

# Using R to interact with the API

This is a tutorial on how to get data and send orders directly to the API of the Winton Climate Prediction Market using the statistical software R. It assumes that the user is familiar with this software. The code was tested on a Windows machine running R version 3.2.3.

## Preparation:

1. You need to have installed the following R packages: [httr],and [jsonlite].
2. You need to have copied your personal API token for your account, available at <https://app.climatepredictionmarket.com/account> (<https://app.climatepredictionmarket.com/account>). In the scripts below the personal token is used in the variable `mytok="Bearer [API token]"`, and will be part of the authorization header when required.

The API can be seen at <https://api.climatepredictionmarket.com/swagger/> (<https://api.climatepredictionmarket.com/swagger/>) (Authenticate using 'Bearer [API token]' to see authorized endpoints).

Now you can begin your R session or script:

```
library(httr)
#Library(curl)
library(jsonlite)
```

In what follows we look at the R commands for several of the available ways of interacting with the API.

## 1) Getting various ids

There are various ids that are required to interact with the API. These include:

- i. **marketId**: identifies the monthly market (2018-04,...,2018-09).
- ii. **eventSpaceId**: identifies the outcomes space for each market. There is a one to one correspondence between marketId and eventSpaceId.
- iii. **eventId**: identifies each of the outcomes in a given outcome space (i.e., for a given marketId or eventSpaceId).
- iv. **variableId**: identifies the variable: either monthly maximum temperature or monthly precipitation.

To find the **marketId** and the **eventSpaceId** go to the Market section:

```
# GET/Market          Get all markets
get.markets<-GET(url = "https://api.climatepredictionmarket.com/Market",
                accept_json()
                )

markets<-content(get.markets)
length(markets) # number of markets available
```

```
## [1] 6
```

markets is a list of 6 elements, one for each market. For each of them we can get their ids as follows:

```
i.market<-1 # choose a market
names(markets[[i.market]])
```

```
## [1] "eventSpaceId"      "expiry"            "liquidityFactor"
## [4] "settled"           "spreadCoefficients" "id"
```

The **marketId** is:

```
markets[[i.market]]$id # market id
```

```
## [1] "72a6c511-41ba-462f-ae66-148b477259e6"
```

and the corresponding **eventSpaceId**:

```
markets[[i.market]]$eventSpaceId # event space id
```

```
## [1] "54ff1cfe-ad53-4f94-a4fe-dd82588f75c0"
```

To find the **variableId** go to the Variables section:

```
# GET/variable          Get all Variables
get.variable<-GET(url="https://api.climatepredictionmarket.com/Variable",
                 accept_json()
                 )

variables<-content(get.variable)
# length(variables) # number of variables
```

There are 2 variables. Their ids can be obtained as follows:

```
variables[[1]]$id # variable id
```

```
## [1] "a3310156-5e31-4637-9595-17734f74d8df"
```

is the **variableId** for the variable named:

```
variables[[1]]$name # variable name
```

```
## [1] "UK Maximum Temperature"
```

To find the **eventId** for a given market

```
# GET/Event/EventSpace/{eventSpaceId} Get all events for an event space
url.tmp<- paste("https://api.climatepredictionmarket.com/Event/EventSpace/",
               markets[[i.market]]$eventSpaceId,sep="")

get.events<-GET(url = url.tmp,
               accept_json())
)
rm(url.tmp)

events.id<-content(get.events)
length(events.id) # list of as many elements as the dimension of the outcomes space.
```

```
## [1] 5207
```

Each element of this list has the following information:

```
names(events.id[[1]])
```

```
## [1] "intervals" "id"
```

```
# events.id[[j]]$id # this is eventId for outcome j in the outcomes space
# events.id[[j]]$intervals[[k]] intervals for variable k , with k=1,2
# names(events.id[[5202]]$intervals[[k]])
# "high"      "Low"      "variableId"
```

## 2) Getting your account information

To find the information about your account balance and your portfolio of contracts:

```
# GET/Account Get the current user's account
get.account<-GET(url = "https://api.climatepredictionmarket.com/Account",
                accept_json(),
                add_headers(Authorization=mytok)
)
account<-content(get.account)
names(account)
```

```
## [1] "balance"      "deposited"    "lastTransactionId"
## [4] "portfolio"    "withdrawn"    "id"
```

Here, "portfolio" is a list, where each element corresponds to a contract and contains the "contractId" and the number of contracts you hold ("quantity") for that contract. Note that your personal API token 'mytok' is required to retrieve the information about your account.

```
names(account$portfolio[[1]])
```

```
## [1] "contractId" "quantity"
```

To find the total number of contracts hold in each outcome in the outcomes space for a given market:

```

# GET/Account/Positions/{marketId}           Get positions in a market
# need market ID
i.market=1 # choose a market
url.tmp<-paste("https://api.climatepredictionmarket.com/Account/Positions/",markets[[i.market
]]$id,sep="")
get.account.pos<-GET(url =url.tmp,
                    accept_json(),
                    add_headers(Authorization=mytok)
)
rm(url.tmp)
account.pos<-content(get.account.pos)
length(account.pos) # each element of the list corresponds to an outcome in the outcomes spa
ce.

```

```
## [1] 70
```

```
names(account.pos[[1]])
```

```
## [1] "eventId" "value"
```

Account.pos is a list. Each element corresponds to an outcome in the outcomes space of market markets[[i.market]]\$id. Each element identifies the outcome (“eventId”) in the outcomes space, and the number of contracts hold (“value”).

The P&L and account valuation can be found as follows:

```

# GET/Account/Valuation Gets the valuation of the account's portfolio
get.account.val<-GET(url ="https://api.climatepredictionmarket.com/Account/Valuation",
                    accept_json(),
                    add_headers(Authorization=mytok)
)
account.val<-content(get.account.val)
# length(account.val) #
names(account.val)

```

```
## [1] "accountId" "key" "pnl" "valuation"
```

The P&L of your account is:

```
account.val$pnl #P&L
```

```
## [1] 0
```

and its instantaneous and liquidation value:

```
account.val$valuation # $instantaneousValue $liquidationValue
```

```
## $instantaneousValue
## [1] 0
##
## $liquidationValue
## [1] 0
```

### 3) Getting the price distribution

The market price distribution for each market can be obtained as follows:

```
# GET/Price/{marketId}          Get the prices for a market
# need market ID
i.market=1 # choose a market
url.tmp<-paste("https://api.climatepredictionmarket.com/Price/",markets[[i.market]]$id,sep=""
)
get.prices<-GET(url =url.tmp,
                accept_json(),
                add_headers(Authorization=mytok)
)
rm(url.tmp)
prices<-content(get.prices)
length(prices) # each element of the list corresponds to an outcome in the outcome space.
```

```
## [1] 5207
```

```
names(prices[[1]])
```

```
## [1] "eventId" "value"
```

For each outcome (“eventId”) in the outcomes space, “value” gives the corresponding price

### 4) Check the market state

```
# GET/MarketOrderStatus          Get the order status of all the markets
get.status<-GET(url ="https://api.climatepredictionmarket.com/MarketOrderStatus",
                accept_json(),
                add_headers(Authorization=mytok)
)

status<-content(get.status)
```

Each element of the list “status” corresponds to a market, there are:

```
length(status)
```

```
## [1] 6
```

```
names(status[[1]])
```

```
## [1] "isOpenForAuctionOrders" "isOpenForRealTimeOrders"  
## [3] "isProcessingAuction"      "marketExpiry"  
## [5] "marketId"
```

markets in total.

Note that each market can be open for Auction orders, Real time orders, or it can be procesing Auction orders.

## 5) Create a contract

To create a contract you need to provide the marketId for which you want to create a contract, the name of the contract, and a list (“weights”) specifying the relative weights (“value”) of the different outcomes (“eventId”) within the region in the outcomes space the contract encloses.

```
# POST/Contract Create a contract  
x<-list(marketId= markets[[i.market]]$id,name="test.contract")  
x$weights<-list(list(eventId= events.id[[1]]$id,value=0.25),  
                list(eventId= events.id[[2]]$id,value=0.75))  
  
post.contract<-POST(url = "https://api.climatepredictionmarket.com/Contract" ,  
                    body = toJSON(x,auto_unbox=TRUE,pretty=TRUE),  
                    add_headers (  
                      "Content-Type" = "application/json-patch+json",  
                      "Accept" = "application/json",  
                      "Authorization" = mytok  
                    ),  
                    encode="json"  
)  
contract<-content(post.contract) # this returns the contract id  
rm(x)
```

contract is the contractId of the created contract.

## 6) Get a quote for buying or selling a contract

To get a quote you need the contractId. If you created a contract as in the previous step, you already know the contract name and its corresponding id. If you created the contract using the UI, you will need to find its id. As described in 2), under Account: Get/Account, you’ll get a list of all the contract ids, and then under Contract: Get/contract{id} you can find the name for each contractId.

```

# POST/Quote      Get a quote

x<-list(list(contractId={contract} ,quantity=1),
        list(contractId={"050dda1a-f0c8-484d-a4e1-ba7528cfbb27"} ,quantity=1))

post.quote<-POST(url ="https://api.climatepredictionmarket.com/Quote" ,
                body = toJSON(x,auto_unbox=TRUE,pretty=TRUE),
                add_headers (
                  "Content-Type" = "application/json-patch+json",
                  "Accept" = "application/json",
                  "Authorization" = mytok
                ),
                encode="json"
)

quote<-content(post.quote) #

rm(x)

```

“quote” returns the quote for buying (if quantity>0) or selling (if quantity <0) the contracts (and quantities) specified above.

## 7) Post a Real time order

To post an order you need the contractId. If you created a contract as in the previous step, you already know the contract name and its corresponding id. If you created the contract using the UI, you will need to find its id. As described in 2), under Account: Get/Account, you'll get a list of all the contract ids, and then under Contract: Get/contract{id} you can find the name for each contractId.

```

# POST/RealTimeOrder      Create a real time order

x<-list(contractQuantities={},limit=0.01)                # slippage limit
x$contractQuantities<-list(list(contractId= contract,quantity=100),      # contract creat
ed above
                            list(contractId="050dda1a-f0c8-484d-a4e1-ba7528cfbb27" ,quantity=100)) # contract
created in UI

post.realtimeorder<-POST(url ="https://api.climatepredictionmarket.com/RealTimeOrder" ,
                        body = toJSON(x,auto_unbox=TRUE,pretty=TRUE),
                        add_headers (
                          "Content-Type" = "application/json-patch+json",
                          "Accept" = "application/json",
                          "Authorization" = mytok
                        ),
                        encode="json"
)

realtimeorder<-content(post.realtimeorder) # this returns the order id

rm(x)

```

## References

[1] R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/> (<https://www.R-project.org/>).

[2] Hadley Wickham (2017). htr: Tools for Working with URLs and HTTP. R package version 1.3.1. <https://CRAN.R-project.org/package/htr> (<https://CRAN.R-project.org/package/htr>).

[3] Jeroen Ooms (2014). The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects. arXiv:1403.2805 [stat.CO] URL <https://arxiv.org/abs/1403.2805> (<https://arxiv.org/abs/1403.2805>).