Package ‘pinnacle.API’

November 27, 2015

Type Package

Version 1.90

Title A Wrapper for the Pinnacle Sports API

URL https://github.com/marcoblume/pinnacle.API

Author Marco Blume, Nicholas Jhirad, Amine Gassem

Maintainer Marco Blume <marco.blume@pinnaclesports.com>

Description An R interface to the Pinnaclesports.com API.

LazyData true

License GPL-3

Imports dplyr, uuid, httr, RCurl, XML, rjson, jsonlite

NeedsCompilation no

Repository CRAN

Date/Publication 2015-11-27 08:53:18

R topics documented:

`AcceptTermsAndConditions` *Accept terms and conditions, only run once per session, must agree to terms or functions will not work*

Description

Accept terms and conditions, only run once per session, must agree to terms or functions will not work

Usage

```
AcceptTermsAndConditions(accepted = FALSE)
```

Arguments

`accepted`	Default=FALSE , BOOLEAN

Examples

```
AcceptTermsAndConditions(accepted=TRUE)
```

`authorization` *Authorization for the Pinnacle API*

Description

Authorization for the Pinnacle API

Usage

```
authorization(user = as.character(GetCredentials()$Value[1]),
  pwd = as.character(GetCredentials()$Value[2]))
```

Arguments

`user`	Pinnacle Username
`pwd`	Pinnacle Password

`cbindNames` *Combines elements of an object, keeping names*

Description

Combines elements of an object, keeping names

Usage

```
cbindNames(x)
```

Arguments

`x`	takes a list of elements with names

Value

data.frame containing combined names

`CheckTermsAndConditions` *Prompts User for Terms and Conditions, otherwise stops running function*

Description

Prompts User for Terms and Conditions, otherwise stops running function

Usage

```
CheckTermsAndConditions()
```

Examples

```
CheckTermsAndConditions()
```

GetBetsList *Get a list of running/settled bets*

Description

Get a list of running/settled bets

Usage

```
GetBetsList(betlist = c("SETTLED", "RUNNING"),
  fromDate = as.POSIXlt(Sys.Date(), tz = "UTC") - 15 * 24 * 60 * 60,
  toDate = as.POSIXlt(Sys.Date(), tz = "UTC") + 24 * 60 * 60)
```

Arguments

betlist	Either 'SETTLED' or 'RUNNING' Default Behavior shows both
fromDate	Iso8061 Date Default: 15 days prior in UTC, as.POSIXct(Sys.Date(), tz = 'UTC')-15*24*60*60
toDate	Iso8061 Date Default: 1 day ahead in UTC (to counter possible fencepost situations), as.POSIXct(Sys.Date(), tz = 'UTC') + 24*60*60

Value

A list of bets and associated details

Examples

```
SetCredentials("TESTAPI","APITEST")
AcceptTermsAndConditions(accepted=TRUE)
GetBetsList()
```

GetClientBalance *Get Client Balance*

Description

Get Client Balance

Usage

```
GetClientBalance(force = TRUE)
```

Arguments

force	Default=TRUE , boolean if TRUE force a reload of the data if FALSE use cached data

Value

vector client balance parameter

Examples

```
SetCredentials("TESTAPI","APITEST")
AcceptTermsAndConditions(accepted=TRUE)
GetClientBalance()
```

`GetCredentials` *Get your credential values*

Description

Get your credential values

Usage

```
GetCredentials()
```

Value

A data.frame with your username and password

Examples

```
SetCredentials("TESTAPI","APITEST")
GetCredentials()
```

`GetCurrencies` *Get the list of supported Currencies*

Description

Get the list of supported Currencies

Usage

```
GetCurrencies(force = FALSE)
```

Arguments

`force` Default=FALSE, boolean if TRUE force a reload of the data if FALSE use cached data

Value

a data frame with these columns:

- Currency Code
- Exchange Rate to USD
- Currency Name

Examples

```
SetCredentials("TESTAPI","APITEST")
AcceptTermsAndConditions(accepted=TRUE)
GetCurrencies()
```

GetFixtures *Get Fixtures*

Description

Get Fixtures

Usage

```
GetFixtures(sportname, leagueIds, since = NULL, isLive = 0)
```

Arguments

sportname	The sport name for which to retrieve the fixtures
leagueIds	integer vector with league IDs.
since	Default=NULL, numeric This is used to receive incremental updates. Use the value of last from previous fixtures response.
isLive	Default=FALSE, boolean if TRUE retrieves ONLY live events if FALSE retrieved all events

Value

returns a data frame with columns:

- SportID
- Last
- League
- LeagueID
- EventID
- StartTime
- HomeTeamName

- AwayTeamName
- Rotation Number
- Live Status
- Status
- Parlay Status

Examples

```
SetCredentials("TESTAPI","APITEST")
AcceptTermsAndConditions(accepted=TRUE)
GetFixtures(sportname = "Badminton",leagueIds = 191545)
```

`GetInrunning` *GetInrunning*

Description

GetInrunning

Usage

```
GetInrunning()
```

Value

A dataframe containing the current State of live events

Examples

```
SetCredentials("TESTAPI","APITEST")
AcceptTermsAndConditions(accepted=TRUE)
GetInrunning()
```

GetLeagues *Get Leagues for Sport(s) by name*

Description

Returns all Leagues for the Sport(s)

Usage

```
GetLeagues(sports, force = FALSE, regex = FALSE)
```

Arguments

`sports`	character vector of sports names.
`force`	Default=FALSE, boolean if TRUE force a reload of the data if FALSE use cached data
`regex`	Default=FALSE, boolean if TRUE , retreives sports id using regular expression on names

Value

a data frame having columns:

- LeagueID
- LinesAvailable
- HomeTeam
- AllowRoundRobin
- LeagueName

Examples

```
SetCredentials("TESTAPI","APITEST")
AcceptTermsAndConditions(accepted=TRUE)
GetLeagues("Badminton")
```

GetLeaguesByID *Get Leagues for Sport(s) by ID*

Description

Returns all Leagues for the Sport(s)

Usage

```
GetLeaguesByID(sportid, force = FALSE)
```

Arguments

`sportid`	integer vector of sports IDs
`force`	boolean whether to get new data (TRUE) or use cached data (FALSE)

Value

a data frame having columns:

- LeagueID
- LinesAvailable
- HomeTeam
- AllowRoundRobin
- LeagueName

Examples

```
SetCredentials("TESTAPI","APITEST")
AcceptTermsAndConditions(accepted=TRUE)
GetLeaguesByID(1)
```

GetLine *Get Lines (Use to get more detail on a single line, but the GetOdds or showOddsDF versions are intended for large amounts of data)*

Description

Get Lines (Use to get more detail on a single line, but the GetOdds or showOddsDF versions are intended for large amounts of data)

Usage

```
GetLine(sportId, leagueids, eventId, periodNumber, betType, team = NULL,
  side = NULL, handicap = NULL, oddsFormat = "AMERICAN")
```

Arguments

`sportId`	The sport ID
`leagueids`	integer vector of leagueids.
`eventId`	numeric xxxxx
`periodNumber`	xxxxx
`betType`	xxxx
`team`	xxxx
`side`	xxx
`handicap`	xxx
`oddsFormat`	xxx

Value

returns a data frame with columns:

- SportID
- Last
- League
- LeagueID
- EventID
- StartTime
- HomeTeamName
- AwayTeamName
- Rotation Number
- Live Status
- Status
- Parlay Status

Examples

```
SetCredentials("TESTAPI","APITEST")
AcceptTermsAndConditions(accepted=TRUE)
 GetLine(sportId=1,leagueids=191545,eventId=495418854,
 periodNumber=0,team="TEAM1",betType="Moneyline")
```

GetOdds *Get Odds*

Description

Get Odds

Usage

```
GetOdds(sportname, leagueIds, since = NULL, isLive = 0, force = TRUE)
```

Arguments

`sportname`	The sport name for which to retrieve the fixutres
`leagueIds`	integer vector of leagueids.
`since`	numeric This is used to receive incremental updates. Use the value of last from previous fixtures response.
`isLive`	boolean if TRUE retrieves ONLY live events
`force`	boolean if FALSE, functions using cached data will use the values since the last force

Value

list of lists

Examples

```

SetCredentials("TESTAPI","APITEST")
AcceptTermsAndConditions(accepted=TRUE)
GetOdds (sportname="Badminton", leagueIds=191545,,isLive=0)
```

GetSports *Get Sports*

Description

Returns all sports with the status whether they currently have lines or not

Usage

```
GetSports(force = FALSE)
```

Arguments

`force`	Default=FALSE, boolean if TRUE force a reload of the data if FALSE use cached data

Value

a data frame with these columns:

- SportID
- LinesAvailable
- SportName

Examples

```
SetCredentials("TESTAPI","APITEST")
AcceptTermsAndConditions(accepted=TRUE)
GetSports()
```

JSONtoDF *Sets the determined depth as a data.frame, and evades R's list-dropping rules*

Description

Sets the determined depth as a data.frame, and evades R's list-dropping rules

Usage

```
JSONtoDF(x, depth = 5, flag = TRUE)
```

Arguments

`x`	a list
`depth`	numeric value of the number of sublist
`flag`	whether to start with combining factors or binding rows

Value

a tree with the defined level as a dataframe

`pinnRawXMLResponse` *pinnRawXMLResponse*

Description

pinnRawXMLResponse

Usage

```
pinnRawXMLResponse()
```

Value

a list containing the raw and the converted XML responses

`PlaceBet` *Place Bet*

Description

Place bet in the system

Usage

```
PlaceBet(stake, sportId, eventId, periodNumber, lineId, betType,
  altLineId = NULL, team = NULL, side = NULL, acceptBetterLine = TRUE,
  winRiskStake = "RISK", oddsFormat = "AMERICAN")
```

Arguments

`stake`	numeric Wager amount in currency
`sportId`	numeric the sport id
`eventId`	numeric the vent id
`periodNumber`	numeric Number of the period , see Pinnacle API manual
`lineId`	numeric ID of the line
`betType`	BET_TYPE - SPREAD - MONEYLINE - TOTAL_POINTS - TEAM_TOTAL_POINTS
`altLineId`	numeric ID of the alternate line (lineID must also be included)
`team`	Default = NULL , , see Pinnacle API manual

- TEAM1
- TEAM2
- DRAW

`side` Defaulat = NULL , , see Pinnacle API manual

- OVER
- UNDER

`acceptBetterLine`
Default=TRUE ,boolean Whether or not to accept a bet when there is a line change in favor of the client

`winRiskStake` Default="RISK", either place the stake to RISK or to WIN

- WIN
- RISK

`oddsFormat` Default="AMERICAN", Desired Odds Format

- AMERICAN
- DECIMAL
- HONGKONG
- INDONESIAN
- MALAY

Value

list containing :

- status If Status is PROCESSED_WITH_ERROR errorCode will be in the response
- errorCode

Examples

```
SetCredentials("TESTAPI","APITEST")
AcceptTermsAndConditions(accepted=TRUE)
PlaceBet (stake=10,
          sportId=1,
          eventId=495418854,
          periodNumber=0,
          lineId=222136736,
          betType="MONEYLINE",
          team="TEAM2",
          acceptBetterLine=TRUE,
          winRiskStake="WIN",
          oddsFormat="AMERICAN")
```

PlaceParlayBet *PlaceParlayBet*

Description

Place parlay or round robin parlay bet in the system

Usage

```
PlaceParlayBet(riskAmount, legslist, roundRobinOptions = c("Parlay",
  "TwoLegRoundRobin", "ThreeLegRoundRobin", "FourLegRoundRobin",
  "FiveLegRoundRobin", "SixLegRoundRobin", "SevenLegRoundRobin",
  "EightLegRoundRobin")[1], oddsFormat = "AMERICAN",
  acceptBetterLine = TRUE)
```

Arguments

`riskAmount` numeric Wager amount in currency

`legslist` A list of wagers, where each wager must be in named list format. Required named values are: legBetType, lineId, either team or side, and periodNumber. Optional named values are: altLineId, pitcher1MustStart, or pitcher2MustStart. See the API Manual for more info

- lineId
- altLineId OPTIONAL
- sportId
- eventId
- periodNumber
- legBetType
  - MONEYLINE
  - SPREAD
  - TOTAL
  - TEAMTOTAL
- team/side using one will invalidate the other
- pitcher1MustStart OPTIONAL
- pitcher2MustStart OPTIONAL

`roundRobinOptions`
one of the round robin options, default is 'Parlay'

- Parlay
- TwoLegRoundRobin
- ThreeLegRoundRobin
- FourLegRoundRobin
- FiveLegRoundRobin
- SixLegRoundRobin

- SevenLegRoundRobin
- EightLegRoundRobin

oddsFormat default:'AMERICAN'

- AMERICAN
- DECIMAL
- HONGKONG
- INDONESIAN
- MALAY

acceptBetterLine

: Default TRUE ,boolean Whether or not to accept a bet when there is a line change in favor of the client

Value

list containing:

- status If Status is PROCESSED_WITH_ERROR errorCode will be in the response
- errorCode

Examples

```
SetCredentials("TESTAPI","APITEST")
AcceptTermsAndConditions(accepted=TRUE)
parlay1 <-  list(lineId = 222136736,
                 sportId=1,
                 eventId = 495418854,
                 periodNumber=0,
                 legBetType = "MONEYLINE",
                 team = 'TEAM1')
parlay2 <- list(lineId = 223187865,
                 sportId=1,
                 eventId = 496997901,
                 periodNumber=0,
                 legBetType = "TOTAL_POINTS",
                 side = 'OVER')
legslist <- list(parlay1,parlay2)


PlaceParlayBet(riskAmount=10,
               legslist=legslist,
               roundRobinOptions="Parlay",
               oddsFormat="AMERICAN" ,
               acceptBetterLine=TRUE)
```

SetCredentials *Set your pinnaclesports.com user credentials*

Description

Set your pinnaclesports.com user credentials

Usage

```
SetCredentials(username, password)
```

Arguments

`username`	Your username
`password`	Your password

Examples

```
SetCredentials("TESTAPI","APITEST")
```

showOddsDF *showOddsDF - Takes a GetOdds JSON response and turns it into a data.frame*

Description

showOddsDF - Takes a GetOdds JSON response and turns it into a data.frame

Usage

```
showOddsDF(sportname, leagueIds = NULL, since = NULL, isLive = 0,
  attachLeagueNames = TRUE, force = TRUE)
```

Arguments

`sportname`	The sport name for which to retrieve the fixutres
`leagueIds`	numeric vector of leagueids - can get as output from GetLeagues
`since`	numeric This is used to receive incremental updates. Use the value of last from previous fixtures response.
`isLive`	boolean if TRUE retrieves ONLY live events
`attachLeagueNames`	boolean default set to true, will attach league names.
`force`	boolean default set to TRUE, forces a reload of the cache. bettable leagues

Value

a dataframe combining GetOdds and GetFixtures data, containing NA's where levels of factors do not have a value. Naming convention is as follows, Example: spread.altLineId.N is the altLineId associated with spread.hdp.(N+1) whereas spread.hdp refers to the mainline. spread.altLineId is the first alternate, and equivalent to spread.altLineId.0

Examples

```
SetCredentials("TESTAPI","APITEST")
AcceptTermsAndConditions(accepted=TRUE)
showOddsDF(sportname="Badminton",leagueIds=191545)
```

Index