds(season_code)

getCompetitionPhases(season_code)

getCompetitionTeams(season_code)

getCompetitionGames(season_code, round, phase_type = "All")

Arguments

competition_code
  One or more competition codes. Admitted values are E for Euroleague and U for Eurocup.

season_code
  One or more season codes as obtained from getCompetitionHistory(). Examples are E2023 for Euroleague or U2023 for Eurocup 2023.

round
  One or more round codes as obtained from getCompetitionRounds().

phase_type
  One or more phase type codes as obtained from getCompetitionPhases(). Admitted values are RS for regular season, PO for playoffs and FF for final four. Default is All for all.

Value

For each function, returns a tibble with information about history, rounds, phases, teams or games of chosen season and competition code.

Examples

## Not run:

getCompetitionHistory(competition_code = c("E", "U")) |> head(5)

getCompetitionRounds(season_code = c("E2023", "E2022")) |> head(5)

getCompetitionPhases(season_code = c("E2023", "U2023")) |> head(5)

getCompetitionTeams(season_code = c("E2023", "U2023")) |> head(5)

getCompetitionGames(season_code = "E2023", round = 1:5) |> head(5)

## End(Not run)
### getCompetitionMargins

*Get competition margins standings*

**Description**

Get competition margins standings

**Usage**

```r
getCompetitionMargins(season_code, round)
```

**Arguments**

- `season_code`  
  One or more season codes as obtained from `getCompetitionHistory()`. Examples are E2023 for Euroleague or U2023 for Eurocup 2023.

- `round`  
  One or more round codes as obtained from `getCompetitionRounds()`.

**Value**

Returns a summary tibble of standing margins for chosen competition and round

Reference webpage: [Margins standings](#)

**Examples**

```r
## Not run:
getCompetitionMargins(season_code = c("E2023", "E2022"), round = 1)
## End(Not run)
```

### getCompetitionStandings

*Get competition traditional standings*

**Description**

Get competition traditional standings

**Usage**

```r
getCompetitionStandings(season_code, round)
```
**getCompetitionStreaks**

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>season_code</td>
<td>One or more season codes as obtained from <code>getCompetitionHistory()</code>.</td>
</tr>
<tr>
<td></td>
<td>Examples are E2023 for Euroleague or U2023 for Eurocup 2023.</td>
</tr>
<tr>
<td>round</td>
<td>One or more round codes as obtained from <code>getCompetitionRounds()</code>.</td>
</tr>
</tbody>
</table>

**Value**

Returns a summary tibble of standings for chosen competitions and rounds.

Reference webpage: [Traditional standings](#)

**Examples**

```r
## Not run:
getCompetitionStandings(season_code = c("E2023", "E2022"), round = 1)

## End(Not run)
```

---

**getCompetitionStreaks**  *Get competition streaks standings*

**Description**

Get competition streaks standings

**Usage**

```r
getCompetitionStreaks(season_code, round)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>season_code</td>
<td>One or more season codes as obtained from <code>getCompetitionHistory()</code>.</td>
</tr>
<tr>
<td></td>
<td>Examples are E2023 for Euroleague or U2023 for Eurocup 2023.</td>
</tr>
<tr>
<td>round</td>
<td>One or more round codes as obtained from <code>getCompetitionRounds()</code>.</td>
</tr>
</tbody>
</table>

**Value**

Returns a summary tibble of streaks for chosen competitions and rounds.

Reference webpage: [Streaks standings](#)
getGameBoxScore

Examples

```r
## Not run:
getCompetitionStreaks(season_code = c("E2023", "E2022"), round = 1)

## End(Not run)
```

getGameBoxScore  

**Get game box-score**

Description  

[Experimental]

Usage  

```r
gameBoxScore(season_code, game_code)
```

Arguments

- `season_code` One or more season codes as obtained from `getCompetitionHistory()`. Examples are E2023 for Euroleague or U2023 for Eurocup 2023.
- `game_code` One or more game codes as obtained from `getCompetitionGames()`.

Value  

Returns a list of elements for the chosen games and seasons

- **Team**, Name of the teams
- **Coach**, Name of the coaches
- **EndOfQuarter**, Team accumulated points by quarter
- **ByQuarter**, Team points for each quarter
- **PlayerStats**, Statistics for each player in the game
- **TeamStats**, Aggregated statistics for each team in the game

Glossary of columns:

<table>
<thead>
<tr>
<th>Column name</th>
<th>Column extended name</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP</td>
<td>Game player</td>
</tr>
<tr>
<td>GS</td>
<td>Game started</td>
</tr>
<tr>
<td>MIN</td>
<td>Minutes played</td>
</tr>
<tr>
<td>PTS</td>
<td>Points scored</td>
</tr>
<tr>
<td>2PM</td>
<td>Two-pointers made</td>
</tr>
</tbody>
</table>
getGameEvolution

2PA  Two-pointers attempted
2P%  Two-point %
3PM  Three-pointers made
3PA  Three-pointers attempted
3P%  Three-point %
FTM  Free throws made
FTA  Free throws attempted
FT%  Free-throw %
OREB Offensive rebounds
DREB Defensive rebounds
TREB Total rebounds
AST  Assists
STL  Steals
TO   Turnovers
BLK  Blocks
BLKA Blocks against
FC   Personal fouls committed
FD   Personal fouls drawn
PIR  Performance Index Rating

Reference webpage: BoxScore

Examples

```r
## Not run:
getGameBoxScore(season_code = c("E2023", "U2023"), game_code = 1)
```

```r
## End(Not run)
```

---

**getGameEvolution**  
*Get game evolution*

Description

*[Experimental]*

Usage

```r
getGameEvolution(season_code, game_code)
```
getGameHeader

Arguments

season_code    One or more season codes as obtained from `getCompetitionHistory()`. Examples are E2023 for Euroleague or U2023 for Eurocup 2023.

game_code      One or more game codes as obtained from `getCompetitionGames()`.

Value

Returns a list of two elements for the chosen games and seasons

- **EvolutionSummary.** Overall information about minimum and maximum difference of scores between teams
- **Evolution.** Minute by minute points of each team

Reference webpage: GraphicStats

Examples

```r
## Not run:
getGameEvolution(season_code = c("E2023", "U2023"), game_code = 1)

## End(Not run)
```

---

**getGameHeader**  
*Game metadata*

Description

[Experimental]

Retrieve contextual information about games. Outputs may be required as arguments of other getGame* functions

Usage

```r
getGameHeader(season_code, game_code)

getGamePlayers(season_code, game_code, team_code)

getGameRound(season_code, game_code)
```
getGamePlayByPlay

Arguments

season_code  One or more season codes as obtained from `getCompetitionHistory()`. Examples are E2023 for Euroleague or U2023 for Eurocup 2023.
game_code  One or more game codes as obtained from `getCompetitionGames()`.
team_code  One or more team codes as obtained from `getCompetitionTeams()`.

Value

For each function, returns a tibble with information about header, player or round of chosen season and game code.

Examples

```r
## Not run:
getGameHeader(season_code = c("E2023", "U2023"), game_code = 1)
getGamePlayers(season_code = c("E2023", "U2023"), team_code = "ASV", game_code = 1)
getGameRound(season_code = c("E2023", "U2023"), game_code = 1)
## End(Not run)
```

getGamePlayByPlay  Get game play-by-play

Description

[Experimental]

Usage

getGamePlayByPlay(season_code, game_code)

Arguments

season_code  One or more season codes as obtained from `getCompetitionHistory()`. Examples are E2023 for Euroleague or U2023 for Eurocup 2023.
game_code  One or more game codes as obtained from `getCompetitionGames()`.
getGamePlayByPlay

Value

Returns a list of two elements for the chosen games and seasons

- **PlayByPlaySummary.** Overall information about the games, teams involved and status (live or not)
- **PlayByPlay.** Detailed information about the games, particularly **NumberOfPlay** and **PlayType**

Glossary of **PlayType**:

<table>
<thead>
<tr>
<th>PlayType</th>
<th>PlayInfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>2PA</td>
<td>Missed Two Pointer</td>
</tr>
<tr>
<td>2PM</td>
<td>Two Pointer</td>
</tr>
<tr>
<td>3PA</td>
<td>Missed Three Pointer</td>
</tr>
<tr>
<td>3PM</td>
<td>Three Pointer</td>
</tr>
<tr>
<td>AG</td>
<td>Shot Rejected</td>
</tr>
<tr>
<td>AS</td>
<td>Assist</td>
</tr>
<tr>
<td>BP</td>
<td>Begin Period</td>
</tr>
<tr>
<td>C</td>
<td>Coach Foul</td>
</tr>
<tr>
<td>CCH</td>
<td>Coach Challenge</td>
</tr>
<tr>
<td>CM</td>
<td>Foul</td>
</tr>
<tr>
<td>CMT</td>
<td>Technical Foul</td>
</tr>
<tr>
<td>CMTI</td>
<td>Throw-In Foul</td>
</tr>
<tr>
<td>CMU</td>
<td>Unsportsmanlike Foul</td>
</tr>
<tr>
<td>D</td>
<td>Def Rebound</td>
</tr>
<tr>
<td>EG</td>
<td>End Game</td>
</tr>
<tr>
<td>EP</td>
<td>End Period</td>
</tr>
<tr>
<td>FTA</td>
<td>Missed Free Throw</td>
</tr>
<tr>
<td>FTM</td>
<td>Free Throw In</td>
</tr>
<tr>
<td>FV</td>
<td>Block</td>
</tr>
<tr>
<td>IN</td>
<td>In</td>
</tr>
<tr>
<td>O</td>
<td>Off Rebound</td>
</tr>
<tr>
<td>OF</td>
<td>Offensive Foul</td>
</tr>
<tr>
<td>OUT</td>
<td>Out</td>
</tr>
<tr>
<td>RV</td>
<td>Foul Drawn</td>
</tr>
<tr>
<td>ST</td>
<td>Steal</td>
</tr>
<tr>
<td>TO</td>
<td>Turnover</td>
</tr>
<tr>
<td>TOUT</td>
<td>Time Out</td>
</tr>
<tr>
<td>TOUT_TV</td>
<td>TV Time Out</td>
</tr>
</tbody>
</table>

Reference webpage: PlayByPlay

Examples

```r
## Not run:

PlayByPlay = getGamePlayByPlay(season_code = c("E2023", "U2023"), game_code = 1)
```
getGamePoints

PlayByPlay$PlayByPlaySummary |> head(5)
PlayByPlay$PlayByPlay |> head(5)

## End(Not run)

---

getGamePoints | Get game points

### Description

[Experimental]

### Usage

getGamePoints(season_code, game_code)

### Arguments

- `season_code` One or more season codes as obtained from getCompetitionHistory(). Examples are E2023 for Euroleague or U2023 for Eurocup 2023.
- `game_code` One or more game codes as obtained from getCompetitionGames().

### Value

Returns scoring information of each player for the chosen games and seasons (subset of play-by-play data). In particular:

- **NumberOfPlay.** Reference id of the action (useful for join with results of getPlayByPlay)
- **CoordX** and **CoordY.** Spatial coordinates of the shot
- **Zone.** Area of the court of the shot

Reference webpage: PlayByPlay

### Examples

```r
## Not run:
getGamePoints(season_code = c("E2023", "U2023"), game_code = 1)
```

## End(Not run)
getPlayerAdvanced

Get player advanced statistics

Description

[Experimental]

Usage

getPlayerAdvanced(
  season_code,
  statistic_mode = c("perGame", "perMinute", "accumulated")
)

Arguments

season_code One or more season codes as obtained from getCompetitionHistory(). Examples are E2023 for Euroleague or U2023 for Eurocup 2023.

statistic_mode One or more aggregation modes of statistics. Admitted values are perGame, perMinute and accumulated.

Value

Returns a summary tibble of advanced players statistics for chosen seasons.

Glossary of columns:

<table>
<thead>
<tr>
<th>Column name</th>
<th>Column extended name</th>
<th>Column description</th>
</tr>
</thead>
<tbody>
<tr>
<td>eFG%</td>
<td>Effective field goal %</td>
<td>Combined two- and three-point shooting effectiveness</td>
</tr>
<tr>
<td>TS%</td>
<td>True shooting %</td>
<td>Percentage of points vs. points attempted</td>
</tr>
<tr>
<td>OREB%</td>
<td>Offensive rebound %</td>
<td>Estimated % of available offensive rebounds obtained while on court</td>
</tr>
<tr>
<td>DREB%</td>
<td>Defensive rebound %</td>
<td>Estimated % of available defensive rebounds obtained while on court</td>
</tr>
<tr>
<td>REB%</td>
<td>Rebound %</td>
<td>Estimated % of available rebounds obtained while on court</td>
</tr>
<tr>
<td>AST/TO</td>
<td>Assist to turnover ratio</td>
<td>Ratio of assists made to turnovers committed</td>
</tr>
<tr>
<td>AST-R</td>
<td>Assist ratio</td>
<td>Estimated % of assists per player’s offensive possessions</td>
</tr>
<tr>
<td>TO-R</td>
<td>Turnover ratio</td>
<td>Estimated % of turnovers per player’s offensive possessions</td>
</tr>
<tr>
<td>2PTA-R</td>
<td>Two-point attempts ratio</td>
<td>Estimated % of two-point attempts per player’s offensive possessions</td>
</tr>
<tr>
<td>3PTA-R</td>
<td>Three-point attempts ratio</td>
<td>Estimated % of three-point attempts per player’s offensive possessions</td>
</tr>
<tr>
<td>FT-RT</td>
<td>Free Throw rate</td>
<td>Measure of free throw attempts vs. field goal attempts</td>
</tr>
</tbody>
</table>

Reference webpage: Stats

Examples

## Not run:
getPlayerAdvanced(season_code = "E2023", statistic_mode = "perGame")

## End(Not run)

---

getPlayerMisc  
Get player miscellaneous statistics

### Description

[Experimental]

### Usage

getPlayerMisc(season_code)

### Arguments

- **season_code**: One or more season codes as obtained from `getCompetitionHistory()`. Examples are E2023 for Euroleague or U2023 for Eurocup 2023.

### Value

Returns a summary tibble of miscellaneous players statistics for chosen seasons.

Glossary of columns:

<table>
<thead>
<tr>
<th>Column name</th>
<th>Column extended name</th>
<th>Column description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD2</td>
<td>Double-doubles</td>
<td>Games with double-digit totals in two of: points, rebounds, assists, steals and blocks.</td>
</tr>
<tr>
<td>TD3</td>
<td>Triple-doubles</td>
<td>Games with double-digit totals in three of: points, rebounds, assists, steals and blocks.</td>
</tr>
</tbody>
</table>

Reference webpage: Stats

### Examples

```r
## Not run:
getPlayerMisc(season_code = "E2023")

## End(Not run)
```
**getPlayerPoints**  
*Get player points statistics*

**Description**  
*Experimental*

**Usage**  
`getPlayerPoints(season_code)`

**Arguments**  
- `season_code`  
  One or more season codes as obtained from `getCompetitionHistory()`. Examples are E2023 for Euroleague or U2023 for Eurocup 2023.

**Value**  
Returns a summary tibble of points players statistics for chosen seasons.

**Glossary of columns:**

<table>
<thead>
<tr>
<th>Column name</th>
<th>Column extended name</th>
<th>Column description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2PA-S</td>
<td>Two-point attempts share</td>
<td>Player’s share of team’s total two-point attempts</td>
</tr>
<tr>
<td>3PA-S</td>
<td>Three-point attempts share</td>
<td>Player’s share of team’s total three-point attempts</td>
</tr>
<tr>
<td>FTA-S</td>
<td>Free throw attempts share</td>
<td>Player’s share of team’s total free throw attempts</td>
</tr>
<tr>
<td>2PM-S</td>
<td>Two-pointers made share</td>
<td>Player’s share of team’s total two-pointers made</td>
</tr>
<tr>
<td>3PM-S</td>
<td>Three-pointers made share</td>
<td>Player’s share of team’s total three-pointers made</td>
</tr>
<tr>
<td>FTM-S</td>
<td>Free throws made share</td>
<td>Player’s share of team’s total free throws made</td>
</tr>
<tr>
<td>2P-RT</td>
<td>Two-Point Rate</td>
<td>% of a player’s field goal attempts that are two-pointers</td>
</tr>
<tr>
<td>3P-RT</td>
<td>Three-Point Rate</td>
<td>% of field goal attempts that are three-pointers</td>
</tr>
<tr>
<td>%2P</td>
<td>% of points from two-pointers</td>
<td>% of points from two-point shots made</td>
</tr>
<tr>
<td>%3P</td>
<td>% of points from three-pointers</td>
<td>% of points from three-point shots made</td>
</tr>
<tr>
<td>%FT</td>
<td>% of points from free throws</td>
<td>% of points from free throws made</td>
</tr>
</tbody>
</table>

Reference webpage: [Stats](#)

**Examples**

```r
## Not run:
getPlayerPoints(season_code = "E2023")
```

## End(Not run)
**getPlayerStats**  
*Get player statistics*

### Description

[Experimental]

### Usage

```r
getPlayerStats(
  season_code,
  statistic_mode = c("perGame", "perMinute", "accumulated")
)
```

### Arguments

- **season_code**  
  One or more season codes as obtained from `getCompetitionHistory()`.  
  Examples are E2023 for Euroleague or U2023 for Eurocup 2023.

- **statistic_mode**  
  One or more aggregation modes of statistics.  
  Admitted values are `perGame`, `perMinute` and `accumulated`.

### Value

Returns a summary tibble of players statistics for chosen seasons.

### Glossary of columns:

<table>
<thead>
<tr>
<th>Column name</th>
<th>Column extended name</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP</td>
<td>Game player</td>
</tr>
<tr>
<td>GS</td>
<td>Game started</td>
</tr>
<tr>
<td>MIN</td>
<td>Minutes played</td>
</tr>
<tr>
<td>PTS</td>
<td>Points scored</td>
</tr>
<tr>
<td>2PM</td>
<td>Two-pointers made</td>
</tr>
<tr>
<td>2PA</td>
<td>Two-pointers attempted</td>
</tr>
<tr>
<td>2P%</td>
<td>Two-point %</td>
</tr>
<tr>
<td>3PM</td>
<td>Three-pointers made</td>
</tr>
<tr>
<td>3PA</td>
<td>Three-pointers attempted</td>
</tr>
<tr>
<td>3P%</td>
<td>Three-point %</td>
</tr>
<tr>
<td>FTM</td>
<td>Free throws made</td>
</tr>
<tr>
<td>FTA</td>
<td>Free throws attempted</td>
</tr>
<tr>
<td>FT%</td>
<td>Free-throw %</td>
</tr>
<tr>
<td>OREB</td>
<td>Offensive rebounds</td>
</tr>
<tr>
<td>DREB</td>
<td>Defensive rebounds</td>
</tr>
<tr>
<td>TREB</td>
<td>Total rebounds</td>
</tr>
<tr>
<td>AST</td>
<td>Assists</td>
</tr>
<tr>
<td>STL</td>
<td>Steals</td>
</tr>
<tr>
<td>TO</td>
<td>Turnovers</td>
</tr>
<tr>
<td>BLK</td>
<td>Blocks</td>
</tr>
</tbody>
</table>
### Description

**[Experimental]**

Retrieve contextual information about teams. Outputs may be required as arguments of other `getTeam*` or `getPlayer*` functions.

### Usage

```r
getTeam(season_code, team_code)
getTeamPeople(season_code, team_code)
getTeamGames(season_code, team_code)
```

### Arguments

- `season_code` One or more season codes as obtained from `getCompetitionHistory()`. Examples are E2023 for Euroleague or U2023 for Eurocup 2023.
- `team_code` One or more team codes as obtained from `getCompetitionTeams()`. Examples are ASV, MAD, ...

### Value

For each function, returns a tibble with information about team, people or games of chosen season and team code.

---

**Reference webpage:** Stats
getTeamLeadStats

Get team lead statistics

Description

[Experimental]

Usage

getTeamLeadStats(season_code, phase_type = "All", subset = "All")

Arguments

- **season_code**: One or more season codes as obtained from `getCompetitionHistory()`. Examples are E2023 for Euroleague or U2023 for Eurocup 2023.
- **phase_type**: One or more phase type codes as obtained from `getCompetitionPhases()`. Admitted values are RS for regular season, PO for playoffs and FF for final four. Default is All for all.
- **subset**: One or more game subsets. Admitted values are HomeGames, AwayGames, GamesWon, GamesLost, ResultsIn5Points (for games resulted in +/-5 points) and All. Default is All.

Value

Returns a list of elements for the chosen seasons, phase_type and subset.

- **TeamAccumulated**: Total sum of statistics of all teams team
- **TeamAveragePerGame**: Average per game of statistics of all teams

Glossary of columns:

<table>
<thead>
<tr>
<th>Column name</th>
<th>Column extended name</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP</td>
<td>Game player</td>
</tr>
<tr>
<td>GS</td>
<td>Game started</td>
</tr>
<tr>
<td>MIN</td>
<td>Minutes played</td>
</tr>
</tbody>
</table>
**getTeamStats**

**Description**

[Experimental]

<table>
<thead>
<tr>
<th>Stat</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PTS</td>
<td>Points scored</td>
</tr>
<tr>
<td>2PM</td>
<td>Two-pointers made</td>
</tr>
<tr>
<td>2PA</td>
<td>Two-pointers attempted</td>
</tr>
<tr>
<td>2P%</td>
<td>Two-point %</td>
</tr>
<tr>
<td>3PM</td>
<td>Three-pointers made</td>
</tr>
<tr>
<td>3PA</td>
<td>Three-pointers attempted</td>
</tr>
<tr>
<td>3P%</td>
<td>Three-point %</td>
</tr>
<tr>
<td>FTM</td>
<td>Free throws made</td>
</tr>
<tr>
<td>FTA</td>
<td>Free throws attempted</td>
</tr>
<tr>
<td>FT%</td>
<td>Free-throw %</td>
</tr>
<tr>
<td>OREB</td>
<td>Offensive rebounds</td>
</tr>
<tr>
<td>DREB</td>
<td>Defensive rebounds</td>
</tr>
<tr>
<td>TREB</td>
<td>Total rebounds</td>
</tr>
<tr>
<td>AST</td>
<td>Assists</td>
</tr>
<tr>
<td>STL</td>
<td>Steals</td>
</tr>
<tr>
<td>TO</td>
<td>Turnovers</td>
</tr>
<tr>
<td>BLK</td>
<td>Blocks</td>
</tr>
<tr>
<td>BLKA</td>
<td>Blocks against</td>
</tr>
<tr>
<td>FC</td>
<td>Personal fouls committed</td>
</tr>
<tr>
<td>FD</td>
<td>Personal fouls drawn</td>
</tr>
<tr>
<td>PIR</td>
<td>Performance Index Rating</td>
</tr>
</tbody>
</table>

Reference webpage: [TeamLead](#)

**Examples**

```r
## Not run:
TeamLeadStats = getTeamLeadStats(season_code = c("E2022", "E2023"), phase_type = "RS")
TeamLeadStats$TeamAccumulated |> head(5)
TeamLeadStats$TeamAveragePerGame |> head(5)
```

## End(Not run)
### getTeamStats

#### Usage

getTeamStats(season_code, team_code, phase_type = "All")

#### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>season_code</td>
<td>One or more season codes as obtained from getCompetitionHistory(). Examples are E2023 for Euroleague or U2023 for Eurocup 2023.</td>
</tr>
<tr>
<td>team_code</td>
<td>One or more team codes as obtained from getCompetitionTeams(). Examples are ASV, MAD, ...</td>
</tr>
<tr>
<td>phase_type</td>
<td>One or more phase type codes as obtained from getCompetitionPhases(). Admitted values are RS for regular season, P0 for playoffs and FF for final four. Default is All for all.</td>
</tr>
</tbody>
</table>

#### Value

Returns a list of elements for the chosen teams and seasons:

- **PlayerAccumulated.** Total sum of statistics by player
- **PlayerAveragePerGame.** Average per game of statistics by player
- **PlayerAveragePer40.** Average per 40 minutes of statistics by player
- **TeamAccumulated.** Total sum of statistics of team
- **TeamAveragePerGame.** Average per game of statistics of teams

#### Glossary of columns:

<table>
<thead>
<tr>
<th>Column name</th>
<th>Column extended name</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP</td>
<td>Game player</td>
</tr>
<tr>
<td>GS</td>
<td>Game started</td>
</tr>
<tr>
<td>MIN</td>
<td>Minutes played</td>
</tr>
<tr>
<td>PTS</td>
<td>Points scored</td>
</tr>
<tr>
<td>2PM</td>
<td>Two-pointers made</td>
</tr>
<tr>
<td>2PA</td>
<td>Two-pointers attempted</td>
</tr>
<tr>
<td>2P%</td>
<td>Two-point %</td>
</tr>
<tr>
<td>3PM</td>
<td>Three-pointers made</td>
</tr>
<tr>
<td>3PA</td>
<td>Three-pointers attempted</td>
</tr>
<tr>
<td>3FG%</td>
<td>Three-point %</td>
</tr>
<tr>
<td>FTM</td>
<td>Free throws made</td>
</tr>
<tr>
<td>FTA</td>
<td>Free throws attempted</td>
</tr>
<tr>
<td>FT%</td>
<td>Free-throw %</td>
</tr>
<tr>
<td>OREB</td>
<td>Offensive rebounds</td>
</tr>
<tr>
<td>DREB</td>
<td>Defensive rebounds</td>
</tr>
<tr>
<td>TREB</td>
<td>Total rebounds</td>
</tr>
<tr>
<td>AST</td>
<td>Assists</td>
</tr>
<tr>
<td>STL</td>
<td>Steals</td>
</tr>
<tr>
<td>TO</td>
<td>Turnovers</td>
</tr>
<tr>
<td>BLK</td>
<td>Blocks</td>
</tr>
<tr>
<td>BLKA</td>
<td>Blocks against</td>
</tr>
</tbody>
</table>
getTeamStats

<table>
<thead>
<tr>
<th>FC</th>
<th>Personal fouls committed</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD</td>
<td>Personal fouls drawn</td>
</tr>
<tr>
<td>PIR</td>
<td>Performance Index Rating</td>
</tr>
</tbody>
</table>

Reference webpage: Team

Examples

```r
## Not run:
TeamStats = getTeamStats(team_code = "ASV", season_code = c("E2023", "E2022"), phase_type = "RS")
TeamStats$PlayerAccumulated |> head(5)
TeamStats$PlayerAveragePerGame |> head(5)
TeamStats$PlayerAveragePer40 |> head(5)
TeamStats$TeamAccumulated |> head(5)
TeamStats$TeamAveragePerGame |> head(5)
```

## End(Not run)
Index

* `r`
  - getCompetitionAheadBehind, 2
  - getCompetitionCalendar, 3
  - getCompetitionMargins, 5
  - getCompetitionStandings, 5
  - getCompetitionStreaks, 6
* `competitionMetadata`
  - getCompetitionHistory, 3
* `competitionStandings`
  - getCompetitionAheadBehind, 2
  - getCompetitionCalendar, 3
  - getCompetitionMargins, 5
  - getCompetitionStandings, 5
  - getCompetitionStreaks, 6
* `gameMetadata`
  - getGameHeader, 9
* `lifecycle::badge('experimental')`
  - getCompetitionAheadBehind, 2
  - getCompetitionCalendar, 3
  - getCompetitionMargins, 5
  - getCompetitionStandings, 5
  - getCompetitionStreaks, 6
* `teamMetadata`
  - getTeam, 17

getCompetitionAheadBehind, 2
getCompetitionCalendar, 3
getCompetitionGames
  (getCompetitionHistory), 3
getCompetitionGames(), 7, 9, 10, 12
getCompetitionHistory, 3
getCompetitionHistory(), 2–7, 9, 10, 12–18, 20
getCompetitionMargins, 5
getCompetitionPhases
  (getCompetitionHistory), 3
getCompetitionPhases(), 4, 18, 20
getCompetitionRounds
  (getCompetitionHistory), 3
getCompetitionRounds(), 2–6
getCompetitionStandings, 5
getCompetitionStreaks, 6
getCompetitionTeams
  (getCompetitionHistory), 3
getCompetitionTeams(), 10, 17, 20
getGameBoxScore, 7
getGameEvolution, 8
getGameHeader, 9
getGamePlayByPlay, 10
getGamePlayers (getGameHeader), 9
getGamePoints, 12
getGameRound (getGameHeader), 9
getPlayerAdvanced, 13
getPlayerMisc, 14
getPlayerPoints, 15
getPlayerStats, 16
getTeam, 17
getTeamGames (getTeam), 17
getTeamLeadStats, 18
getTeamPeople (getTeam), 17
getTeamStats, 19