
pbpstats

Oct 19, 2022

1	Features	3
2	Quickstart Guide	5
2.1	Installation	5
2.2	Setup data directory (optional but recommended)	5
2.2.1	Download data	5
2.2.2	data.nba.com vs stats.nba.com	5
2.3	Basic Usage	6
2.4	Code Examples	6
2.4.1	Game Data	6
2.4.2	All Final Games For Season	7
2.4.3	All Final Games For Day	7
2.4.4	Doing Detailed Stuff with Possession Data	7
2.4.5	Doing Detailed Stuff with Enhanced PBP Data	7
2.5	Note on Ids	8
2.6	Issues with raw play-by-play	8
3	Contents	9
3.1	Documentation	9
3.1.1	Data Loaders	9
3.1.1.1	data.nba.com Data Loaders	9
3.1.1.2	live Data Loaders	15
3.1.1.3	stats.nba.com Data Loaders	19
3.1.1.4	Data Loader Factory	29
3.1.1.5	Enhanced PBP Loader	30
3.1.1.6	Possession Loader	30
3.1.2	Objects	30
3.1.2.1	Day	30
3.1.2.2	Game	31
3.1.2.3	Season	31
3.1.3	Resources	32
3.1.3.1	Boxscore	32
3.1.3.2	Enhanced PBP	34
3.1.3.3	Games	59
3.1.3.4	PBP	60
3.1.3.5	Possessions	62
3.1.3.6	Shots	63

3.1.4 Client	64
Python Module Index	65
Index	69

A package to scrape and parse NBA, WNBA and G-League play-by-play data.

CHAPTER 1

Features

- Adds lineup on floor for all events
- Adds detailed data for each possession including start time, end time, score margin, how the previous possession ended
- Shots, rebounds and assists broken down by shot zone
- Supports both stats.nba.com and data.nba.com endpoints
- Supports NBA, WNBA and G-League stats
- All stats on pbpstats.com are derived from these stats
- Fixes order of events for some common cases in which events are out of order

2.1 Installation

```
$ pip install pbpstats
```

2.2 Setup data directory (optional but recommended)

To avoid repeating the same requests multiple times, and to allow for manually fixing issues with the raw play-by-play data I recommend setting up a data directory to save the response data. This will limit the number of requests you make to the NBA Stats API and allow you to manually fix issues with the raw data. Within this directory you will need to add four subdirectories(game_details, overrides, pbp and schedule). To use the directory, just include it in your settings when initializing the client like in the example below. This directory is also where [override files](#) to fix periods starters and handle issues with parsing pbp that can't be fixed by editing pbp file are placed.

2.2.1 Download data

If you want data with pbp event order fixed as well as overrides that are up to date as of this release you can download the files from [here](#). Unzip it and use the unzipped directory as your data directory.

2.2.2 data.nba.com vs stats.nba.com

The data from both these sources is mostly the same but there are a few small differences. The data.nba.com play-by-play has the offense team id in all events, which makes it easier to track possession changes. The stats.nba.com events do not have this attribute and to get possession counts the play-by-play needs to be parsed, which makes it more sensitive to events being in the correct order. The stats.nba.com play-by-play also has lots of shots and rebounds that are out of order that need to be fixed manually. If you download the data from the link above, I have fixed these for previous seasons, but going forward if you use this you will have to keep up with fixing them manually yourself. If

you don't care if there may be an occasional possession count being off or you don't want to deal with manually fixing the event order I suggest using `data.nba.com` for your play-by-play provider.

The `stats.nba.com` has more older season data available. So it should be used if you want to work with older data.

Another difference is that the `data.nba.com` play-by-play updates in real time, so if you are looking to do live stats during a game you can use it for that.

2.3 Basic Usage

Within your settings that will be passed into the client when you instantiate it you can set the data directory and set the resources and data source you want.

Options for `source` are 'file' and 'web'. When source is 'file', it will pull data from the `dir` specified in the settings. When source is 'web' it will make an API request to get the data. If you want to save the response data to disk set `dir` within the settings and it will be saved in the appropriate subdirectory.

Options for `data_provider` are 'stats_nba' and 'data_nba' and 'live'.

See the code examples below for some examples settings.

Resource options are:

- `Boxscore` - basic boxscore stats
- `EnhancedPbp` - more detail than the basic play-by-play
- `Games` - for getting all games for a season or date
- `Pbp` - raw play-by-play
- `Possessions` - splits enhanced pbp data up into possessions.
- `Shots` (stats.nba.com only)

2.4 Code Examples

2.4.1 Game Data

The following code will instantiate the client and instantiate the Game data object for the given game id with boxscore and possession data.

```
from pbpstats.client import Client

settings = {
    "dir": "/response_data",
    "Boxscore": {"source": "file", "data_provider": "stats_nba"},
    "Possessions": {"source": "file", "data_provider": "stats_nba"},
}
client = Client(settings)
game = client.Game("0021900001")
```

Resource data can be accessed by calling `game.<snake_case_resource_name>.items`. In this case, since 'Boxscore' and 'Possessions' were provided in the settings dict boxscore and possessions data can be accessed via `game.boxscore.items` and `game.possessions.items`. See [Possessions](#) for properties for working with possessions data. See [Boxscore](#) for properties for working with boxscore data.

2.4.2 All Final Games For Season

The following code can be used to get all final games for a season.

```
from pbpstats.client import Client

settings = {
    "Games": {"source": "web", "data_provider": "data_nba"},
}
client = Client(settings)
season = client.Season("nba", "2019-20", "Regular Season")

for final_game in season.games.final_games:
    print(final_game)
```

2.4.3 All Final Games For Day

The following code can be used to get all final games for a season. Note that for day, data_provider must be stats_nba

```
from pbpstats.client import Client

settings = {
    "Games": {"source": "web", "data_provider": "stats_nba"},
}
client = Client(settings)
day = client.Day("12/05/2019", "nba")

for final_game in day.games.final_games:
    print(final_game)
```

2.4.4 Doing Detailed Stuff with Possession Data

The following code will get all possessions that start off a missed field goal.

```
off_rim_miss = [possession for possession in game.possessions.items if possession.
    ↪possession_start_type == "OffAtRimMiss"]
```

For more on what is available for possession data see *Possession* docs

2.4.5 Doing Detailed Stuff with Enhanced PBP Data

The following code will get the average 2pt shot distance on all missed field goals.

```
from pbpstats.resources.enhanced_pbp import FieldGoal

...

shot_dists = []
for possession in game.possessions.items:
    for possession_event in possession.events:
        if isinstance(possession_event, FieldGoal) and not possession_event.is_made_
    ↪and possession_event.shot_value == 2:
```

(continues on next page)

(continued from previous page)

```
shot_dists.append(possesion_event.distance)
print(sum(shot_dists) / len(shot_dists))
```

For more on what is available for enhanced pbp data see `pbpstats.resources.enhanced_pbp` docs

2.5 Note on Ids

Player and team Ids the same player and team ids used by stats.nba.com. Lineup ids are '-' separated player ids (with player ids sorted as strings).

2.6 Issues with raw play-by-play

If you need to fix event order in the play-by-play file you will need to open the pbp file for the game in your data directory and change the order of the events list. Each event is a list and searching for the event number (the event number should be in the exception text somewhere) to find the event is a good place to start to figure out which event needs to be moved around.

3.1 Documentation

3.1.1 Data Loaders

The data loader modules are used to load data for a specific resource either from file stored on disk or via an API request.

3.1.1.1 data.nba.com Data Loaders

data.nba.com data loaders are used to load data for a specific resource either from file stored on disk or via an API request to data.nba.com.

Boxscore

Boxscore

DataNbaBoxscoreLoader loads boxscore data for a game and creates *DataNbaBoxscoreItem* objects for each player and team

The following code will load boxscore data for game id “0021900001” from a file located in a subdirectory of the /data directory

```
from pbpstats.data_loader import DataNbaBoxscoreFileLoader, DataNbaBoxscoreLoader

source_loader = DataNbaBoxscoreFileLoader("/data")
boxscore_loader = DataNbaBoxscoreLoader("0021900001", source_loader)
print(boxscore_loader.items[0].data) # prints dict with a player's boxscore data for_
↪ game
```

```
class pbpstats.data_loader.data_nba.boxscore.loader.DataNbaBoxscoreLoader (game_id,  
                                                                    source_loader)
```

Bases: `pbpstats.data_loader.data_nba.base.DataNbaLoaderBase`

Loads data.nba.com source boxscore data for game. Team/Player data is stored in items attribute as `DataNbaBoxscoreItem` objects

Parameters

- **game_id** (*str*) – NBA Stats Game Id
- **source_loader** – `DataNbaBoxscoreFileLoader` or `DataNbaBoxscoreWebLoader` object

data
returns raw JSON response data

data_provider = 'data_nba'

parent_object = 'Game'

resource = 'Boxscore'

```
class pbpstats.data_loader.data_nba.boxscore.file.DataNbaBoxscoreFileLoader (file_directory)  
Bases: pbpstats.data_loader.data_nba.file_loader.DataNbaFileLoader
```

A `DataNbaBoxscoreFileLoader` object should be instantiated and passed into `DataNbaBoxscoreLoader` when loading data from file

Parameters **file_directory** (*str*) – Directory in which data should be loaded from. The specific file location will be `data_<game_id>.json` in the `/game_details` subdirectory.

load_data (*game_id*)

```
class pbpstats.data_loader.data_nba.boxscore.web.DataNbaBoxscoreWebLoader (file_directory=None)  
Bases: pbpstats.data_loader.data_nba.web_loader.DataNbaWebLoader
```

A `DataNbaBoxscoreWebLoader` object should be instantiated and passed into `DataNbaBoxscoreLoader` when loading data directly from the NBA Stats API

Parameters **file_directory** (*str*) – (optional, use it if you want to store the response data on disk) Directory in which data should be either stored. The specific file location will be `data_<game_id>.json` in the `/game_details` subdirectory. If not provided response data will not be saved on disk.

load_data (*game_id*)

Enhanced PBP

Enhanced PBP

`DataNbaEnhancedPbpLoader` loads pbp data for a game and creates `EnhancedPbpItem` objects for each event

Enhanced data for each event includes current players on floor, score, fouls to give and number of fouls committed by each player, plus additional data depending on event type

The following code will load pbp data for game id “0021900001” from a file located in a subdirectory of the `/data` directory

```

from pbpstats.data_loader import DataNbaEnhancedPbpFileLoader, ↳
↳DataNbaEnhancedPbpLoader

source_loader = DataNbaEnhancedPbpFileLoader("/data")
pbp_loader = DataNbaEnhancedPbpLoader("0021900001", source_loader)
print(pbp_loader.items[0].data) # prints dict with the first event of the game

```

class pbpstats.data_loader.data_nba.enhanced_pbp.loader.**DataNbaEnhancedPbpLoader** (*game_id*, *source_loader*)

Bases: *pbpstats.data_loader.data_nba.pbp.loader.DataNbaPbpLoader*, *pbpstats.data_loader.nba_enhanced_pbp_loader.NbaEnhancedPbpLoader*

Loads data.nba.com source enhanced pbp data for game. Events are stored in items attribute as *EnhancedPbpItem* objects

Parameters

- **game_id** (*str*) – NBA Stats Game Id
- **source_loader** – *DataNbaEnhancedPbpFileLoader* or *DataNbaEnhancedPbpWebLoader* object

Raises *InvalidNumberOfStartersException*: If all 5 players that start the period for a team can't be determined. You can add the correct period starters to over-rides/missing_period_starters.json in your data directory to fix this.

data_provider = 'data_nba'

parent_object = 'Game'

resource = 'EnhancedPbp'

class pbpstats.data_loader.data_nba.enhanced_pbp.file.**DataNbaEnhancedPbpFileLoader** (*file_directory*)

Bases: *pbpstats.data_loader.data_nba.pbp.file.DataNbaPbpFileLoader*

A *DataNbaEnhancedPbpFileLoader* object should be instantiated and passed into *DataNbaEnhancedPbpLoader* when loading data from file

Parameters **file_directory** (*str*) – Directory in which data should be loaded from. The specific file location will be *data_<game_id>.json* in the */pbp* subdirectory.

class pbpstats.data_loader.data_nba.enhanced_pbp.web.**DataNbaEnhancedPbpWebLoader** (*file_directory*)

Bases: *pbpstats.data_loader.data_nba.pbp.web.DataNbaPbpWebLoader*

A *DataNbaEnhancedPbpWebLoader* object should be instantiated and passed into *DataNbaEnhancedPbpLoader* when loading data directly from the NBA Stats API

Parameters **file_directory** (*str*) – (optional, use it if you want to store the response data on disk) Directory in which data should be either stored. The specific file location will be *data_<game_id>.json* in the */pbp* subdirectory. If not provided response data will not be saved on disk.

PBP

PBP

DataNbaPbpLoader loads pbp data for a game and creates *DataNbaPbpItem* objects for each event

The following code will load pbp data for game id “0021900001” from a file located in a subdirectory of the */data* directory

```
from pbpstats.data_loader import DataNbaPbpFileLoader, DataNbaPbpLoader

source_loader = DataNbaPbpFileLoader("/data")
pbp_loader = DataNbaPbpLoader("0021900001", source_loader)
print(pbp_loader.items[0].data)  # prints dict with the first event of the game
```

[illegible]

Loads data.nba.com source pbp data for game. Events are stored in items attribute as *DataNbaPbpItem* objects

Parameters

- **game_id**(*str*) – NBA Stats Game Id
- **source_loader** – *DataNbaPbpFileLoader* or *DataNbaPbpWebLoader* object

data
returns raw JSON response data

```
data_provider = 'data_nba'
```

```
parent_object = 'Game'
```

```
resource = 'Pbp'
```

```
class pbpstats.data_loader.data_nba.pbp.file.DataNbaPbpFileLoader (file_directory=None)
    Bases: pbpstats.data_loader.data_nba.file_loader.DataNbaFileLoader
```

A `DataNbaPbpFileLoader` object should be instantiated and passed into `DataNbaPbpLoader` when loading data from file

Parameters `file_directory` (*str*) – Directory in which data should be loaded from. The specific file location will be `data_<game_id>.json` in the `/pbp` subdirectory.

```
load_data (game_id)
```

```
class pbpstats.data_loader.data_nba.pbp.web.DataNbaPbpWebLoader (file_directory=None)
    Bases: pbpstats.data_loader.data_nba.web_loader.DataNbaWebLoader
```

A `DataNbaPbpWebLoader` object should be instantiated and passed into `DataNbaPbpLoader` when loading data directly from the NBA Stats API

Parameters **file_directory** (*str*) – (optional, use it if you want to store the response data on disk) Directory in which data should be either stored. The specific file location will be *data_<game_id>.json* in the */pbp* subdirectory. If not provided response data will not be saved on disk.

```
load_data(game_id)
```

Possessions

Possessions

`DataNbaPossessionLoader` loads possession data for a game and creates *Possession* objects for each possession

The following code will load possession data for game id “0021900001” from a pbp file located in the /pbp subdirectory of the /data directory


```

from pbpstats.data_loader import DataNbaPossessionFileLoader, DataNbaPossessionLoader

source_loader = DataNbaPossessionFileLoader("/data")
possession_loader = DataNbaPossessionLoader("0021900001", source_loader)
print(possession_loader.items[0].data)  # prints dict with the first possession of
↳ the game

```

class pbpstats.data_loader.data_nba.possessions.loader.**DataNbaPossessionLoader** (*game_id*, *source_loader*)

Bases: *pbpstats.data_loader.nba_possession_loader.NbaPossessionLoader*

Loads data.nba.com source possession data for game. Possessions are stored in items attribute as *Possession* objects

Parameters

- **game_id** (*str*) – NBA Stats Game Id
- **source_loader** – *DataNbaPossessionFileLoader* or *DataNbaPossessionWebLoader* object

data_provider = 'data_nba'

parent_object = 'Game'

resource = 'Possessions'

class pbpstats.data_loader.data_nba.possessions.file.**DataNbaPossessionFileLoader** (*file_directory*)

Bases: object

A *DataNbaPossessionFileLoader* object should be instantiated and passed into *DataNbaPossessionLoader* when loading data from file

Parameters **file_directory** (*str*) – Directory in which data should be loaded from. The specific file location will be *data_<game_id>.json* in the */pbp* subdirectory.

class pbpstats.data_loader.data_nba.possessions.web.**DataNbaPossessionWebLoader** (*file_directory*=None)

Bases: object

A *DataNbaPbpWebLoader* object should be instantiated and passed into *DataNbaPossessionLoader* when loading data directly from the NBA Stats API

Parameters **file_directory** (*str*) – (optional, use it if you want to store the response data on disk) Directory in which data should be either stored. The specific file location will be *data_<game_id>.json* in the */pbp* subdirectory. If not provided response data will not be saved on disk.

Schedule

Schedule

DataNbaScheduleLoader loads schedule data for a season and creates *DataNbaGameItem* objects for each game

The following code will load schedule data for 2019-20 NBA Regular Season

```

from pbpstats.data_loader import DataNbaScheduleFileLoader, DataNbaScheduleLoader

source_loader = DataNbaScheduleFileLoader("/data")

```

(continues on next page)

(continued from previous page)

```
schedule_loader = DataNbaScheduleLoader("nba", "2019-20", "Regular Season", source_
↪loader)
print(schedule_loader.items[0].data)  # prints dict with the first game of the season
```

```
class pbpstats.data_loader.data_nba.schedule.loader.DataNbaScheduleLoader(league,
                                                                           sea-
                                                                           son,
                                                                           sea-
                                                                           son_type,
                                                                           source_loader)
```

Bases: `pbpstats.data_loader.data_nba.base.DataNbaLoaderBase`

Loads `data.nba.com` source schedule data for season. Games are stored in `items` attribute as `DataNbaGameItem` objects

Parameters

- **league** (*str*) – Options are ‘nba’, ‘wnba’ or ‘gleague’
- **season** (*str*) – Can be formatted as either 2019-20 or 2019.
- **season_type** (*str*) – Options are ‘Regular Season’ or ‘Playoffs’ or ‘Play In’
- **source_loader** – `DataNbaScheduleFileLoader` or `DataNbaScheduleWebLoader` object

data

returns raw JSON response data

data_provider = 'data_nba'

parent_object = 'Season'

resource = 'Games'

```
class pbpstats.data_loader.data_nba.schedule.file.DataNbaScheduleFileLoader(file_directory=None)
```

Bases: `pbpstats.data_loader.data_nba.file_loader.DataNbaFileLoader`

A `DataNbaScheduleFileLoader` object should be instantiated and passed into `DataNbaScheduleLoader` when loading data from file

Parameters file_directory (*str*) – Directory in which data should be loaded from. The specific file location will be `data_<league>_<season_year>.json` in the `/schedule` subdirectory.

load_data (*league, season*)

```
class pbpstats.data_loader.data_nba.schedule.web.DataNbaScheduleWebLoader(file_directory=None)
```

Bases: `pbpstats.data_loader.data_nba.web_loader.DataNbaWebLoader`

A `DataNbaScheduleWebLoader` object should be instantiated and passed into `DataNbaScheduleLoader` when loading data directly from the NBA Stats API

Parameters file_directory (*str*) – (optional, use it if you want to store the response data on disk) Directory in which data should be either stored. The specific file location will be `data_<league>_<season_year>.json` in the `/schedule` subdirectory. If not provided response data will not be saved on disk.

league_id

Returns League Id for league.

00 for nba, 10 for wnba, 20 for g-league

`load_data (league, season)`

File Loader

class `pbpstats.data_loader.data_nba.file_loader.DataNbaFileLoader`

Bases: `pbpstats.data_loader.data_nba.base.DataNbaLoaderBase`

Base class for loading data.nba.com files saved on disk.

All data.nba.com data loader classes should inherit from this class.

This class should not be instantiated directly.

Web Loader

class `pbpstats.data_loader.data_nba.web_loader.DataNbaWebLoader`

Bases: `pbpstats.data_loader.data_nba.base.DataNbaLoaderBase`

Base class for loading data from data.nba.com API request.

All data.nba.com data loader classes should inherit from this class.

This class should not be instantiated directly.

3.1.1.2 live Data Loaders

live data loaders are used to load data for a specific resource either from file stored on disk or via an API request to live endpoint at [nba.com](https://data.nba.com).

Boxscore

Boxscore

`LiveBoxscoreLoader` loads boxscore data for a game and creates `LiveBoxscoreItem` objects for each player and team

The following code will load boxscore data for game id “0021900001” from a file located in a subdirectory of the /data directory

```
from pbpstats.data_loader import LiveBoxscoreFileLoader, LiveBoxscoreLoader

source_loader = LiveBoxscoreFileLoader("/data")
boxscore_loader = LiveBoxscoreLoader("0021900001", source_loader)
print(boxscore_loader.items[0].data) # prints dict with a player's boxscore data for_
↪ game
```

class `pbpstats.data_loader.live.boxscore.loader.LiveBoxscoreLoader` (*game_id*, *source_loader*)

Bases: `pbpstats.data_loader.live.base.LiveLoaderBase`

Loads data.nba.com source boxscore data for game. Team/Player data is stored in items attribute as `LiveBoxscoreItem` objects

Parameters

- **game_id** (*str*) – NBA Stats Game Id

- **source_loader** – *LiveBoxscoreFileLoader* or *LiveBoxscoreWebLoader* object

data
returns raw JSON response data

data_provider = 'live'

parent_object = 'Game'

resource = 'Boxscore'

class `pbpstats.data_loader.live.boxscore.file.LiveBoxscoreFileLoader` (*file_directory*)
Bases: `pbpstats.data_loader.live.file_loader.LiveFileLoader`

A *LiveBoxscoreFileLoader* object should be instantiated and passed into *LiveBoxscoreLoader* when loading data from file

Parameters **file_directory** (*str*) – Directory in which data should be loaded from. The specific file location will be *live_<game_id>.json* in the */game_details* subdirectory.

load_data (*game_id*)

class `pbpstats.data_loader.live.boxscore.web.LiveBoxscoreWebLoader` (*file_directory=None*)
Bases: `pbpstats.data_loader.live.web_loader.LiveWebLoader`

LiveBoxscoreWebLoader object should be instantiated and passed into *LiveBoxscoreLoader* when loading data directly from the NBA Stats API

Parameters **file_directory** (*str*) – (optional, use it if you want to store the response data on disk) Directory in which data should be either stored. The specific file location will be *live_<game_id>.json* in the */game_details* subdirectory. If not provided response data will not be saved on disk.

load_data (*game_id*)

Enhanced PBP

Enhanced PBP

LiveEnhancedPbpLoader loads pbp data for a game and creates *EnhancedPbpItem* objects for each event

Enhanced data for each event includes current players on floor, score, fouls to give and number of fouls committed by each player, plus additional data depending on event type

The following code will load pbp data for game id “0021900001” from a file located in a subdirectory of the /data directory

```
from pbpstats.data_loader import LiveEnhancedPbpFileLoader, LiveEnhancedPbpLoader

source_loader = LiveEnhancedPbpFileLoader("/data")
pbp_loader = LiveEnhancedPbpLoader("0021900001", source_loader)
print(pbp_loader.items[0].data)  # prints dict with the first event of the game
```

class `pbpstats.data_loader.live.enhanced_pbp.loader.LiveEnhancedPbpLoader` (*game_id*, *source_loader*)
Bases: `pbpstats.data_loader.live.pbp.loader.LivePbpLoader`, `pbpstats.data_loader.nba_enhanced_pbp_loader.NbaEnhancedPbpLoader`

Loads data.nba.com source enhanced pbp data for game. Events are stored in items attribute as *EnhancedPbpItem* objects

Parameters

- **game_id**(*str*) – NBA Stats Game Id
- **source_loader** – *LiveEnhancedPbpFileLoader* or *LiveEnhancedPbpWebLoader* object

data_provider = 'live'

parent_object = 'Game'

resource = 'EnhancedPbp'

class `pbpstats.data_loader.live.enhanced_pbp.file.LiveEnhancedPbpFileLoader` (*file_directory*)
 Bases: `pbpstats.data_loader.live.pbp.file.LivePbpFileLoader`

A `LiveEnhancedPbpFileLoader` object should be instantiated and passed into `LiveEnhancedPbpLoader` when loading data from file

Parameters

- **game_id**(*str*) – NBA Stats Game Id
- **file_directory**(*str*) – Directory in which data should be loaded from. The specific file location will be *live_<game_id>.json* in the */pbp* subdirectory.

class `pbpstats.data_loader.live.enhanced_pbp.web.LiveEnhancedPbpWebLoader` (*file_directory=None*)
 Bases: `pbpstats.data_loader.live.pbp.web.LivePbpWebLoader`

A `LiveEnhancedPbpWebLoader` object should be instantiated and passed into `LiveEnhancedPbpLoader` when loading data directly from the NBA Stats API

Parameters **file_directory** (*str*) – (optional, use it if you want to store the response data on disk) Directory in which data should be either stored. The specific file location will be *live_<game_id>.json* in the */pbp* subdirectory. If not provided response data will not be saved on disk.

PBP**PBP**

`LivePbpLoader` loads pbp data for a game and creates `LivePbpItem` objects for each event

The following code will load pbp data for game id “0021900001” from a file located in a subdirectory of the */data* directory

```
from pbpstats.data_loader import LivePbpFileLoader, LivePbpLoader

source_loader = LivePbpFileLoader("/data")
pbp_loader = LivePbpLoader("0021900001", source_loader)
print(pbp_loader.items[0].data)  # prints dict with the first event of the game
```

class `pbpstats.data_loader.live.pbp.loader.LivePbpLoader` (*game_id*, *source_loader*)
 Bases: `pbpstats.data_loader.live.base.LiveLoaderBase`

Loads live data source pbp data for game. Events are stored in *items* attribute as `LivePbpItem` objects

Parameters

- **game_id**(*str*) – NBA Stats Game Id
- **source_loader** – *LivePbpFileLoader* or *LivePbpWebLoader* object

data
returns raw JSON response data

data_provider = 'live'

parent_object = 'Game'

resource = 'Pbp'

class `pbpstats.data_loader.live.pbp.file.LivePbpFileLoader` (*file_directory*)

Bases: `pbpstats.data_loader.live.file_loader.LiveFileLoader`

A `LivePbpFileLoader` object should be instantiated and passed into `LivePbpLoader` when loading data from file

Parameters

- **game_id** (*str*) – NBA Stats Game Id
- **file_directory** (*str*) – Directory in which data should be loaded from. The specific file location will be `live_<game_id>.json` in the `/pbp` subdirectory.

load_data (*game_id*)

class `pbpstats.data_loader.live.pbp.web.LivePbpWebLoader` (*file_directory=None*)

Bases: `pbpstats.data_loader.live.web_loader.LiveWebLoader`

A `LivePbpWebLoader` object should be instantiated and passed into `LivePbpLoader` when loading data directly from the NBA Stats API

Parameters **file_directory** (*str*) – (optional, use it if you want to store the response data on disk) Directory in which data should be either stored. The specific file location will be `live_<game_id>.json` in the `/pbp` subdirectory. If not provided response data will not be saved on disk.

load_data (*game_id*)

Possessions

Possessions

`LivePossessionLoader` loads possession data for a game and creates `Possession` objects for each possession

The following code will load possession data for game id “0021900001” from a pbp file located in the `/pbp` subdirectory of the `/data` directory

```
from pbpstats.data_loader import LivePossessionFileLoader, LivePossessionLoader

source_loader = LivePossessionFileLoader("/data")
pbp_loader = LivePossessionLoader("0021900001", source_loader)
print(pbp_loader.items[0].data)  # prints dict with the first event of the game
```

class `pbpstats.data_loader.live.possessions.loader.LivePossessionLoader` (*game_id*, *source_loader*)

Bases: `pbpstats.data_loader.nba_possession_loader.NbaPossessionLoader`

Loads live data source possession data for game. Possessions are stored in `items` attribute as `Possession` objects

Parameters

- **game_id** (*str*) – NBA Stats Game Id

- **source_loader** – *LivePossessionFileLoader* or *LivePossessionWebLoader* object

data_provider = 'live'

parent_object = 'Game'

resource = 'Possessions'

class pbpstats.data_loader.live.possessions.file.**LivePossessionFileLoader** (*file_directory*)
Bases: object

A *LivePossessionFileLoader* object should be instantiated and passed into *LivePossessionLoader* when loading data from file

Parameters

- **game_id** (*str*) – NBA Stats Game Id
- **file_directory** (*str*) – Directory in which data should be loaded from. The specific file location will be *live_<game_id>.json* in the */pbp* subdirectory.

class pbpstats.data_loader.live.possessions.web.**LivePossessionWebLoader** (*file_directory=None*)
Bases: object

A *LivePossessionWebLoader* object should be instantiated and passed into *LivePossessionLoader* when loading data directly from the NBA Stats API

Parameters **file_directory** (*str*) – (optional, use it if you want to store the response data on disk) Directory in which data should be either stored. The specific file location will be *live_<game_id>.json* in the */pbp* subdirectory. If not provided response data will not be saved on disk.

File Loader

class pbpstats.data_loader.live.file_loader.**LiveFileLoader**
Bases: pbpstats.data_loader.live.base.LiveLoaderBase

Base class for loading live data files saved on disk.

All live data data loader classes should inherit from this class.

This class should not be instantiated directly.

Web Loader

class pbpstats.data_loader.live.web_loader.**LiveWebLoader**
Bases: pbpstats.data_loader.live.base.LiveLoaderBase

Base class for loading data from live data API request.

All live data data loader classes should inherit from this class.

This class should not be instantiated directly.

3.1.1.3 stats.nba.com Data Loaders

stats.nba.com data loaders are used to load data for a specific resource either from file stored on disk or via an API request to stats.nba.com.

Base

class `pbpstats.data_loader.stats_nba.base.StatsNbaLoaderBase`

Bases: `object`

Base Class for all stats.nba.com data loaders

This class should not be instantiated directly

data

returns data from response JSON as a list of dicts

static dedupe_events_row_set (*events_row_set*)

Dedupes list of results while preserving order

Used to dedupe events rowSets pbp response because some games have duplicate events

Parameters *events_row_set* (*list*) – List of results from API Response

Returns deduped list of results

league

Returns League for game id.

First 2 in game id represent league - 00 for nba, 10 for wnba, 20 for g-league

league_id

Returns League Id for league.

00 for nba, 10 for wnba, 20 for g-league

make_list_of_dicts (*results_set_index=0*)

Creates list of dicts from source data

Parameters *results_set_index* (*int*) – Index results are in. Default is 0

Returns list of dicts with data for results

season

Returns season for game id

4th and 5th characters in game id represent season year ex. for 2016-17 season 4th and 5th characters would be 16 and season should return 2016-17 For WNBA just returns season year

season_type

Returns season type for game id

3rd character in game id represent season type - 2 for reg season, 4 for playoffs, 5 for play in

Boxscore

Boxscore

`StatsNbaBoxscoreLoader` loads boxscore data for a game and creates `StatsNbaBoxscoreItem` objects for each player and team

The following code will load boxscore data for game id “0021900001” from a file located in a subdirectory of the /data directory


```

from pbpstats.data_loader import StatsNbaBoxscoreFileLoader, StatsNbaBoxscoreLoader

source_loader = StatsNbaBoxscoreFileLoader("/data")
boxscore_loader = StatsNbaBoxscoreLoader("0021900001", source_loader)
print(boxscore_loader.items[0].data) # prints dict with a player's boxscore data for
↪ game

```

class pbpstats.data_loader.stats_nba.boxscore.loader.**StatsNbaBoxscoreLoader** (*game_id*, *source_loader*)

Bases: *pbpstats.data_loader.stats_nba.base.StatsNbaLoaderBase*

Loads stats.nba.com source boxscore data for game. Team/Player data is stored in items attribute as *StatsNbaBoxscoreItem* objects

Parameters

- **game_id** (*str*) – NBA Stats Game Id
- **source_loader** – *StatsNbaBoxscoreFileLoader* or *StatsNbaBoxscoreWebLoader* object

data_provider = 'stats_nba'

parent_object = 'Game'

resource = 'Boxscore'

class pbpstats.data_loader.stats_nba.boxscore.file.**StatsNbaBoxscoreFileLoader** (*file_directory*)

Bases: *pbpstats.data_loader.stats_nba.file_loader.StatsNbaFileLoader*

A *StatsNbaBoxscoreFileLoader* object should be instantiated and passed into *StatsNbaBoxscoreLoader* when loading data from file

Parameters **file_directory** (*str*) – Directory in which data should be loaded from. The specific file location will be *stats_boxscore_<game_id>.json* in the */game_details* subdirectory.

load_data (*game_id*)

class pbpstats.data_loader.stats_nba.boxscore.web.**StatsNbaBoxscoreWebLoader** (*file_directory=None*)

Bases: *pbpstats.data_loader.stats_nba.web_loader.StatsNbaWebLoader*

A *StatsNbaBoxscoreWebLoader* object should be instantiated and passed into *StatsNbaBoxscoreLoader* when loading data directly from the NBA Stats API

Parameters **file_directory** (*str*) – (optional, use it if you want to store the response data on disk) Directory in which data should be either stored. The specific file location will be *stats_boxscore_<game_id>.json* in the */game_details* subdirectory. If not provided response data will not be saved on disk.

load_data (*game_id*)

Enhanced PBP

Enhanced PBP

StatsNbaEnhancedPbpLoader loads pbp data for a game and creates *EnhancedPbpItem* objects for each event

Enhanced data for each event includes current players on floor, score, fouls to give and number of fouls committed by each player, plus additional data depending on event type

The following code will load pbp data for game id “0021900001” from a file located in a subdirectory of the /data directory

```
from pbpstats.data_loader import StatsNbaEnhancedPbpFileLoader, \
↳ StatsNbaEnhancedPbpLoader

source_loader = StatsNbaEnhancedPbpFileLoader("/data")
pbp_loader = StatsNbaEnhancedPbpLoader("0021900001", source_loader)
print(pbp_loader.items[0].data) # prints dict with the first event of the game
```

class pbpstats.data_loader.stats_nba.enhanced_pbp.loader.**StatsNbaEnhancedPbpLoader** (*game_id*, *source_loader*)

Bases: *pbpstats.data_loader.stats_nba.pbp.loader.StatsNbaPbpLoader*, *pbpstats.data_loader.nba_enhanced_pbp_loader.NbaEnhancedPbpLoader*

Loads stats.nba.com source enhanced pbp data for game. Events are stored in items attribute as *EnhancedPbpItem* objects

Parameters

- **game_id** (*str*) – NBA Stats Game Id
- **source_loader** – *StatsNbaEnhancedPbpFileLoader* or *StatsNbaEnhancedPbpWebLoader* object

Raises *InvalidNumberOfStartersException*: If all 5 players that start the period for a team can’t be determined. You can add the correct period starters to overrides/missing_period_starters.json in your data directory to fix this.

Raises *EventOrderError*: If rebound event is not immediately following a missed shot event. You can manually edit the event order in the pbp file stored on disk to fix this.

data_provider = 'stats_nba'

parent_object = 'Game'

resource = 'EnhancedPbp'

class pbpstats.data_loader.stats_nba.enhanced_pbp.file.**StatsNbaEnhancedPbpFileLoader** (*file_directory*)

Bases: *pbpstats.data_loader.stats_nba.pbp.file.StatsNbaPbpFileLoader*

A *StatsNbaEnhancedPbpFileLoader* object should be instantiated and passed into *StatsNbaEnhancedPbpLoader* when loading data from file

Parameters **file_directory** (*str*) – Directory in which data should be loaded from. The specific file location will be *stats_<game_id>.json* in the */pbp* subdirectory.

class pbpstats.data_loader.stats_nba.enhanced_pbp.web.**StatsNbaEnhancedPbpWebLoader** (*file_directory*)

Bases: *pbpstats.data_loader.stats_nba.pbp.web.StatsNbaPbpWebLoader*

A *StatsNbaEnhancedPbpWebLoader* object should be instantiated and passed into *StatsNbaEnhancedPbpLoader* when loading data directly from the NBA Stats API

Parameters **file_directory** (*str*) – (optional, use it if you want to store the response data on disk) Directory in which data should be either stored. The specific file location will be *stats_<game_id>.json* in the */pbp* subdirectory. If not provided response data will not be saved on disk.

PBP

PBP

StatsNbaPbpLoader loads pbp data for a game and creates *StatsNbaPbpItem* objects for each event

The following code will load pbp data for game id “0021900001” from a file located in a subdirectory of the /data directory

```
from pbpstats.data_loader import StatsNbaPbpFileLoader, StatsNbaPbpLoader

source_loader = StatsNbaPbpFileLoader("/data")
pbp_loader = StatsNbaPbpLoader("0021900001", source_loader)
print(pbp_loader.items[0].data)  # prints dict with the first event of the game
```

```
class pbpstats.data_loader.stats_nba.pbp.loader.StatsNbaPbpLoader (game_id,
                                                                    source_loader)
```

Bases: *pbpstats.data_loader.stats_nba.base.StatsNbaLoaderBase*

Loads stats.nba.com source pbp data for game. Events are stored in items attribute as *StatsNbaPbpItem* objects

Parameters

- **game_id** (*str*) – NBA Stats Game Id
- **source_loader** – *StatsNbaPbpFileLoader* or *StatsNbaPbpWebLoader* object

data_provider = 'stats_nba'

parent_object = 'Game'

resource = 'Pbp'

```
class pbpstats.data_loader.stats_nba.pbp.file.StatsNbaPbpFileLoader (file_directory)
```

Bases: *pbpstats.data_loader.stats_nba.file_loader.StatsNbaFileLoader*

A *StatsNbaPbpFileLoader* object should be instantiated and passed into *StatsNbaPbpLoader* when loading data from file

Parameters file_directory (*str*) – Directory in which data should be loaded from. The specific file location will be *stats_<game_id>.json* in the */pbp* subdirectory.

load_data (*game_id*)

```
class pbpstats.data_loader.stats_nba.pbp.web.StatsNbaPbpWebLoader (file_directory=None)
```

Bases: *pbpstats.data_loader.stats_nba.web_loader.StatsNbaWebLoader*

A *StatsNbaPbpWebLoader* object should be instantiated and passed into *StatsNbaPbpLoader* when loading data directly from the NBA Stats API

Parameters file_directory (*str*) – (optional, use it if you want to store the response data on disk) Directory in which data should be either stored. The specific file location will be *stats_<game_id>.json* in the */pbp* subdirectory. If not provided response data will not be saved on disk.

load_data (*game_id*)

Possessions

Possessions

StatsNbaPossessionLoader loads possession data for a game and creates *Possession* objects for each possession

The following code will load possession data for game id “0021900001” from a pbp file located in the /pbp subdirectory of the /data directory

```
from pbpstats.data_loader import StatsNbaPossessionFileLoader, \
    StatsNbaPossessionLoader

source_loader = StatsNbaPossessionFileLoader("/data")
pbp_loader = StatsNbaPossessionLoader("0021900001", source_loader)
print(pbp_loader.items[0].data)  # prints dict with the first event of the game
```

class pbpstats.data_loader.stats_nba.possessions.loader.**StatsNbaPossessionLoader** (*game_id*, *source_loader*)

Bases: *pbpstats.data_loader.nba_possession_loader.NbaPossessionLoader*

Loads stats.nba.com source possession data for game. Possessions are stored in items attribute as *Possession* objects

Parameters

- **game_id** (*str*) – NBA Stats Game Id
- **source_loader** – *StatsNbaPossessionFileLoader* or *StatsNbaPossessionWebLoader* object

Raises TeamHasBackToBackPossessionsException: If team has the ball on back-to-back possessions.

data_provider = 'stats_nba'

league

Returns League for game id.

First 2 in game id represent league - 00 for nba, 10 for wnba, 20 for g-league

parent_object = 'Game'

resource = 'Possessions'

exception pbpstats.data_loader.stats_nba.possessions.loader.**TeamHasBackToBackPossessionsException**

Bases: Exception

Class for exception when a team is credited with back-to-back possessions.

You can manually edit the event order in the pbp file stored on disk or add an event to the overrides file in your data directory to fix this.

class pbpstats.data_loader.stats_nba.possessions.file.**StatsNbaPossessionFileLoader** (*file_directory*)

Bases: object

A *StatsNbaPossessionFileLoader* object should be instantiated and passed into *StatsNbaPossessionLoader* when loading data from file

Parameters **file_directory** (*str*) – Directory in which data should be loaded from. The specific file location will be *stats_<game_id>.json* in the /pbp subdirectory.

class pbpstats.data_loader.stats_nba.possessions.web.**StatsNbaPossessionWebLoader** (*file_directory*)

Bases: object

A `StatsNbaPossessionWebLoader` object should be instantiated and passed into `StatsNbaPossessionLoader` when loading data directly from the NBA Stats API

Parameters `file_directory` (*str*) – (optional, use it if you want to store the response data on disk) Directory in which data should be either stored. The specific file location will be `stats_<game_id>.json` in the `/pbp` subdirectory. If not provided response data will not be saved on disk.

Game Finder

Boxscore

`StatsNbaGameFinderLoader` loads all games for a season and creates `StatsNbaGameItem` objects for each game

The following code will load data for the 2019-20 NBA Regular Season

```
from pbpstats.data_loader import StatsNbaGameFinderWebLoader, StatsNbaGameFinderLoader

source_loader = StatsNbaGameFinderWebLoader("/data")
game_finder_loader = StatsNbaGameFinderLoader("nba", "2019-20", "Regular Season",
↪source_loader)
print(game_finder_loader.items[0].data) # prints dict for first game
```

```
class pbpstats.data_loader.stats_nba.game_finder.loader.StatsNbaGameFinderLoader(league,
sea-
son,
sea-
son_type,
source_loader)
```

Bases: `pbpstats.data_loader.stats_nba.base.StatsNbaLoaderBase`

Loads stats.nba.com source data for season. Games are stored in items attribute as `StatsNbaGameItem` objects

Parameters

- **league** (*str*) – Options are 'nba', 'wnba' or 'gleague'
- **season** (*str*) – Formatted as 2019-20 for NBA and G-League, 2019 of WNBA.
- **season_type** (*str*) – Options are 'Regular Season' or 'Playoffs' or 'Play In'
- **source_loader** – `StatsNbaGameFinderFileLoader` or `StatsNbaGameFinderWebLoader` object

`data_provider = 'stats_nba'`

`parent_object = 'Season'`

`resource = 'Games'`

```
class pbpstats.data_loader.stats_nba.game_finder.file.StatsNbaGameFinderFileLoader(file_directory)
```

Bases: `pbpstats.data_loader.stats_nba.file_loader.StatsNbaFileLoader`

A `StatsNbaGameFinderFileLoader` object should be instantiated and passed into `StatsNbaGameFinderLoader` when loading data from file

Parameters `file_directory` (*str*) – Directory in which data should be loaded from. The specific file location will be `stats_<league>_<season>_<season_type>.json` in the `/schedule` subdirectory.

`load_data (league, season, season_type)`

class `pbpstats.data_loader.stats_nba.game_finder.web.StatsNbaGameFinderWebLoader` (*file_directory*):
Bases: `pbpstats.data_loader.stats_nba.web_loader.StatsNbaWebLoader`

A `StatsNbaGameFinderWebLoader` object should be instantiated and passed into `StatsNbaGameFinderLoader` when loading data directly from the NBA Stats API

Parameters `file_directory (str)` – (optional, use it if you want to store the response data on disk) Directory in which data should be either stored. The specific file location will be `stats_<league>_<season>_<season_type>.json` in the `/schedule` subdirectory. If not provided response data will not be saved on disk.

`load_data (league, season, season_type)`

Scoreboard

Boxscore

`StatsNbaScoreboardLoader` loads all games for a date and creates `StatsNbaGameItem` objects for each game

The following code will load data for 02/03/2020

```
from pbpstats.data_loader import StatsNbaScoreboardFileLoader, \
    StatsNbaScoreboardLoader

source_loader = StatsNbaScoreboardFileLoader("/data")
game_finder_loader = StatsNbaScoreboardLoader("02/03/2020", "nba", source_loader)
print(game_finder_loader.items[0].data) # prints dict for first game
```

class `pbpstats.data_loader.stats_nba.scoreboard.loader.StatsNbaScoreboardLoader` (*date*, *league_string*, *source_loader*):

Bases: `pbpstats.data_loader.stats_nba.file_loader.StatsNbaFileLoader`,
`pbpstats.data_loader.stats_nba.web_loader.StatsNbaWebLoader`

Loads stats.nba.com source data for date. Games are stored in items attribute as `StatsNbaGameItem` objects

Parameters

- **date** (*str*) – Formatted as MM/DD/YYYY
- **league_string** (*str*) – Options are 'nba', 'wnba' or 'gleague'
- **source_loader** – `StatsNbaScoreboardFileLoader` or `StatsNbaScoreboardWebLoader` object

`data_provider = 'stats_nba'`

`parent_object = 'Day'`

`resource = 'Games'`

class `pbpstats.data_loader.stats_nba.scoreboard.file.StatsNbaScoreboardFileLoader` (*file_directory*):
Bases: `pbpstats.data_loader.stats_nba.file_loader.StatsNbaFileLoader`

A `StatsNbaScoreboardFileLoader` object should be instantiated and passed into `StatsNbaScoreboardLoader` when loading data from file

Parameters `file_directory (str)` – Directory in which data should be loaded from. The specific file location will be `stats_<league>_<date>.json` in the `/schedule` subdirectory.

`load_data (date, league_string)`

class `pbpstats.data_loader.stats_nba.scoreboard.web.StatsNbaScoreboardWebLoader` (*file_directory=*
Bases: `pbpstats.data_loader.stats_nba.web_loader.StatsNbaWebLoader`

A `StatsNbaScoreboardWebLoader` object should be instantiated and passed into `StatsNbaScoreboardLoader` when loading data directly from the NBA Stats API

Parameters `file_directory (str)` – (optional, use it if you want to store the response data on disk) Directory in which data should be either stored. The specific file location will be `stats_<league>_<date>.json` in the `/schedule` subdirectory. If not provided response data will not be saved on disk.

league_id

Returns League Id for league.

00 for nba, 10 for wnba, 20 for g-league

`load_data (date, league_string)`

Shots

Boxscore

`StatsNbaShotsLoader` loads shot data for a game and creates `StatsNbaShot` objects for all shots

The following code will load shot data for game id “0021900001” from a file located in a subdirectory of the `/data` directory

```
from pbpstats.data_loader import StatsNbaShotsFileLoader, StatsNbaShotsLoader

source_loader = StatsNbaShotsFileLoader("/data")
shot_loader = StatsNbaShotsLoader("0021900001", source_loader)
print(shot_loader.items[0].data) # prints dict with data for one shot from game
```

class `pbpstats.data_loader.stats_nba.shots.loader.StatsNbaShotsLoader` (*game_id,*
source_loader)
Bases: `pbpstats.data_loader.stats_nba.base.StatsNbaLoaderBase`

Loads stats.nba.com source shot data for game. Shots are stored in items attribute as `StatsNbaShot` objects

Parameters

- **game_id** (*str*) – NBA Stats Game Id
- **source_loader** – `StatsNbaShotsFileLoader` or `StatsNbaShotsWebLoader` object

data_provider = 'stats_nba'

make_list_of_dicts (*results_set_index=0*)

Creates list of dicts from home and away source data

Parameters `results_set_index (int)` – Index results are in. Default is 0

Returns list of dicts with shot data for all shots

parent_object = 'Game'

resource = 'Shots'

```
class pbpstats.data_loader.stats_nba.shots.file.StatsNbaShotsFileLoader (file_directory)
    Bases: pbpstats.data_loader.stats_nba.file_loader.StatsNbaFileLoader
```

A StatsNbaShotsFileLoader object should be instantiated and passed into StatsNbaShotsLoader when loading data from file

Parameters *file_directory* (*str*) – Directory in which data should be loaded from. The specific file location will be *stats_home_shots_<game_id>.json* and *stats_away_shots_<game_id>.json* in the */game_details* subdirectory.

load_data (*game_id*)

```
class pbpstats.data_loader.stats_nba.shots.web.StatsNbaShotsWebLoader (file_directory=None)
    Bases: pbpstats.data_loader.stats_nba.web_loader.StatsNbaWebLoader
```

A StatsNbaShotsWebLoader object should be instantiated and passed into StatsNbaShotsLoader when loading data directly from the NBA Stats API

Parameters *file_directory* (*str*) – (optional, use it if you want to store the response data on disk) Directory in which data should be either stored. The specific file location will be *stats_home_shots_<game_id>.json* and *stats_away_shots_<game_id>.json* in the */game_details* subdirectory. If not provided response data will not be saved on disk.

load_data (*game_id*)

Game Summary

Boxscore

StatsNbaSummaryLoader loads summary data for a game and creates *StatsNbaGameItem* objects for game. The following code will load summary data for game id “0021900001” from a file located in a subdirectory of the */data* directory

```
from pbpstats.data_loader.stats_nba.summary.file import StatsNbaSummaryFileLoader
from pbpstats.data_loader.stats_nba.summary.loader import StatsNbaSummaryLoader

source_loader = StatsNbaSummaryFileLoader("/data")
summary_loader = StatsNbaSummaryLoader("0021900001", source_loader)
print(summary_loader.items[0].data) # prints game summary dict for game
```

```
class pbpstats.data_loader.stats_nba.summary.loader.StatsNbaSummaryLoader (game_id,
                                                                           source_loader)
```

Bases: *pbpstats.data_loader.stats_nba.base.StatsNbaLoaderBase*

Loads stats.nba.com source summary data for game. Summary data is stored in items attribute as *StatsNbaGameItem* objects

Parameters

- **game_id** (*str*) – NBA Stats Game Id
- **source_loader** – *StatsNbaSummaryFileLoader* or *StatsNbaSummaryWebLoader* object

data_provider = 'stats_nba'

parent_object = 'Game'

resource = 'Games'


```
class pbpstats.data_loader.stats_nba.summary.file.StatsNbaSummaryFileLoader (file_directory)
    Bases: pbpstats.data_loader.stats_nba.file_loader.StatsNbaFileLoader
```

A StatsNbaSummaryFileLoader object should be instantiated and passed into StatsNbaSummaryLoader when loading data from file

Parameters *file_directory* (*str*) – Directory in which data should be loaded from. The specific file location will be *stats_summary_<game_id>.json* in the */game_details* subdirectory.

```
load_data (game_id)
```

```
class pbpstats.data_loader.stats_nba.summary.web.StatsNbaSummaryWebLoader (file_directory=None)
    Bases: pbpstats.data_loader.stats_nba.web_loader.StatsNbaWebLoader
```

A StatsNbaSummaryWebLoader object should be instantiated and passed into StatsNbaSummaryLoader when loading data directly from the NBA Stats API

Parameters *file_directory* (*str*) – (optional, use it if you want to store the response data on disk) Directory in which data should be either stored. The specific file location will be *stats_summary_<game_id>.json* in the */game_details* subdirectory. If not provided response data will not be saved on disk.

```
load_data (game_id)
```

File Loader

```
class pbpstats.data_loader.stats_nba.file_loader.StatsNbaFileLoader
    Bases: pbpstats.data_loader.stats_nba.base.StatsNbaLoaderBase
```

Base class for loading stats.nba.com files saved on disk.

All stats.nba.com data loader classes should inherit from this class.

This class should not be instantiated directly.

Web Loader

```
class pbpstats.data_loader.stats_nba.web_loader.StatsNbaWebLoader
    Bases: pbpstats.data_loader.stats_nba.base.StatsNbaLoaderBase
```

Base class for loading data from data.nba.com API request.

All stats.nba.com data loader classes should inherit from this class.

This class should not be instantiated directly.

3.1.1.4 Data Loader Factory

DataLoaderFactory can be used to create data loader objects from the data_loader module.

The following code will create a data loader object for loading enhanced pbp from stats.nba.com.

```
from pbpstats.data_loader.factory import DataLoaderFactory

data_loader = DataLoaderFactory()
stats_enhanced_pbp_data_loader = data_loader.get_data_loader("stats_nba", "EnhancedPbp
↪")
```

(continues on next page)

(continued from previous page)

```
print(stats_enhanced_pbp_data_loader[0])
# prints "<class 'pbpstats.data_loader.stats_nba.enhanced_pbp_loader.
↳ StatsNbaEnhancedPbpLoader'>"
```

class pbpstats.data_loader.factory.DataLoaderFactory

Bases: object

Class for factory of data loader classes. On initialization will load in all data loader classes in data_loader module

get_data_loader (data_provider, resource)

Gets data loader classes for given data provider and resource.

Parameters

- **data_provider** (*str*) – Which data provider should data be loaded from. Options are 'stats_nba' or 'data_nba' or 'live'
- **resource** (*str*) – Name of class from resources directory

Returns list of data loader classes**Return type** list

3.1.1.5 Enhanced PBP Loader

class pbpstats.data_loader.nba_enhanced_pbp_loader.NbaEnhancedPbpLoader

Bases: object

Class for shared methods between DataNbaEnhancedPbpLoader and StatsNbaEnhancedPbpLoader

Both DataNbaEnhancedPbpLoader and StatsNbaEnhancedPbpLoader should inherit from this class

This class should not be instantiated directly

3.1.1.6 Possession Loader

class pbpstats.data_loader.nba_possession_loader.NbaPossessionLoader

Bases: object

Class for shared methods between DataNbaPossessionLoader and StatsNbaPossessionLoader

Both DataNbaPossessionLoader and StatsNbaPossessionLoader should inherit from this class

This class should not be instantiated directly

3.1.2 Objects

3.1.2.1 Day

Instantiating a Day object will load all resources for the Day object that were set in the settings when the client was instantiated

The following code will instantiate the client and get game data for games played on 02/03/2020

```

from pbpstats.client import Client

settings = {
    "dir": "/response_data",
    "Games": {"source": "web", "data_provider": "stats_nba"}
}
client = Client(settings)
day = client.Day("02/03/2020", "nba")
for game in day.games.items:
    print(game.data)

```

class pbpstats.objects.day.**Day** (*date, league*)

Bases: object

Class for loading resource data from data loaders with a parent_object of Day

Parameters

- **date** (*str*) – Formatted as MM/DD/YYYY
- **league** (*str*) – Options are 'nba', 'wnba' or 'gleague'

3.1.2.2 Game

Instantiating a Game object will load all resources for the Game object that were set in the settings when the client was instantiated

The following code will instantiate the client and get possession data for game id 0021900001 from files in /response_data subdirectories

```

from pbpstats.client import Client

settings = {
    "dir": "/response_data",
    "Possessions": {"source": "file", "data_provider": "stats_nba"}
}
client = Client(settings)
game = client.Game('0021900001')
for possession in game.possessions.items:
    print(possession)

```

class pbpstats.objects.game.**Game** (*game_id*)

Bases: object

Class for loading resource data from data loaders with a parent_object of Game

Parameters **game_id** (*str*) – NBA Stats Game Id

3.1.2.3 Season

Instantiating a Season object will load all resources for the Season object that were set in the settings when the client was instantiated

The following code will instantiate the client and get all games for the 2019-20 NBA Regular Season and store the schedule response in a /response_data subdirectory

```
from pbpstats.client import Client

settings = {
    "dir": "/response_data",
    "Games": {"source": "web", "data_provider": "data_nba"}
}
client = Client(settings)
season = client.Season("nba", "2019-20", "Regular Season")
for game in season.games.items:
    print(game)
```

class pbpstats.objects.season.**Season** (*league, season, season_type*)
Bases: object

Class for loading resource data from data loaders with a parent_object of Season

Parameters

- **league** (*str*) – Options are ‘nba’, ‘wnba’ or ‘gleague’
- **season** (*str*) – Can be formatted as either 2019-20 or 2019.
- **season_type** (*str*) – Options are ‘Regular Season’ or ‘Playoffs’ or ‘Play In’

3.1.3 Resources

3.1.3.1 Boxscore

Boxscore

The Boxscore class has some basic properties for handling boxscore data

class pbpstats.resources.boxscore.boxscore.**Boxscore** (*items*)
Bases: pbpstats.resources.base.Base

Class for boxscore items

Parameters **items** (*list*) – list of either *StatsNbaBoxscoreItem* or *LiveBoxscoreItem* or *DataNbaBoxscoreItem* items, typically from a boxscore data loader

data
returns dict with boxscore items split up by player and team

player_items
returns list of player boxscore items

player_name_map
returns dict mapping player id to player name

player_team_map
returns dict mapping player id to team id

team_items
returns list of team boxscore items

data.nba.com Boxscore Item

```
class pbpstats.resources.boxscore.data_nba_boxscore_item.DataNbaBoxscoreItem(item,
                                                                              team_id=None,
                                                                              team_abbreviation=None)
```

Bases: object

Class for boxscore items from data.nba.com

Parameters

- **item** (*dict*) – dict with boxscore stats from response
- **team_id** (*int*) – (optional) team id is not in dict with stats and can be added in here
- **team_abbreviation** (*str*) – (optional) team abbreviation is not in dict with stats and can be added in here

data
returns boxscore data dict

live Boxscore Item

```
class pbpstats.resources.boxscore.live_boxscore_item.LiveBoxscoreItem(item,
                                                                           team_id=None,
                                                                           team_abbreviation=None)
```

Bases: object

Class for boxscore items from live data endpoint

Parameters

- **item** (*dict*) – dict with boxscore stats from response
- **team_id** (*int*) – (optional) team id is not in dict with stats and can be added in here
- **team_abbreviation** (*str*) – (optional) team abbreviation is not in dict with stats and can be added in here

data
returns boxscore data dict

total_seconds

stats.nba.com Boxscore Item

```
class pbpstats.resources.boxscore.stats_nba_boxscore_item.StatsNbaBoxscoreItem(item)
Bases: object
```

Class for boxscore items from stats.nba.com

Parameters **item** (*dict*) – dict with boxscore stats from response

data
returns boxscore data dict

3.1.3.2 Enhanced PBP

data.nba.com Enhanced PBP

Ejection

```
class pbpstats.resources.enhanced_pbp.data_nba.ejection.DataEjection(*args)
    Bases: pbpstats.resources.enhanced_pbp.ejection.Ejection, pbpstats.resources.enhanced_pbp.data_nba.enhanced_pbp_item.DataEnhancedPbpItem

    Class for Ejection events

    event_type = 11
```

End of Period

```
class pbpstats.resources.enhanced_pbp.data_nba.end_of_period.DataEndOfPeriod(*args)
    Bases: pbpstats.resources.enhanced_pbp.end_of_period.EndOfPeriod, pbpstats.resources.enhanced_pbp.data_nba.enhanced_pbp_item.DataEnhancedPbpItem

    Class for end of period events

    event_type = 13
```

Enhanced PBP Factory

DataNbaEnhancedPbpFactory can be used to create enhanced pbp event objects from the pbpstats.resources.enhanced_pbp.data_nba module based on the event type.

The following code will get the event class for event type 1 (which is the event type for a field goal make)

```
from pbpstats.resources.enhanced_pbp.data_nba.enhanced_pbp_factory import _
↳ DataNbaEnhancedPbpFactory

factory = DataNbaEnhancedPbpFactory()
event_class = factory.get_event_class(1)
print(event_class) # prints "<class 'pbpstats.resources.enhanced_pbp.data_nba.field_
↳ goal.DataFieldGoal'>"
```

```
class pbpstats.resources.enhanced_pbp.data_nba.enhanced_pbp_factory.DataNbaEnhancedPbpFactory
    Bases: object

    Class for factory of event type classes. On initialization will load in all event classes in the pbpstats.resources.enhanced_pbp.data_nba module

    get_event_class(event_type)
        Gets the class for the event based on the event_type

        Parameters event_type (int) – event action type for the event

        Returns class for event type
```

Enhanced PBP Item

DataEnhancedPbpItem is the base class for all data.nba.com enhanced pbp event types

```
class pbpstats.resources.enhanced_pbp.data_nba.enhanced_pbp_item.DataEnhancedPbpItem(item,  
pe-  
riod,  
game_id
```

Bases: `pbpstats.resources.enhanced_pbp.enhanced_pbp_item.EnhancedPbpItem`

Base class for enhanced pbp events from data.nba.com

Parameters

- **item** (*dict*) – dict with event data
- **period** (*int*) – period in which event occurs
- **game_id** (*str*) – NBA Stats Game Id

data

returns event as a dict

event_stats

returns list of dicts with all stats for event

get_offense_team_id()

returns team id for team on offense for event

is_possession_ending_event

returns True if event ends a possession, False otherwise

seconds_remaining

returns seconds remaining in period as a float

Field Goal

```
class pbpstats.resources.enhanced_pbp.data_nba.field_goal.DataFieldGoal(*args)  
Bases:      pbpstats.resources.enhanced_pbp.field_goal.FieldGoal,      pbpstats.  
resources.enhanced_pbp.data_nba.enhanced_pbp_item.DataEnhancedPbpItem
```

Class for field goal events

event_type = [1, 2]

is_made

returns True if shot was made, False otherwise

shot_value

returns 3 if shot is a 3 point attempt, 2 otherwise

Foul

```
class pbpstats.resources.enhanced_pbp.data_nba.foul.DataFoul(*args)  
Bases:      pbpstats.resources.enhanced_pbp.foul.Foul,      pbpstats.resources.  
enhanced_pbp.data_nba.enhanced_pbp_item.DataEnhancedPbpItem
```

Class for foul events

event_type = 6

is_away_from_play_foul

is_charge

is_clear_path_foul

`is_defensive_3_seconds`
`is_delay_of_game`
`is_double_foul`
`is_double_technical`
`is_flagrant1`
`is_flagrant2`
`is_inbound_foul`
`is_loose_ball_foul`
`is_offensive_foul`
`is_personal_block_foul`
`is_personal_foul`
`is_personal_take_foul`
`is_shooting_block_foul`
`is_shooting_foul`
`is_technical`
`is_transition_take_foul`
`number_of_fta_for_foul`
 returns the number of free throws resulting from the foul

Free Throw

```
class pbpstats.resources.enhanced_pbp.data_nba.free_throw.DataFreeThrow(*args)
    Bases: pbpstats.resources.enhanced_pbp.free_throw.FreeThrow, pbpstats.
resources.enhanced_pbp.data_nba.enhanced_pbp_item.DataEnhancedPbpItem

    Class for free throw events

    event_type = 3

    is_flagrant_ft
    is_ft_1_of_1
    is_ft_1_of_2
    is_ft_1_of_3
    is_ft_1pt
        returns True if free throw is a 1 point free throw, False otherwise Only used in g-league, starting in 2019-20
        season
    is_ft_2_of_2
    is_ft_2_of_3
    is_ft_2pt
        returns True if free throw is a 2 point free throw, False otherwise Only used in g-league, starting in 2019-20
        season
    is_ft_3_of_3
```


is_ft_3pt
returns True if free throw is a 3 point free throw, False otherwise Only used in g-league, starting in 2019-20 season

is_made
returns True if shot was made, False otherwise

is_technical_ft

Jump Ball

class `pbpstats.resources.enhanced_pbp.data_nba.jump_ball.DataJumpBall(*args)`
Bases: `pbpstats.resources.enhanced_pbp.jump_ball.JumpBall`, `pbpstats.resources.enhanced_pbp.data_nba.enhanced_pbp_item.DataEnhancedPbpItem`
Class for jump ball events
event_type = 10

Rebound

class `pbpstats.resources.enhanced_pbp.data_nba.rebound.DataRebound(*args)`
Bases: `pbpstats.resources.enhanced_pbp.rebound.Rebound`, `pbpstats.resources.enhanced_pbp.data_nba.enhanced_pbp_item.DataEnhancedPbpItem`
Class for rebound events
event_type = 4
is_placeholder
returns True if rebound is a placeholder event, False otherwise.
These are team rebounds on for example missed FT 1 of 2
missed_shot
returns *FieldGoal* or *FreeThrow* object for shot that was missed
Raises *EventOrderError*: If rebound event is not immediately following a missed shot event.
oreb
returns True if rebound is an offensive rebound, False otherwise

Replay

class `pbpstats.resources.enhanced_pbp.data_nba.replay.DataReplay(*args)`
Bases: `pbpstats.resources.enhanced_pbp.replay.Replay`, `pbpstats.resources.enhanced_pbp.data_nba.enhanced_pbp_item.DataEnhancedPbpItem`
Class for replay events
event_type = 18
overturn_ruling
ruling_stands
support_ruling

Start of Period

```
class pbpstats.resources.enhanced_pbp.data_nba.start_of_period.DataStartOfPeriod(*args)
    Bases: pbpstats.resources.enhanced_pbp.start_of_period.StartOfPeriod,
pbpstats.resources.enhanced_pbp.data_nba.enhanced_pbp_item.
DataEnhancedPbpItem
```

Class for start of period events

event_type = 12

get_period_starters (*file_directory=None*)

Gets player ids of players who started the period for each team If players can't be determined from parsing pbp, will try to find them by making API request to stats.nba.com boxscore filtered by time.

Parameters **file_directory** (*str*) – directory in which overrides subdirectory exists containing period starter overrides when period starters can't be determined from parsing pbp events

Returns dict with list of player ids for each team with players on the floor at start of period

Raises *InvalidNumberOfStartersException*: If all 5 players that start the period for a team can't be determined.

Substitution

```
class pbpstats.resources.enhanced_pbp.data_nba.substitution.DataSubstitution(*args)
    Bases: pbpstats.resources.enhanced_pbp.substitution.Substitution, pbpstats.
resources.enhanced_pbp.data_nba.enhanced_pbp_item.DataEnhancedPbpItem
```

Class for Substitution events

event_type = 8

incoming_player_id

returns player id of player coming in to the game

outgoing_player_id

returns player id of player going out of the game

Timeout

```
class pbpstats.resources.enhanced_pbp.data_nba.timeout.DataTimeout(*args)
    Bases: pbpstats.resources.enhanced_pbp.timeout.Timeout, pbpstats.resources.
enhanced_pbp.data_nba.enhanced_pbp_item.DataEnhancedPbpItem
```

Class for timeout events

event_type = 9

Turnover

```
class pbpstats.resources.enhanced_pbp.data_nba.turnover.DataTurnover(*args)
    Bases: pbpstats.resources.enhanced_pbp.turnover.Turnover, pbpstats.resources.
enhanced_pbp.data_nba.enhanced_pbp_item.DataEnhancedPbpItem
```

Class for Turnover events

```
event_type = 5
is_3_second_violation
is_bad_pass
is_bad_pass_out_of_bounds
is_kicked_ball
is_lane_violation
is_lost_ball
is_lost_ball_out_of_bounds
is_no_turnover
is_offensive_goaltending
is_shot_clock_violation
is_step_out_of_bounds
is_travel
```

Violation

```
class pbpstats.resources.enhanced_pbp.data_nba.violation.DataViolation(*args)
    Bases: pbpstats.resources.enhanced_pbp.violation.Violation, pbpstats.
            resources.enhanced_pbp.data_nba.enhanced_pbp_item.DataEnhancedPbpItem
    Class for violation events
    event_type = 7
    is_delay_of_game
    is_double_lane_violation
    is_goaltend_violation
    is_jumpball_violation
    is_kicked_ball_violation
    is_lane_violation
```

data.nba.com Enhanced PBP

Ejection

```
class pbpstats.resources.enhanced_pbp.live.ejection.LiveEjection(*args)
    Bases: pbpstats.resources.enhanced_pbp.ejection.Ejection, pbpstats.resources.
            enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem
    Class for Ejection events
    action_type = 'ejection'
```

End of Period

```
class pbpstats.resources.enhanced_pbp.live.end_of_period.LiveEndOfPeriod(*args)
    Bases: pbpstats.resources.enhanced_pbp.end_of_period.EndOfPeriod, pbpstats.
            resources.enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem
```

Class for end of period events

action_type = 'period'

sub_type = 'end'

Enhanced PBP Factory

LiveEnhancedPbpFactory can be used to create enhanced pbp event objects from the pbpstats.resources.enhanced_pbp.live module based on the event type.

The following code will get the event class for a turnover event

```
from pbpstats.resources.enhanced_pbp.live.enhanced_pbp_factory import _
↳ LiveEnhancedPbpFactory

factory = LiveEnhancedPbpFactory()
event_class = factory.get_event_class('turnover', 'out-of-bounds')
print(event_class)  # prints "<class 'pbpstats.resources.enhanced_pbp.live.turnover.
↳ LiveTurnover'>"
```

```
class pbpstats.resources.enhanced_pbp.live.enhanced_pbp_factory.LiveEnhancedPbpFactory
    Bases: object
```

Class for factory of event type classes. On initialization will load in all event classes in the pbpstats.resources.enhanced_pbp.live module

get_event_class (*action_type*, *sub_type*)

Gets the class for the event based on the event_type

Parameters **action_type** (*str*) – event action type for the event

Returns class for event type

Enhanced PBP Item

LiveEnhancedPbpItem is the base class for all live data enhanced pbp event types

```
class pbpstats.resources.enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem(item,
                                                                                   game_id)
```

Bases: *pbpstats.resources.enhanced_pbp.enhanced_pbp_item.EnhancedPbpItem*

Base class for enhanced pbp events from live data

Parameters

- **item** (*dict*) – dict with event data
- **period** (*int*) – period in which event occurs
- **game_id** (*str*) – NBA Stats Game Id

data

returns event as a dict

event_stats
returns list of dicts with all stats for event

get_offense_team_id()
returns team id for team on offense for event

is_possession_ending_event
returns True if event ends a possession, False otherwise

seconds_remaining
returns seconds remaining in period as a float

stripped_descriptor

stripped_sub_type

Field Goal

```
class pbpstats.resources.enhanced_pbp.live.field_goal.LiveFieldGoal(*args)
    Bases: pbpstats.resources.enhanced_pbp.field_goal.FieldGoal, pbpstats.
            resources.enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem

    Class for field goal events

    action_type = ['2pt', '3pt']

    is_made
        returns True if shot was made, False otherwise

    shot_value
        returns 3 if shot is a 3 point attempt, 2 otherwise
```

Foul

```
class pbpstats.resources.enhanced_pbp.live.foul.LiveFoul(*args)
    Bases: pbpstats.resources.enhanced_pbp.foul.Foul, pbpstats.resources.
            enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem

    Class for foul events

    action_type = 'foul'

    is_away_from_play_foul

    is_charge

    is_clear_path_foul

    is_defensive_3_seconds

    is_delay_of_game

    is_double_foul

    is_double_technical

    is_flagrant1

    is_flagrant2

    is_inbound_foul

    is_loose_ball_foul
```

```
is_offensive_foul
is_personal_block_foul
is_personal_foul
is_personal_take_foul
is_shooting_block_foul
is_shooting_foul
is_technical
is_transition_take_foul
number_of_fta_for_foul
    returns the number of free throws resulting from the foul
```

Free Throw

```
class pbpstats.resources.enhanced_pbp.live.free_throw.LiveFreeThrow(*args)
    Bases: pbpstats.resources.enhanced_pbp.free_throw.FreeThrow, pbpstats.
resources.enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem
    Class for free throw events
    action_type = 'freethrow'
    is_away_from_play_ft
        returns True if free throw is from an away from the play foul, False otherwise.
    is_flagrant_ft
        returns True if free throw is from a flagrant foul, False otherwise.
    is_ft_1_of_1
    is_ft_1_of_2
    is_ft_1_of_3
    is_ft_1pt
        returns True if free throw is a 1 point free throw, False otherwise Only used in g-league, starting in 2019-20
        season
    is_ft_2_of_2
    is_ft_2_of_3
    is_ft_2pt
        returns True if free throw is a 2 point free throw, False otherwise Only used in g-league, starting in 2019-20
        season
    is_ft_3_of_3
    is_ft_3pt
        returns True if free throw is a 3 point free throw, False otherwise Only used in g-league, starting in 2019-20
        season
    is_made
        returns True if shot was made, False otherwise
    is_technical_ft
```

Jump Ball

```
class pbpstats.resources.enhanced_pbp.live.jump_ball.LiveJumpBall(*args)
    Bases: pbpstats.resources.enhanced_pbp.jump_ball.JumpBall, pbpstats.
            resources.enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem

    Class for jump ball events

    action_type = 'jumpball'
```

Rebound

```
class pbpstats.resources.enhanced_pbp.live.rebound.LiveRebound(*args)
    Bases: pbpstats.resources.enhanced_pbp.rebound.Rebound, pbpstats.resources.
            enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem

    Class for rebound events

    action_type = 'rebound'

    is_placeholder
        returns True if rebound is a placeholder event, False otherwise.

        These are team rebounds on for example missed FT 1 of 2

    missed_shot
        returns FieldGoal or FreeThrow object for shot that was missed

        Raises EventOrderError: If rebound event is not immediately following a missed shot
        event.

    oreb
        returns True if rebound is an offensive rebound, False otherwise
```

Replay

```
class pbpstats.resources.enhanced_pbp.live.replay.LiveReplay(*args)
    Bases: pbpstats.resources.enhanced_pbp.replay.Replay, pbpstats.resources.
            enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem

    Class for replay events

    action_type = 'replay'

    overturn_ruling

    ruling_stands

    support_ruling
```

Start of Period

```
class pbpstats.resources.enhanced_pbp.live.start_of_period.LiveStartOfPeriod(*args)
    Bases: pbpstats.resources.enhanced_pbp.start_of_period.StartOfPeriod,
            pbpstats.resources.enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem

    Class for start of period events

    action_type = 'period'
```

get_period_starters (*file_directory=None, ignore_missing_starters=False*)

Gets player ids of players who started the period for each team

Parameters

- **file_directory** (*str*) – directory in which overrides subdirectory exists containing period starter overrides when period starters can't be determined from parsing pbp events
- **ignore_missing_starters** (*bool*) – when True won't raise missing starters exception

Returns dict with list of player ids for each team with players on the floor at start of period

Raises *InvalidNumberOfStartersException*: If all 5 players that start the period for a team can't be determined.

sub_type = 'start'

Substitution

class pbpstats.resources.enhanced_pbp.live.substitution.**LiveSubstitution** (*args)
Bases: *pbpstats.resources.enhanced_pbp.substitution.Substitution, pbpstats.resources.enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem*

Class for Substitution events

action_type = 'substitution'

current_players

returns dict with list of player ids for each team with players on the floor following the sub

incoming_player_id

returns player id of player coming in to the game

outgoing_player_id

returns player id of player coming in to the game

Timeout

class pbpstats.resources.enhanced_pbp.live.timeout.**LiveTimeout** (*args)
Bases: *pbpstats.resources.enhanced_pbp.timeout.Timeout, pbpstats.resources.enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem*

Class for timeout events

action_type = 'timeout'

Turnover

class pbpstats.resources.enhanced_pbp.live.turnover.**LiveTurnover** (*args)
Bases: *pbpstats.resources.enhanced_pbp.turnover.Turnover, pbpstats.resources.enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem*

Class for Turnover events

action_type = 'turnover'

is_3_second_violation


```
is_bad_pass
is_bad_pass_out_of_bounds
is_kicked_ball
is_lane_violation
is_lost_ball
is_lost_ball_out_of_bounds
is_no_turnover
is_offensive_goaltending
is_shot_clock_violation
is_step_out_of_bounds
is_travel
```

Violation

```
class pbpstats.resources.enhanced_pbp.live.violation.LiveViolation(*args)
    Bases: pbpstats.resources.enhanced_pbp.violation.Violation, pbpstats.
            resources.enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem
    Class for violation events
    action_type = 'violation'
    is_delay_of_game
    is_double_lane_violation
    is_goaltend_violation
    is_jumpball_violation
    is_kicked_ball_violation
    is_lane_violation
```

stats.nba.com Enhanced PBP

Ejection

```
class pbpstats.resources.enhanced_pbp.stats_nba.ejection.StatsEjection(*args)
    Bases: pbpstats.resources.enhanced_pbp.ejection.Ejection, pbpstats.resources.
            enhanced_pbp.stats_nba.enhanced_pbp_item.StatsEnhancedPbpItem
    Class for Ejection events
    event_type = 11
```

End of Period

```
class pbpstats.resources.enhanced_pbp.stats_nba.end_of_period.StatsEndOfPeriod(*args)
    Bases: pbpstats.resources.enhanced_pbp.end_of_period.EndOfPeriod, pbpstats.
            resources.enhanced_pbp.stats_nba.enhanced_pbp_item.StatsEnhancedPbpItem

    Class for end of period events

    event_type = 13
```

Enhanced PBP Factory

StatsNbaEnhancedPbpFactory can be used to create enhanced pbp event objects from the pbpstats.resources.enhanced_pbp.stats_nba module based on the event type.

The following code will get the event class for event type 1 (which is the event type for a field goal make)

```
from pbpstats.resources.enhanced_pbp.stats_nba.enhanced_pbp_factory import _
↳ StatsNbaEnhancedPbpFactory

factory = StatsNbaEnhancedPbpFactory()
event_class = factory.get_event_class(1)
print(event_class)  # prints "<class 'pbpstats.resources.enhanced_pbp.stats_nba.field_
↳ goal.StatsFieldGoal'>"
```

```
class pbpstats.resources.enhanced_pbp.stats_nba.enhanced_pbp_factory.StatsNbaEnhancedPbpFactory
    Bases: object

    Class for factory of event type classes. On initialization will load in all event classes in the pbpstats.
    resources.enhanced_pbp.stats_nba module

    get_event_class(event_type)
        Gets the class for the event based on the event_type

        Parameters event_type (int) – event action type for the event

        Returns class for event type
```

Enhanced PBP Item

StatsEnhancedPbpItem is the base class for all stats.nba.com enhanced pbp event types

```
class pbpstats.resources.enhanced_pbp.stats_nba.enhanced_pbp_item.StatsEnhancedPbpItem(event_data, order)
    Bases: pbpstats.resources.enhanced_pbp.enhanced_pbp_item.EnhancedPbpItem

    Base class for enhanced pbp events from stats.nba.com

    Parameters
        • event (dict) – dict with event data
        • order (int) – sequential order in which event occurs

    data
        returns event as a dict

    get_offense_team_id()
        returns team id for team on offense for event
```

is_possession_ending_event
returns True if event ends a possession, False otherwise

seconds_remaining
returns seconds remaining in period as a float

video_url
returns url for mp4 video of play, if available

Field Goal

```
class pbpstats.resources.enhanced_pbp.stats_nba.field_goal.StatsFieldGoal(*args)
    Bases: pbpstats.resources.enhanced_pbp.field_goal.FieldGoal, pbpstats.
resources.enhanced_pbp.stats_nba.enhanced_pbp_item.StatsEnhancedPbpItem

    Class for field goal events

    event_type = [1, 2]

    get_offense_team_id()
        returns team id that took the shot

    is_made
        returns True if shot was made, False otherwise

    shot_value
        returns 3 if shot is a 3 point attempt, 2 otherwise
```

Foul

```
class pbpstats.resources.enhanced_pbp.stats_nba.foul.StatsFoul(*args)
    Bases: pbpstats.resources.enhanced_pbp.foul.Foul, pbpstats.resources.
enhanced_pbp.stats_nba.enhanced_pbp_item.StatsEnhancedPbpItem

    Class for foul events

    event_type = 6

    is_away_from_play_foul

    is_charge

    is_clear_path_foul

    is_defensive_3_seconds

    is_delay_of_game

    is_double_foul

    is_double_technical

    is_flagrant1

    is_flagrant2

    is_inbound_foul

    is_loose_ball_foul

    is_offensive_foul

    is_personal_block_foul
```

`is_personal_foul`
`is_personal_take_foul`
`is_shooting_block_foul`
`is_shooting_foul`
`is_technical`
`is_transition_take_foul`
`number_of_fta_for_foul`
returns the number of free throws resulting from the foul

Free Throw

```
class pbpstats.resources.enhanced_pbp.stats_nba.free_throw.StatsFreeThrow(*args)
    Bases: pbpstats.resources.enhanced_pbp.free_throw.FreeThrow, pbpstats.
resources.enhanced_pbp.stats_nba.enhanced_pbp_item.StatsEnhancedPbpItem
    Class for free throw events

    event_type = 3

    get_offense_team_id()
        returns team id that took the shot

    is_flagrant_ft
    is_ft_1_of_1
    is_ft_1_of_2
    is_ft_1_of_3
    is_ft_1pt
        returns True if free throw is a 1 point free throw, False otherwise Only used in g-league, starting in 2019-20
        season
    is_ft_2_of_2
    is_ft_2_of_3
    is_ft_2pt
        returns True if free throw is a 2 point free throw, False otherwise Only used in g-league, starting in 2019-20
        season
    is_ft_3_of_3
    is_ft_3pt
        returns True if free throw is a 3 point free throw, False otherwise Only used in g-league, starting in 2019-20
        season
    is_made
        returns True if shot was made, False otherwise
    is_technical_ft
```

Jump Ball

```
class pbpstats.resources.enhanced_pbp.stats_nba.jump_ball.StatsJumpBall(*args)
    Bases: pbpstats.resources.enhanced_pbp.jump_ball.JumpBall, pbpstats.resources.enhanced_pbp.stats_nba.enhanced_pbp_item.StatsEnhancedPbpItem

    Class for jump ball events

    event_type = 10

    get_offense_team_id()
        returns team id for team on offense for event
```

Rebound

```
class pbpstats.resources.enhanced_pbp.stats_nba.rebound.StatsRebound(*args)
    Bases: pbpstats.resources.enhanced_pbp.rebound.Rebound, pbpstats.resources.enhanced_pbp.stats_nba.enhanced_pbp_item.StatsEnhancedPbpItem

    Class for rebound events

    event_type = 4

    get_offense_team_id()
        returns team id for team on offense for the shot that was rebounded

    is_placeholder
        returns True if rebound is a placeholder event, False otherwise.

        These are team rebounds on for example missed FT 1 of 2

    missed_shot
        returns FieldGoal or FreeThrow object for shot that was missed

        Raises EventOrderError: If rebound event is not immediately following a missed shot event.

    oreb
        returns True if rebound is an offensive rebound, False otherwise
```

Replay

```
class pbpstats.resources.enhanced_pbp.stats_nba.replay.StatsReplay(*args)
    Bases: pbpstats.resources.enhanced_pbp.replay.Replay, pbpstats.resources.enhanced_pbp.stats_nba.enhanced_pbp_item.StatsEnhancedPbpItem

    Class for replay events

    event_type = 18

    overturn_ruling

    ruling_stands

    support_ruling
```

Start of Period

```
class pbpstats.resources.enhanced_pbp.stats_nba.start_of_period.StatsStartOfPeriod(*args)
    Bases: pbpstats.resources.enhanced_pbp.start_of_period.StartOfPeriod,
pbpstats.resources.enhanced_pbp.stats_nba.enhanced_pbp_item.
StatsEnhancedPbpItem
```

Class for start of period events

event_type = 12

get_period_starters (*file_directory=None*)

Gets player ids of players who started the period for each team. If players can't be determined from parsing pbp, will try to find them by making API request to stats.nba.com boxscore filtered by time.

Parameters **file_directory** (*str*) – directory in which overrides subdirectory exists containing period starter overrides when period starters can't be determined from parsing pbp events

Returns dict with list of player ids for each team with players on the floor at start of period

Raises *InvalidNumberOfStartersException*: If all 5 players that start the period for a team can't be determined.

Substitution

```
class pbpstats.resources.enhanced_pbp.stats_nba.substitution.StatsSubstitution(*args)
    Bases: pbpstats.resources.enhanced_pbp.substitution.Substitution, pbpstats.
resources.enhanced_pbp.stats_nba.enhanced_pbp_item.StatsEnhancedPbpItem
```

Class for Substitution events

event_type = 8

incoming_player_id

returns player id of player coming in to the game

outgoing_player_id

returns player id of player going out of the game

Timeout

```
class pbpstats.resources.enhanced_pbp.stats_nba.timeout.StatsTimeout(*args)
    Bases: pbpstats.resources.enhanced_pbp.timeout.Timeout, pbpstats.resources.
enhanced_pbp.stats_nba.enhanced_pbp_item.StatsEnhancedPbpItem
```

Class for timeout events

event_type = 9

Turnover

```
class pbpstats.resources.enhanced_pbp.stats_nba.turnover.StatsTurnover(*args)
    Bases: pbpstats.resources.enhanced_pbp.turnover.Turnover, pbpstats.resources.
enhanced_pbp.stats_nba.enhanced_pbp_item.StatsEnhancedPbpItem
```

Class for Turnover events

```
event_type = 5
get_offense_team_id()
    returns team id for team on offense for event
is_3_second_violation
is_bad_pass
is_bad_pass_out_of_bounds
is_kicked_ball
is_lane_violation
is_lost_ball
is_lost_ball_out_of_bounds
is_no_turnover
is_offensive_goaltending
is_shot_clock_violation
is_step_out_of_bounds
is_travel
```

Violation

```
class pbpstats.resources.enhanced_pbp.stats_nba.violation.StatsViolation(*args)
    Bases: pbpstats.resources.enhanced\_pbp.violation.Violation, pbpstats.resources.enhanced\_pbp.stats\_nba.enhanced\_pbp\_item.StatsEnhancedPbpItem
    Class for violation events
    event_type = 7
    is_delay_of_game
    is_double_lane_violation
    is_goaltend_violation
    is_jumpball_violation
    is_kicked_ball_violation
    is_lane_violation
```

Ejection

```
class pbpstats.resources.enhanced_pbp.ejection.Ejection
    Bases: object
    Class for Ejection events
    event_stats
        returns list of dicts with all stats for event
```

End of Period

```
class pbpstats.resources.enhanced_pbp.end_of_period.EndOfPeriod
    Bases: object

    Class for end of period events

    event_stats
        returns list of dicts with all stats for event
```

Enhanced PBP

The EnhancedPbp class has some basic properties for handling enhanced pbp data

```
class pbpstats.resources.enhanced_pbp.enhanced_pbp.EnhancedPbp(items)
    Bases: pbpstats.resources.base.Base

    Class for enhanced play-by-play events

        Parameters events (list) – list of EnhancedPbpItem items, typically from a enhanced pbp
            data loader

    data
        returns list of pbp event dicts

    fgas
        returns list of FieldGoal events

    fgms
        returns list of FieldGoal events with all made FGs

    ftas
        returns list of FreeThrow events

    rebounds
        returns list of Rebound events

    turnovers
        returns list of Turnover events
```

Enhanced PBP Item

EnhancedPbpItem is an abstract base class for all enhanced pbp event types

```
class pbpstats.resources.enhanced_pbp.enhanced_pbp_item.EnhancedPbpItem
    Bases: object

    base_stats
        returns list of dicts with all seconds played and possession count stats for event

    count_as_possession
        returns True if event is possession changing event that should count as a real possession, False otherwise.

        In order to not include possessions which a very low probability of scoring in possession counts, possession
        won't be counted as a possession if it starts with <= 2 seconds left and no points are scored before period
        ends

    current_players
        returns dict with list of player ids for each team with players on the floor for current event
```


For all non substitution events current players are just the same as previous event

This gets overwritten in *Substitution* since those are the only event types where players are not the same as the previous event

event_stats

returns list of dicts with all stats for event

get_all_events_at_current_time()

returns list of all events that take place at the same time as the current event

get_offense_team_id()

returns team id for team on offense for event

is_penalty_event()

returns True if the team on offense is in the penalty, False otherwise

is_possession_ending_event

returns True if event ends a possession, False otherwise

is_second_chance_event()

returns True if the event takes place after an offensive rebound on the current possession, False otherwise

lineup_ids

returns dict with lineup ids for each team for current event. Lineup ids are hyphen separated sorted player id strings.

score_margin

returns the score margin from perspective of offense team before the event took place

seconds_remaining

returns seconds remaining in period as a float

seconds_since_previous_event

returns the number of seconds that have elapsed since the previous event

Field Goal

```
class pbpstats.resources.enhanced_pbp.field_goal.FieldGoal
```

Bases: object

Class for field goal events

distance

returns shot distance in feet

event_stats

returns list of dicts with all stats for event

is_and1

returns True if shot was an and 1, False otherwise

is_assisted

returns True if shot was assisted, False otherwise

is_blocked

returns True if shot was blocked, False otherwise

is_corner_3

returns True if shot was a corner 3, False otherwise

is_heave
returns True if shot was a last second heave, False otherwise

is_made
returns True if shot was made, False otherwise

is_make_that_does_not_end_possession
returns True if shot is a made shot that does not end the possession due to a foul, False otherwise

is_putback
returns True if shot is a 2pt attempt within 2 seconds of an offensive rebound attempted by the same player who got the rebound

rebound
returns *Rebound* item for the rebound of the shot, if it was missed, None otherwise

shot_data
returns a dict with detailed shot data

shot_type
returns shot type string ('AtRim', 'ShortMidRange', 'LongMidRange', 'Arc3' or 'Corner3')

shot_value
returns 3 if shot is a 3 point attempt, 2 otherwise

Foul

```
class pbpstats.resources.enhanced_pbp.foul.Foul
    Bases: object

    Class for foul events

    counts_as_personal_foul
        returns True if fouls is a foul type that counts as a personal foul, False otherwise

    counts_towards_penalty
        returns True if foul is a foul type that counts towards the penalty, False otherwise

    event_stats
        returns list of dicts with all stats for event

    foul_type_string
        returns string description of foul type

    is_away_from_play_foul

    is_charge

    is_clear_path_foul

    is_defensive_3_seconds

    is_delay_of_game

    is_double_foul

    is_double_technical

    is_flagrant

    is_flagrant1

    is_flagrant2
```

`is_inbound_foul`
`is_loose_ball_foul`
`is_offensive_foul`
`is_personal_block_foul`
`is_personal_foul`
`is_personal_take_foul`
`is_shooting_block_foul`
`is_shooting_foul`
`is_technical`
`is_transition_take_foul`
`number_of_fta_for_foul`
returns the number of free throws resulting from the foul

Free Throw

class `pbpstats.resources.enhanced_pbp.free_throw.FreeThrow`

Bases: `object`

Class for free throw events

event_for_efficiency_stats

returns *Foul* object for the foul that resulted in the free throw. Plus/minus points should go to the players on the floor at the time of the foul, not the free throw.

event_stats

returns list of dicts with all stats for event

foul_that_led_to_ft

returns *Foul* object for the foul that resulted in the free throw

free_throw_type

returns string description of free throw type

is_away_from_play_ft

returns True if free throw is from an away from the play foul, False otherwise.

is_end_ft

returns True if free throw is last of trip to the free throw line, False otherwise

is_first_ft

returns True if free throw is first of trip to the free throw line, False otherwise

is_ft_1_of_1

is_ft_1_of_2

is_ft_1_of_3

is_ft_1pt

is_ft_2_of_2

is_ft_2_of_3

is_ft_2pt

is_ft_3_of_3

is_ft_3pt

is_inbound_foul_ft

returns True if free throw is from an inbound foul, False otherwise.

is_made

returns True if shot was made, False otherwise

is_technical_ft

is_transition_take_foul_ft

returns True if free throw is from an transition take foul, False otherwise.

num_ft_for_trip

returns number of shots for the trip to the free throw line

shot_type = 'FT'

shot_value

returns shot value of a free throw Starting in 2019-20 season, the G-League added 2 and 3 point FTs

Jump Ball

class pbpstats.resources.enhanced_pbp.jump_ball.**JumpBall**

Bases: object

Class for jump ball events

event_stats

returns list of dicts with all stats for event

winning_team

returns team id that won the jump ball

Rebound

exception pbpstats.resources.enhanced_pbp.rebound.**EventOrderError**

Bases: Exception

Class for exception raised when rebound event is not immediately following a missed shot event.

You can manually edit the event order in the pbp file stored on disk to fix this.

class pbpstats.resources.enhanced_pbp.rebound.**Rebound**

Bases: object

Class for rebound events

event_stats

returns list of dicts with all stats for event

is_buzzer_beater_placeholder

returns True if rebound is a placeholder event after a missed buzzer beater, False otherwise.

Rebounds occur after time has expired but are still logged in play-by-play, but should not be counted in rebound totals

is_buzzer_beater_rebound_at_shot_time

returns True if rebound is a placeholder event after a missed buzzer beater, False otherwise.

Sometimes rebound on buzzer beater is given the same time as shot - don't count these. Only don't count if rebound is last event before end of period event, ignoring replay events

is_non_live_ft_placeholder

returns True if rebound is a placeholder event after a missed free throw that is not a live ball, False otherwise.

Example: rebound after missed flagrant FT 2 of 2

is_placeholder

returns True if rebound is a placeholder event, False otherwise.

These are team rebounds on for example missed FT 1 of 2

is_real_rebound

Returns True if rebound should be counted as a rebound, False otherwise.

All missed shots have a rebound in the play-by-play but not all of these rebounds should be counted as actual rebounds. Some are just placeholder events.

is_turnover_placeholder

returns True if rebound is a placeholder event when a turnover occurs, False otherwise.

Example shot clock violation or kicked ball turnover at time of team rebound

missed_shot**oreb****self_reb**

returns True if rebound was gotten by player who missed the shot, False otherwise

Replay

class pbpstats.resources.enhanced_pbp.replay.Replay

Bases: object

Class for replay events

event_stats

returns list of dicts with all stats for event

overturn_ruling**ruling_stands****support_ruling**

Start of Period

exception pbpstats.resources.enhanced_pbp.start_of_period.InvalidNumberOfStartersException

Bases: Exception

Class for exception when a team's 5 period starters can't be determined.

You can add the correct period starters to overrides/missing_period_starters.json in your data directory to fix this.

```
class pbpstats.resources.enhanced_pbp.start_of_period.StartOfPeriod
    Bases: object

    Class for start of period events

    current_players
        returns period starters

    event_stats
        returns list of dicts with all stats for event

    get_offense_team_id()
        returns team id for team on starting period on offense

    classmethod get_period_starters (file_directory)
        Gets player ids of players who started the period for each team

        Parameters file_directory (str) – directory in which overrides subdirectory exists containing period starter overrides when period starters can't be determined from parsing pbp events

        Returns dict with list of player ids for each team with players on the floor at start of period

        Raises InvalidNumberOfStartersException: If all 5 players that start the period for a team can't be determined.

    get_team_starting_with_ball()
        returns team id for team on starting period with the ball

    league
        Returns League for game id.

        First 2 in game id represent league - 00 for nba, 10 for wnba, 20 for g-league

    league_url_part
```

Substitution

```
class pbpstats.resources.enhanced_pbp.substitution.Substitution
    Bases: object

    Class for Substitution events

    current_players
        returns dict with list of player ids for each team with players on the floor following the sub

    event_stats
        returns list of dicts with all stats for event

    incoming_player_id

    outgoing_player_id
```

Timeout

```
class pbpstats.resources.enhanced_pbp.timeout.Timeout
    Bases: object

    Class for timeout events

    event_stats
        returns list of dicts with all stats for event
```

Turnover

```
class pbpstats.resources.enhanced_pbp.turnover.Turnover
    Bases: object

    Class for Turnover events

    event_stats
        returns list of dicts with all stats for event

    classmethod is_3_second_violation()

    classmethod is_bad_pass()

    classmethod is_bad_pass_out_of_bounds()

    classmethod is_kicked_ball()

    classmethod is_lane_violation()

    classmethod is_lost_ball()

    classmethod is_lost_ball_out_of_bounds()

    classmethod is_no_turnover()

    classmethod is_offensive_goaltending()

    classmethod is_shot_clock_violation()

    is_steal

    classmethod is_step_out_of_bounds()

    classmethod is_travel()
```

Violation

```
class pbpstats.resources.enhanced_pbp.violation.Violation
    Bases: object

    Class for violation events

    event_stats
        returns list of dicts with all stats for event

    classmethod is_delay_of_game()

    classmethod is_double_lane_violation()

    classmethod is_goaltend_violation()

    classmethod is_jumpball_violation()

    classmethod is_kicked_ball_violation()

    classmethod is_lane_violation()
```

3.1.3.3 Games

data.nba.com Game Item

class pbpstats.resources.games.data_nba_game_item.**DataNbaGameItem**(*item*)

Bases: object

Class for game data from data.nba.com

Parameters *item* (*dict*) – dict with game data

data

returns game dict

is_final

returns True if game is final, False otherwise

Games

The Games class has some basic properties for handling game data

class pbpstats.resources.games.games.**Games**(*items*)

Bases: pbpstats.resources.base.Base

Class for games items

Parameters *items* (*list*) – list of either *StatsNbaGameItem* or *DataNbaGameItem* items, typically from a games data loader

data

returns list of dicts with game items

final_games

returns list of dicts with final game items

stats.nba.com Game Item

class pbpstats.resources.games.stats_nba_game_item.**StatsNbaGameItem**(*item*)

Bases: object

Class for game data from stats.nba.com

Parameters *item* (*dict*) – dict with game data

data

returns game dict

is_final

returns True if game is final, False otherwise

3.1.3.4 PBP

data.nba.com PBP Item

class pbpstats.resources.pbp.data_nba_pbp_item.**DataNbaPbpItem**(*item*, *period*)

Bases: object

Class for pbp events from data.nba.com

Parameters

- **item** (*dict*) – dict with event data
- **period** (*int*) – period in which event occurs

data
returns pbp event dict

live PBP Item

class pbpstats.resources.pbp.live_pbp_item.**LivePbpItem** (*item*)

Bases: object

Class for pbp events from live data

Parameters

- **item** (*dict*) – dict with event data
- **period** (*int*) – period in which event occurs

data
returns pbp event dict

PBP

The Pbp class has some basic properties for handling pbp data

class pbpstats.resources.pbp.pbp.**Pbp** (*items*)

Bases: pbpstats.resources.base.Base

Class for pbp items

Parameters **items** (*list*) – list of either *StatsNbaPbpItem* or *DataNbaPbpItem* items, typically from a pbp data loader

data
returns list of dicts with each event

stats.nba.com PBP Item

class pbpstats.resources.pbp.stats_nba_pbp_item.**StatsNbaPbpItem** (*event*, *order*)

Bases: object

Class for pbp events from stats.nba.com

Parameters

- **event** (*dict*) – dict with event data
- **order** (*int*) – sequential order in which event occurs

data
returns pbp event dict

3.1.3.5 Possessions

Possession

The `Possession` class has some basic properties for handling possession data

class `pbpstats.resources.possessions.possession.Possession(events)`

Bases: `object`

Class for possession

Parameters `events` (*list*) – list of *EnhancedPbpItem* items for possession, typically from a possession data loader

data

returns dict possession data

end_time

returns the time remaining (MM:SS) in the period when the possession ended

get_team_ids()

returns a list with the team ids of both teams playing

offense_team_id

returns team id for team on offense on possession

possession_has_timeout

returns True if there was a timeout called on the current possession, False otherwise

possession_start_type

returns possession start type string

possession_stats

returns list of dicts with aggregate stats for possession

previous_possession_end_rebound_player_id

returns player id of player who got rebound that ended previous possession. returns 0 if previous possession did not end with a live ball rebound

previous_possession_end_shooter_player_id

returns player id of player who took shot (make or miss) that ended previous possession. returns 0 if previous possession did not end with made field goal or live ball rebound

previous_possession_end_steal_player_id

returns player id of player who got steal that ended previous possession. returns 0 if previous possession did not end with a live ball turnover

previous_possession_end_turnover_player_id

returns player id of player who turned ball over that ended previous possession. returns 0 if previous possession did not end with a live ball turnover

previous_possession_ending_event

returns previous possession ending event - ignoring subs

previous_possession_has_timeout

returns True if there was a timeout called at same time as possession ended, False otherwise

start_score_margin

returns the score margin from the perspective of the team on offense when the possession started

start_time

returns the time remaining (MM:SS) in the period when the possession started

Possessions

The `Possessions` class has some basic properties for aggregating possession stats

```
class pbpstats.resources.possessions.possessions.Possessions (items)
```

```
    Bases: pbpstats.resources.base.Base
```

```
    Class for possession items
```

```
        Parameters items (list) – list of Possession items, typically from a possession data loader
```

```
    data
```

```
        returns possessions dict
```

```
    lineup_opponent_stats
```

```
        returns list of dicts with aggregated stats by lineup opponent
```

```
    lineup_stats
```

```
        returns list of dicts with aggregated stats by lineup
```

```
    opponent_stats
```

```
        returns list of dicts with aggregated stats by opponent
```

```
    player_stats
```

```
        returns list of dicts with aggregated stats by player
```

```
    team_stats
```

```
        returns list of dicts with aggregated stats by team
```

3.1.3.6 Shots

Shots

The `Shots` class has some basic properties for handling shot data

```
class pbpstats.resources.shots.shots.Shots (items)
```

```
    Bases: pbpstats.resources.base.Base
```

```
    Class for games items
```

```
        Parameters items (list) – list of StatsNbaShot items, typically from a shots data loader
```

```
    data
```

```
        returns list of dicts with shots
```

stats.nba.com Shot

```
class pbpstats.resources.shots.stats_nba_shot.StatsNbaShot (item)
```

```
    Bases: object
```

```
    Class for shot data from stats.nba.com
```

```
        Parameters item (dict) – dict with shot data
```

```
    data
```

```
        returns shot data dict
```

3.1.4 Client

Instantiating a `Client` object will load data loader objects for resources specified in settings dict.

The following code will instantiate the client and get Possession data for game id 0021900001 from files in `/response_data` subdirectories

```
from pbpstats.client import Client

settings = {
    "dir": "/response_data",
    "Possessions": {"source": "file", "data_provider": "stats_nba"}
}
client = Client(settings)
game = client.Game('0021900001')
for possession in game.possessions.items:
    print(possession)
```

```
class pbpstats.client.Client(settings)
```

Bases: `object`

Parameters `settings` (*dict*) – Dict with data that specifies which data loaders should be used. `dir` key is optional, but recommended and should point to the directory you have set up that either already contains response data or where you want to store the response data. Other keys in the settings dict should be resources from the `resources` module and their values should be a dict with `source` ('file' or 'web') and `data_provider` ('stats_nba' or 'data_nba' or 'live')

Python Module Index

p

pbpstats.client, 64

pbpstats.data_loader.data_nba.boxscore.file, 10

pbpstats.data_loader.data_nba.boxscore.loader, 9

pbpstats.data_loader.data_nba.boxscore.web, 10

pbpstats.data_loader.data_nba.enhanced_pbp.file, 11

pbpstats.data_loader.data_nba.enhanced_pbp.loader, 10

pbpstats.data_loader.data_nba.enhanced_pbp.web, 11

pbpstats.data_loader.data_nba.pbp.file, 12

pbpstats.data_loader.data_nba.pbp.loader, 11

pbpstats.data_loader.data_nba.pbp.web, 12

pbpstats.data_loader.data_nba.possessions.file, 13

pbpstats.data_loader.data_nba.possessions.loader, 12

pbpstats.data_loader.data_nba.possessions.web, 13

pbpstats.data_loader.data_nba.schedule.file, 14

pbpstats.data_loader.data_nba.schedule.loader, 13

pbpstats.data_loader.data_nba.schedule.web, 14

pbpstats.data_loader.data_nba.web_loader, 15

pbpstats.data_loader.factory, 29

pbpstats.data_loader.live.boxscore.file, 16

pbpstats.data_loader.live.boxscore.loader, 15

pbpstats.data_loader.live.boxscore.web, 16

pbpstats.data_loader.live.enhanced_pbp.file, 17

pbpstats.data_loader.live.enhanced_pbp.loader, 16

pbpstats.data_loader.live.enhanced_pbp.web, 17

pbpstats.data_loader.live.file_loader, 19

pbpstats.data_loader.live.pbp.file, 18

pbpstats.data_loader.live.pbp.loader, 17

pbpstats.data_loader.live.pbp.web, 18

pbpstats.data_loader.live.possessions.file, 19

pbpstats.data_loader.live.possessions.loader, 18

pbpstats.data_loader.live.possessions.web, 19

pbpstats.data_loader.live.web_loader, 19

pbpstats.data_loader.nba_enhanced_pbp_loader, 30

pbpstats.data_loader.nba_possession_loader, 30

pbpstats.data_loader.stats_nba.base, 20

pbpstats.data_loader.stats_nba.boxscore.file, 21

pbpstats.data_loader.stats_nba.boxscore.loader, 20

pbpstats.data_loader.stats_nba.boxscore.web, 21

pbpstats.data_loader.stats_nba.enhanced_pbp.file, 22

pbpstats.data_loader.stats_nba.enhanced_pbp.loader, 21

pbpstats.data_loader.stats_nba.enhanced_pbp.web,

22 pbpstats.resources.enhanced_pbp.data_nba.end_of_per
pbpstats.data_loader.stats_nba.file_loader, 34
29 pbpstats.resources.enhanced_pbp.data_nba.enhanced_p
pbpstats.data_loader.stats_nba.game_finder.file, 34
25 pbpstats.resources.enhanced_pbp.data_nba.enhanced_p
pbpstats.data_loader.stats_nba.game_finder.loader, 34
25 pbpstats.resources.enhanced_pbp.data_nba.field_goal,
pbpstats.data_loader.stats_nba.game_finder.web, 35
26 pbpstats.resources.enhanced_pbp.data_nba.foul,
pbpstats.data_loader.stats_nba.pbp.file, 35
23 pbpstats.resources.enhanced_pbp.data_nba.free_throw,
pbpstats.data_loader.stats_nba.pbp.loader, 36
23 pbpstats.resources.enhanced_pbp.data_nba.jump_ball,
pbpstats.data_loader.stats_nba.pbp.web, 37
23 pbpstats.resources.enhanced_pbp.data_nba.rebound,
pbpstats.data_loader.stats_nba.possessions.file, 37
24 pbpstats.resources.enhanced_pbp.data_nba.replay,
pbpstats.data_loader.stats_nba.possessions.loader, 37
24 pbpstats.resources.enhanced_pbp.data_nba.start_of_p
pbpstats.data_loader.stats_nba.possessions.web, 38
24 pbpstats.resources.enhanced_pbp.data_nba.substitut
pbpstats.data_loader.stats_nba.scoreboard.file, 38
26 pbpstats.resources.enhanced_pbp.data_nba.timeout,
pbpstats.data_loader.stats_nba.scoreboard.loader, 38
26 pbpstats.resources.enhanced_pbp.data_nba.turnover,
pbpstats.data_loader.stats_nba.scoreboard.web, 38
27 pbpstats.resources.enhanced_pbp.data_nba.violation,
pbpstats.data_loader.stats_nba.shots.file, 39
27 pbpstats.resources.enhanced_pbp.ejection,
pbpstats.data_loader.stats_nba.shots.loader, 51
27 pbpstats.resources.enhanced_pbp.end_of_period,
pbpstats.data_loader.stats_nba.shots.web, 52
28 pbpstats.resources.enhanced_pbp.enhanced_pbp,
pbpstats.data_loader.stats_nba.summary.file, 52
28 pbpstats.resources.enhanced_pbp.enhanced_pbp_item,
pbpstats.data_loader.stats_nba.summary.loader, 52
28 pbpstats.resources.enhanced_pbp.field_goal,
pbpstats.data_loader.stats_nba.summary.web, 53
29 pbpstats.resources.enhanced_pbp.foul,
pbpstats.data_loader.stats_nba.web_loader, 54
29 pbpstats.resources.enhanced_pbp.free_throw,
pbpstats.objects.day, 30 55
pbpstats.objects.game, 31 pbpstats.resources.enhanced_pbp.jump_ball,
pbpstats.objects.season, 31 56
pbpstats.resources.boxscore.boxscore, pbpstats.resources.enhanced_pbp.live.ejection,
32 39
pbpstats.resources.boxscore.data_nba_boxscore_item, pbpstats.resources.enhanced_pbp.live.end_of_period,
33 40
pbpstats.resources.boxscore.live_boxscore_item, pbpstats.resources.enhanced_pbp.live.enhanced_pbp_
33 40
pbpstats.resources.boxscore.stats_nba_boxscore_item, pbpstats.resources.enhanced_pbp.live.enhanced_pbp_
33 40
pbpstats.resources.enhanced_pbp.data_nba_boxscore_item, pbpstats.resources.enhanced_pbp.live.field_goal,
34 41

pbpstats.resources.enhanced_pbp.live.fouls, 41
pbpstats.resources.enhanced_pbp.live.free_throws, 42
pbpstats.resources.enhanced_pbp.live.jump_balls, 43
pbpstats.resources.enhanced_pbp.live.rebounds, 43
pbpstats.resources.enhanced_pbp.live.replays, 43
pbpstats.resources.enhanced_pbp.live.start_of_period, 43
pbpstats.resources.enhanced_pbp.live.substitution, 44
pbpstats.resources.enhanced_pbp.live.timeout, 44
pbpstats.resources.enhanced_pbp.live.turnover, 44
pbpstats.resources.enhanced_pbp.live.violation, 45
pbpstats.resources.enhanced_pbp.rebound, 56
pbpstats.resources.enhanced_pbp.replay, 57
pbpstats.resources.enhanced_pbp.start_of_period, 57
pbpstats.resources.enhanced_pbp.stats_nba.assists, 45
pbpstats.resources.enhanced_pbp.stats_nba.end_of_period, 46
pbpstats.resources.enhanced_pbp.stats_nba.enhanced_pbp_factory, 46
pbpstats.resources.enhanced_pbp.stats_nba.enhanced_pbp_item, 46
pbpstats.resources.enhanced_pbp.stats_nba.field_goal, 47
pbpstats.resources.enhanced_pbp.stats_nba.foul, 47
pbpstats.resources.enhanced_pbp.stats_nba.free_throw, 48
pbpstats.resources.enhanced_pbp.stats_nba.jump_ball, 49
pbpstats.resources.enhanced_pbp.stats_nba.rebound, 49
pbpstats.resources.enhanced_pbp.stats_nba.replay, 49
pbpstats.resources.enhanced_pbp.stats_nba.start_of_period, 50
pbpstats.resources.enhanced_pbp.stats_nba.substitution, 50
pbpstats.resources.enhanced_pbp.stats_nba.timeout, 50
pbpstats.resources.enhanced_pbp.stats_nba.turnover, 50

A

counts_as_personal_foul (pbp-
 action_type (pbpstats.resources.enhanced_pbp.live.ejection.LiveEjection
 attribute), 39 stats.resources.enhanced_pbp.foul.Foul
 attribute), 54
 counts_towards_penalty (pbp-
 action_type (pbpstats.resources.enhanced_pbp.live.end_of_period.LiveEndOfPeriod
 attribute), 40 stats.resources.enhanced_pbp.foul.Foul
 attribute), 54
 current_players (pbp-
 action_type (pbpstats.resources.enhanced_pbp.live.field_goal.LiveFieldGoal
 attribute), 41 stats.resources.enhanced_pbp.enhanced_pbp_item.EnhancedPbpItem
 attribute), 52
 current_players (pbp-
 action_type (pbpstats.resources.enhanced_pbp.live.free_throw.LiveFreeThrow
 attribute), 42 stats.resources.enhanced_pbp.live.substitution.LiveSubstitution
 attribute), 44
 current_players (pbp-
 action_type (pbpstats.resources.enhanced_pbp.live.jump_ball.LiveJumpBall
 attribute), 43 stats.resources.enhanced_pbp.start_of_period.StartOfPeriod
 attribute), 58
 current_players (pbp-
 action_type (pbpstats.resources.enhanced_pbp.live.rebound.LiveRebound
 attribute), 43 stats.resources.enhanced_pbp.substitution.Substitution
 attribute), 58
 current_players (pbp-
 action_type (pbpstats.resources.enhanced_pbp.live.replay.LiveReplay
 attribute), 43 stats.resources.enhanced_pbp.substitution.Substitution
 attribute), 58
 current_players (pbp-
 action_type (pbpstats.resources.enhanced_pbp.live.start_of_period.LiveStartOfPeriod
 attribute), 43 stats.resources.enhanced_pbp.substitution.Substitution
 attribute), 58
 data (pbpstats.data_loader.data_nba.boxscore.loader.DataNbaBoxscoreLoader
 attribute), 10
 data (pbpstats.data_loader.data_nba.pbp.loader.DataNbaPbpLoader
 attribute), 12
 data (pbpstats.data_loader.data_nba.schedule.loader.DataNbaScheduleLoader
 attribute), 14
 data (pbpstats.data_loader.live.boxscore.loader.LiveBoxscoreLoader
 attribute), 16
 data (pbpstats.data_loader.live.pbp.loader.LivePbpLoader
 attribute), 17
 data (pbpstats.data_loader.stats_nba.base.StatsNbaLoaderBase
 attribute), 20
 data (pbpstats.resources.boxscore.boxscore.Boxscore
 attribute), 32
 data (pbpstats.resources.boxscore.data_nba_boxscore_item.DataNbaBoxscoreItem
 attribute), 33
 data (pbpstats.resources.boxscore.live_boxscore_item.LiveBoxscoreItem
 attribute), 33
 data (pbpstats.resources.boxscore.stats_nba_boxscore_item.StatsNbaBoxscoreItem
 attribute), 33

B

base_stats (pbpstats.resources.enhanced_pbp.enhanced_pbp_item.EnhancedPbpItem
 attribute), 52
 Boxscore (class in pbpstats.resources.boxscore.boxscore), 32

C

Client (class in pbpstats.client), 64
 count_as_possession (pbp-
 stats.resources.enhanced_pbp.enhanced_pbp_item.EnhancedPbpItem
 attribute), 52
 data (pbpstats.data_loader.data_nba.boxscore.loader.DataNbaBoxscoreLoader
 attribute), 10
 data (pbpstats.data_loader.data_nba.pbp.loader.DataNbaPbpLoader
 attribute), 12
 data (pbpstats.data_loader.data_nba.schedule.loader.DataNbaScheduleLoader
 attribute), 14
 data (pbpstats.data_loader.live.boxscore.loader.LiveBoxscoreLoader
 attribute), 16
 data (pbpstats.data_loader.live.pbp.loader.LivePbpLoader
 attribute), 17
 data (pbpstats.data_loader.stats_nba.base.StatsNbaLoaderBase
 attribute), 20
 data (pbpstats.resources.boxscore.boxscore.Boxscore
 attribute), 32
 data (pbpstats.resources.boxscore.data_nba_boxscore_item.DataNbaBoxscoreItem
 attribute), 33
 data (pbpstats.resources.boxscore.live_boxscore_item.LiveBoxscoreItem
 attribute), 33
 data (pbpstats.resources.boxscore.stats_nba_boxscore_item.StatsNbaBoxscoreItem
 attribute), 33

data (pbpstats.resources.enhanced_pbp.data_nba.enhanced_pbp_item.DataNbaEnhancedPbpItem (pbp-
attribute), 35 data_provider
data (pbpstats.resources.enhanced_pbp.enhanced_pbp.EnhancedPbpItem stats.data_loader.stats_nba.boxscore.loader.StatsNbaBoxscoreLo
attribute), 52 attribute), 21
data (pbpstats.resources.enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem (pbp-
attribute), 40 stats.data_loader.stats_nba.enhanced_pbp.loader.StatsNbaEnhancedPbpItem
data (pbpstats.resources.enhanced_pbp.stats_nba.enhanced_pbp_item.StatsNbaEnhancedPbpItem
attribute), 46 data_provider (pbp-
data (pbpstats.resources.games.data_nba_game_item.DataNbaGameItem stats.data_loader.stats_nba.game_finder.loader.StatsNbaGameFin
attribute), 60 attribute), 25
data (pbpstats.resources.games.games.Games attribute), 60 data_provider (pbp-
stats.data_loader.stats_nba.pbp.loader.StatsNbaPbpLoader
data (pbpstats.resources.games.stats_nba_game_item.StatsNbaGameItem attribute), 23
attribute), 60 data_provider (pbp-
data (pbpstats.resources.pbp.data_nba_pbp_item.DataNbaPbpItem stats.data_loader.stats_nba.possessions.loader.StatsNbaPossession
attribute), 61 attribute), 24
data (pbpstats.resources.pbp.live_pbp_item.LivePbpItem data_provider (pbp-
attribute), 61 stats.data_loader.stats_nba.scoreboard.loader.StatsNbaScoreboard
data (pbpstats.resources.pbp.pbp.Pbp attribute), 61 attribute), 26
data (pbpstats.resources.pbp.stats_nba_pbp_item.StatsNbaPbpItem data_provider (pbp-
attribute), 61 stats.data_loader.stats_nba.shots.loader.StatsNbaShotsLoader
data (pbpstats.resources.possessions.possession.Possession attribute), 27
attribute), 62 data_provider (pbp-
data (pbpstats.resources.possessions.possessions.Possessions stats.data_loader.stats_nba.summary.loader.StatsNbaSummaryLo
attribute), 63 attribute), 28
data (pbpstats.resources.shots.shots.Shots attribute), 63 DataEjection (class in pbp-
data (pbpstats.resources.shots.stats_nba_shot.StatsNbaShot stats.resources.enhanced_pbp.data_nba.ejection),
attribute), 63 34
data_provider (pbp- DataEndOfPeriod (class in pbp-
stats.data_loader.data_nba.boxscore.loader.DataNbaBoxscoreLoader stats.resources.enhanced_pbp.data_nba.end_of_period),
attribute), 10 34
data_provider (pbp- DataEnhancedPbpItem (class in pbp-
stats.data_loader.data_nba.enhanced_pbp.loader.DataNbaEnhancedPbpItem stats.resources.enhanced_pbp.data_nba.enhanced_pbp_item),
attribute), 11 34
data_provider (pbp- DataFieldGoal (class in pbp-
stats.data_loader.data_nba.pbp.loader.DataNbaPbpLoader stats.resources.enhanced_pbp.data_nba.field_goal),
attribute), 12 35
data_provider (pbp- DataFoul (class in pbp-
stats.data_loader.data_nba.possessions.loader.DataNbaPossessionsLoader stats.resources.enhanced_pbp.data_nba.foul),
attribute), 13 35
data_provider (pbp- DataFreeThrow (class in pbp-
stats.data_loader.data_nba.schedule.loader.DataNbaScheduleLoader stats.resources.enhanced_pbp.data_nba.free_throw),
attribute), 14 36
data_provider (pbp- DataJumpBall (class in pbp-
stats.data_loader.live.boxscore.loader.LiveBoxscoreLoader stats.resources.enhanced_pbp.data_nba.jump_ball),
attribute), 16 37
data_provider (pbp- DataLoaderFactory (class in pbp-
stats.data_loader.live.enhanced_pbp.loader.LiveEnhancedPbpLoader stats.data_loader.factory), 30
attribute), 17 DataNbaBoxscoreFileLoader (class in pbp-
data_provider (pbp- stats.data_loader.data_nba.boxscore.file),
stats.data_loader.live.pbp.loader.LivePbpLoader 10
attribute), 18 DataNbaBoxscoreItem (class in pbp-
data_provider (pbp- stats.resources.boxscore.data_nba_boxscore_item),
stats.data_loader.live.possessions.loader.LivePossessionLoader 11

DataNbaBoxscoreLoader (class in <i>pbpstats.data_loader.data_nba.boxscore.loader</i>), 9	DataRebound (class in <i>pbpstats.resources.enhanced_pbp.data_nba.rebound</i>), 37
DataNbaBoxscoreWebLoader (class in <i>pbpstats.data_loader.data_nba.boxscore.web</i>), 10	DataReplay (class in <i>pbpstats.resources.enhanced_pbp.data_nba.replay</i>), 37
DataNbaEnhancedPbpFactory (class in <i>pbpstats.resources.enhanced_pbp.data_nba.enhanced_pbp_factory</i>), 34	EndOfPeriod (class in <i>pbpstats.resources.enhanced_pbp.data_nba.start_of_period</i>), 38
DataNbaEnhancedPbpFileLoader (class in <i>pbpstats.data_loader.data_nba.enhanced_pbp.file</i>), 11	DataSubstitution (class in <i>pbpstats.resources.enhanced_pbp.data_nba.substitution</i>), 38
DataNbaEnhancedPbpLoader (class in <i>pbpstats.data_loader.data_nba.enhanced_pbp.loader</i>), 11	DataTimeout (class in <i>pbpstats.resources.enhanced_pbp.data_nba.timeout</i>), 38
DataNbaEnhancedPbpWebLoader (class in <i>pbpstats.data_loader.data_nba.enhanced_pbp.web</i>), 11	DataTurnover (class in <i>pbpstats.resources.enhanced_pbp.data_nba.turnover</i>), 38
DataNbaFileLoader (class in <i>pbpstats.data_loader.data_nba.file_loader</i>), 15	DataViolation (class in <i>pbpstats.resources.enhanced_pbp.data_nba.violation</i>), 39
DataNbaGameItem (class in <i>pbpstats.resources.games.data_nba_game_item</i>), 60	Day (class in <i>pbpstats.objects.day</i>), 31
DataNbaPbpFileLoader (class in <i>pbpstats.data_loader.data_nba.pbp.file</i>), 12	dedupe_events_row_set () (pbpstats.data_loader.stats_nba.base.StatsNbaLoaderBase static method), 20
DataNbaPbpItem (class in <i>pbpstats.resources.pbp.data_nba_pbp_item</i>), 60	distance (pbpstats.resources.enhanced_pbp.field_goal.FieldGoal attribute), 53
DataNbaPbpLoader (class in <i>pbpstats.data_loader.data_nba.pbp.loader</i>), 12	E
DataNbaPbpWebLoader (class in <i>pbpstats.data_loader.data_nba.pbp.web</i>), 12	Ejection (class in <i>pbpstats.resources.enhanced_pbp.ejection</i>), 51
DataNbaPossessionFileLoader (class in <i>pbpstats.data_loader.data_nba.possessions.file</i>), 13	end_time (pbpstats.resources.possessions.possession.Possession attribute), 62
DataNbaPossessionLoader (class in <i>pbpstats.data_loader.data_nba.possessions.loader</i>), 13	EndOfPeriod (class in <i>pbpstats.resources.enhanced_pbp.end_of_period</i>), 52
DataNbaPossessionWebLoader (class in <i>pbpstats.data_loader.data_nba.possessions.web</i>), 13	EnhancedPbp (class in <i>pbpstats.resources.enhanced_pbp.enhanced_pbp</i>), 52
DataNbaScheduleFileLoader (class in <i>pbpstats.data_loader.data_nba.schedule.file</i>), 14	EnhancedPbpItem (class in <i>pbpstats.resources.enhanced_pbp.enhanced_pbp_item</i>), 52
DataNbaScheduleLoader (class in <i>pbpstats.data_loader.data_nba.schedule.loader</i>), 14	event_for_efficiency_stats (pbpstats.resources.enhanced_pbp.free_throw.FreeThrow attribute), 55
DataNbaScheduleWebLoader (class in <i>pbpstats.data_loader.data_nba.schedule.web</i>), 14	event_stats (pbpstats.resources.enhanced_pbp.data_nba.enhanced_pbp attribute), 35
DataNbaWebLoader (class in <i>pbpstats.data_loader.data_nba.web_loader</i>),	event_stats (pbpstats.resources.enhanced_pbp.ejection.Ejection attribute), 51
	event_stats (pbpstats.resources.enhanced_pbp.end_of_period.EndOfPeriod attribute), 52
	event_stats (pbpstats.resources.enhanced_pbp.enhanced_pbp_item.En attribute), 53

event_stats (pbpstats.resources.enhanced_pbp.field_goal.FieldGoal (pbpstats.resources.enhanced_pbp.stats_nba.field_goal.StatsFieldGoal attribute), 53
 event_stats (pbpstats.resources.enhanced_pbp.foul.Foul (pbpstats.resources.enhanced_pbp.stats_nba.foul.StatsFoul attribute), 54
 event_stats (pbpstats.resources.enhanced_pbp.free_throw.FreeThrow (pbpstats.resources.enhanced_pbp.stats_nba.free_throw.StatsFreeThrow attribute), 55
 event_stats (pbpstats.resources.enhanced_pbp.jump_ball.JumpBall (pbpstats.resources.enhanced_pbp.stats_nba.jump_ball.StatsJumpBall attribute), 56
 event_stats (pbpstats.resources.enhanced_pbp.live.enhanced_pbp.live.enhanced_pbp.enhanced_pbp.stats_nba.rebound.StatsRebound attribute), 40
 event_stats (pbpstats.resources.enhanced_pbp.rebound.Rebound (pbpstats.resources.enhanced_pbp.stats_nba.replay.StatsReplay attribute), 56
 event_stats (pbpstats.resources.enhanced_pbp.replay.Replay (pbpstats.resources.enhanced_pbp.stats_nba.start_of_period.StatsStartOfPeriod attribute), 57
 event_stats (pbpstats.resources.enhanced_pbp.start_of_period.StartOfPeriod (pbpstats.resources.enhanced_pbp.stats_nba.substitution.StatsSubstitution attribute), 58
 event_stats (pbpstats.resources.enhanced_pbp.substitution.Substitution (pbpstats.resources.enhanced_pbp.stats_nba.timeout.StatsTimeout attribute), 58
 event_stats (pbpstats.resources.enhanced_pbp.timeout.Timeout (pbpstats.resources.enhanced_pbp.stats_nba.turnover.StatsTurnover attribute), 58
 event_stats (pbpstats.resources.enhanced_pbp.turnover.Turnover (pbpstats.resources.enhanced_pbp.stats_nba.violation.StatsViolation attribute), 59
 event_stats (pbpstats.resources.enhanced_pbp.violation.Violation (pbpstats.resources.enhanced_pbp.stats_nba.violation.StatsViolation attribute), 59
 event_type (pbpstats.resources.enhanced_pbp.data_nba.Ejection.DataEjection (pbpstats.resources.enhanced_pbp.enhanced_pbp.EnhancedPbp attribute), 34
 event_type (pbpstats.resources.enhanced_pbp.data_nba.end_of_period.EndOfPeriod (pbpstats.resources.enhanced_pbp.enhanced_pbp.EnhancedPbp attribute), 34
 event_type (pbpstats.resources.enhanced_pbp.data_nba.field_goal.FieldGoal (pbpstats.resources.enhanced_pbp.enhanced_pbp.EnhancedPbp attribute), 35
 event_type (pbpstats.resources.enhanced_pbp.data_nba.foul.Foul (pbpstats.resources.enhanced_pbp.enhanced_pbp.EnhancedPbp attribute), 35
 event_type (pbpstats.resources.enhanced_pbp.data_nba.free_throw.FreeThrow (pbpstats.resources.enhanced_pbp.enhanced_pbp.EnhancedPbp attribute), 36
 event_type (pbpstats.resources.enhanced_pbp.data_nba.jump_ball.JumpBall (pbpstats.resources.enhanced_pbp.enhanced_pbp.EnhancedPbp attribute), 37
 event_type (pbpstats.resources.enhanced_pbp.data_nba.rebound.Rebound (pbpstats.resources.enhanced_pbp.enhanced_pbp.EnhancedPbp attribute), 37
 event_type (pbpstats.resources.enhanced_pbp.data_nba.replay.Replay (pbpstats.resources.enhanced_pbp.enhanced_pbp.EnhancedPbp attribute), 37
 event_type (pbpstats.resources.enhanced_pbp.data_nba.start_of_period.StartOfPeriod (pbpstats.resources.enhanced_pbp.enhanced_pbp.EnhancedPbp attribute), 38
 event_type (pbpstats.resources.enhanced_pbp.data_nba.substitution.Substitution (pbpstats.resources.enhanced_pbp.enhanced_pbp.EnhancedPbp attribute), 38
 event_type (pbpstats.resources.enhanced_pbp.data_nba.timeout.Timeout (pbpstats.resources.enhanced_pbp.enhanced_pbp.EnhancedPbp attribute), 38
 event_type (pbpstats.resources.enhanced_pbp.data_nba.turnover.Turnover (pbpstats.resources.enhanced_pbp.enhanced_pbp.EnhancedPbp attribute), 38
 event_type (pbpstats.resources.enhanced_pbp.data_nba.violation.Violation (pbpstats.resources.enhanced_pbp.enhanced_pbp.EnhancedPbp attribute), 39
 event_type (pbpstats.resources.enhanced_pbp.stats_nba.ejection.StatsEjection (pbpstats.resources.enhanced_pbp.stats_nba.ejection.StatsEjection attribute), 45
 event_type (pbpstats.resources.enhanced_pbp.stats_nba.end_of_period.StatsEndOfPeriod (pbpstats.resources.enhanced_pbp.stats_nba.end_of_period.StatsEndOfPeriod attribute), 46
 Games (class in pbpstats.resources.games.games), 60

get_all_events_at_current_time() (pbp- stats.resources.enhanced_pbp.enhanced_pbp_item.EnhancedPbpItem method), 53
 get_data_loader() (pbp- stats.data_loader.factory.DataLoaderFactory method), 30
 get_event_class() (pbp- stats.resources.enhanced_pbp.data_nba.enhanced_pbp_factory.DataNbaEnhancedPbpFactory method), 34
 get_event_class() (pbp- stats.resources.enhanced_pbp.live.enhanced_pbp_factory.LiveEnhancedPbpFactory method), 40
 get_event_class() (pbp- stats.resources.enhanced_pbp.stats_nba.enhanced_pbp_factory.StatsNbaEnhancedPbpFactory method), 46
 get_offense_team_id() (pbp- stats.resources.enhanced_pbp.data_nba.enhanced_pbp_item.DataNbaEnhancedPbpItem method), 35
 get_offense_team_id() (pbp- stats.resources.enhanced_pbp.enhanced_pbp_item.EnhancedPbpItem method), 53
 get_offense_team_id() (pbp- stats.resources.enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem method), 41
 get_offense_team_id() (pbp- stats.resources.enhanced_pbp.start_of_period.StartOfPeriod method), 58
 get_offense_team_id() (pbp- stats.resources.enhanced_pbp.stats_nba.enhanced_pbp_item.StatsNbaEnhancedPbpItem method), 46
 get_offense_team_id() (pbp- stats.resources.enhanced_pbp.stats_nba.field_goal.StatsFieldGoal method), 47
 get_offense_team_id() (pbp- stats.resources.enhanced_pbp.stats_nba.free_throw.StatsFreeThrow method), 48
 get_offense_team_id() (pbp- stats.resources.enhanced_pbp.stats_nba.jump_ball.StatsJumpBall method), 49
 get_offense_team_id() (pbp- stats.resources.enhanced_pbp.stats_nba.rebound.StatsRebound method), 49
 get_offense_team_id() (pbp- stats.resources.enhanced_pbp.stats_nba.turnover.StatsTurnover method), 51
 get_period_starters() (pbp- stats.resources.enhanced_pbp.data_nba.start_of_period.DataStartOfPeriod method), 38
 get_period_starters() (pbp- stats.resources.enhanced_pbp.live.start_of_period.LiveStartOfPeriod method), 44
 get_period_starters() (pbp- stats.resources.enhanced_pbp.start_of_period.StartOfPeriod class method), 58
 get_period_starters() (pbp- stats.resources.enhanced_pbp.stats_nba.start_of_period.StatsStartOfPeriod method), 50
 get_team_ids() (pbp- stats.resources.possessions.possession.Possession method), 62
 get_team_starting_with_ball() (pbp- stats.data_nba.enhanced_pbp_factory.DataNbaEnhancedPbpFactory method), 58
 incoming_player_id (pbp- stats.resources.enhanced_pbp.data_nba.substitution.DataSubstitution attribute), 46
 incoming_player_id (pbp- stats.resources.enhanced_pbp.live.substitution.LiveSubstitution attribute), 46
 incoming_player_id (pbp- stats.resources.enhanced_pbp.stats_nba.substitution.StatsSubstitution attribute), 50
 incoming_player_id (pbp- stats.resources.enhanced_pbp.substitution.Substitution attribute), 46
 InvalidNumberOfStartersException, 57
 is_3_second_violation (pbp- stats.resources.enhanced_pbp.data_nba.turnover.DataTurnover attribute), 39
 is_3_second_violation (pbp- stats.resources.enhanced_pbp.live.turnover.LiveTurnover attribute), 44
 is_3_second_violation (pbp- stats.resources.enhanced_pbp.stats_nba.turnover.StatsTurnover attribute), 51
 is_3_second_violation() (pbp- stats.resources.enhanced_pbp.turnover.Turnover class method), 59
 is_and1 (pbpstats.resources.enhanced_pbp.field_goal.FieldGoal attribute), 53
 is_assisted (pbpstats.resources.enhanced_pbp.field_goal.FieldGoal attribute), 53
 is_away_from_play_foul (pbp- stats.resources.enhanced_pbp.data_nba.foul.DataFoul attribute), 35
 is_away_from_play_foul (pbp- stats.resources.enhanced_pbp.foul.Foul attribute), 54
 is_away_from_play_foul (pbp- stats.resources.enhanced_pbp.live.foul.LiveFoul attribute), 41
 is_away_from_play_foul (pbp- stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul attribute), 47
 is_away_from_play_ft (pbp- stats.resources.enhanced_pbp.free_throw.FreeThrow

attribute), 55
 is_away_from_play_ft (pbp- stats.resources.enhanced_pbp.live.free_throw.LiveFreeThrow attribute), 42
 is_bad_pass (pbpstats.resources.enhanced_pbp.data_nba.turnover.DataTurnover attribute), 39
 is_bad_pass (pbpstats.resources.enhanced_pbp.live.turnover.LiveTurnover attribute), 44
 is_bad_pass (pbpstats.resources.enhanced_pbp.stats_nba.turnover.StatsTurnover attribute), 51
 is_bad_pass () (pbp- stats.resources.enhanced_pbp.turnover.Turnover class method), 59
 is_bad_pass_out_of_bounds (pbp- stats.resources.enhanced_pbp.data_nba.turnover.DataTurnover attribute), 39
 is_bad_pass_out_of_bounds (pbp- stats.resources.enhanced_pbp.live.turnover.LiveTurnover attribute), 45
 is_bad_pass_out_of_bounds (pbp- stats.resources.enhanced_pbp.stats_nba.turnover.StatsTurnover attribute), 51
 is_bad_pass_out_of_bounds () (pbp- stats.resources.enhanced_pbp.turnover.Turnover class method), 59
 is_blocked (pbpstats.resources.enhanced_pbp.field_goal.FieldGoal attribute), 53
 is_buzzer_beater_placeholder (pbp- stats.resources.enhanced_pbp.rebound.Rebound attribute), 56
 is_buzzer_beater_rebound_at_shot_time (pbpstats.resources.enhanced_pbp.rebound.Rebound attribute), 56
 is_charge (pbpstats.resources.enhanced_pbp.data_nba.foul.DataFoul attribute), 35
 is_charge (pbpstats.resources.enhanced_pbp.foul.Foul attribute), 54
 is_charge (pbpstats.resources.enhanced_pbp.live.foul.LiveFoul attribute), 41
 is_charge (pbpstats.resources.enhanced_pbp.stats_nba.foul.StatsFoul attribute), 47
 is_clear_path_foul (pbp- stats.resources.enhanced_pbp.data_nba.foul.DataFoul attribute), 35
 is_clear_path_foul (pbp- stats.resources.enhanced_pbp.foul.Foul attribute), 54
 is_clear_path_foul (pbp- stats.resources.enhanced_pbp.live.foul.LiveFoul attribute), 41
 is_clear_path_foul (pbp- stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul attribute), 47
 is_corner_3 (pbpstats.resources.enhanced_pbp.field_goal.FieldGoal attribute), 53
 is_defensive_3_seconds (pbp- stats.resources.enhanced_pbp.data_nba.foul.DataFoul attribute), 36
 is_double_foul (pbp- stats.resources.enhanced_pbp.foul.Foul attribute), 54
 is_defensive_3_seconds (pbp- stats.resources.enhanced_pbp.live.foul.LiveFoul attribute), 41
 is_defensive_3_seconds (pbp- stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul attribute), 47
 is_delay_of_game (pbp- stats.resources.enhanced_pbp.data_nba.foul.DataFoul attribute), 36
 is_delay_of_game (pbp- stats.resources.enhanced_pbp.data_nba.violation.DataViolation attribute), 39
 is_delay_of_game (pbp- stats.resources.enhanced_pbp.foul.Foul attribute), 54
 is_delay_of_game (pbp- stats.resources.enhanced_pbp.live.foul.LiveFoul attribute), 41
 is_delay_of_game (pbp- stats.resources.enhanced_pbp.live.violation.LiveViolation attribute), 45
 is_delay_of_game (pbp- stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul attribute), 47
 is_delay_of_game (pbp- stats.resources.enhanced_pbp.stats_nba.violation.StatsViolation attribute), 51
 is_double_foul (pbp- stats.resources.enhanced_pbp.foul.Foul attribute), 54
 is_double_foul (pbp- stats.resources.enhanced_pbp.live.foul.LiveFoul attribute), 41
 is_double_foul (pbp- stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul attribute), 47
 is_double_lane_violation (pbp- stats.resources.enhanced_pbp.data_nba.violation.DataViolation attribute), 39
 is_double_lane_violation (pbp- stats.resources.enhanced_pbp.live.violation.LiveViolation attribute), 39

attribute), 45
 is_double_lane_violation (pbp- stats.resources.enhanced_pbp.stats_nba.violation.StatsViolation attribute), 36
 stats.resources.enhanced_pbp.stats_nba.violation.StatsViolation attribute), 51
 stats.resources.enhanced_pbp.live.free_throw.LiveFreeThrow
 is_double_lane_violation() (pbp- attribute), 42
 stats.resources.enhanced_pbp.violation.Violation is_flagrant_ft (pbp- attribute), 48
 stats.resources.enhanced_pbp.stats_nba.free_throw.StatsFreeThrow
 is_double_technical (pbp- attribute), 36
 stats.resources.enhanced_pbp.data_nba.foul.DataFoul ft_1_of_1 (pbp- attribute), 36
 stats.resources.enhanced_pbp.data_nba.free_throw.DataFreeThrow
 is_double_technical (pbp- attribute), 36
 stats.resources.enhanced_pbp.foul.Foul at is_ft_1_of_1 (pbp- attribute), 54
 stats.resources.enhanced_pbp.free_throw.FreeThrow
 is_double_technical (pbp- attribute), 55
 stats.resources.enhanced_pbp.live.foul.LiveFoul is_ft_1_of_1 (pbp- attribute), 41
 stats.resources.enhanced_pbp.live.free_throw.LiveFreeThrow
 is_double_technical (pbp- attribute), 42
 stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul ft_1_of_1 (pbp- attribute), 47
 stats.resources.enhanced_pbp.stats_nba.free_throw.StatsFreeThrow
 is_end_ft (pbpstats.resources.enhanced_pbp.free_throw.FreeThrow attribute), 48
 attribute), 55
 is_ft_1_of_2 (pbp- attribute), 36
 is_final (pbpstats.resources.games.data_nba_game_item.DataNbaGameItem stats.resources.enhanced_pbp.data_nba.free_throw.DataFreeThrow attribute), 60
 attribute), 36
 is_final (pbpstats.resources.games.stats_nba_game_item.StatsNbaGameItem (pbp- attribute), 60
 stats.resources.enhanced_pbp.free_throw.FreeThrow
 is_first_ft (pbpstats.resources.enhanced_pbp.free_throw.FreeThrow attribute), 55
 attribute), 55
 is_ft_1_of_2 (pbp- attribute), 42
 is_flagrant (pbpstats.resources.enhanced_pbp.foul.Foul stats.resources.enhanced_pbp.live.free_throw.LiveFreeThrow attribute), 54
 attribute), 42
 is_flagrant1 (pbp- is_ft_1_of_2 (pbp- attribute), 36
 stats.resources.enhanced_pbp.data_nba.foul.DataFoul stats.resources.enhanced_pbp.stats_nba.free_throw.StatsFreeThrow attribute), 48
 is_flagrant1 (pbp- is_ft_1_of_3 (pbp- attribute), 36
 stats.resources.enhanced_pbp.foul.Foul at stats.resources.enhanced_pbp.data_nba.free_throw.DataFreeThrow attribute), 54
 attribute), 36
 is_flagrant1 (pbp- is_ft_1_of_3 (pbp- attribute), 55
 stats.resources.enhanced_pbp.live.foul.LiveFoul stats.resources.enhanced_pbp.free_throw.FreeThrow attribute), 41
 attribute), 55
 is_flagrant1 (pbp- is_ft_1_of_3 (pbp- attribute), 42
 stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul stats.resources.enhanced_pbp.live.free_throw.LiveFreeThrow attribute), 47
 attribute), 42
 is_flagrant2 (pbp- is_ft_1_of_3 (pbp- attribute), 48
 stats.resources.enhanced_pbp.data_nba.foul.DataFoul stats.resources.enhanced_pbp.stats_nba.free_throw.StatsFreeThrow attribute), 36
 attribute), 48
 is_flagrant2 (pbp- is_ft_1pt (pbpstats.resources.enhanced_pbp.data_nba.free_throw.Data attribute), 36
 stats.resources.enhanced_pbp.foul.Foul at attribute), 36
 is_flagrant2 (pbp- is_ft_1pt (pbpstats.resources.enhanced_pbp.free_throw.FreeThrow attribute), 55
 stats.resources.enhanced_pbp.live.foul.LiveFoul is_ft_1pt (pbpstats.resources.enhanced_pbp.free_throw.FreeThrow attribute), 41
 attribute), 42
 is_flagrant2 (pbp- is_ft_1pt (pbpstats.resources.enhanced_pbp.stats_nba.free_throw.Stats attribute), 48
 stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul attribute), 47
 attribute), 47
 is_flagrant_ft (pbp- is_ft_2_of_2 (pbp- stats.resources.enhanced_pbp.data_nba.free_throw.DataFreeThrow

attribute), 36		stats.resources.enhanced_pbp.live.violation.LiveViolation
is_ft_2_of_2	(pbp- stats.resources.enhanced_pbp.free_throw.FreeThrow	attribute), 45
attribute), 55		is_goaltend_violation (pbp-
is_ft_2_of_2	(pbp- stats.resources.enhanced_pbp.free_throw.FreeThrow	attribute), 51
attribute), 42		stats.resources.enhanced_pbp.live.free_throw.LiveFreeThrow
is_ft_2_of_2	(pbp- stats.resources.enhanced_pbp.live.free_throw.LiveFreeThrow	attribute), 42
attribute), 48		stats.resources.enhanced_pbp.stats_nba.free_throw.StatsFreeThrow
is_ft_2_of_3	(pbp- stats.resources.enhanced_pbp.stats_nba.free_throw.StatsFreeThrow	attribute), 48
attribute), 36		stats.resources.enhanced_pbp.field_goal.FieldGoal
is_ft_2_of_3	(pbp- stats.resources.enhanced_pbp.data_nba.free_throw.DataFreeThrow	attribute), 36
attribute), 36		stats.resources.enhanced_pbp.data_nba.foul.DataFoul
is_ft_2_of_3	(pbp- stats.resources.enhanced_pbp.free_throw.FreeThrow	attribute), 55
attribute), 55		stats.resources.enhanced_pbp.foul.Foul
is_ft_2_of_3	(pbp- stats.resources.enhanced_pbp.live.free_throw.LiveFreeThrow	attribute), 42
attribute), 42		stats.resources.enhanced_pbp.live.foul.LiveFoul
is_ft_2_of_3	(pbp- stats.resources.enhanced_pbp.stats_nba.free_throw.StatsFreeThrow	attribute), 48
attribute), 48		stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul
is_ft_2pt (pbpstats.resources.enhanced_pbp.data_nba.free_throw.DataFreeThrow		attribute), 36
attribute), 36		stats.resources.enhanced_pbp.free_throw.FreeThrow
is_ft_2pt (pbpstats.resources.enhanced_pbp.free_throw.FreeThrow		attribute), 55
attribute), 55		is_jumpball_violation (pbp-
is_ft_2pt (pbpstats.resources.enhanced_pbp.live.free_throw.LiveFreeThrow		attribute), 42
attribute), 42		stats.resources.enhanced_pbp.data_nba.violation.DataViolation
is_ft_2pt (pbpstats.resources.enhanced_pbp.stats_nba.free_throw.StatsFreeThrow		attribute), 48
attribute), 48		stats.resources.enhanced_pbp.live.violation.LiveViolation
is_ft_3_of_3	(pbp- stats.resources.enhanced_pbp.data_nba.free_throw.DataFreeThrow	attribute), 36
attribute), 36		stats.resources.enhanced_pbp.stats_nba.violation.StatsViolation
is_ft_3_of_3	(pbp- stats.resources.enhanced_pbp.free_throw.FreeThrow	attribute), 55
attribute), 55		stats.resources.enhanced_pbp.violation.Violation
is_ft_3_of_3	(pbp- stats.resources.enhanced_pbp.live.free_throw.LiveFreeThrow	attribute), 42
attribute), 42		stats.resources.enhanced_pbp.data_nba.turnover.DataTurnover
is_ft_3_of_3	(pbp- stats.resources.enhanced_pbp.stats_nba.free_throw.StatsFreeThrow	attribute), 48
attribute), 48		stats.resources.enhanced_pbp.live.turnover.LiveTurnover
is_ft_3pt (pbpstats.resources.enhanced_pbp.data_nba.free_throw.DataFreeThrow		attribute), 36
attribute), 36		is_kicked_ball (pbp-
is_ft_3pt (pbpstats.resources.enhanced_pbp.free_throw.FreeThrow		attribute), 56
attribute), 56		stats.resources.enhanced_pbp.stats_nba.turnover.StatsTurnover
is_ft_3pt (pbpstats.resources.enhanced_pbp.live.free_throw.LiveFreeThrow		attribute), 42
attribute), 42		stats.resources.enhanced_pbp.turnover.Turnover
is_ft_3pt (pbpstats.resources.enhanced_pbp.stats_nba.free_throw.StatsFreeThrow		attribute), 48
attribute), 48		is_kicked_ball_violation (pbp-
is_goaltend_violation	(pbp- stats.resources.enhanced_pbp.data_nba.violation.DataViolation	attribute), 39
attribute), 39		is_kicked_ball_violation (pbp-
is_goaltend_violation	(pbp- stats.resources.enhanced_pbp.live.violation.LiveViolation	

Index	77
--------------	-----------

<i>stats.resources.enhanced_pbp.live.foul.LiveFoul</i> <i>attribute</i>), 41	<i>stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul</i> <i>attribute</i>), 48
<i>is_offensive_foul</i> (<i>pbp- stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul</i> <i>attribute</i>), 47	<i>is_placeholder</i> (<i>pbp- stats.resources.enhanced_pbp.data_nba.rebound.DataRebound</i> <i>attribute</i>), 37
<i>is_offensive_goaltending</i> (<i>pbp- stats.resources.enhanced_pbp.data_nba.turnover.DataTurnover</i> <i>attribute</i>), 39	<i>is_placeholder</i> (<i>pbp- stats.resources.enhanced_pbp.live.rebound.LiveRebound</i> <i>attribute</i>), 43
<i>is_offensive_goaltending</i> (<i>pbp- stats.resources.enhanced_pbp.live.turnover.LiveTurnover</i> <i>attribute</i>), 45	<i>is_placeholder</i> (<i>pbp- stats.resources.enhanced_pbp.rebound.Rebound</i> <i>attribute</i>), 57
<i>is_offensive_goaltending</i> (<i>pbp- stats.resources.enhanced_pbp.stats_nba.turnover.StatsTurnover</i> <i>attribute</i>), 51	<i>is_placeholder</i> (<i>pbp- stats.resources.enhanced_pbp.stats_nba.rebound.StatsRebound</i> <i>attribute</i>), 49
<i>is_offensive_goaltending()</i> (<i>pbp- stats.resources.enhanced_pbp.turnover.Turnover</i> <i>class method</i>), 59	<i>is_possession_ending_event</i> (<i>pbp- stats.resources.enhanced_pbp.data_nba.enhanced_pbp_item.Data</i> <i>attribute</i>), 35
<i>is_penalty_event()</i> (<i>pbp- stats.resources.enhanced_pbp.enhanced_pbp_item.EnhancedPbpItem</i> <i>method</i>), 53	<i>is_possession_ending_event</i> (<i>pbp- stats.resources.enhanced_pbp.enhanced_pbp_item.EnhancedPbp</i> <i>attribute</i>), 53
<i>is_personal_block_foul</i> (<i>pbp- stats.resources.enhanced_pbp.data_nba.foul.DataFoul</i> <i>attribute</i>), 36	<i>is_possession_ending_event</i> (<i>pbp- stats.resources.enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem</i> <i>attribute</i>), 41
<i>is_personal_block_foul</i> (<i>pbp- stats.resources.enhanced_pbp.foul.Foul</i> <i>at-</i> <i>tribute</i>), 55	<i>is_possession_ending_event</i> (<i>pbp- stats.resources.enhanced_pbp.stats_nba.enhanced_pbp_item.StatsEnhancedPbpItem</i> <i>attribute</i>), 46
<i>is_personal_block_foul</i> (<i>pbp- stats.resources.enhanced_pbp.live.foul.LiveFoul</i> <i>attribute</i>), 42	<i>is_putback</i> (<i>pbpstats.resources.enhanced_pbp.field_goal.FieldGoal</i> <i>attribute</i>), 54
<i>is_personal_block_foul</i> (<i>pbp- stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul</i> <i>attribute</i>), 47	<i>is_real_rebound</i> (<i>pbp- stats.resources.enhanced_pbp.rebound.Rebound</i> <i>attribute</i>), 57
<i>is_personal_foul</i> (<i>pbp- stats.resources.enhanced_pbp.data_nba.foul.DataFoul</i> <i>attribute</i>), 36	<i>is_second_chance_event()</i> (<i>pbp- stats.resources.enhanced_pbp.enhanced_pbp_item.EnhancedPbp</i> <i>method</i>), 53
<i>is_personal_foul</i> (<i>pbp- stats.resources.enhanced_pbp.foul.Foul</i> <i>at-</i> <i>tribute</i>), 55	<i>is_shooting_block_foul</i> (<i>pbp- stats.resources.enhanced_pbp.data_nba.foul.DataFoul</i> <i>attribute</i>), 36
<i>is_personal_foul</i> (<i>pbp- stats.resources.enhanced_pbp.live.foul.LiveFoul</i> <i>attribute</i>), 42	<i>is_shooting_block_foul</i> (<i>pbp- stats.resources.enhanced_pbp.foul.Foul</i> <i>at-</i> <i>tribute</i>), 55
<i>is_personal_foul</i> (<i>pbp- stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul</i> <i>attribute</i>), 47	<i>is_shooting_block_foul</i> (<i>pbp- stats.resources.enhanced_pbp.live.foul.LiveFoul</i> <i>attribute</i>), 42
<i>is_personal_take_foul</i> (<i>pbp- stats.resources.enhanced_pbp.data_nba.foul.DataFoul</i> <i>attribute</i>), 36	<i>is_shooting_block_foul</i> (<i>pbp- stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul</i> <i>attribute</i>), 48
<i>is_personal_take_foul</i> (<i>pbp- stats.resources.enhanced_pbp.foul.Foul</i> <i>at-</i> <i>tribute</i>), 55	<i>is_shooting_foul</i> (<i>pbp- stats.resources.enhanced_pbp.data_nba.foul.DataFoul</i> <i>attribute</i>), 36
<i>is_personal_take_foul</i> (<i>pbp- stats.resources.enhanced_pbp.live.foul.LiveFoul</i> <i>attribute</i>), 42	<i>is_shooting_foul</i> (<i>pbp- stats.resources.enhanced_pbp.foul.Foul</i> <i>at-</i> <i>tribute</i>), 55
<i>is_personal_take_foul</i> (<i>pbp- stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul</i> <i>attribute</i>), 47	<i>is_shooting_foul</i> (<i>pbp- stats.resources.enhanced_pbp.live.foul.LiveFoul</i> <i>attribute</i>), 42

attribute), 42
 is_shooting_foul (pbp- stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul attribute), 36
 attribute), 48
 is_shot_clock_violation (pbp- stats.resources.enhanced_pbp.data_nba.turnover.DataTurnover attribute), 55
 attribute), 39
 is_shot_clock_violation (pbp- stats.resources.enhanced_pbp.live.turnover.LiveTurnover attribute), 42
 attribute), 45
 is_shot_clock_violation (pbp- stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul attribute), 48
 attribute), 51
 is_shot_clock_violation() (pbp- stats.resources.enhanced_pbp.turnover.Turnover class method), 59
 is_steal (pbpstats.resources.enhanced_pbp.turnover.Turnover attribute), 39
 attribute), 59
 is_step_out_of_bounds (pbp- stats.resources.enhanced_pbp.data_nba.turnover.DataTurnover attribute), 39
 attribute), 39
 is_step_out_of_bounds (pbp- stats.resources.enhanced_pbp.live.turnover.LiveTurnover attribute), 45
 is_step_out_of_bounds (pbp- stats.resources.enhanced_pbp.stats_nba.turnover.StatsTurnover attribute), 57
 attribute), 51
 is_step_out_of_bounds() (pbp- stats.resources.enhanced_pbp.turnover.Turnover class method), 59
 is_technical (pbp- stats.resources.enhanced_pbp.data_nba.foul.DataFoul attribute), 36
 is_technical (pbp- stats.resources.enhanced_pbp.foul.Foul attribute), 55
 is_technical (pbp- stats.resources.enhanced_pbp.live.foul.LiveFoul attribute), 42
 is_technical (pbp- stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul attribute), 48
 is_technical_ft (pbp- stats.resources.enhanced_pbp.data_nba.free_throw.DataFreeThrow attribute), 37
 is_technical_ft (pbp- stats.resources.enhanced_pbp.free_throw.FreeThrow attribute), 56
 is_technical_ft (pbp- stats.resources.enhanced_pbp.live.free_throw.LiveFreeThrow attribute), 42
 is_technical_ft (pbp- stats.resources.enhanced_pbp.stats_nba.free_throw.StatsFreeThrow attribute), 48
 is_transition_take_foul (pbp- stats.resources.enhanced_pbp.data_nba.foul.DataFoul attribute), 36
 is_transition_take_foul (pbp- stats.resources.enhanced_pbp.foul.Foul attribute), 55
 is_transition_take_foul (pbp- stats.resources.enhanced_pbp.live.foul.LiveFoul attribute), 42
 is_transition_take_foul (pbp- stats.resources.enhanced_pbp.stats_nba.foul.StatsFoul attribute), 48
 is_transition_take_foul_ft (pbp- stats.resources.enhanced_pbp.free_throw.FreeThrow attribute), 56
 is_travel (pbpstats.resources.enhanced_pbp.data_nba.turnover.DataTurnover attribute), 39
 is_travel (pbpstats.resources.enhanced_pbp.live.turnover.LiveTurnover attribute), 45
 is_travel (pbpstats.resources.enhanced_pbp.stats_nba.turnover.StatsTurnover attribute), 51
 is_travel() (pbpstats.resources.enhanced_pbp.turnover.Turnover class method), 59
 is_turnover_placeholder (pbp- stats.resources.enhanced_pbp.rebound.Rebound attribute), 57
 J
 JumpBall (class in pbp- stats.resources.enhanced_pbp.jump_ball), 56
 L
 league (pbpstats.data_loader.stats_nba.base.StatsNbaLoaderBase attribute), 20
 league (pbpstats.data_loader.stats_nba.possessions.loader.StatsNbaPossessions attribute), 24
 league (pbpstats.resources.enhanced_pbp.start_of_period.StartOfPeriod attribute), 58
 league_id (pbpstats.data_loader.data_nba.schedule.web.DataNbaSchedule attribute), 14
 league_id (pbpstats.data_loader.stats_nba.base.StatsNbaLoaderBase attribute), 20
 league_id (pbpstats.data_loader.stats_nba.scoreboard.web.StatsNbaScoreboard attribute), 27
 league_url_part (pbp- stats.resources.enhanced_pbp.start_of_period.StartOfPeriod attribute), 58
 lineup_ids (pbpstats.resources.enhanced_pbp.enhanced_pbp_item.EnhancedPbpItem attribute), 53
 lineup_opponent_stats (pbp- stats.resources.possessions.possessions.Possessions attribute), 63

lineup_stats (pbp- stats.data_loader.live.pbp.web), 18
 stats.resources.possessions.possessions.PossessionLivePossessionFileLoader (class in pbp- stats.data_loader.live.possessions.file), 19
 attribute), 63
 LiveBoxscoreFileLoader (class in pbp- stats.data_loader.live.boxscore.file), 16
 LiveBoxscoreItem (class in pbp- stats.resources.boxscore.live_boxscore_item), 33
 LiveBoxscoreLoader (class in pbp- stats.data_loader.live.boxscore.loader), 15
 LiveBoxscoreWebLoader (class in pbp- stats.data_loader.live.boxscore.web), 16
 LiveEjection (class in pbp- stats.resources.enhanced_pbp.live.ejection), 39
 LiveEndOfPeriod (class in pbp- stats.resources.enhanced_pbp.live.end_of_period), 40
 LiveEnhancedPbpFactory (class in pbp- stats.resources.enhanced_pbp.live.enhanced_pbp_factory), 44
 LiveEnhancedPbpFileLoader (class in pbp- stats.data_loader.live.enhanced_pbp.file), 17
 LiveEnhancedPbpItem (class in pbp- stats.resources.enhanced_pbp.live.enhanced_pbp_item), 40
 LiveEnhancedPbpLoader (class in pbp- stats.data_loader.live.enhanced_pbp.loader), 16
 LiveEnhancedPbpWebLoader (class in pbp- stats.data_loader.live.enhanced_pbp.web), 17
 LiveFieldGoal (class in pbp- stats.resources.enhanced_pbp.live.field_goal), 41
 LiveFileLoader (class in pbp- stats.data_loader.live.file_loader), 19
 LiveFoul (class in pbp- stats.resources.enhanced_pbp.live.foul), 41
 LiveFreeThrow (class in pbp- stats.resources.enhanced_pbp.live.free_throw), 42
 LiveJumpBall (class in pbp- stats.resources.enhanced_pbp.live.jump_ball), 43
 LivePbpFileLoader (class in pbp- stats.data_loader.live.pbp.file), 18
 LivePbpItem (class in pbp- stats.resources.pbp.live_pbp_item), 61
 LivePbpLoader (class in pbp- stats.data_loader.live.pbp.loader), 17
 LivePbpWebLoader (class in pbp- stats.data_loader.live.pbp.web), 18
 stats.data_loader.live.possessions.loader), 18
 LivePossessionWebLoader (class in pbp- stats.data_loader.live.possessions.web), 19
 LiveRebound (class in pbp- stats.resources.enhanced_pbp.live.rebound), 43
 LiveReplay (class in pbp- stats.resources.enhanced_pbp.live.replay), 43
 LiveStartOfPeriod (class in pbp- stats.resources.enhanced_pbp.live.start_of_period), 43
 LiveSubstitution (class in pbp- stats.resources.enhanced_pbp.live.substitution), 44
 LiveTimeout (class in pbp- stats.resources.enhanced_pbp.live.timeout), 44
 LiveTurnover (class in pbp- stats.resources.enhanced_pbp.live.turnover), 44
 LiveViolation (class in pbp- stats.resources.enhanced_pbp.live.violation), 45
 LiveWebLoader (class in pbp- stats.data_loader.live.web_loader), 19
 load_data () (pbpstats.data_loader.data_nba.boxscore.file.DataNbaBoxscoreFileLoader method), 10
 load_data () (pbpstats.data_loader.data_nba.boxscore.web.DataNbaBoxscoreWebLoader method), 10
 load_data () (pbpstats.data_loader.data_nba.pbp.file.DataNbaPbpFileLoader method), 12
 load_data () (pbpstats.data_loader.data_nba.pbp.web.DataNbaPbpWebLoader method), 12
 load_data () (pbpstats.data_loader.data_nba.schedule.file.DataNbaScheduleFileLoader method), 14
 load_data () (pbpstats.data_loader.data_nba.schedule.web.DataNbaScheduleWebLoader method), 14
 load_data () (pbpstats.data_loader.live.boxscore.file.LiveBoxscoreFileLoader method), 16
 load_data () (pbpstats.data_loader.live.boxscore.web.LiveBoxscoreWebLoader method), 16
 load_data () (pbpstats.data_loader.live.pbp.file.LivePbpFileLoader method), 18
 load_data () (pbpstats.data_loader.live.pbp.web.LivePbpWebLoader method), 18
 load_data () (pbpstats.data_loader.stats_nba.boxscore.file.StatsNbaBoxscoreFileLoader method), 21
 load_data () (pbpstats.data_loader.stats_nba.boxscore.web.StatsNbaBoxscoreWebLoader method), 21

`method`), 21
`load_data()` (`pbpstats.data_loader.stats_nba.game_finder.file.StatsNbaGameFinderFileLoader` (pbp-
`method`), 26
`load_data()` (`pbpstats.data_loader.stats_nba.game_finder.web.StatsNbaGameFinderWebLoader` (pbp-
`method`), 26
`load_data()` (`pbpstats.data_loader.stats_nba.pbp.file.StatsNbaPbpFileLoader` (pbp-
`method`), 23
`load_data()` (`pbpstats.data_loader.stats_nba.pbp.web.StatsNbaPbpWebLoader` (pbp-
`method`), 23
`load_data()` (`pbpstats.data_loader.stats_nba.scoreboard.file.StatsNbaScoreboardFileLoader` (pbp-
`method`), 27
`load_data()` (`pbpstats.data_loader.stats_nba.scoreboard.web.StatsNbaScoreboardWebLoader` (pbp-
`method`), 27
`load_data()` (`pbpstats.data_loader.stats_nba.shots.file.StatsNbaShotsFileLoader` (pbp-
`method`), 28
`load_data()` (`pbpstats.data_loader.stats_nba.shots.web.StatsNbaShotsWebLoader` (pbp-
`method`), 28
`load_data()` (`pbpstats.data_loader.stats_nba.summary.file.StatsNbaSummaryFileLoader` (pbp-
`method`), 29
`load_data()` (`pbpstats.data_loader.stats_nba.summary.web.StatsNbaSummaryWebLoader` (pbp-
`method`), 29

M

`make_list_of_dicts()` (`pbpstats.data_loader.stats_nba.base.StatsNbaLoaderBase` (pbp-
`method`), 20
`make_list_of_dicts()` (`pbpstats.data_loader.stats_nba.shots.loader.StatsNbaShotsLoader` (pbp-
`method`), 27
`missed_shot` (`pbpstats.resources.enhanced_pbp.data_nba.rebound.DataRebound` (pbp-
`attribute`), 37
`missed_shot` (`pbpstats.resources.enhanced_pbp.live.rebound.LiveRebound` (pbp-
`attribute`), 43
`missed_shot` (`pbpstats.resources.enhanced_pbp.rebound.Rebound` (pbp-
`attribute`), 57
`missed_shot` (`pbpstats.resources.enhanced_pbp.stats_nba.rebound.StatsRebound` (pbp-
`attribute`), 49

N

`NbaEnhancedPbpLoader` (class in `pbpstats.data_loader.nba_enhanced_pbp_loader`), 30
`NbaPossessionLoader` (class in `pbpstats.data_loader.nba_possession_loader`), 30
`num_ft_for_trip` (`pbpstats.resources.enhanced_pbp.free_throw.FreeThrow` (pbp-
`attribute`), 56

P

`number_of_fta_for_foul` (`pbpstats.resources.enhanced_pbp.data_nba.foul.DataFoul` (pbp-
`attribute`), 36
`number_of_fta_for_foul` (`pbpstats.resources.enhanced_pbp.foul.Foul` (pbp-
`attribute`), 55

parent_object (pbp- (module), 11
 stats.data_loader.data_nba.pbp.loader.DataNbaPbpLoader (module), 10
 attribute), 12

parent_object (pbp- pbpstats.data_loader.data_nba.enhanced_pbp.web
 stats.data_loader.data_nba.possessions.loader.DataNbaPossessionsLoader
 attribute), 13 pbpstats.data_loader.data_nba.file_loader

parent_object (pbp- (module), 15
 stats.data_loader.data_nba.schedule.loader.DataNbaScheduleLoader (module), 12
 attribute), 14

parent_object (pbp- pbpstats.data_loader.data_nba.pbp.loader
 stats.data_loader.live.boxscore.loader.LiveBoxscoreLoader (module), 11
 attribute), 16 pbpstats.data_loader.data_nba.pbp.web

parent_object (pbp- (module), 12
 stats.data_loader.live.enhanced_pbp.loader.LiveEnhancedPbpLoader (module), 13
 attribute), 17

parent_object (pbp- pbpstats.data_loader.data_nba.possessions.loader
 stats.data_loader.live.pbp.loader.LivePbpLoader (module), 12
 attribute), 18 pbpstats.data_loader.data_nba.possessions.web

parent_object (pbp- (module), 13
 stats.data_loader.live.possessions.loader.LivePossessionsLoader (module), 14
 attribute), 19

parent_object (pbp- pbpstats.data_loader.data_nba.schedule.loader
 stats.data_loader.stats_nba.boxscore.loader.StatsNbaBoxscoreLoader (module), 13
 attribute), 21 pbpstats.data_loader.data_nba.schedule.web

parent_object (pbp- (module), 14
 stats.data_loader.stats_nba.enhanced_pbp.loader.StatsNbaEnhancedPbpLoader (module), 15
 attribute), 22

parent_object (pbp- pbpstats.data_loader.factory (module), 29
 stats.data_loader.stats_nba.game_finder.loader.StatsNbaGameFinderLoader (module), 16
 attribute), 25

parent_object (pbp- pbpstats.data_loader.live.boxscore.loader
 stats.data_loader.stats_nba.pbp.loader.StatsNbaPbpLoader (module), 15
 attribute), 23 pbpstats.data_loader.live.boxscore.web

parent_object (pbp- (module), 16
 stats.data_loader.stats_nba.possessions.loader.StatsNbaPossessionsLoader (module), 17
 attribute), 24

parent_object (pbp- pbpstats.data_loader.live.enhanced_pbp.loader
 stats.data_loader.stats_nba.scoreboard.loader.StatsNbaScoreboardLoader (module), 18
 attribute), 26 pbpstats.data_loader.live.enhanced_pbp.web

parent_object (pbp- (module), 17
 stats.data_loader.stats_nba.shots.loader.StatsNbaShotsLoader (module), 19
 attribute), 27

parent_object (pbp- pbpstats.data_loader.live.pbp.file (mod-
 stats.data_loader.stats_nba.summary.loader.StatsNbaSummaryLoader (module), 17
 attribute), 28 pbpstats.data_loader.live.pbp.loader

Pbp (class in pbpstats.resources.pbp.pbp), 61 (module), 17

pbpstats.client (module), 64 pbpstats.data_loader.live.pbp.web (mod-
 pbpstats.data_loader.data_nba.boxscore.file (module), 18
 (module), 10 pbpstats.data_loader.live.possessions.file

pbpstats.data_loader.data_nba.boxscore.loader (module), 19
 (module), 9 pbpstats.data_loader.live.possessions.loader

pbpstats.data_loader.data_nba.boxscore.web (module), 18
 (module), 10 pbpstats.data_loader.live.possessions.web

pbpstats.data_loader.data_nba.enhanced_pbp.file (module), 19

pbpstats.data_loader.live.web_loader (module), 19	pbpstats.data_loader.stats_nba.summary.loader (module), 28
pbpstats.data_loader.nba_enhanced_pbp_loader (module), 30	pbpstats.data_loader.stats_nba.summary.web (module), 29
pbpstats.data_loader.nba_possession_loader (module), 30	pbpstats.data_loader.stats_nba.web_loader (module), 29
pbpstats.data_loader.stats_nba.base (module), 20	pbpstats.objects.day (module), 30
pbpstats.data_loader.stats_nba.boxscore.file (module), 21	pbpstats.objects.game (module), 31
pbpstats.data_loader.stats_nba.boxscore.loader (module), 20	pbpstats.objects.season (module), 31
pbpstats.data_loader.stats_nba.boxscore.web (module), 21	pbpstats.resources.boxscore.boxscore (module), 32
pbpstats.data_loader.stats_nba.enhanced_pbp.file (module), 22	pbpstats.resources.boxscore.data_nba_boxscore_item (module), 33
pbpstats.data_loader.stats_nba.enhanced_pbp.loader (module), 21	pbpstats.resources.boxscore.live_boxscore_item (module), 33
pbpstats.data_loader.stats_nba.enhanced_pbp.web (module), 22	pbpstats.resources.boxscore.stats_nba_boxscore_item (module), 34
pbpstats.data_loader.stats_nba.file_loader (module), 29	pbpstats.resources.enhanced_pbp.data_nba.ejection (module), 34
pbpstats.data_loader.stats_nba.game_finder.file (module), 25	pbpstats.resources.enhanced_pbp.data_nba.end_of_per (module), 34
pbpstats.data_loader.stats_nba.game_finder.loader (module), 25	pbpstats.resources.enhanced_pbp.data_nba.enhanced_p (module), 34
pbpstats.data_loader.stats_nba.game_finder.web (module), 26	pbpstats.resources.enhanced_pbp.data_nba.enhanced_p (module), 35
pbpstats.data_loader.stats_nba.pbp.file (module), 23	pbpstats.resources.enhanced_pbp.data_nba.field_goa (module), 35
pbpstats.data_loader.stats_nba.pbp.loader (module), 23	pbpstats.resources.enhanced_pbp.data_nba.foul (module), 35
pbpstats.data_loader.stats_nba.pbp.web (module), 23	pbpstats.resources.enhanced_pbp.data_nba.free_throv (module), 36
pbpstats.data_loader.stats_nba.possessions.file (module), 24	pbpstats.resources.enhanced_pbp.data_nba.jump_ball (module), 37
pbpstats.data_loader.stats_nba.possessions.loader (module), 24	pbpstats.resources.enhanced_pbp.data_nba.rebound (module), 37
pbpstats.data_loader.stats_nba.possessions.web (module), 24	pbpstats.resources.enhanced_pbp.data_nba.replay (module), 38
pbpstats.data_loader.stats_nba.scoreboard.file (module), 26	pbpstats.resources.enhanced_pbp.data_nba.start_of_p (module), 38
pbpstats.data_loader.stats_nba.scoreboard.loader (module), 26	pbpstats.resources.enhanced_pbp.data_nba.substitut (module), 38
pbpstats.data_loader.stats_nba.scoreboard.web (module), 27	pbpstats.resources.enhanced_pbp.data_nba.timeout (module), 38
pbpstats.data_loader.stats_nba.shots.file (module), 27	pbpstats.resources.enhanced_pbp.data_nba.turnover (module), 39
pbpstats.data_loader.stats_nba.shots.loader (module), 27	pbpstats.resources.enhanced_pbp.data_nba.violation (module), 51
pbpstats.data_loader.stats_nba.shots.web (module), 28	pbpstats.resources.enhanced_pbp.ejection (module), 52
pbpstats.data_loader.stats_nba.summary.file (module), 28	pbpstats.resources.enhanced_pbp.end_of_period (module), 52
	pbpstats.resources.enhanced_pbp.enhanced_pbp (module), 52
	pbpstats.resources.enhanced_pbp.enhanced_pbp_item (module), 52

<code>stats.resources.boxscore.boxscore.Boxscore</code> attribute), 32	resource (<code>pbpstats.data_loader.data_nba.enhanced_pbp.loader.DataNbaPbpLoader</code> attribute), 11
player_stats (pbp- <code>stats.resources.possessions.possessions.Possessions</code> attribute), 63	resource (<code>pbpstats.data_loader.data_nba.pbp.loader.DataNbaPbpLoader</code> attribute), 12
player_team_map (pbp- <code>stats.resources.boxscore.boxscore.Boxscore</code> attribute), 32	resource (<code>pbpstats.data_loader.data_nba.possessions.loader.DataNbaPbpLoader</code> attribute), 13
Possession (class in pbp- <code>stats.resources.possessions.possession</code>), 62	resource (<code>pbpstats.data_loader.data_nba.schedule.loader.DataNbaScheduleLoader</code> attribute), 14
possession_has_timeout (pbp- <code>stats.resources.possessions.possession.Possession</code> attribute), 62	resource (<code>pbpstats.data_loader.live.boxscore.loader.LiveBoxscoreLoader</code> attribute), 16
possession_start_type (pbp- <code>stats.resources.possessions.possession.Possession</code> attribute), 62	resource (<code>pbpstats.data_loader.live.enhanced_pbp.loader.LiveEnhancedPbpLoader</code> attribute), 17
possession_stats (pbp- <code>stats.resources.possessions.possession.Possession</code> attribute), 62	resource (<code>pbpstats.data_loader.live.pbp.loader.LivePbpLoader</code> attribute), 18
Possessions (class in pbp- <code>stats.resources.possessions.possessions</code>), 63	resource (<code>pbpstats.data_loader.live.possessions.loader.LivePossessionLoader</code> attribute), 19
previous_possession_end_rebound_player_id (pbpstats.resources.possessions.possession.Possession attribute), 62	resource (<code>pbpstats.data_loader.stats_nba.boxscore.loader.StatsNbaBoxscoreLoader</code> attribute), 21
previous_possession_end_shooter_player_id (pbpstats.resources.possessions.possession.Possession attribute), 62	resource (<code>pbpstats.data_loader.stats_nba.enhanced_pbp.loader.StatsNbaPbpLoader</code> attribute), 22
previous_possession_end_steal_player_id (pbpstats.resources.possessions.possession.Possession attribute), 62	resource (<code>pbpstats.data_loader.stats_nba.game_finder.loader.StatsNbaGameFinder</code> attribute), 25
previous_possession_end_turnover_player_id (pbpstats.resources.possessions.possession.Possession attribute), 62	resource (<code>pbpstats.data_loader.stats_nba.pbp.loader.StatsNbaPbpLoader</code> attribute), 23
previous_possession_ending_event (pbp- <code>stats.resources.possessions.possession.Possession</code> attribute), 62	resource (<code>pbpstats.data_loader.stats_nba.possessions.loader.StatsNbaPbpLoader</code> attribute), 24
previous_possession_has_timeout (pbp- <code>stats.resources.possessions.possession.Possession</code> attribute), 62	resource (<code>pbpstats.data_loader.stats_nba.scoreboard.loader.StatsNbaScoreboardLoader</code> attribute), 26
Rebound (class in pbp- <code>stats.resources.enhanced_pbp.rebound</code>), 56	resource (<code>pbpstats.data_loader.stats_nba.shots.loader.StatsNbaShotsLoader</code> attribute), 27
rebound (pbpstats.resources.enhanced_pbp.field_goal.FieldGoal attribute), 54	resource (<code>pbpstats.data_loader.stats_nba.summary.loader.StatsNbaSummaryLoader</code> attribute), 28
rebounds (pbpstats.resources.enhanced_pbp.enhanced_pbp.EnhancedPbpItem attribute), 52	ruling_stands (pbp- <code>stats.resources.enhanced_pbp.data_nba.replay.DataReplay</code> attribute), 37
Replay (class in pbp- <code>stats.resources.enhanced_pbp.replay</code>), 57	ruling_stands (pbp- <code>stats.resources.enhanced_pbp.live.replay.LiveReplay</code> attribute), 43
resource (<code>pbpstats.data_loader.data_nba.boxscore.loader.DataNbaBoxscoreLoader</code> attribute), 10	ruling_stands (pbp- <code>stats.resources.enhanced_pbp.replay.Replay</code> attribute), 57
	ruling_stands (pbp- <code>stats.resources.enhanced_pbp.stats_nba.replay.StatsReplay</code> attribute), 49
	score_margin (pbp- <code>stats.resources.enhanced_pbp.enhanced_pbp_item.EnhancedPbpItem</code> attribute), 53
	Season (class in <code>pbpstats.objects.season</code>), 32
	season (<code>pbpstats.data_loader.stats_nba.base.StatsNbaLoaderBase</code> attribute), 20
	season_type (<code>pbpstats.data_loader.stats_nba.base.StatsNbaLoaderBase</code> attribute), 20

seconds_remaining (pbp- StatsFoul (class in pbp- stats.resources.enhanced_pbp.data_nba.enhanced_pbp_item.StatsEnhancedPbpItem (class in pbp- stats.resources.enhanced_pbp.stats_nba.foul), attribute), 35 47

seconds_remaining (pbp- StatsFreeThrow (class in pbp- stats.resources.enhanced_pbp.enhanced_pbp_item.EnhancedPbpItem (class in pbp- stats.resources.enhanced_pbp.stats_nba.free_throw), attribute), 53 48

seconds_remaining (pbp- StatsJumpBall (class in pbp- stats.resources.enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem (class in pbp- stats.resources.enhanced_pbp.stats_nba.jump_ball), attribute), 41 49

seconds_remaining (pbp- StatsNbaBoxscoreFileLoader (class in pbp- stats.resources.enhanced_pbp.stats_nba.enhanced_pbp_item.StatsEnhancedPbpItem (class in pbp- stats.resources.enhanced_pbp.stats_nba.boxscore.file), attribute), 47 21

seconds_since_previous_event (pbp- StatsNbaBoxscoreItem (class in pbp- stats.resources.boxscore.stats_nba_boxscore_item), StatsNbaBoxscoreLoader (class in pbp- stats.resources.enhanced_pbp.enhanced_pbp_item.EnhancedPbpItem (class in pbp- stats.resources.enhanced_pbp.stats_nba.enhanced_pbp_item), attribute), 53 21

self_reb (pbpstats.resources.enhanced_pbp.rebound.Rebound (class in pbp- stats.data_loader.stats_nba.boxscore.loader), attribute), 57 21

shot_data (pbpstats.resources.enhanced_pbp.field_goal.FieldGoal (class in pbp- stats.data_loader.stats_nba.boxscore.web), attribute), 54 21

shot_type (pbpstats.resources.enhanced_pbp.field_goal.FieldGoal (class in pbp- stats.data_loader.stats_nba.enhanced_pbp_factory), attribute), 54 46

shot_type (pbpstats.resources.enhanced_pbp.free_throw.FreeThrow (class in pbp- stats.resources.enhanced_pbp.stats_nba.enhanced_pbp_factory), attribute), 56 46

shot_value (pbpstats.resources.enhanced_pbp.data_nba.field_goal.FieldGoal (class in pbp- stats.data_loader.stats_nba.enhanced_pbp.file), attribute), 35 22

shot_value (pbpstats.resources.enhanced_pbp.field_goal.FieldGoal (class in pbp- stats.data_loader.stats_nba.enhanced_pbp.loader), attribute), 54 22

shot_value (pbpstats.resources.enhanced_pbp.free_throw.FreeThrow (class in pbp- stats.data_loader.stats_nba.enhanced_pbp.loader), attribute), 56 22

shot_value (pbpstats.resources.enhanced_pbp.live.field_goal.LiveFieldGoal (class in pbp- stats.data_loader.stats_nba.enhanced_pbp.web), attribute), 41 23

shot_value (pbpstats.resources.enhanced_pbp.stats_nba.field_goal.StatsFieldGoal (class in pbp- stats.data_loader.stats_nba.enhanced_pbp.web), attribute), 47 23

Shots (class in pbpstats.resources.shots.shots), 63 29

start_score_margin (pbp- StatsNbaGameFinderFileLoader (class in pbp- stats.data_loader.stats_nba.game_finder.file), StatsNbaGameFinderWebLoader (class in pbp- stats.data_loader.stats_nba.game_finder.web), attribute), 62 26

start_time (pbpstats.resources.possessions.possession.Possession (class in pbp- stats.data_loader.stats_nba.game_finder.web), attribute), 62 26

StartOfPeriod (class in pbp- StatsNbaGameFinderLoader (class in pbp- stats.data_loader.stats_nba.game_finder.loader), stats.resources.enhanced_pbp.start_of_period), 57 25

StatsEjection (class in pbp- StatsNbaGameFinderWebLoader (class in pbp- stats.data_loader.stats_nba.game_finder.web), stats.resources.enhanced_pbp.stats_nba.ejection), 45 26

StatsEndOfPeriod (class in pbp- StatsNbaGameItem (class in pbp- stats.resources.games.stats_nba_game_item), stats.resources.enhanced_pbp.stats_nba.end_of_period), 46 60

StatsEnhancedPbpItem (class in pbp- StatsNbaLoaderBase (class in pbp- stats.data_loader.stats_nba.base), stats.resources.enhanced_pbp.stats_nba.enhanced_pbp_item), 46 20

StatsFieldGoal (class in pbp- StatsNbaPbpItem (class in pbp- stats.data_loader.stats_nba.pbp.file), stats.resources.enhanced_pbp.stats_nba.field_goal), 47 23

StatsFieldGoal (class in pbp- StatsNbaPbpItem (class in pbp- stats.resources.pbp.stats_nba_pbp_item), stats.resources.pbp.stats_nba_pbp_item), 47 61

StatsNbaPbpLoader (class in <i>pbpstats.data_loader.stats_nba.pbp.loader</i>), 23	StatsTimeout (class in <i>pbpstats.resources.enhanced_pbp.stats_nba.timeout</i>), 50
StatsNbaPbpWebLoader (class in <i>pbpstats.data_loader.stats_nba.pbp.web</i>), 23	StatsTurnover (class in <i>pbpstats.resources.enhanced_pbp.stats_nba.turnover</i>), 50
StatsNbaPossessionFileLoader (class in <i>pbpstats.data_loader.stats_nba.possessions.file</i>), 24	StatsViolation (class in <i>pbpstats.resources.enhanced_pbp.stats_nba.violation</i>), 51
StatsNbaPossessionLoader (class in <i>pbpstats.data_loader.stats_nba.possessions.loader</i>), 24	stripped_descriptor (<i>pbpstats.resources.enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem</i> attribute), 41
StatsNbaPossessionWebLoader (class in <i>pbpstats.data_loader.stats_nba.possessions.web</i>), 24	stripped_sub_type (<i>pbpstats.resources.enhanced_pbp.live.enhanced_pbp_item.LiveEnhancedPbpItem</i> attribute), 41
StatsNbaScoreboardFileLoader (class in <i>pbpstats.data_loader.stats_nba.scoreboard.file</i>), 26	sub_type (<i>pbpstats.resources.enhanced_pbp.live.end_of_period.LiveEndOfPeriod</i> attribute), 40
StatsNbaScoreboardLoader (class in <i>pbpstats.data_loader.stats_nba.scoreboard.loader</i>), 26	sub_type (<i>pbpstats.resources.enhanced_pbp.live.start_of_period.LiveStartOfPeriod</i> attribute), 44
StatsNbaScoreboardWebLoader (class in <i>pbpstats.data_loader.stats_nba.scoreboard.web</i>), 27	Substitution (class in <i>pbpstats.resources.enhanced_pbp.substitution</i>), 58
StatsNbaShot (class in <i>pbpstats.resources.shots.stats_nba_shot</i>), 63	support_ruling (<i>pbpstats.resources.enhanced_pbp.data_nba.replay.DataReplay</i> attribute), 37
StatsNbaShotsFileLoader (class in <i>pbpstats.data_loader.stats_nba.shots.file</i>), 27	support_ruling (<i>pbpstats.resources.enhanced_pbp.live.replay.LiveReplay</i> attribute), 43
StatsNbaShotsLoader (class in <i>pbpstats.data_loader.stats_nba.shots.loader</i>), 27	support_ruling (<i>pbpstats.resources.enhanced_pbp.replay.Replay</i> attribute), 57
StatsNbaShotsWebLoader (class in <i>pbpstats.data_loader.stats_nba.shots.web</i>), 28	support_ruling (<i>pbpstats.resources.enhanced_pbp.stats_nba.replay.StatsReplay</i> attribute), 49
StatsNbaSummaryFileLoader (class in <i>pbpstats.data_loader.stats_nba.summary.file</i>), 28	T
StatsNbaSummaryLoader (class in <i>pbpstats.data_loader.stats_nba.summary.loader</i>), 28	team_items (<i>pbpstats.resources.boxscore.boxscore.Boxscore</i> attribute), 32
StatsNbaSummaryWebLoader (class in <i>pbpstats.data_loader.stats_nba.summary.web</i>), 29	team_stats (<i>pbpstats.resources.possessions.possessions.Possessions</i> attribute), 63
StatsNbaWebLoader (class in <i>pbpstats.data_loader.stats_nba.web_loader</i>), 29	TeamHasBackToBackPossessionsException, 24
StatsRebound (class in <i>pbpstats.resources.enhanced_pbp.stats_nba.rebound</i>), 49	Timeout (class in <i>pbpstats.resources.enhanced_pbp.timeout</i>), 58
StatsReplay (class in <i>pbpstats.resources.enhanced_pbp.stats_nba.replay</i>), 49	total_seconds (<i>pbpstats.resources.boxscore.live_boxscore_item.LiveBoxscoreItem</i> attribute), 33
StatsStartOfPeriod (class in <i>pbpstats.resources.enhanced_pbp.stats_nba.start_of_period</i>), 50	Turnover (class in <i>pbpstats.resources.enhanced_pbp.turnover</i>), 59
StatsSubstitution (class in <i>pbpstats.resources.enhanced_pbp.stats_nba.substitution</i>), 50	turnovers (<i>pbpstats.resources.enhanced_pbp.enhanced_pbp.EnhancedPbpItem</i> attribute), 52

V

`video_url` (*pbpstats.resources.enhanced_pbp.stats_nba.enhanced_pbp_item.StatsEnhancedPbpItem* attribute), [47](#)

`Violation` (class in *pbpstats.resources.enhanced_pbp.violation*), [59](#)

W

`winning_team` (*pbpstats.resources.enhanced_pbp.jump_ball.JumpBall* attribute), [56](#)