Package ‘sleeperapi’

June 1, 2024

Type  Package
Title  Wrapper Functions Around 'Sleeper' (Fantasy Sports) API
Version  1.1.1
Description  For those wishing to interact with the 'Sleeper' (Fantasy Sports) API (<https://docs.sleeper.com/>) without looking too much into its documentation (found at <https://docs.sleeper.com/>), this package offers wrapper functions around the available API calls to make it easier.
License  MIT + file LICENSE
Encoding  UTF-8
RoxygenNote  7.3.1
Imports  httr, jsonlite, plotly, purrr, scales, DT, shiny,
         htmlwidgets, shinyWidgets, RColorBrewer,
         htmltools, tidyr, stringr, rlang
Suggests  testthat (>= 2.0.0), covr
NeedsCompilation  no
Author  Nick Bultman [aut, cre, cph]
Maintainer  Nick Bultman <njbultman74@gmail.com>
Repository  CRAN
Date/Publication  2024-06-01 21:00:02 UTC

R topics documented:

get_all_nfl_players ............................................. 2
get_avatar_picture ............................................. 3
get_draft ......................................................... 4
get_draft_picks .................................................. 4
get_draft_pick_trades .......................................... 5
get_league ......................................................... 6
get_league_drafts .............................................. 6
get_league_users .............................................. 7
get_loser_playoff Bracket ...................................... 8
get_all_nfl_players

Gather NFL Player Data

Description

Gather all information concerning NFL players in list or data frame. Per Sleeper, this should be called once per day at most given the amount of data that it returns. By modifying the 'clean' parameter, either a data frame will be returned or a list. When 'clean' is TRUE, a data frame will be returned. Note that while most data is returned for the data frame, metadata will not be due to its dynamic nature. For example, metadata can be data frames on their own, so attempting to put them in a single data frame is not feasible. One might notice that a metadata column is present, but this is not properly cleaned since there could be NULLs, rookie years, injury overrides, or other miscellaneous columns that could add additional explanatory information. If either of these pieces will be problematic for you, it is recommended to keep 'clean' set to FALSE and parse the default list as you see fit.
Usage
get_all_nfl_players(clean = FALSE)

Arguments
   clean   Specifies whether a data frame or the default list will be returned (logical)

Value
Returns a list or data frame containing information about the NFL players.

Author(s)
Nick Bultman, <njbultman74@gmail.com>, December 2021

Examples
## Not run: get_all_nfl_players(clean = FALSE)
## Not run: get_all_nfl_players(clean = TRUE)

get_avatar_picture
Display Avatar

Description
Given an avatar ID, grab the picture (full or thumbnail) concerning that particular avatar. Note that this function downloads the image into your temporary directory to display it. Then, the function deletes the file before completing. If it would be preferred to keep the image at the temporary location, set the keep_image argument to TRUE (default is FALSE).

Usage
get_avatar_picture(avatar_id, type = "full", keep_image = FALSE)

Arguments
   avatar_id    Avatar ID generated by Sleeper (character)
   type         String that is either "full" or "thumbnail"
   keep_image   FALSE to delete the image after displaying it or TRUE to keep it

Value
Returns a picture displaying the avatar picture

Author(s)
Nick Bultman, <njbultman74@gmail.com>, October 2021
get_draft_picks

Examples

```r
## Not run: get_avatar_picture("c751b27d9158c1dd41bd33dc7e4bcdde",
    type = "full",
    keep_image = FALSE)
## End(Not run)
```

get_draft  

_Gather Draft Information_

**Description**

Given a draft ID, gather draft information for selection.

**Usage**

```r
get_draft(draft_id)
```

**Arguments**

- `draft_id`  
  Draft ID generated by Sleeper (integer)

**Value**

Returns a list containing information about the draft.

**Author(s)**

Nick Bultman, njbultman74@gmail.com, December 2021

**Examples**

```r
## Not run: get_draft(688281872463106048)
```

get_draft_picks  

_Gather Draft Picks_

**Description**

Given a draft ID, gather draft picks (along with other metadata concerning the picks).

**Usage**

```r
get_draft_picks(draft_id)
```
get_draft_pick_trades

Arguments

draft_id
Draft ID generated by Sleeper (numeric or character)

Value
Returns a data frame containing information about the draft picks.

Author(s)
Nick Bultman, <njbultman74@gmail.com>, December 2021

Examples
## Not run: get_draft_picks(688281872463106048)

get_draft_pick_trades
Gather Draft Pick Trades

Description
Given a draft ID, gather draft pick trades for that draft.

Usage
get_draft_pick_trades(draft_id)

Arguments

draft_id
Draft ID generated by Sleeper (numeric or character)

Value
Returns a data frame containing information about the draft pick trade(s).

Author(s)
Nick Bultman, <njbultman74@gmail.com>, December 2021

Examples
## Not run: get_draft_pick_trades(688281872463106048)
**get_league**  
*Gather League Data*

**Description**
Given a league ID, grab the data for that league. Includes roster positions, season type, if it is in season, and more.

**Usage**
```r
gt_league(league_id)
```

**Arguments**
- `league_id`  
  League ID generated by Sleeper (numeric or character)

**Value**
Returns a data frame containing information about the league.

**Author(s)**
Nick Bultman, <njbultman74@gmail.com>, September 2021

**Examples**
```r
# Not run: get_league(688281863499907072)
```

---

**get_league_drafts**  
*Gather League Draft Information*

**Description**
Given a league ID, gather draft information for that league.

**Usage**
```r
gt_league_drafts(league_id)
```

**Arguments**
- `league_id`  
  League ID generated by Sleeper (numeric or character)

**Value**
Returns a data frame containing draft information for that league.
get_league_users

Author(s)

Nick Bultman, <njbultman74@gmail.com>, December 2021

Examples

## Not run: get_league_drafts(688281863499907072)

---

get_league_users Gather User Data for League

Description

Given a league ID, grab the data concerning each user. This includes information about the users’ mascot, display name, avatar, if they are an owner, if they are a bot, and more.

Usage

get_league_users(league_id)

Arguments

league_id League ID generated by Sleeper (numeric or character)

Value

Returns a data frame containing information about the users in the league.

Author(s)

Nick Bultman, <njbultman74@gmail.com>, September 2021

Examples

## Not run: get_league_users(688281863499907072)
get_loser_playoff_bracket

Gather Loser Playoff Bracket

Description
Given a league ID, get the loser playoff bracket for the league.

Usage
get_loser_playoff_bracket(league_id)

Arguments
- league_id: League ID generated by Sleeper (numeric or character)

Value
Returns a data frame containing playoff information for the loser bracket.

Author(s)
Nick Bultman, <njbultman74@gmail.com>, December 2021

Examples
## Not run: get_loser_playoff_bracket(688281863499907072)

get_main_data

Generate Main Data

Description
Given the league ID, generate a main data set that can be used for generating quick insights. This includes teams, fantasy points, avatars, divisions, owner IDs, and more.

Usage
get_main_data(league_id)

Arguments
- league_id: League ID generated by Sleeper (numeric).
**get_matchups**

**Value**

Returns a data frame containing roster, league user, and overall league information.

**Author(s)**

Nick Bultman, <njbultman74@gmail.com>, December 2023

**Examples**

```r
## Not run: get_main_data(688281863499907072)
```

---

**get_matchups**  
*Gather Matchup Data in a League for a Week*

**Description**

Given a league ID and week number, grab the matchup data, including player points, total points, starting rosters, and more.

**Usage**

```r
get_matchups(league_id, week)
```

**Arguments**

- `league_id` League ID generated by Sleeper (numeric or character)
- `week` Matchup week (numeric)

**Value**

Returns a data frame containing information about the matchups.

**Author(s)**

Nick Bultman, <njbultman74@gmail.com>, September 2021

**Examples**

```r
## Not run: get_matchups(688281863499907072, 2)
## Not run: get_matchups("688281863499907072", 1)
```
get_rosters  
Gather Roster Data for League

Description

Given a league ID, grab the data concerning each roster in that league. This includes information with the roster wins, losses, ties, fantasy points, and more.

Usage

get_rosters(league_id)

Arguments

league_id  
League ID generated by Sleeper (numeric or character)

Value

Returns a data frame containing information about the rosters in the league.

Author(s)

Nick Bultman, <njbultman74@gmail.com>, September 2021

Examples

## Not run: get_rosters(688281863499907072)

get_sport_state  
Gather Sport State

Description

Given a sport abbreviation (NFL, NBA, etc.), return information about the current sport state.

Usage

get_sport_state(sport)

Arguments

sport  
Sport that you would like to query (nfl, nba, lcs, etc.) (character)

Value

Returns a list containing information about the sport state.
get_traded_picks

Author(s)

Nick Bultman, <njbultman74@gmail.com>, December 2021

Examples

## Not run: get_sport_state("nfl")

-----------------------------------------------------------------------------------

get_traded_picks Gather League Trades

Description

Given a league ID, grab the trades that have occurred.

Usage

get_traded_picks(league_id)

Arguments

league_id League ID generated by Sleeper (numeric or character)

Value

Returns a list containing trades.

Author(s)

Nick Bultman, <njbultman74@gmail.com>, September 2021

Examples

## Not run: get_traded_picks(688281863499907072)
get_transactions  
*Gather League Transactions for Specific Round*

**Description**

Given a league ID and round, grab the transaction data concerning that particular league. This includes adds or drops, whether it was a free agent or a waiver, notes, and more.

**Usage**

```r
get_transactions(league_id, round)
```

**Arguments**

- `league_id` League ID generated by Sleeper (numeric or character)
- `round` round of transactions to gather (can also be thought of as week depending on when your league processes transactions) (numeric)

**Value**

Returns a data frame containing information about the round of transactions for that league.

**Author(s)**

Nick Bultman, <njbultman74@gmail.com>, September 2021

**Examples**

```r
## Not run: get_transactions(688281863499907072, 2)
```

---

get_trending_players  
*Gather Trending Player Information*

**Description**

Given a sport and type (add or drop), return the top trending players for that selection. More specifically, the player ID and count are returned in a data frame, allowing one to see which players have been added/dropped the most given the particular lookback period (default lookback period of one day).

**Usage**

```r
get_trending_players(sport, type, lookback_hours = 24, limit = 25)
```
get_user

Arguments

<table>
<thead>
<tr>
<th>sport</th>
<th>Sport that you would like to query (nfl, nba, etc.) (character)</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>Either &quot;add&quot; or &quot;drop&quot; (character)</td>
</tr>
<tr>
<td>lookback_hours</td>
<td>Number of hours to look back (numeric). Default is 24</td>
</tr>
<tr>
<td>limit</td>
<td>Number of results you would like (numeric). Default is 25</td>
</tr>
</tbody>
</table>

Value

Returns a data frame containing information (player ID and count) about the trending players.

Author(s)

Nick Bultman, <njbultman74@gmail.com>, December 2021

Examples

```r
## Not run: get_trending_players("nfl", "add", 24, 25)
```

get_user

Gather User Data

Description

Given a user ID, grab the data concerning that particular user. This includes username, user ID, display name, and more.

Usage

```r
get_user(user_id)
```

Arguments

| user_id    | User ID generated by Sleeper (numeric or character) |

Value

Returns a list containing information about the user.

Author(s)

Nick Bultman, <njbultman74@gmail.com>, September 2021

Examples

```r
## Not run: get_user(688556535013502976)
```
get_user_drafts  
*Gather User Draft Information*

**Description**

Given a user ID, sport abbreviation, and season number, gather draft information for that selection.

**Usage**

```r
get_user_drafts(user_id, sport, season)
```

**Arguments**

- `user_id`: User ID generated by Sleeper (numeric)
- `sport`: Sport that you would like to query ("nfl" only supported currently) (character)
- `season`: Season that you would like to select (numeric)

**Value**

Returns a data frame containing information about the draft(s).

**Author(s)**

Nick Bultman, <njbultman74@gmail.com>, December 2021

**Examples**

```r
## Not run: get_user_drafts(68856535013502976, "nfl", 2021)
```

get_user_leagues  
*Gather Leagues for User*

**Description**

Given a user ID and optional sport and season, grab the league data concerning that particular user. This includes total rosters, scoring settings, roster positions, and more.

**Usage**

```r
get_user_leagues(
  user_id,
  sport = "nfl",
  season = substr(Sys.Date(), start = 1, stop = 4)
)
```
get_weekly_matchups

Arguments

user_id  User ID generated by Sleeper (numeric or character)
sport  Sport for league information (character)
season  Season for league information (numeric or character)

Value

Returns a data frame containing information about the user.

Author(s)

Nick Bultman, <njbultman74@gmail.com>, September 2021

Examples

## Not run: get_user_leagues(688556535013502976)
## Not run: get_user_leagues("688556535013502976", sport = "nfl", season = "2021")

get_weekly_matchups

Gather All Weekly Matchup Data Available for League

Description

Given a league ID and the type of data (all, team, or player data), grab the matchup data for each week that matchup data is available.

Usage

get_weekly_matchups(league_id, type = "all")

Arguments

league_id  League ID generated by Sleeper (numeric or character)
type  Type of data returned ("all", "team", "player") (string).

Value

If type is "all", will return a list containing data frames for both team and player data. Otherwise, will return team or player data depending on what is specified in the type argument. Note that with the player data, the aggregated player points could differ from the total team points due to the custom (manual) points that can be present.

Author(s)

Nick Bultman, <njbultman74@gmail.com>, April 2024
Examples

```r
## Not run: get_weekly_matchups(688281863499907072)
## Not run: get_weekly_matchups(688281863499907072, type = "player")
```

---

`get_winner_playoff_bracket`

*Gather Winner Playoff Bracket*

### Description

Given a league ID, get the winner playoff bracket for the league.

### Usage

```r
get_winner_playoff_bracket(league_id)
```

### Arguments

- `league_id`: League ID generated by Sleeper (numeric or character)

### Value

Returns a data frame containing playoff information for the winner bracket.

### Author(s)

Nick Bultman, <njbultman74@gmail.com>, December 2021

### Examples

```r
## Not run: get_winner_playoff_bracket(688281863499907072)
```

---

`plot_generate_nfl_player_data`

*Generate NFL Player Information for Plotting*

### Description

When using some of the plotting functions, it is recommended to run this function before any plotting begins to ensure the latest player information is present with the package. Per Sleeper, gathering these data should only be done once per day, which is why this is not recommended to implement a call every time to gather player information when a plot is generated. This function will refresh the latest NFL player information and place it in the appropriate place for the plotting functions to grab. Moreover, it will keep that information stored for the duration of the R session.
Usage

plot_generate_nfl_player_data()

Value

Returns a message stating if the data refresh was successful or not.

Author(s)

Nick Bultman, <njbultman74@gmail.com>, September 2023

Examples

## Not run: plot_generate_nfl_player_data()

---

plot_league_dashboard  Plot League Dashboard

Description

Given the league ID, generate a dashboard with various plots/tables of the current league. Note that since this is a dashboard, your R session will continuously run until the console is stopped. Stoppage can be achieved by pressing Control + C on the keyboard.

Usage

plot_league_dashboard(league_id)

Arguments

league_id       League ID generated by Sleeper (numeric).

Value

Returns a dashboard of the current league state.

Author(s)

Nick Bultman, <njbultman74@gmail.com>, March 2024

Examples

## Not run: plot_league_dashboard(688281863499907072)
plot_league_information_table

Plot League Information Table

Description

Given the league ID, generate a table showing team names and their associated statistics (wins, losses, points for/against, etc.).

Usage

plot_league_information_table(
  league_id,
  font_color = "inherit",
  win_loss_fill = "RdYlGn",
  fpts_for_fill = "lightgreen",
  fpts_against_fill = "#f68383"
)

Arguments

league_id League ID generated from Sleeper (numeric).
font_color Font color, hex code or name (string).
win_loss_fill String describing the brewer.pal color palette selection. Can see all options through RColorBrewer::display.brewer.all() (string).
fpts_for_fill Bar color, name or hex, for each fantasy points for cell by team (string).
fpts_against_fill Bar color, name or hex, for each fantasy points against cell by team (string).

Value

Returns a table containing team names and their associated statistics.

Author(s)

Nick Bultman, <njbultman74@gmail.com>, January 2024

Examples

## Not run: plot_league_information_table(688281863499907072)
## Not run: plot_league_information_table(688281863499907072, font_color = "#ee5050")
**plot_league_streaks_table**

*Plot League Streaks Table*

**Description**

Given the league ID, generate a table showing display names and their associated streaks (2W, 5L, etc.).

**Usage**

```r
plot_league_streaks_table(
  league_id,
  font_color = "inherit",
  win_streak_font_color = "lightgreen",
  lose_streak_font_color = "#f68383"
)
```

**Arguments**

- **league_id**: League ID generated from Sleeper (numeric).
- **font_color**: Header font color, hex code or name (string).
- **win_streak_font_color**: Font color, hex code or name, for teams with winning streak (string).
- **lose_streak_font_color**: Font color, hex code or name, for teams with losing streak (string).

**Value**

Returns a table containing display names and their streaks.

**Author(s)**

Nick Bultman, <njbultman74@gmail.com>, January 2024

**Examples**

```r
## Not run: plot_league_streaks_table(688281863499907072)
## Not run: plot_league_streaks_table(688281863499907072, font_color = "red")
```
**plot_nfl_player_data_table**

*Plot NFL Player Information in Table*

### Description

This function will generate a table containing general information (name, age, college, etc.) concerning all the NFL players that Sleeper has data on.

### Usage

```r
plot_nfl_player_data_table(font_color = "inherit")
```

### Arguments

- **font_color**: Font color, name or hex (string).

### Value

Returns a table showing general NFL player information.

### Author(s)

Nick Bultman, <njbultman74@gmail.com>, March 2024

### Examples

```r
## Not run: plot_nfl_player_data_table()
## Not run: plot_nfl_player_data_table(font_color = "green")
```

---

**plot_nfl_player_high_school_state**

*Plot NFL Player High School State Information*

### Description

This function will generate a plot showing counts of NFL players by high school state. Colors can be manually defined for high and low values of each state’s count.

### Usage

```r
plot_nfl_player_high_school_state(
  title = "&lt;b&gt;NFL Players by High School State&lt;/b&gt;",
  high_fill = "lightgreen",
  low_fill = "white"
)
```
Arguments

- title: Title for plot (string).
- high_fill: Fill color, name or hex, for high value states (string).
- low_fill: Fill color, name or hex, for low value states (string).

Value

Returns a plot containing information about counts for NFL players by their high school state.

Author(s)

Nick Bultman, <njbultman74@gmail.com>, March 2024

Examples

```r
## Not run: plot_nfl_player_high_school_state()
## Not run: plot_nfl_player_high_school_state(title = "Test")
```

---

plot_nfl_top_colleges  Plot NFL Player College Information

Description

This function will generate a bar plot showing counts of NFL players by college. The total number of colleges returned is specified by the user.

Usage

```r
plot_nfl_top_colleges(
  title = "<b>NFL Players by College</b>",
  tick_color = "black",
  number = 10,
  fill_color = "black"
)
```

Arguments

- title: Title for plot (string).
- tick_color: Color for tick labels (string).
- number: Number of colleges to show (numeric).
- fill_color: Bar color, number or hex (string).

Value

Returns a plot containing information about counts for NFL players by their college.
plot_regular_season_rankings

Author(s)
Nick Bultman, <njbultman74@gmail.com>, March 2024

Examples

## Not run: plot_nfl_player_college()
## Not run: plot_nfl_player_college(title = "TestTitle", tick_color = "green", number = 5)

---

plot_regular_season_rankings

Plot Regular Season League Rankings

Description

Given the league ID, plot the regular season league rankings. This is done by assigning positive points to wins and negative points to losses and then also adding in fantasy points for. The calculation can be seen in the get_main_data function.

Usage

plot_regular_season_rankings(
  league_id,
  title = "<b>Regular Season Rankings</b>",
  tick_color = "black",
  rank_high_fill = "lightgreen",
  rank_low_fill = "#f68383"
)

Arguments

league_id  League ID assigned by Sleeper (numeric).
title      Title for plot, which can include HTML formatting (string).
tick_color Font color, name or hex, for display names (string).
rank_high_fill  Bar color, name or hex, for highest rank (string).
rank_low_fill   Bar color, name or hex, for lowest rank (string).

Value

Returns a plot containing the regular season league rankings.

Author(s)
Nick Bultman, <njbultman74@gmail.com>, January 2024
plot_trending_players  

**Plot Trending NFL Player Information**

**Description**

Given the number of hours to look back, a limit of how many players to return, a color for labels, and a title, display a figure showing the top NFL additions and drops.

**Usage**

```r
plot_trending_players(
  lookback_hours = 24,
  limit = 10,
  tick_color = "black",
  title = "Trending Adds/Drops",
  drop_fill = "#f68383",
  add_fill = "lightgreen"
)
```

**Arguments**

- `lookback_hours`  Number of hours to look back (numeric).
- `limit` Number of results you would like (numeric).
- `tick_color` Font color, name or hex, for display names (string).
- `title` Plot title - default is "Trending Adds/Drops" (string).
- `drop_fill` Bar color, name or hex, for drops (string).
- `add_fill` Bar color, name or hex, for adds (string).

**Value**

Returns a plot containing information (name and count) about the trending players.

**Author(s)**

Nick Bultman, <njbultman74@gmail.com>, September 2023

**Examples**

```r
## Not run: plot_trending_players(lookback_hours = 24, limit = 10)
## Not run: plot_trending_players(lookback_hours = 24, limit = 10, tick_color = "white")
```
plot_trending_players_table

*Plot Trending NFL Player Information in Table*

**Description**

Given the number of hours to look back, and a limit of how many players to return, display a figure showing the top additions and drops in a table.

**Usage**

```r
plot_trending_players_table(
    lookback_hours = 24,
    limit = 10,
    font_color = "inherit",
    drop_fill = "#f68383",
    add_fill = "lightgreen"
)
```

**Arguments**

- `lookback_hours`: Number of hours to look back (numeric).
- `limit`: Number of players returned for add/drop. Should be less than 50 (numeric).
- `font_color`: Font color, name or hex (string).
- `drop_fill`: Bar color, name or hex, for drops (string).
- `add_fill`: Bar color, name or hex, for adds (string).

**Value**

Returns a plot containing information (name and count) about the trending NFL players.

**Author(s)**

Nick Bultman, <njbultman74@gmail.com>, February 2024

**Examples**

```r
# Not run: plot_trending_players_table(lookback_hours = 24, limit = 10)
# Not run: plot_trending_players_table(lookback_hours = 24, limit = 10, font_color = "white")
```
plot_user_fantasy_points_against  
Plot User Fantasy Points Against

Description

Given the league ID and display name, plot the current total fantasy points against that user while also plotting the current total fantasy points against the rest of the league in a different color (assuming the colors are not set to the same by the user).

Usage

plot_user_fantasy_points_against(
    league_id, 
    display_name, 
    title = "<b>Total Fantasy Points Against by User</b>", 
    tick_color = "black", 
    user_fill = "#f68383", 
    other_fill = "lightgrey"
)

Arguments

- **league_id**: League ID assigned by Sleeper (numeric).
- **display_name**: Display name created by user (string).
- **title**: Title for plot, which can include HTML formatting (string).
- **tick_color**: Font color, name or hex, for display names (string).
- **user_fill**: Bar color, name or hex, for display name specified (string).
- **other_fill**: Bar color, name or hex, for other display names (string).

Value

Returns a plot containing the total fantasy points against by user.

Author(s)

Nick Bultman, <njbultman74@gmail.com>, March 2024

Examples

```r
## Not run: plot_user_fantasy_points_against(688281863499907072, "njbultman74")
```
Description

Given the league ID and display name, plot the current total fantasy points for, against, and differential between the two together in one bar chart.

Usage

```r
plot_user_fantasy_points_differential(
  league_id,
  display_name,
  title = paste0("<b>Fantasy Point Differential: ", display_name, "</b>"),
  tick_color = "black",
  fill_points_for = "lightgreen",
  fill_points_against = "#f68383"
)
```

Arguments

- `league_id`: League ID assigned by Sleeper (numeric).
- `display_name`: Display name created by user (string).
- `title`: Title for plot, which can include HTML formatting (string).
- `tick_color`: Font color, name or hex, for display names (string).
- `fill_points_for`: Bar color, name or hex, for bar containing fantasy points for (string).
- `fill_points_against`: Bar color, name or hex, for bar containing fantasy points against (string).

Value

Returns a plot containing the total fantasy points differential by user.

Author(s)

Nick Bultman, <njbultman74@gmail.com>, March 2024

Examples

```r
## Not run: plot_user_fantasy_points_differential(688281863499907072, "njbultman74")
```
Description

Given the league ID and display name, plot the current total fantasy points for that user while also plotting the current total fantasy points for the rest of the league, with the specified user’s bar colored in a different color than the others (assuming the colors are not set to the same by the user).

Usage

```r
plot_user_fantasy_points_for(
  league_id,
  display_name,
  title = "<b>Total Fantasy Points For by User</b>",
  tick_color = "black",
  user_fill = "#f68383",
  other_fill = "lightgrey"
)
```

Arguments

- `league_id`: League ID assigned by Sleeper (numeric).
- `display_name`: Display name created by user (string).
- `title`: Title for plot, which can include HTML formatting (string).
- `tick_color`: Font color, name or hex, for display names (string).
- `user_fill`: Bar color, name or hex, for display name specified (string).
- `other_fill`: Bar color, name or hex, for other display names (string).

Value

Returns a plot containing the total fantasy points for by user.

Author(s)

Nick Bultman, <njbultman74@gmail.com>, March 2024

Examples

```r
## Not run: plot_user_fantasy_points_for(688281863499907072, "njbultman74")
```
plot_user_waiver_budget

Plot User Waiver Budget

Description

Given the league ID and display name, plot the current waiver budget for the user (meaning total waiver budget available versus how much the user has remaining).

Usage

plot_user_waiver_budget(
  league_id,
  display_name,
  title = "&lt;b&gt; Waiver Budget&lt;/b&gt;",
  tick_color = "black",
  budget_total_line_color = "black",
  fill_color = "lightgreen"
)

Arguments

league_id      League ID assigned by Sleeper (numeric).
display_name  Display name created by user (string).
title          Title for plot, which can include HTML formatting (string).
tick_color     Font color, name or hex, for display names (string).
budget_total_line_color
                Line color for max budget, name or hex (string).
fill_color     Fill color for bar, name or hex (string).

Value

Returns a plot containing the waiver budget status.

Author(s)

Nick Bultman, <njbultman74@gmail.com>, January 2024

Examples

## Not run: plot_user_waiver_budget(688281863499907072, "njbultman74")
Description

Given the league ID and display name, plot the total points scored by week for the display name and colored by whether the result was a win or a loss. The x-axis labels show the opposing display name for that specific week.

Usage

```r
plot_user_weekly_matchups(
  league_id,
  display_name,
  title = paste0("<b>", display_name, ": Weekly Matchups, Filled by Win/Loss</b>"),
  tick_color = "inherit",
  win_fill = "lightgreen",
  lose_fill = "#f68383"
)
```

Arguments

- `league_id`: League ID assigned by Sleeper (numeric).
- `display_name`: Display name created by user (string).
- `title`: Title for plot, which can include HTML formatting (string).
- `tick_color`: Font color, name or hex, for weeks (string).
- `win_fill`: Fill color, name or hex, for wins (string).
- `lose_fill`: Fill color, name or hex, for losses (string).

Value

Returns a plot containing the display name’s weekly matchups.

Author(s)

Nick Bultman, <njbultman74@gmail.com>, May 2024

Examples

```r
## Not run: plot_user_weekly_matchups(688281863499907072, "datcommish")
```
Description

Given the league ID and display name, plot the total points scored by week for each player ever associated with the display name.

Usage

```r
plot_user_weekly_players(
  league_id,
  display_name,
  title = paste0("<b">", display_name, ": Player Points by Week</b">"),
  tick_color = "inherit"
)
```

Arguments

- `league_id`: League ID assigned by Sleeper (numeric).
- `display_name`: Display name created by user (string).
- `title`: Title for plot, which can include HTML formatting (string).
- `tick_color`: Font color, name or hex, for players (string).

Value

Returns a plot containing the display name's weekly players and their respective points for each week.

Author(s)

Nick Bultman, <njbultman74@gmail.com>, May 2024

Examples

```r
## Not run: plot_user_weekly_players(688281863499907072, "datcommish")
```
Index

* avatar
  * get_avatar_picture, 3
* college
  * plot_nfl_top_colleges, 21
* dashboard
  * plot_league_dashboard, 17
* data
  * plot_generate_nfl_player_data, 16
* draft
  * get_draft, 4
  * get_draft_pick_trades, 5
  * get_draft_picks, 4
  * get_league_drafts, 6
  * get_user_drafts, 14
* fantasy
  * plot_user_fantasy_points_against, 25
  * plot_user_fantasy_points_differential, 26
  * plot_user_fantasy_points_for, 27
* full
  * get_main_data, 8
* general
  * plot_nfl_player_data_table, 20
* high
  * plot_nfl_player_high_school_state, 20
* information
  * plot_league_information_table, 18
  * plot_league_streaks_table, 19
* league
  * get_league_drafts, 6
  * get_league_users, 7
  * get_loser_playoff_bracket, 8
  * get_main_data, 8
  * get_matchups, 9
  * get_rosters, 10
  * get_traded_picks, 11
  * get_transactions, 12
  * get_user_leagues, 14
  * get_weekly_matchups, 15
  * get_winner_playoff_bracket, 16
  * plot_league_dashboard, 17
  * plot_league_information_table, 18
  * plot_league_streaks_table, 19
  * plot_regular_season_rankings, 22
  * plot_user_fantasy_points_against, 25
  * plot_user_fantasy_points_differential, 26
  * plot_user_fantasy_points_for, 27
  * plot_user_weekly_matchups, 29
  * plot_user_weekly_players, 30
* loser
  * get_loser_playoff_bracket, 8
* main
  * get_main_data, 8
* matchups
  * get_matchups, 9
  * get_weekly_matchups, 15
  * plot_user_weekly_matchups, 29
  * plot_user_weekly_players, 30
* nfl
  * get_all_nfl_players, 2
  * plot_generate_nfl_player_data, 16
* picks
  * get_draft_picks, 4
* pick
  * get_draft_pick_trades, 5
* picture
  * get_avatar_picture, 3
* players
  * get_all_nfl_players, 2
  * get_trending_players, 12
  * plot_generate_nfl_player_data, 16
  * plot_nfl_player_data_table, 20
  * plot_nfl_player_high_school_state,
INDEX

plot_nfl_top_colleges, 21
plot_trending_players, 23
plot_trending_players_table, 24
plot_user_weekly_players, 30

* playoff
  get_loser_playoff_bracket, 8
  get_winner_playoff_bracket, 16

* plot
  plot_league_dashboard, 17

* points
  plot_user_fantasy_points_against, 25
  plot_user_fantasy_points_differential, 26
  plot_user_fantasy_points_for, 27

* rankings
  plot_regular_season_rankings, 22
  plot_user_waiver_budget, 28

* refresh
  plot_generate_nfl_player_data, 16

* regular
  plot_regular_season_rankings, 22
  plot_user_waiver_budget, 28

* rosters
  get_rosters, 10

* roster
  get_main_data, 8

* school
  plot_nfl_player_high_school_state, 20

* season
  plot_regular_season_rankings, 22
  plot_user_waiver_budget, 28

* sport
  get_sport_state, 10

* state
  get_sport_state, 10
  plot_nfl_player_high_school_state, 20

* statistics
  plot_league_information_table, 18
  plot_league_streaks_table, 19

* trades
  get_draft_pick_trades, 5
  get_traded_picks, 11

* transactions
  get_transactions, 12

* trending
  get_trending_players, 12
  plot_trending_players, 23
  plot_trending_players_table, 24

* users
  get_league, 6
  get_league_users, 7
  get_rosters, 10
  get_user, 13

* user
  get_league, 6
  get_main_data, 8
  get_user, 13
  get_user_drafts, 14
  get_user_leagues, 14
  plot_user_fantasy_points_against, 25
  plot_user_fantasy_points_differential, 26
  plot_user_fantasy_points_for, 27
  plot_user_weekly_matchups, 29
  plot_user_weekly_players, 30

* weekly
  plot_user_weekly_matchups, 29
  plot_user_weekly_players, 30

* winner
  get_winner_playoff_bracket, 16

get_all_nfl_players, 2
get_avatar_picture, 3
get_draft, 4
get_draft_pick_trades, 5
get_draft_picks, 4
get_league, 6
get_league_drafts, 6
get_league_users, 7
get_loser_playoff_bracket, 8
get_main_data, 8
get_matchups, 9
get_rosters, 10
get_sport_state, 10
get_traded_picks, 11
get_transactions, 12
get_trending_players, 12
get_user, 13
get_user_drafts, 14
get_user_leagues, 14
get_weekly_matchups, 15
get_winner_playoff_bracket, 16
plot_generate_nfl_player_data, 16
plot_league_dashboard, 17
plot_league_information_table, 18
plot_league_streaks_table, 19
plot_nfl_player_data_table, 20
plot_nfl_player_high_school_state, 20
plot_nfl_top_colleges, 21
plot_regular_season_rankings, 22
plot_trending_players, 23
plot_trending_players_table, 24
plot_user_fantasy_points_against, 25
plot_user_fantasy_points_differential, 26
plot_user_fantasy_points_for, 27
plot_user_waiver_budget, 28
plot_user_weekly_matchups, 29
plot_user_weekly_players, 30