nba_py Documentation

Release 0.1a2

nba_py

May 11, 2017

Contents

Contents:

CHAPTER 1

nba_py package

class nba_py.**Scoreboard**(*month=5*, *day=11*, *year=2017*, *league_id='00'*, *offset=0*)

A scoreboard for all games for a given day Displays current games plus info for a given day

Args:

month Specified month (1-12)

day Specified day (1-31)

year Specified year (YYYY)

league_id ID for the league to look in (Default is 00)

offset Day offset from which to operate

Attributes:

json Contains the full json dump to play around with

available()

east_conf_standings_by_day()

game_header()

last_meeting()

line_score()

series_standings()

west_conf_standings_by_day()

CHAPTER 2

nba_py.player module

class nba_py.player.**PlayerCareer**(*player_id*, *per_mode='PerGame'*, *league_id='00'*)

Contains stats based on several parameters such as career regular season totals, post season career totals, all star season careers totals, college season career totals, etc.

Args:

player_id Player ID to look up

per_mode Mode to measure statistics (Totals, PerGame, Per36, etc.)

league_id ID for the league to look in (Default is 00)

Attributes:

json Contains the full json dump to play around with

all_star_season_totals()

career_all_star_season_totals()

college_season_career_totals()

college_season_totals()

post_season_career_totals()

post_season_rankings()

post_season_totals()

preseason_career_totals()

preseason_season_totals()

regular_season_career_totals()

regular_season_rankings()

regular_season_totals()

class nba_py.player.**PlayerClutchSplits**(*player_id*, *team_id=0*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.player._PlayerDashboard

Contains a lot of methods for last n minutes with a deficit of x points

Args:

player_id ID of the player to look up

team_id ID of the team to look up

measure_type Specifies type of measure to use (Base, Advanced, etc.)

per_mode Mode to measure statistics (Totals, PerGame, Per36, etc.)

plus_minus Whether or not to consider plus minus (Y or N)

pace_adjust Whether or not to pace adjust stats (Y or N)

rank Whether or not to consider rank (Y or N)

league_id ID for the league to look in (Default is 00)

season Season given to look up

season_type Season type to consider (Regular / Playoffs)

po_round Playoff round

outcome Filter out by wins or losses

location Filter out by home or away

month Specify month to filter by

season_segment Filter by pre/post all star break

date_from Filter out games before a specific date

date_to Filter out games after a specific date

opponent_team_id Opponent team ID to look up

vs_conference Filter by conference

vs_division Filter by division

game_segment Filter by half / overtime

period Filter by quarter / specific overtime

shot_clock_range Filter statistics by range in shot clock

last_n_games Filter by number of games specified in N

Attributes:

json Contains the full json dump to play around with

last10sec_deficit_3point()

Results in last 5 minutes <= 5 points

last1min_deficit_5point()
Results in last 5 minutes <= 5 points

last1min_plusminus_5point()
Last 1 minutes +/= 5 points

last30sec_deficit_3point()
Results in last 5 minutes <= 5 points

last30sec_plusminus_5point()
Last 30 seconds +/= 3 points

last3min_deficit_5point()
Results in last 5 minutes <= 5 points

last3min_plusminus_5point()
Last 3 minutes +/= 5 points

last5min_deficit_5point()
Results in last 5 minutes <= 5 points

last5min_plusminus_5point()
Last 5 minutes +/= 5 points

class nba_py.player.**PlayerDefenseTracking**(*player_id*, *team_id=0*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.player._PlayerDashboard

Tracking data for defense for a given player

Args:

player_id ID of the player to look up

team_id ID of the team to look up

measure_type Specifies type of measure to use (Base, Advanced, etc.)

per_mode Mode to measure statistics (Totals, PerGame, Per36, etc.)

plus_minus Whether or not to consider plus minus (Y or N)

pace_adjust Whether or not to pace adjust stats (Y or N)

rank Whether or not to consider rank (Y or N)

league_id ID for the league to look in (Default is 00)

season Season given to look up

season_type Season type to consider (Regular / Playoffs)

po_round Playoff round

outcome Filter out by wins or losses

location Filter out by home or away

month Specify month to filter by

season_segment Filter by pre/post all star break

date_from Filter out games before a specific date

date_to Filter out games after a specific date

opponent_team_id Opponent team ID to look up

vs_conference Filter by conference

vs_division Filter by division

game_segment Filter by half / overtime

period Filter by quarter / specific overtime

shot_clock_range Filter statistics by range in shot clock

last_n_games Filter by number of games specified in N

Attributes:

json Contains the full json dump to play around with

class nba_py.player.**PlayerGameLogs**(*player_id*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*)

Contains a full log of all the games for a player for a given season

Args:

player_id ID of the player to look up

league_id ID for the league to look in (Default is 00)

season Season given to look up

season_type Season type to consider (Regular / Playoffs)

Attributes:

json Contains the full json dump to play around with

info()

class nba_py.player.**PlayerGeneralSplits**(*player_id*, *team_id=0*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.player._PlayerDashboard

Contains stats pertaining to location, wins and losses, pre/post all star break, starting position, and numbers of days rest

Args:

player_id ID of the player to look up

team_id ID of the team to look up

measure_type Specifies type of measure to use (Base, Advanced, etc.)

per_mode Mode to measure statistics (Totals, PerGame, Per36, etc.)

plus_minus Whether or not to consider plus minus (Y or N)

pace_adjust Whether or not to pace adjust stats (Y or N)

rank Whether or not to consider rank (Y or N)

league_id ID for the league to look in (Default is 00)

season Season given to look up

season_type Season type to consider (Regular / Playoffs)

po_round Playoff round

outcome Filter out by wins or losses

location Filter out by home or away

month Specify month to filter by

season_segment Filter by pre/post all star break

date_from Filter out games before a specific date

date_to Filter out games after a specific date

opponent_team_id Opponent team ID to look up

vs_conference Filter by conference

vs_division Filter by division

game_segment Filter by half / overtime

period Filter by quarter / specific overtime

shot_clock_range Filter statistics by range in shot clock

last_n_games Filter by number of games specified in N

Attributes:

json Contains the full json dump to play around with

days_rest()

location()

month()

pre_post_all_star()

starting_position()

win_losses()

class nba_py.player.**PlayerInGameSplits**(*player_id*, *team_id=0*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.player._PlayerDashboard

Contains player stats by half, by quarter, by score margin, and by actual margins.

Args:

player_id ID of the player to look up

team_id ID of the team to look up

measure_type Specifies type of measure to use (Base, Advanced, etc.)

per_mode Mode to measure statistics (Totals, PerGame, Per36, etc.)

plus_minus Whether or not to consider plus minus (Y or N)

pace_adjust Whether or not to pace adjust stats (Y or N)

rank Whether or not to consider rank (Y or N)

league_id ID for the league to look in (Default is 00)

season Season given to look up

season_type Season type to consider (Regular / Playoffs)

po_round Playoff round

outcome Filter out by wins or losses

location Filter out by home or away

month Specify month to filter by

season_segment Filter by pre/post all star break

date_from Filter out games before a specific date

date_to Filter out games after a specific date

opponent_team_id Opponent team ID to look up

vs_conference Filter by conference

vs_division Filter by division

game_segment Filter by half / overtime

period Filter by quarter / specific overtime

shot_clock_range Filter statistics by range in shot clock

last_n_games Filter by number of games specified in N

Attributes:

json Contains the full json dump to play around with

by_actual_margin()

by_half()

by_period()

by_score_margin()

class nba_py.player.**PlayerLastNGamesSplits**(*player_id*, *team_id=0*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.player._PlayerDashboard

Contains players stats per last 5, 10, 15, and 20 games, or specified number of games.

Args:

player_id ID of the player to look up

team_id ID of the team to look up

measure_type Specifies type of measure to use (Base, Advanced, etc.)

per_mode Mode to measure statistics (Totals, PerGame, Per36, etc.)

plus_minus Whether or not to consider plus minus (Y or N)

pace_adjust Whether or not to pace adjust stats (Y or N)

rank Whether or not to consider rank (Y or N)

league_id ID for the league to look in (Default is 00)

season Season given to look up

season_type Season type to consider (Regular / Playoffs)

po_round Playoff round

outcome Filter out by wins or losses

location Filter out by home or away

month Specify month to filter by

season_segment Filter by pre/post all star break

date_from Filter out games before a specific date

date_to Filter out games after a specific date

opponent_team_id Opponent team ID to look up

vs_conference Filter by conference

vs_division Filter by division

game_segment Filter by half / overtime

period Filter by quarter / specific overtime

shot_clock_range Filter statistics by range in shot clock

last_n_games Filter by number of games specified in N

Attributes:

json Contains the full json dump to play around with

gamenumber()

last10()

last15()

last20()

last5()

class nba_py.player.**PlayerList**(*league_id='00'*, *season='2016-17'*, *only_current=1*)
Contains a list of all players for a season, if specified, and will only contain current players if specified as well

Args:

league_id ID for the league to look in (Default is 00)

season Season given to look up

only_current Restrict lookup to only current players

Attributes:

json Contains the full json dump to play around with

info()

exception nba_py.player.**PlayerNotFoundException**
Bases: exceptions.Exception

class nba_py.player.**PlayerOpponentSplits**(*player_id*, *team_id=0*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)
Bases: nba_py.player._PlayerDashboard

Contains stats pertaining to player stats vs certain opponents by division, conference, and by specific team opponent

Args:

player_id ID of the player to look up

team_id ID of the team to look up

measure_type Specifies type of measure to use (Base, Advanced, etc.)

per_mode Mode to measure statistics (Totals, PerGame, Per36, etc.)

plus_minus Whether or not to consider plus minus (Y or N)

pace_adjust Whether or not to pace adjust stats (Y or N)

rank Whether or not to consider rank (Y or N)

league_id ID for the league to look in (Default is 00)

season Season given to look up

season_type Season type to consider (Regular / Playoffs)

po_round Playoff round

outcome Filter out by wins or losses

location Filter out by home or away

month Specify month to filter by

season_segment Filter by pre/post all star break

date_from Filter out games before a specific date

date_to Filter out games after a specific date

opponent_team_id Opponent team ID to look up

vs_conference Filter by conference

vs_division Filter by division

game_segment Filter by half / overtime

period Filter by quarter / specific overtime

shot_clock_range Filter statistics by range in shot clock

last_n_games Filter by number of games specified in N

Attributes:

json Contains the full json dump to play around with

by_conference()

by_division()

by_opponent()

class nba_py.player.**PlayerPassTracking**(*player_id*, *team_id=0*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.player._PlayerDashboard

Tracking data for passing for a given player

Args:

player_id ID of the player to look up

team_id ID of the team to look up

measure_type Specifies type of measure to use (Base, Advanced, etc.)

per_mode Mode to measure statistics (Totals, PerGame, Per36, etc.)

plus_minus Whether or not to consider plus minus (Y or N)

pace_adjust Whether or not to pace adjust stats (Y or N)

rank Whether or not to consider rank (Y or N)

league_id ID for the league to look in (Default is 00)

season Season given to look up

season_type Season type to consider (Regular / Playoffs)

po_round Playoff round

outcome Filter out by wins or losses

location Filter out by home or away

month Specify month to filter by

season_segment Filter by pre/post all star break

date_from Filter out games before a specific date

date_to Filter out games after a specific date

opponent_team_id Opponent team ID to look up

vs_conference Filter by conference

vs_division Filter by division

game_segment Filter by half / overtime

period Filter by quarter / specific overtime

shot_clock_range Filter statistics by range in shot clock

last_n_games Filter by number of games specified in N

Attributes:

json Contains the full json dump to play around with

passes_made()

passes_received()

class nba_py.player.**PlayerPerformanceSplits**(*player_id*, *team_id=0*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.player._PlayerDashboard

Player stats by different performance metrics such as score differntial, points scored, and points scored against

Args:

player_id ID of the player to look up

team_id ID of the team to look up

measure_type Specifies type of measure to use (Base, Advanced, etc.)

per_mode Mode to measure statistics (Totals, PerGame, Per36, etc.)

plus_minus Whether or not to consider plus minus (Y or N)

pace_adjust Whether or not to pace adjust stats (Y or N)

rank Whether or not to consider rank (Y or N)

league_id ID for the league to look in (Default is 00)

season Season given to look up

season_type Season type to consider (Regular / Playoffs)

po_round Playoff round

outcome Filter out by wins or losses

location Filter out by home or away

month Specify month to filter by

season_segment Filter by pre/post all star break

date_from Filter out games before a specific date

date_to Filter out games after a specific date

opponent_team_id Opponent team ID to look up

vs_conference Filter by conference

vs_division Filter by division

game_segment Filter by half / overtime

period Filter by quarter / specific overtime

shot_clock_range Filter statistics by range in shot clock

last_n_games Filter by number of games specified in N

Attributes:

json Contains the full json dump to play around with

points_against()

points_scored()

score_differential()

class nba_py.player.**PlayerProfile**(*player_id*, *per_mode='PerGame'*, *league_id='00'*)

Bases: *nba_py.player.PlayerCareer*

Contains a more in depth version of player career stats with season highs, career highs, and when the player's next game is

Args:

player_id Player ID to look up

per_mode Mode to measure statistics (Totals, PerGame, Per36, etc.)

league_id ID for the league to look in (Default is 00)

Attributes:

json Contains the full json dump to play around with

career_highs()

next_game()

season_highs()

class nba_py.player.**PlayerReboundLogTracking**(*player_id*, *team_id=0*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.player._PlayerDashboard

Contains a log for every rebound for a given season for a given player

Args:

player_id ID of the player to look up

team_id ID of the team to look up

measure_type Specifies type of measure to use (Base, Advanced, etc.)

per_mode Mode to measure statistics (Totals, PerGame, Per36, etc.)

plus_minus Whether or not to consider plus minus (Y or N)

pace_adjust Whether or not to pace adjust stats (Y or N)

rank Whether or not to consider rank (Y or N)

league_id ID for the league to look in (Default is 00)

season Season given to look up

season_type Season type to consider (Regular / Playoffs)

po_round Playoff round

outcome Filter out by wins or losses

location Filter out by home or away

month Specify month to filter by

season_segment Filter by pre/post all star break

date_from Filter out games before a specific date

date_to Filter out games after a specific date

opponent_team_id Opponent team ID to look up

vs_conference Filter by conference

vs_division Filter by division

game_segment Filter by half / overtime

period Filter by quarter / specific overtime

shot_clock_range Filter statistics by range in shot clock

last_n_games Filter by number of games specified in N

Attributes:

json Contains the full json dump to play around with

class nba_py.player.**PlayerReboundTracking**(*player_id*, *team_id=0*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.player._PlayerDashboard

Tracking data for rebounding for a given player

Args:

player_id ID of the player to look up

team_id ID of the team to look up

measure_type Specifies type of measure to use (Base, Advanced, etc.)

per_mode Mode to measure statistics (Totals, PerGame, Per36, etc.)

plus_minus Whether or not to consider plus minus (Y or N)

pace_adjust Whether or not to pace adjust stats (Y or N)

rank Whether or not to consider rank (Y or N)

league_id ID for the league to look in (Default is 00)

season Season given to look up

season_type Season type to consider (Regular / Playoffs)

po_round Playoff round

outcome Filter out by wins or losses

location Filter out by home or away

month Specify month to filter by

season_segment Filter by pre/post all star break

date_from Filter out games before a specific date

date_to Filter out games after a specific date

opponent_team_id Opponent team ID to look up

vs_conference Filter by conference

vs_division Filter by division

game_segment Filter by half / overtime

period Filter by quarter / specific overtime

shot_clock_range Filter statistics by range in shot clock

last_n_games Filter by number of games specified in N

Attributes:

json Contains the full json dump to play around with

num_contested_rebounding()

rebound_distance_rebounding()

shot_distance_rebounding()

shot_type_rebounding()

class nba_py.player.**PlayerShootingSplits**(*player_id*, *team_id=0*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.player._PlayerDashboard

Shooting stats based on distance, area, assisted to, shot types, and assisted by.

Args:

player_id ID of the player to look up

team_id ID of the team to look up

measure_type Specifies type of measure to use (Base, Advanced, etc.)

per_mode Mode to measure statistics (Totals, PerGame, Per36, etc.)

plus_minus Whether or not to consider plus minus (Y or N)

pace_adjust Whether or not to pace adjust stats (Y or N)

rank Whether or not to consider rank (Y or N)

league_id ID for the league to look in (Default is 00)

season Season given to look up

season_type Season type to consider (Regular / Playoffs)

po_round Playoff round

outcome Filter out by wins or losses

location Filter out by home or away

month Specify month to filter by

season_segment Filter by pre/post all star break

date_from Filter out games before a specific date

date_to Filter out games after a specific date

opponent_team_id Opponent team ID to look up

vs_conference Filter by conference

vs_division Filter by division

game_segment Filter by half / overtime

period Filter by quarter / specific overtime

shot_clock_range Filter statistics by range in shot clock

last_n_games Filter by number of games specified in N

Attributes:

json Contains the full json dump to play around with

assisted_by()

assisted_shots()

shot_5ft()

shot_8ft()

shot_areas()

shot_types_detail()

shot_types_summary()

class nba_py.player.**PlayerShotLogTracking**(*player_id*, *team_id=0*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.player._PlayerDashboard

Contains a log for every shot for a given season for a given player

Args:

player_id ID of the player to look up

team_id ID of the team to look up

measure_type Specifies type of measure to use (Base, Advanced, etc.)

per_mode Mode to measure statistics (Totals, PerGame, Per36, etc.)

plus_minus Whether or not to consider plus minus (Y or N)

pace_adjust Whether or not to pace adjust stats (Y or N)

rank Whether or not to consider rank (Y or N)

league_id ID for the league to look in (Default is 00)

season Season given to look up

season_type Season type to consider (Regular / Playoffs)

po_round Playoff round

outcome Filter out by wins or losses

location Filter out by home or away

month Specify month to filter by

season_segment Filter by pre/post all star break

date_from Filter out games before a specific date

date_to Filter out games after a specific date

opponent_team_id Opponent team ID to look up

vs_conference Filter by conference

vs_division Filter by division

game_segment Filter by half / overtime

period Filter by quarter / specific overtime

shot_clock_range Filter statistics by range in shot clock

last_n_games Filter by number of games specified in N

Attributes:

json Contains the full json dump to play around with

class nba_py.player.**PlayerShotTracking**(*player_id*, *team_id=0*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.player._PlayerDashboard

Tracking data for shooting for a given player

Args:

player_id ID of the player to look up

team_id ID of the team to look up

measure_type Specifies type of measure to use (Base, Advanced, etc.)

per_mode Mode to measure statistics (Totals, PerGame, Per36, etc.)

plus_minus Whether or not to consider plus minus (Y or N)

pace_adjust Whether or not to pace adjust stats (Y or N)

rank Whether or not to consider rank (Y or N)

league_id ID for the league to look in (Default is 00)

season Season given to look up

season_type Season type to consider (Regular / Playoffs)

po_round Playoff round

outcome Filter out by wins or losses

location Filter out by home or away

month Specify month to filter by

season_segment Filter by pre/post all star break

date_from Filter out games before a specific date

date_to Filter out games after a specific date

opponent_team_id Opponent team ID to look up

vs_conference Filter by conference

vs_division Filter by division

game_segment Filter by half / overtime

period Filter by quarter / specific overtime

shot_clock_range Filter statistics by range in shot clock

last_n_games Filter by number of games specified in N

Attributes:

json Contains the full json dump to play around with

closest_defender_shooting()

closest_defender_shooting_long()

dribble_shooting()

general_shooting()

shot_clock_shooting()

touch_time_shooting()

class nba_py.player.**PlayerSummary**(*player_id*)

Contains common player information like headline stats, weight, etc.

Args:

player_id ID of the player to look up

Attributes:

json Contains the full json dump to play around with

headline_stats()

info()

class nba_py.player.**PlayerVsPlayer**(*player_id*, *vs_player_id*, *team_id=0*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Contains general stats that pertain to players going against other players

Args:

player_id ID of the player to look up

vs_player_id ID of the vs player to look up

team_id ID of the team to look up

measure_type Specifies type of measure to use (Base, Advanced, etc.)

per_mode Mode to measure statistics (Totals, PerGame, Per36, etc.)

plus_minus Whether or not to consider plus minus (Y or N)

pace_adjust Whether or not to pace adjust stats (Y or N)

rank Whether or not to consider rank (Y or N)

league_id ID for the league to look in (Default is 00)

season Season given to look up

season_type Season type to consider (Regular / Playoffs)

po_round Playoff round

outcome Filter out by wins or losses

location Filter out by home or away

month Specify month to filter by

season_segment Filter by pre/post all star break

date_from Filter out games before a specific date

date_to Filter out games after a specific date

opponent_team_id Opponent team ID to look up

vs_conference Filter by conference

vs_division Filter by division

game_segment Filter by half / overtime

period Filter by quarter / specific overtime

shot_clock_range Filter statistics by range in shot clock

last_n_games Filter by number of games specified in N

Attributes: json: Contains the full json dump to play around with

on_off_court()

overall()

player_info()

shot_area_off_court()

shot_area_on_court()

shot_area_overall()

shot_distance_off_court()

shot_distance_on_court()

shot_distance_overall()

vs_player_info()

class nba_py.player.**PlayerYearOverYearSplits**(*player_id*, *team_id=0*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.player._PlayerDashboard

Displays player stats over the given season and over all seasons in the given league

Args:

player_id ID of the player to look up

team_id ID of the team to look up

measure_type Specifies type of measure to use (Base, Advanced, etc.)

per_mode Mode to measure statistics (Totals, PerGame, Per36, etc.)

plus_minus Whether or not to consider plus minus (Y or N)

pace_adjust Whether or not to pace adjust stats (Y or N)

rank Whether or not to consider rank (Y or N)

league_id ID for the league to look in (Default is 00)

season Season given to look up

season_type Season type to consider (Regular / Playoffs)

po_round Playoff round

outcome Filter out by wins or losses

location Filter out by home or away

month Specify month to filter by

season_segment Filter by pre/post all star break

date_from Filter out games before a specific date

date_to Filter out games after a specific date

opponent_team_id Opponent team ID to look up

vs_conference Filter by conference

vs_division Filter by division

game_segment Filter by half / overtime

period Filter by quarter / specific overtime

shot_clock_range Filter statistics by range in shot clock

last_n_games Filter by number of games specified in N

Attributes:

json Contains the full json dump to play around with

by_year()

nba_py.player.**get_player**(*first_name*, *last_name=None*, *season='2016-17'*, *only_current=0*, *just_id=True*)

Calls our PlayerList class to get a full list of players and then returns just an id if specified or the full row of player information

Args:

first_name First name of the player

last_name Last name of the player

(this is None if the player only has first name [Nene]) :only_current: Only wants the current list of players :just_id: Only wants the id of the player

Returns: Either the ID or full row of information of the player inputted

Raises: :PlayerNotFoundException:

CHAPTER 3

nba_py.game module

class nba_py.game.**Boxscore**(*game_id*, *season='2016-17'*, *season_type='Regular Season'*, *range_type='0'*, *start_period='0'*, *end_period='0'*, *start_range='0'*, *end_range='0'*)

Bases: nba_py.game._BaseBoxcore

player_stats()

team_starter_bench_stats()

team_stats()

class nba_py.game.**BoxscoreAdvanced**(*game_id*, *season='2016-17'*, *season_type='Regular Season'*, *range_type='0'*, *start_period='0'*, *end_period='0'*, *start_range='0'*, *end_range='0'*)

Bases: nba_py.game._BaseBoxcore

sql_players_advanced()

sql_team_advanced()

class nba_py.game.**BoxscoreFourFactors**(*game_id*, *season='2016-17'*, *season_type='Regular Season'*, *range_type='0'*, *start_period='0'*, *end_period='0'*, *start_range='0'*, *end_range='0'*)

Bases: nba_py.game._BaseBoxcore

sql_players_four_factors()

sql_team_four_factors()

class nba_py.game.**BoxscoreMisc**(*game_id*, *season='2016-17'*, *season_type='Regular Season'*, *range_type='0'*, *start_period='0'*, *end_period='0'*, *start_range='0'*, *end_range='0'*)

Bases: nba_py.game._BaseBoxcore

sql_players_misc()

sql_team_misc()

class nba_py.game.**BoxscoreScoring**(*game_id*, *season='2016-17'*, *season_type='Regular Season'*, *range_type='0'*, *start_period='0'*, *end_period='0'*, *start_range='0'*, *end_range='0'*)

Bases: nba_py.game._BaseBoxcore

sql_players_scoring()

sql_team_scoring()

class nba_py.game.**BoxscoreSummary**(*game_id*, *season='2016-17'*, *season_type='Regular Season'*, *range_type='0'*, *start_period='0'*, *end_period='0'*, *start_range='0'*, *end_range='0'*)

available_video()

game_info()

game_summary()

inactive_players()

last_meeting()

line_score()

officials()

other_stats()

season_series()

class nba_py.game.**BoxscoreUsage**(*game_id*, *season='2016-17'*, *season_type='Regular Season'*, *range_type='0'*, *start_period='0'*, *end_period='0'*, *start_range='0'*, *end_range='0'*)

Bases: nba_py.game._BaseBoxcore

sql_players_usage()

sql_team_usage()

class nba_py.game.**HustleStats**(*game_id*)

hustle_stats_available()

hustle_stats_player_box_score()

hustle_stats_team_box_score()

class nba_py.game.**PlayByPlay**(*game_id*, *start_period='0'*, *end_period='0'*)

available_video()

info()

class nba_py.game.**PlayerTracking**(*game_id*)

info()

CHAPTER 4

nba_py.team module

class nba_py.team.**TeamClutchSplits**(*team_id*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.team._TeamDashboard

This is a weird endpoint, to be honest. It's got a lot of cool little stats and there are two extra fields in the json that I have no idea what they do.

If you know please tell me.

- Last30Sec3Point2TeamDashboard
- Last10Sec3Point2TeamDashboard

last10sec_deficit_3point()
Results in last 5 minutes <= 5 points

last1min_deficit_5point()
Results in last 5 minutes <= 5 points

last1min_plusminus_5point()
Last 1 minutes +/= 5 points

last30sec_deficit_3point()
Results in last 5 minutes <= 5 points

last30sec_plusminus_5point()
Last 30 seconds +/= 3 points

last3min_deficit_5point()
Results in last 5 minutes <= 5 points

last3min_plusminus_5point()
Last 3 minutes +/= 5 points

last5min_deficit_5point()
Results in last 5 minutes <= 5 points

last5min_plusminus_5point()
Last 5 minutes +/= 5 points

class nba_py.team.**TeamCommonRoster**(*team_id*, *season='2016-17'*)

coaches()

roster()

class nba_py.team.**TeamDetails**(*team_id*)

awards_championships()

awards_conf()

awards_div()

background()

history()

hof()

retired()

social_sites()

class nba_py.team.**TeamGameLogs**(*team_id*, *season='2016-17'*, *season_type='Regular Season'*)

info()

class nba_py.team.**TeamGeneralSplits**(*team_id*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.team._TeamDashboard

days_rest()

location()

monthly()

pre_post_all_star()

wins_losses()

class nba_py.team.**TeamInGameSplits**(*team_id*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.team._TeamDashboard

by_actual_margin()

by_half()

by_period()

by_score_margin()

class nba_py.team.**TeamLastNGamesSplits**(*team_id*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.team._TeamDashboard

gamenumber()

last10()

last15()

last20()

last5()

class nba_py.team.**TeamLineups**(*team_id*, *game_id=''*, *group_quantity=5*, *season='2016-17'*, *season_type='Regular Season'*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *last_n_games='0'*)

lineups()

overall()

class nba_py.team.**TeamList**(*league_id='00'*)

info()

class nba_py.team.**TeamOpponentSplits**(*team_id*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.team._TeamDashboard

by_conference()

by_division()

by_opponent()

class nba_py.team.**TeamPassTracking**(*team_id*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.team._TeamDashboard

passes_made()

passes_recieved()

class nba_py.team.**TeamPerformanceSplits**(*team_id*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.team._TeamDashboard

points_against()

points_scored()

score_differential()

class nba_py.team.**TeamPlayerOnOffDetail**(*team_id*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.team._TeamDashboard

off_court()

on_court()

class nba_py.team.**TeamPlayerOnOffSummary**(*team_id*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.team._TeamDashboard

off_court()

on_court()

class nba_py.team.**TeamPlayers**(*team_id*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.team._TeamDashboard

season_totals()

class nba_py.team.**TeamReboundTracking**(*team_id*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.team._TeamDashboard

contested_rebounding()

rebound_distance_rebounding()

shot_distance_rebounding()

shot_type_rebounding()

class nba_py.team.**TeamSeasons**(*team_id*, *league_id='00'*, *season_type='Regular Season'*, *per_mode='PerGame'*)

info()

class nba_py.team.**TeamShootingSplits**(*team_id*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.team._TeamDashboard

assisted_by()

assisted_shots()

shot_5ft()

shot_8ft()

shot_areas()

shot_type_summary()

class nba_py.team.**TeamShotTracking**(*team_id*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.team._TeamDashboard

closest_defender_shooting()

closest_defender_shooting_long()

dribble_shooting()

shot_clock_shooting()

touch_time_shooting()

class nba_py.team.**TeamSummary**(*team_id*, *season='2016-17'*, *league_id='00'*, *season_type='Regular Season'*)

info()

season_ranks()

class nba_py.team.**TeamVsPlayer**(*team_id*, *vs_player_id*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

on_off_court()

overall()

shot_area_off_court()

shot_area_on_court()

shot_area_overall()

shot_distance_off_court()

shot_distance_on_court()

shot_distance_overall()

vs_player_overall()

class nba_py.team.**TeamYearOverYearSplits**(*team_id*, *measure_type='Base'*, *per_mode='PerGame'*, *plus_minus='N'*, *pace_adjust='N'*, *rank='N'*, *league_id='00'*, *season='2016-17'*, *season_type='Regular Season'*, *po_round='0'*, *outcome=''*, *location=''*, *month='0'*, *season_segment=''*, *date_from=''*, *date_to=''*, *opponent_team_id='0'*, *vs_conference=''*, *vs_division=''*, *game_segment=''*, *period='0'*, *shot_clock_range=''*, *last_n_games='0'*)

Bases: nba_py.team._TeamDashboard

by_year()

CHAPTER 5

nba_py.constants module

class nba_py.constants.**AheadBehind**
Bases: nba_py.constants._DefaultBlank

AheadOrBehind = 'Ahead or Behind'

AheadOrTied = 'Ahead or Tied'

BehindOrTied = 'Behind or Tied'

class nba_py.constants.**ClutchTime**
Bases: nba_py.constants._DefaultBlank

Last10Sec = 'Last 10 Seconds'

Last1Min = 'Last 1 Minutes'

Last2Min = 'Last 2 Minutes'

Last30Sec = 'Last 30 Seconds'

Last3Min = 'Last 3 Minutes'

Last4Min = 'Last 4 Minutes'

Last5Min = 'Last 5 Minutes'

class nba_py.constants.**College**
Bases: nba_py.constants._DefaultBlank

class nba_py.constants.**Conference**
Bases: *nba_py.constants.VsConference*

class nba_py.constants.**ContextMeasure**

Default = 'FGM'

EFG_PCT = 'EFG_PCT'

FG3A = 'FG3A'

FG3M = 'FG3m'

FG3_PCT = 'FG3_PCT'

FGA = 'FGA'

FGM = 'FGM'

FG_PCT = 'FG_PCT'

PF = 'PF'

PTS_2ND_CHANCE = 'PTS_2ND_CHANCE'

PTS_FB = 'PTS_FB'

PTS_OFF_TOV = 'PTS_OFF_TOV'

TS_PCT = 'TS_PCT'

class nba_py.constants.**Counter**

Default = '1000'

class nba_py.constants.**Country**
Bases: nba_py.constants._DefaultBlank

class nba_py.constants.**DateFrom**
Bases: nba_py.constants._DefaultBlank

class nba_py.constants.**DateTo**
Bases: nba_py.constants._DefaultBlank

class nba_py.constants.**Direction**

ASC = 'ASC'

DESC = 'DESC'

Default = 'DESC'

class nba_py.constants.**Division**
Bases: *nba_py.constants.VsDivision*

class nba_py.constants.**DraftPick**
Bases: nba_py.constants._DefaultBlank

FirstPick = '1st+Pick'

FirstRound = '1st+Round'

Lottery = 'Lottery+Pick'

Picks11Thru20 = 'Picks+11+Thru+20'

Picks21Thru30 = 'Picks+21+Thru+30'

SecondRound = '2nd+Round'

Top10 = 'Top+10+Pick'

Top15 = 'Top+15+Pick'

Top20 = 'Top+20+Pick'

Top25 = 'Top+25+Pick'

Top5 = 'Top+5+Pick'

Undrafted = 'Undrafted'

class nba_py.constants.**DraftYear**
Bases: nba_py.constants._DefaultBlank

class nba_py.constants.**EndPeriod**
Bases: *nba_py.constants.Period*

class nba_py.constants.**EndRange**
Bases: nba_py.constants._DefaultZero

class nba_py.constants.**GameID**
Bases: nba_py.constants._DefaultBlank

class nba_py.constants.**GameScope**

Default = 'Season'

Finals = 'Finals'

Last10 = 'Last 10'

Season = 'Season'

Yesterday = 'Yesterday'

class nba_py.constants.**GameSegment**
Bases: nba_py.constants._DefaultBlank

EntireGame = ''

FirstHalf = 'First Half'

Overtime = 'Overtime'

SecondHalf = 'Second Half'

class nba_py.constants.**Game_Scope**
Bases: nba_py.constants._DefaultBlank

Last10 = 'Last 10'

Yesterday = 'Yesterday'

class nba_py.constants.**GroupQuantity**

Default = 5

class nba_py.constants.**Height**
Bases: nba_py.constants._DefaultBlank

Example: for greater than 6ft8 api call should be GT+6-8 for lower than 7ft3 api call should be LT+7-3

class nba_py.constants.**LastNGames**
Bases: nba_py.constants._DefaultZero

class nba_py.constants.**League**

Default = '00'

NBA = '00'

class nba_py.constants.**Location**
Bases: nba_py.constants._DefaultBlank

Away = 'Away'

Home = 'Home'

class nba_py.constants.**MeasureType**

Advanced = 'Advanced'

Base = 'Base'

Default = 'Base'

FourFactors = 'Four Factors'

Misc = 'Misc'

Opponent = 'Opponent'

Scoring = 'Scoring'

Usage = 'Usage'

class nba_py.constants.**Month**
Bases: nba_py.constants._DefaultZero

All = '0'

April = '7'

August = '11'

December = '3'

February = '5'

January = '4'

July = '10'

June = '9'

March = '6'

May = '8'

November = '2'

October = '1'

September = '12'

class nba_py.constants.**OpponentTeamID**
Bases: nba_py.constants._DefaultZero

class nba_py.constants.**Outcome**
Bases: nba_py.constants._DefaultBlank

Loss = 'L'

Win = 'W'

class nba_py.constants.**PaceAdjust**
Bases: nba_py.constants._DefaultN

class nba_py.constants.**PerMode**

Default = 'PerGame'

MinutesPer = 'MinutesPer'

Per100Plays = 'Per100Plays'

Per100Possessions = 'Per100Possessions'

Per36 = 'Per36'

Per40 = 'Per40'

Per48 = 'Per48'

PerGame = 'PerGame'

PerMinute = 'PerMinute'

PerPlay = 'PerPlay'

PerPossession = 'PerPossession'

Totals = 'Totals'

class nba_py.constants.**Period**
Bases: nba_py.constants._DefaultZero

AllQuarters = '0'

FirstQuarter = '1'

FourthQuarter = '4'

Overtime(*n*)

SecondQuarter = '2'

ThirdQuarter = '3'

class nba_py.constants.**PlayerExperience**
Bases: nba_py.constants._DefaultBlank

Rookie = 'Rookie'

Sophomore = 'Sophomore'

Veteran = 'Veteran'

class nba_py.constants.**PlayerOrTeam**

Default = 'Player'

Player = 'Player'

Team = 'Team'

class nba_py.constants.**PlayerPosition**
Bases: nba_py.constants._DefaultBlank

Center = 'C'

Forward = 'F'

Guard = 'G'

class nba_py.constants.**PlayerScope**

AllPlayers = 'All Players'

Default = 'All Players'

Rookies = 'Rookie'

class nba_py.constants.**Player_or_Team**

Default = 'P'

Player = 'P'

Team = 'T'

class nba_py.constants.**PlayoffRound**
Bases: nba_py.constants._DefaultZero

All = '0'

ConferenceFinals = '3'

Finals = '4'

QuarterFinals = '1'

SemiFinals = '2'

class nba_py.constants.**PlusMinus**
Bases: nba_py.constants._DefaultN

class nba_py.constants.**PtMeasureType**

SpeedDistance = 'SpeedDistance'

class nba_py.constants.**RangeType**
Bases: nba_py.constants._DefaultZero

class nba_py.constants.**Rank**
Bases: nba_py.constants._DefaultN

class nba_py.constants.**RookieYear**
Bases: nba_py.constants._DefaultBlank

class nba_py.constants.**Scope**

AllPlayers = 'S'

Default = 'S'

Rookies = 'Rookies'

class nba_py.constants.**SeasonSegment**
Bases: nba_py.constants._DefaultBlank

EntireSeason = ''

PostAllStar = 'Post All-Star'

PreAllStar = 'Pre All-Star'

class nba_py.constants.**SeasonType**

Default = 'Regular Season'

Playoffs = 'Playoffs'

Regular = 'Regular Season'

class nba_py.constants.**ShotClockRange**
Bases: nba_py.constants._DefaultBlank

AllRanges = ''

ShotClockOff = 'ShotClock Off'

get(*n*)

class nba_py.constants.**Sorter**

AST = 'AST'

BLK = 'BLK'

DREB = 'DREB'

Default = 'PTS'

FG3A = 'FG3A'

FG3M = 'FG3M'

FG3_PCT = 'FG3_PCT'

FGA = 'FGA'

FGM = 'FGM'

FG_PCT = 'FG_PCT'

FTA = 'FTA'

FTM = 'FTM'

FT_PCT = 'FT_PCT'

OREB = 'OREB'

PTS = 'PTS'

REB = 'REB'

STL = 'STL'

TOV = 'TOV'

class nba_py.constants.**StartPeriod**
Bases: *nba_py.constants.Period*

class nba_py.constants.**StartRange**
Bases: nba_py.constants._DefaultZero

class nba_py.constants.**StarterBench**
Bases: nba_py.constants._DefaultBlank

Bench = 'Bench'

Starters = 'Starters'

class nba_py.constants.**StatCategory**

AST = 'AST'

AST_TOV = 'AST/TO'

BLK = 'BLK'

DREB = 'DREB'

Default = 'PTS'

EFF = 'EFF'

FG3A = '3PA'

FG3M = '3PM'

FG3_PCT = '3P%'

FGA = 'FGA'

FGM = 'FGM'

FG_PCT = 'FG%'

FTA = 'FTA'

FTM = 'FTM'

FT_PCT = 'FT%'

OREB = 'OREB'

PF = 'PF'

PTS = 'PTS'

REB = 'REB'

STL = 'STL'

STL_TOV = 'STL/TOV'

TOV = 'TOV'

class nba_py.constants.**TeamID**
Bases: nba_py.constants._DefaultZero

class nba_py.constants.**VsConference**
Bases: nba_py.constants._DefaultBlank

All = ''

East = 'East'

West = 'West'

class nba_py.constants.**VsDivision**
Bases: nba_py.constants._DefaultBlank

All = ''

Atlantic = 'Atlantic'

Central = 'Central'

Northwest = 'Northwest'

Pacific = 'Pacific'

Southeast = 'Southeast'

Southwest = 'Southwest'

class nba_py.constants.**Weight**
Bases: nba_py.constants._DefaultBlank

Example: for greater than 225lbs api call should be GT+225lbs

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

Python Module Index

n

Index

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

Y