Package ‘matchbook’

October 13, 2022

Type Package

Title Wrapper for the 'Matchbook' API

Version 1.0.7

Date 2017-10-05

Author Niall Fitzgerald

Maintainer Niall Fitzgerald <nfitzgerald@xanadu.ie>

URL https://github.com/xanadunf/matchbook

BugReports https://github.com/xanadunf/matchbook/issues

Description Provides a wrapper for the some basic functionality around the 'Matchbook' <http://www.matchbook.com> REST API. It features calls to get events, markets and runners in data frame format. It features functions for bet placement and position management and also allows reporting of settled bet transactions. Note: this package uses the back-lay format. The default for odds type, currency and language are set according to those of the registered user account.

Depends R (>= 3.0)

Suggests testthat, httr, jsonlite

License MIT + file LICENSE

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-10-05 12:42:15 UTC

R topics documented:

`mb_bet_cancel` *Perform a Bet Cancel Action*

Description

This function provides bet cancellation functionality. It is possible to cancel a single bet by entering in a single value for the bet_id parameter. It is also possible to cancel multiple bets at once by passing a vector of the bet_id parameter. It is also possible to cancel all bets from a given market, event or runner by entering the corresponding ids. NOTE: bets with status 'matched' or 'revised' can not be cancelled.

Usage

```
mb_bet_cancel(session_data, bet_id = NULL, event_id = NULL,
  market_id = NULL, runner_id = NULL, cancel_all = FALSE)
```

Arguments

`session_data`	A session object returned from a successful mb_login attempt. It contains details about your user preferences and security details.
`bet_id`	The bet_id or vector of bet_ids that you want to cancel.
`event_id`	The event_id or vector of event_ids that you want to cancel.
`market_id`	The market_id or vector of market_ids that you want to cancel.
`runner_id`	The runner_id or vector of runner_ids that you want to cancel.
`cancel_all`	Boolean variable. Parameter to allow cancellation of all bets on all events/markets/runners. Default is FALSE.

Value

The status and details of your bet cancellations are returned. The data frame has the following fields:

id the bet id

event-id the event id on which the original bet was placed

event-name the name of the event on which the original bet was placed

event-id the event id on which the original bet was placed

market-id the market id on which the original bet was placed

market-name the name of the market on which the original bet was placed

runner-id the runner id on which the original bet was placed

runner-name the name of the runner on which the original bet was placed

temp-id the temporary id of the update

exchange-type the exchange type. This should always be 'back-lay'

side the side the bet was placed on

odds the odds the bet was placed on

odds-type the odds-type of the odds field

decimal-odds the decimal version of the odds

stake the stake placed

potential-profit the potential profit if the matched component of this wager is successful

remaining-potential-profit the potential profit if the un-matched component of this wager is first matched and then has a successful outcome

currency The currency the bet stake was placed with

created-at The date the bet was placed

status The bet status. Status 'open' indicates an unmatched bet, 'matched' indicates a fully matched bet, 'cancelled' indicates a cancelled bet. For bets with status='open', the 'stake' and 'remaining' fields are key to determining the exact status. If the 'remaining' value is less than 'stake' but greater than zero, then the bet has been partially matched for a 'stake'-'remaining' amount. If the bet is fully un-matched, then the 'stake' and 'remaining' values will be equal.

If no bets have been cancelled the 'offers' object will be an empty list.

See Also

mb_get_bets,mb_bet_place,mb_bet_update

Examples

```
## Not run: my_session <- mb_login("my_user_name","verysafepassword");
mb_bet_cancel(session_data=my_session,odds=2.5,stake=5,runner_id=12345)
## End(Not run)
```

`mb_bet_place` *Perform a Bet Placement Action*

Description

This function provide the core bet placement functionality. Its possible to place a single bet by entering in a single value for each of the runner_id, side, odds and stake parameters. It also possible to place multiple bets at once by passing a vector for each of the runner_id, side, odds and stake parameters. In this instance, its imperative that the order in each of the vectors is preserved i.e. that the runner_id, side, odds and stake for the first bet are the first elements of each parameter vector, and that the runner_id, side, odds and stake values for the second bet are the second elements of each parameter vector and so on. There is a cap of 20 on the number of bets that can be placed at one time. NOTE: its is very important to note the 'status' field returned after executing this function to ensure that you be has been matched as you expect. See description for more details.

Usage

```
mb_bet_place(session_data, runner_id, side, stake, odds)
```

Arguments

`session_data`	A session object returned from a successful mb_login attempt. It contains details about your user preferences and security details.
`runner_id`	The id of the runner that you want to place a bet on.
`side`	The side that you want place a bet. This has to be one of either 'back' or 'lay'.
`stake`	The amount that want to stake for this bet. The currency used is the currency you specified when you set up your Matchbook.com account. A real number format is required.
`odds`	The odds you want to place a bet at. The odds type is based on the information from your session_data, which is the default setting saved for your account.

Value

The status and details of your bet placement are returned. The data frame has the following fields:

id the bet id

event-id the event id on which the bet was placed

event-name the name of the event on which the bet was placed

event-id the event id on which the bet was placed

market-id the market id on which the bet was placed

market-name the name of the market on which the bet was placed

runner-id the runner id on which the bet was placed

runner-name the name of the runner on which the bet was placed

side the side the bet was placed on

odds the odds the bet was placed on

decimal-odds the decimal version of the odds

stake the stake placed

remaining this field indicates how much of the original stake placed remains unmatched. If this value is equal to the original stake, the the bet is fully un-matched. If this value is zero, then the bet has been fully matched. Any value in-between indicates a partial match

potential-profit the potential profit if the matched component of this wager is successful

remaining-potential-profit the potential profit if the un-matched component of this wager is first matched and then has a successful outcome

currency The currency the bet stake was placed with

created-at The date the bet was placed

status The bet status. Status 'open' indicates an unmatched bet, 'matched' indicates a fully matched bet, 'cancelled' indicates a cancelled bet, 'failed' indicates a failed bet placement. For bets with status='open', the 'stake' and 'remaining' fields are key to determining the exact status. If the 'remaining' value is less than 'stake' but greater than zero, then the bet has been partially matched for a 'stake'-'remaining' amount. If the bet is fully un-matched, then the 'stake' and 'remaining' values will be equal.

See Also

`mb_get_bets`

Examples

```
## Not run: my_session <- mb_login("my_user_name","verysafepassword");
mb_bet_place(session_data=my_session,odds=2.5,stake=5,runner_id=12345)
## End(Not run)
```

`mb_bet_update` *Perform a Bet Update Action*

Description

This function provides bet update functionality. Its possible to update a single bet by entering in a single value for each of the bet_id, side, odds and stake parameters. It also possible to update multiple bets at once by passing a vector for each of the bet_id, side, odds and stake parameters. In this instance, its imperative that the order in each of the vectors is preserved i.e. that the runner_id, side, odds and stake for the first update are the first elements of each parameter vector, and that the bet_id, side, odds and stake values for the second update are the second elements of each parameter vector and so on. There is a cap of 25 on the number of updates that can be placed at one time.

Usage

```
mb_bet_update(session_data, bet_id, side, stake, odds)
```

Arguments

`session_data`	A session object returned from a successful mb_login attempt. It contains details about your user preferences and security details.
`bet_id`	The id of the bet that you want to update.
`side`	The side that you want place a bet. This has to be one of either 'back' or 'lay'.
`stake`	The amount that want to stake for this bet. The currency used is the currency you specified when you set up your Matchbook.com account. A real number format is required.
`odds`	The odds you want to place a bet at. The odds type is based on the information from your session_data, which is the default setting saved for your account.

Value

The status and details of your bet updates are returned. The data frame has the following fields:

id the bet id

event-id the event id on which the original bet was placed

event-name the name of the event on which the original bet was placed

event-id the event id on which the original bet was placed

market-id the market id on which the original bet was placed

market-name the name of the market on which the original bet was placed

runner-id the runner id on which the original bet was placed

runner-name the name of the runner on which the original bet was placed

temp-id the temporary id of the update

exchange-type the exchange type. This should always be 'back-lay'

side the side the bet was placed on

odds the odds the bet was placed on

odds-type the odds-type of the odds field

decimal-odds the decimal version of the odds

stake the stake placed

potential-profit the potential profit if the matched component of this wager is successful

remaining-potential-profit the potential profit if the un-matched component of this wager is first matched and then has a successful outcome

currency The currency the bet stake was placed with

created-at The date the bet was placed

status The bet status. Status 'open' indicates an unmatched bet, 'matched' indicates a fully matched bet, 'cancelled' indicates a cancelled bet. For bets with status='open', the 'stake' and 'remaining' fields are key to determining the exact status. If the 'remaining' value is less than 'stake' but greater than zero, then the bet has been partially matched for a 'stake'-'remaining' amount. If the bet is fully un-matched, then the 'stake' and 'remaining' values will be equal.

See Also

mb_get_bets,mb_bet_place,mb_bet_cancel

Examples

```
## Not run: my_session <- mb_login("my_user_name","verysafepassword");
new_odds_value <- 20
mb_bet_update(session_data=my_session,bet_id=12345,odds=new_odds_value)
## End(Not run)
```

---

mb_get_balance *Get List of Current Bets on Matchbook*

Description

Get account balance/exposure etc.

Usage

```
mb_get_balance(session_data)
```

Arguments

`session_data`	A session object returned from a successful mb_login attempt. It contains details about your user preferences and security details.

Value

If successful, a list with account balance information. The data frame has the following fields:

id the account id

balance the account balance in the currency of the account

exposure the account exposure in the currency of the account

commission-reserve the commission-reserve in the currency of the account

free-funds the free-funds in the currency of the account

See Also

mb_login

Examples

```
## Not run: my_session <- mb_login("my_user_name","my_password");
mb_get_balance(session_data=my_session)
## End(Not run)
```

`mb_get_bets` *Get List of Current Bets on Matchbook*

Description

List the first 500 bets that have been made on Matchbook events that have not yet settled.

Usage

```
mb_get_bets(session_data, event_ids = NULL, market_ids = NULL,
  runner_ids = NULL, sides = NULL, status = NULL, interval = 0)
```

Arguments

`session_data`	A session object returned from a successful mb_login attempt. It contains details about your user preferences and security details.
`event_ids`	A vector of event_ids for which a list of current bets is required. This is an optional parameter and the default is to return bets from all events unless market_ids or runner_ids are specified.
`market_ids`	A vector of market_ids for which a list of current bets is required. This is an optional parameter and the default is to return bets from all markets unless event_ids or runner_ids are specified.
`runner_ids`	A vector of runner_ids for which a list of current bets is required. This is an optional parameter and the default is to return bets from all runners unless event_ids or market_ids are specified.
`sides`	A filter to allow selection of either 'back' or 'lay' bets. The default is to return both types.
`status`	The bet status from one of the possible options ('matched','unmatched','cancelled','expired','open','pause By default matched and unmatched bets are returned. Bets with status 'expired' can no longer be matched.
`interval`	Time filter (in seconds) to allow selection of bets that were created or updated in the period between the currnet time and the current time minus the specified number of seconds.

Value

If successful, a dataframe with first 500 bets and associated information. Only 500 bets are permitted at one time. Pagination is possible but not implemented in this version. The data frame has the following fields:

id the bet id

event-id the event id on which the bet was placed

event-name the name of the event on which the bet was placed

event-id the event id on which the bet was placed

market-id the market id on which the bet was placed

market-name the name of the market on which the bet was placed

runner-id the runner id on which the bet was placed

runner-name the name of the runner on which the bet was placed

exchange-type the exchange type. This should always be 'back-lay'

side the side the bet was placed on

odds the odds the bet was placed on

odds-type the odds-type of the odds field

decimal-odds the decimal version of the odds

stake the stake placed

remaining this field indicates how much of the original stake placed remains un-matched. If this value is equal to the original stake, the the bet is fully un-matched. If this value is zero, then the bet has been fully matched. Any value in-between indicates a partial match

potential-profit the potential profit if the matched component of this wager is successful

remaining-potential-profit the potential profit if the un-matched component of this wager is first matched and then has a successful outcome

currency The currency the bet stake was placed with

created-at The date the bet was placed

status The bet status. Status 'open' indicates an unmatched bet, 'matched' indicates a fully matched bet, 'cancelled' indicates a cancelled bet. For bets with status='open', the 'stake' and 'remaining' fields are key to determining the exact status. If the 'remaining' value is less than 'stake' but greater than zero, then the bet has been partially matched for a 'stake'-'remaining' amount. If the bet is fully un-matched, then the 'stake' and 'remaining' values will be equal.

temp-id the temporary id of the bet

See Also

`mb_get_sports`,`mb_get_events`,`mb_get_markets`

Examples

```
## Not run: my_session <- mb_login("my_user_name","my_password");
mb_get_bets(session_data=my_session)
## End(Not run)
```

mb_get_currencies *Get List of Available Currencies*

Description

List the Currencies currently available on Matchbook.com.

Usage

```
mb_get_currencies(session_data)
```

Arguments

session_data	A session object returned from a successful mb_login attempt. It contains security details and your account preferences.

Value

If successful, a data frame with the following fields:

currency-id	currency name
short-name	short version of currency name
long-name	long version of currency name

See Also

mb_login

Examples

```
## Not run: my_session <- mb_login("my_user_name","versafepassword");
mb_get_currencies(session_data=my_session)
## End(Not run)
```

mb_get_events *Get List of Available Events*

Description

List the Events Available on Matchbook.com

Usage

```
mb_get_events(session_data, sport_ids = NULL, start_date = NULL,
  end_date = NULL, market_states = c("open", "suspended"))
```

Arguments

`session_data`	A session object returned from a successful mb_login attempt. It contains details about your user preferences and security details.
`sport_ids`	A vector of integer sport_ids that indicate sports for which event details are required. e.g. c(15,1) gives Soccer and Pro Football (NFL)
`start_date`	A string (or date/POSIXct) value with format YYYY-mm-dd or YYYY-mm-dd HH:MM:SS format.
`end_date`	A string (or date/POSIXct) value with format YYYY-mm-dd or YYYY-mm-dd HH:MM:SS format.
`market_states`	A vector of string containing the market states to return. Defaults to 'open' or 'suspended' market types.

Value

If successful, a dataframe with first 500 events and associated information. Only 500 events are permitted at one time. Pagination is possible but not implemented in this version. The data frame has the following fields:

id Event id

name Event name

start The start date of the event

status If betting is still available on this event it will have status='open'

sport-id The sport id of this event

category-id The category of the event e.g. Premier League is a category within Football/Soccer

in-running-flag Is the market currently in-running

allow-live-betting Is it possible for this market to go in running

market-ids The ids of the markets within this event

meta-tags Tags describing the event

See Also

`mb_get_sports`,`mb_get_markets`

Examples

```
## Not run: my_session <- mb_login("my_user_name","verysafepassword");
mb_get_events(session_data=my_session,sport_ids=15)
## End(Not run)
```

`mb_get_markets` *Get List of Available Markets for a given Event*

Description

List the Markets Available on Matchbook.com for a given Event

Usage

```
mb_get_markets(session_data, event_id, market_states = c("open", "suspended"),
  market_types = c("multirunner", "binary"), grading_types = NULL,
  include_runners = FALSE, include_prices = FALSE)
```

Arguments

`session_data`	A session object returned from a successful mb_login attempt. It contains details about your user preferences and security details.
`event_id`	The event_id integer for which a list of associated markets is required.
`market_states`	A vector of string containing the market states to return. Defaults to 'open' or 'suspended' market types.
`market_types`	A vector of strings containing the required market types. Valid market types are either 'multirunner' or 'binary'. Both are returned by default.
`grading_types`	An optional vector of strings containing the required grading types. Valid grading types are 'one_x_two','asian-handicap','high-score-wins','low-score-wins','point-spread','point-total','single-winner-wins'. All are returned by default.
`include_runners`	A boolean parameter indicating if the runner names and id should be returned or not. Defaults to FALSE.
`include_prices`	A boolean parameter indicating if the runner prices should be returned or not. Defaults to FALSE.

Value

If successful, a dataframe with first 500 markets and associated information. Only 500 markets are permitted at one time. Pagination is possible but not implemented here. The data frame has the following fields:

name Market name

start The start date of the market

status If betting is still available on this market it will have status='open'

type Market name

event-id Event id

id Market id

runner-ids The ids of runners in this market

grading-type The type of grading

in-running-flag Is the market currently in-running

allow-live-betting Is it possible for this market to go in running

handicap The handicap associated with this market, if any

If include_runners=TRUE, then additional runner information is returned. Also if include_prices=TRUE then price data for the associated runner is returned nested within the data frame.

See Also

`mb_get_sports`,`mb_get_events`,`mb_get_runners`

Examples

```
## Not run: my_session <- mb_login("my_user_name","my_password");
mb_get_markets(event_id=309912)
## End(Not run)
```

`mb_get_runners` *Get List of Available Runners for a given Market*

Description

List the Runners Available on Matchbook.com for a given Market

Usage

```
mb_get_runners(session_data, event_id, market_id, runner_id = NULL,
  runner_states = c("open", "suspended"), include_prices = TRUE,
  side = NULL, include_withdrawn = TRUE)
```

Arguments

`session_data`	A session object returned from a successful mb_login attempt. It contains details about your user preferences and security details.
`event_id`	The event_id integer for the associated market.
`market_id`	The market_id integer for which a list of associated runners is required.
`runner_id`	If you only require details for a single runner, specify this optional runner_id integer.
`runner_states`	A vector of string containing the runner states to return. Defaults to 'open' or 'suspended' market types.
`include_prices`	A boolean for returning runner prices in the response. Defaults to TRUE.
`side`	A filter to view the selected 'back' or 'lay' prices. The default is to return both.
`include_withdrawn`	A boolean for returning or not the withdrawn runners in the response. Defaults to TRUE.

Value

If successful, a dataframe with associated runner information. The data frame has the following fields:

name Runner name

status If betting is still available on this runner, it will have status='open'

event-id Event id

id Runner id

market-id Market id

If include_prices=TRUE then available prices for each runner are returned nested within the data frame.

See Also

`mb_get_sports`,`mb_get_events`,`mb_get_markets`

Examples

```
## Not run: my_session <- mb_login("my_user_name","my_password");
mb_get_runners(session_data=my_session,event_id=123456,market_id=1234567)
## End(Not run)
```

`mb_get_settled` *Get Details of Settled Bets*

Description

Get Details of all Settled bets

Usage

```
mb_get_settled(session_data, sport_id = NULL, period = NULL,
  start_date = Sys.Date() - 90, end_date = Sys.Date())
```

Arguments

`session_data`	A session object returned from a successful mb_login attempt. It contains details about your user preferences and security details.
`sport_id`	A vector of integer sport_ids that indicated sports for which event details are required. e.g. c(15,1) gives Soccer and Pro Football (NFL)
`period`	A value with one of 'today', '1-day', '2-day', 'yesterday', 'week', 'month', '1-month', '3-month'
`start_date`	A string (or date/POSIXct) value with format YYYY-mm-dd or YYYY-mm-dd HH:MM:SS format. Defaults to 90 days ago.
`end_date`	A string (or date/POSIXct) value with format YYYY-mm-dd or YYYY-mm-dd HH:MM:SS format. Defaults to today.

Value

If successful, a dataframe with first 500 settled bets. Only 500 bets are permitted at one time. Pagination is possible but not implemented in this version. The data frame has the following fields:

stake the amount bet

sport-id The sport id on which the bet was placed

event-id The event id on which the bet was placed

market-id The market id on which the bet was placed

market-type The market type on which the bet was placed

profit-and-loss The amount won or lost on the market

settled-at The date the bet was settled

See Also

`mb_get_bets`,`mb_get_events`

Examples

```
## Not run: my_session <- mb_login("my_user_name","verysafepassword");
mb_get_settled(session_data=my_session)
## End(Not run)
```

`mb_get_sports` *Get List of Available Sports*

Description

List the Sports currently available on Matchbook.com.

Usage

```
mb_get_sports(session_data, nsports = 1000)
```

Arguments

`session_data`	A session object returned from a successful `mb_login` attempt. It contains security details and your account preferences.
`nsports`	The number of sports for which details are required. Default is 1000.

Value

If successful, a data frame with the following fields:

name sport name

type the type of the sport

id sport id

See Also

mb_get_events

Examples

```
## Not run: my_session <- mb_login("my_user_name","versafepassword");
mb_get_sports(session_data=my_session)
## End(Not run)
```

mb_login *Login and Authenticate with the Matchbook.com API*

Description

Validate your session with the Matchbook API.

Usage

```
mb_login(username, password, print_balance_details = TRUE)
```

Arguments

`username`	Your Matchbook.com username. If you are not already registered go to Matchbook.com.
`password`	The password set by you during the registration process.
`print_balance_details`	optional parameter to allow balance and exposure display at the time of login

Value

A list with status_code and other fields relating to your session and account. If the log-in is successful, a status_code of 200 is returned.

See Also

mb_logout

Examples

```
## Not run: mb_login("my_username","verysafepassword")
```

mb_logout — *Terminate a Matchbook.com API session*

Description

End the Matchbook.com session.

Usage

```
mb_logout(session_data)
```

Arguments

`session_data`	This is a required parameter containing security and preference information. It must take the exact format of the object returned by the mb_login request.

Value

The response integer status_code.

See Also

`mb_login`

Examples

```
## Not run: my_session <- mb_login("my_user_name","verysafepassword");
mb_logout(my_session)
## End(Not run)
```

Overview — *Wrapper for the 'Matchbook' API*

Description

Provides a wrapper for the some basic functionality around the 'Matchbook' <http://www.matchbook.com> REST API. It features calls to get events, markets and runners in data frame format. It features functions for bet placement and position management and also allows reporting of settled bet transactions. Note: this package uses the back-lay format. The default for odds type, currency and language are set according to those of the registered user account.

Details

The package provides the basic functionality for betting on Matchbook.com via the RESTful API service. More details and examples can be found on this Bitbucket repository. A registered Matchbook.com account is required to use this package. Please read the documentation carefully for each individual function before use.

In order to login and log out, use the following functions:

- `mb_login`
- `mb_logout`

In order to get information on events/marketsor runners, use the following functions:

- `mb_get_events`
- `mb_get_markets`
- `mb_get_runners`

In order to perform a bet action, use the following functions:

- `mb_bet_place`
- `mb_bet_cancel`
- `mb_bet_update`

To see details on bets that have not yet settled, use the following function:

- `mb_get_bets`

To see details on bets that have already settled, use the following function:

- `mb_get_settled`

Your balance/exposure can be displayed on login and they are also returned as part of the `mb_bet_place`,`mb_bet_cancel` and `mb_bet_update` function calls.

Author(s)

Niall Fitzgerald

Maintainer: Niall Fitzgerald <nfitzgerald@xanadu.ie>

Index