
robinhood Documentation

Release 0.0

Kanat Bekt

May 16, 2017

Contents

1	API Reference	1
	Python Module Index	13

CHAPTER 1

API Reference

The API reference below is auto-generated from the source code. Unofficial Robinhood API Python client.

Until the official documentation is out, all requests and responses are simple JSON models. It does not make sense to be opiniated and split endpoints into different resources and models right now.

`robinhood.api_token_auth()`

```
@POST("/api-token-auth/")
AuthToken login(@Field("username") String paramString1,
               @Field("password") String paramString2)
```

`robinhood.api_token_logout()`

```
@POST("/api-token-logout/")
Response logout()
```

`robinhood.login()`

```
@POST("/api-token-auth/")
AuthToken login(@Field("username") String paramString1,
               @Field("password") String paramString2)
```

`robinhood.logout()`

```
@POST("/api-token-logout/")
Response logout()
```

`class robinhood.User`

classmethod additional_info()

```
@GET("/user/additional_info/")
UserAdditionalInfo getUserAdditionalInfo()
```

classmethod answer_cip_questions()

```
@PUT("/user/cip_questions/")
Response answerCipQuestions(@Body CipAnswerRequest paramCipAnswerRequest)
```

classmethod basic_info()

```
@GET("/user/basic_info/")
UserBasicInfo getUserBasicInfo()
```

classmethod clip_questions()

```
@GET("/user/cip_questions/")
List<CipQuestion>> getCipQuestions()
```

classmethod create()

```
@PUT("/user/")
User createUser(@Body User paramUser)
```

classmethod employment()

```
@GET("/user/employment/")
UserEmployment getUserEmployment()
```

classmethod get()

```
@GET("/user/")
User getUser();
```

classmethod identity_mismatch()

```
@GET("/user/identity_mismatch/")
PaginatedResult<IdentityMismatch> getMismatches()
```

classmethod investment_profile()

```
@GET("/user/investment_profile/")
UserInvestmentProfile getUserInvestmentProfile()
```

classmethod patch_basic_info()

```
@PATCH("/user/basic_info/")
UserBasicInfo updateUserBasicInfo(@Body UserBasicInfo paramUserBasicInfo)
```

classmethod `patch_user()`

```
@PATCH("/user/")
User updateUser(@Body User.UpdateRequest paramUpdateRequest)
```

classmethod `update_additional_info()`

```
@PUT("/user/additional_info/")
UserAdditionalInfo submitUserAdditionalInfo(@Body UserAdditionalInfo_
↳paramUserAdditionalInfo)
```

classmethod `update_basic_info()`

```
@PUT("/user/basic_info/")
UserBasicInfo submitUserBasicInfo(@Body UserBasicInfo paramUserBasicInfo)
```

classmethod `update_employment()`

```
@PUT("/user/employment/")
UserEmployment submitUserEmployment(@Body UserEmployment paramUserEmployment)
```

classmethod `update_investment_profile()`

```
@PUT("/user/investment_profile/")
Response submitUserInvestmentProfile(@Body UserInvestmentProfile_
↳paramUserInvestmentProfile)
```

class `robinhood.Account`

classmethod `all()`

```
@GET("/accounts/")
PaginatedResult<Account> getAccounts()
```

classmethod `get(account_id)`

```
@GET("/accounts/{id}/")
Account getAccount(@Path("id") String paramString)
```

classmethod `recent_day_trades(account_id)`

```
@GET("/accounts/{id}/recent_day_trades/")
PaginatedResult<DayTrade> getRecentDayTrades(@Path("id") String paramString)
```

class `robinhood.Order`

classmethod `all()`

```
@GET("/orders/")
PaginatedResult<Order> getOrders(@Query("cursor") String paramString)
```

classmethod `cancel(order_id)`

```
@POST("/orders/{orderId}/cancel/")
Response cancelOrder(@Path("orderId") String paramString)
```

classmethod `get(order_id)`

```
@GET("/orders/{orderId}/")
Order getOrder(@Path("orderId") String paramString)
```

classmethod `post()`

```
@POST("/orders/")
Order postOrder(@Body Order.Request paramRequest)
```

class `robinhood.Notification`

classmethod `devices()`

```
@GET("/notifications/devices/")
PaginatedResult<Device> getDevices()
```

classmethod `get_device(device_id)`

```
@DELETE("/notifications/devices/{deviceId}/")
Response deleteDevice(@Path("deviceId") String paramString)
```

classmethod `post_device()`

```
@POST("/notifications/devices/")
Device postDevice(@Body Device paramDevice)
```

class `robinhood.Application`

classmethod `all()`

```
@GET("/applications/")
PaginatedResult<Application> getApplications()
```

classmethod `create()`

```
@PUT("/applications/individual/")
Application createApplication()
```

classmethod `get_by_type(kind)`

```
@GET("/applications/{type}/")
Application getApplicationByType(@Path("type") String paramString)
```

class robinhood.**Dividend**

classmethod `all()`

```
@GET("/dividends/")
PaginatedResult<Dividend> getDividends(@Query("cursor") String paramString)
```

classmethod `get(dividend_id)`

```
@GET("/dividends/{dividendId}/")
Dividend getDividend(@Path("dividendId") String paramString)
```

class robinhood.**Document**

classmethod `all()`

```
@GET("/documents/")
PaginatedResult<Document> getDocuments()
```

classmethod `download(document_id)`

```
@GET("/documents/{id}/download/")
Response getDocumentDownloadUrl(@Path("id") String paramString)
```

classmethod `get(document_id)`

```
@GET("/documents/{id}/")
Document getDocument(@Path("id") String paramString)
```

class robinhood.**Ach**

classmethod `add_automatic_deposit()`

```
@POST("/ach/deposit_schedules/")
AutomaticDeposit addAutomaticDeposit(@Body AutomaticDeposit.Request_
↳ paramRequest)
```

classmethod `automatic_deposits()`

```
@GET("/ach/deposit_schedules/")
PaginatedResult<AutomaticDeposit> getAutomaticDeposits(@Query("cursor")
↳String paramString)
```

classmethod create_iav_relationship()

```
@POST("/ach/iav/create/")
AchRelationship createIavRelationship(@Field("access_token") String
↳paramString1,
    @Field("iav_account_id") String paramString2,
    @Field("account") String paramString3,
    @Field("bank_account_type") String paramString4,
    @Field("bank_account_holder_name") String paramString5)
```

classmethod create_relationship()

```
@POST("/ach/relationships/")
AchRelationship createAchRelationship(@Field("account") String paramString1,
    @Field("bank_routing_number") String paramString2,
    @Field("bank_account_number") String paramString3,
    @Field("bank_account_type") String paramString4,
    @Field("bank_account_holder_name") String paramString5,
    @Field("bank_account_nickname") String paramString6)
```

classmethod delete_automatic_deposit()

```
@DELETE("/ach/deposit_schedules/{automaticDepositId}/")
Response deleteAutomaticDeposit(@Path("automaticDepositId") String
↳paramString)
```

classmethod get_automatic_deposit(*deposit_id*)

```
@GET("/ach/deposit_schedules/{automaticDepositId}/")
AutomaticDeposit getAutomaticDeposit(@Path("automaticDepositId") String
↳paramString)
```

classmethod get_relationship(*relationship_id*)

```
@GET("/ach/relationships/{achRelationshipId}/")
AchRelationship getAchRelationship(@Path("achRelationshipId") String
↳paramString)
```

classmethod get_transfer(*transfer_id*)

```
@GET("/ach/transfers/{id}/")
AchTransfer getAchTransfer(@Path("id") String paramString)
```

classmethod post_iav_auth_request()

```
@POST("/ach/iav/auth/")
IavResponse postIavAuthRequest(@Field("bank_institution") String paramString1,
    @Field("username") String paramString2,
    @Field("password") String paramString3,
    @Field("pin") String paramString4)
```

classmethod `post_iav_mfa_answer()`

```
@POST("/ach/iav/auth/mfa/")
IavResponse postIavAuthMfaAnswer(@Field("bank_institution") String_
↳paramString1,
    @Field("access_token") String paramString2,
    @Field("mfa") String paramString3)
```

classmethod `post_queued_deposit()`

```
@POST("/ach/iav/queued_deposit/")
Response postQueuedAchDeposit(@Body QueuedAchDepositRequest_
↳paramQueuedAchDepositRequest)
```

classmethod `post_transfer()`

```
@POST("/ach/transfers/")
AchTransfer postAchTransfer(@Body AchTransfer.Request paramRequest)
```

classmethod `queued_deposit()`

```
@GET("/ach/iav/queued_deposit/")
QueuedAchDeposit getQueuedAchDeposit()
```

classmethod `relationships()`

```
@GET("/ach/relationships/")
PaginatedResult<AchRelationship> getAchRelationships(
    @Query("cursor") String paramString)
```

classmethod `transfers()`

```
@GET("/ach/transfers/")
PaginatedResult<AchTransfer> getAchTransfers(@Query("cursor") String_
↳paramString)
```

classmethod `unlink_relationship(relationship_id)`

```
@POST("/ach/relationships/{id}/unlink/")
Response unlinkAchRelationship(@Path("id") String paramString)
```

classmethod `verify_microdeposits(relationship_id)`

```
@POST("/ach/relationships/{id}/micro_deposits/verify/")
Response verifyMicrodeposits(@Path("id") String paramString1,
    @Field("first_amount_cents") String paramString2,
    @Field("second_amount_cents") String paramString3)
```

class robinhood.**Watchlist**

classmethod **add_instrument** (*watchlist*, *instrument*)

```
@POST("/watchlists/{watchlistName}/")
WatchlistInstrument addInstrument(@Path("watchlistName") String paramString1,
    @Field("instrument") String paramString2)
```

classmethod **all** ()

```
@GET("/watchlists/")
PaginatedResult<Watchlist> getWatchlists()
```

classmethod **create** ()

```
@POST("/watchlists/")
Watchlist.EmptyWatchlist createWatchlist(@Field("name") String paramString)
```

classmethod **delete_instrument** (*watchlist*, *instrument_id*)

```
@DELETE("/watchlists/{watchlistName}/{instrumentId}/")
Void deleteInstrument(@Path("watchlistName") String paramString1,
    @Path("instrumentId") String paramString2)
```

classmethod **get** (*watchlist*)

```
@GET("/watchlists/{watchlistName}/")
PaginatedResult<WatchlistInstrument> getWatchlist(
    @Path("watchlistName") String paramString1,
    @Query("cursor") String paramString2)
```

classmethod **populate** ()

```
@POST("/watchlists/{watchlist}/bulk_add/")
List<WatchlistInstrument> populateWatchlist(@Path("watchlist") String_
    ↪paramString1,
    @Field("symbols") String paramString2)
```

class robinhood.**Quote**

classmethod **all** ()

```
@GET("/quotes/")
PaginatedResult<Quote> getQuotes(@Query("symbols") String paramString1,
    @Query("cursor") String paramString2)
```

classmethod `get` (*symbol*)

```
@GET("/quotes/{symbol}/")
Quote getQuote(@Path("symbol") String paramString)
```

classmethod `get_historical` (*symbol*)

```
@GET("/quotes/historical/{symbol}/")
HistoricalQuoteResult getHistoricalQuotes(@Path("symbol") String paramString1,
    @Query("interval") String paramString2)
```

class `robinhood.Market`

classmethod `all` ()

```
@GET("/markets/")
PaginatedResult<Market> getMarkets()
```

classmethod `get` (*mic*)

```
@GET("/markets/{mic}/")
Market getMarket(@Path("mic") String paramString)
```

classmethod `xnys_market_hours` (*date*)

```
@GET("/markets/XNYS/hours/{date}/")
MarketHours getMarketHours(@Path("date") String paramString)
```

class `robinhood.Upload`

classmethod `document_requests` ()

```
@GET("/upload/document_requests/")
PaginatedResult<DocumentRequest> getDocumentRequests()
```

classmethod `get_document_request` (*document_id*)

```
@GET("/upload/document_requests/{id}/")
DocumentRequest getDocumentRequest(@Path("id") String paramString)
```

classmethod `mark_document_request_uploaded` (*rhid*)

```
@PATCH("/upload/document_requests/{rhid}/")
DocumentRequest markDocumentRequestUploaded(@Path("rhid") String paramString1,
    @Field("state") String paramString2)
```

class robinhood.**Midland**

classmethod **card_stack** ()

```
@GET("/midlands/notifications/stack/")
Card.Stack getCardStack()
```

classmethod **dismiss_card** (*card_id*)

```
@POST("/midlands/notifications/stack/{cardId}/dismiss/")
Response dismissCard(@Path("cardId") String paramString)
```

classmethod **news** (*symbol*)

```
@GET("/midlands/news/{instrumentSymbol}/")
PaginatedResult<News> getNews(@Path("instrumentSymbol") String paramString)
```

classmethod **top_mover** (*direction*)

```
@GET("/midlands/movers/sp500/")
PaginatedResult<TopMover> getTopMover(@Query("direction") String paramString)
```

class robinhood.**Portfolio**

classmethod **get** (*account_id*)

```
@GET("/portfolios/{accountNumber}/")
Portfolio getPortfolio(@Path("accountNumber") String paramString)
```

classmethod **get_historicals** (*account_id*)

```
@GET("/portfolios/historicals/{accountNumber}/")
HistoricalPortfolio getHistoricalPortfolios(@Path("accountNumber") String
↳ paramString1,
    @Query("interval") String paramString2)
```

class robinhood.**Position**

classmethod **all** (*account*)

```
@GET("/positions/?nonzero=true")
PaginatedResult<Position> getPositions(@Query("account") String paramString1,
    @Query("cursor") String paramString2)
```

classmethod `get` (*account_id*, *instrument*)

```
@GET("/positions/{accountNumber}/{instrumentId}/")
Position getPosition(@Path("accountNumber") String paramString1,
    @Path("instrumentId") String paramString2)
```

class `robinhood.Instrument`

classmethod `all` ()

```
@GET("/instruments/")
PaginatedResult<Instrument> queryInstruments(@Query("query") String_
↳ paramString)
```

classmethod `get` (*instrument_id*)

```
@GET("/instruments/{instrumentId}/")
Instrument getInstrument(@Path("instrumentId") String paramString)
```


r

robinhood, [1](#)

A

Account (class in robinhood), 3
 Ach (class in robinhood), 5
 add_automatic_deposit() (robinhood.Ach class method), 5
 add_instrument() (robinhood.Watchlist class method), 8
 additional_info() (robinhood.User class method), 1
 all() (robinhood.Account class method), 3
 all() (robinhood.Application class method), 4
 all() (robinhood.Dividend class method), 5
 all() (robinhood.Document class method), 5
 all() (robinhood.Instrument class method), 11
 all() (robinhood.Market class method), 9
 all() (robinhood.Order class method), 3
 all() (robinhood.Position class method), 10
 all() (robinhood.Quote class method), 8
 all() (robinhood.Watchlist class method), 8
 answer_cip_questions() (robinhood.User class method), 2
 api_token_auth() (in module robinhood), 1
 api_token_logout() (in module robinhood), 1
 Application (class in robinhood), 4
 automatic_deposits() (robinhood.Ach class method), 5

B

basic_info() (robinhood.User class method), 2

C

cancel() (robinhood.Order class method), 4
 card_stack() (robinhood.Midland class method), 10
 clip_questions() (robinhood.User class method), 2
 create() (robinhood.Application class method), 4
 create() (robinhood.User class method), 2
 create() (robinhood.Watchlist class method), 8
 create_iav_relationship() (robinhood.Ach class method), 6
 create_relationship() (robinhood.Ach class method), 6

D

delete_automatic_deposit() (robinhood.Ach class method), 6

delete_instrument() (robinhood.Watchlist class method), 8
 devices() (robinhood.Notification class method), 4
 dismiss_card() (robinhood.Midland class method), 10
 Dividend (class in robinhood), 5
 Document (class in robinhood), 5
 document_requests() (robinhood.Upload class method), 9
 download() (robinhood.Document class method), 5

E

employment() (robinhood.User class method), 2

G

get() (robinhood.Account class method), 3
 get() (robinhood.Dividend class method), 5
 get() (robinhood.Document class method), 5
 get() (robinhood.Instrument class method), 11
 get() (robinhood.Market class method), 9
 get() (robinhood.Order class method), 4
 get() (robinhood.Portfolio class method), 10
 get() (robinhood.Position class method), 10
 get() (robinhood.Quote class method), 9
 get() (robinhood.User class method), 2
 get() (robinhood.Watchlist class method), 8
 get_automatic_deposit() (robinhood.Ach class method), 6
 get_by_type() (robinhood.Application class method), 5
 get_device() (robinhood.Notification class method), 4
 get_document_request() (robinhood.Upload class method), 9
 get_historical() (robinhood.Quote class method), 9
 get_historicals() (robinhood.Portfolio class method), 10
 get_relationship() (robinhood.Ach class method), 6
 get_transfer() (robinhood.Ach class method), 6

I

identity_mismatch() (robinhood.User class method), 2
 Instrument (class in robinhood), 11
 investment_profile() (robinhood.User class method), 2

L

login() (in module robinhood), [1](#)
logout() (in module robinhood), [1](#)

M

mark_document_request_uploaded() (robinhood.Upload class method), [9](#)
Market (class in robinhood), [9](#)
Midland (class in robinhood), [10](#)

N

news() (robinhood.Midland class method), [10](#)
Notification (class in robinhood), [4](#)

O

Order (class in robinhood), [3](#)

P

patch_basic_info() (robinhood.User class method), [2](#)
patch_user() (robinhood.User class method), [3](#)
populate() (robinhood.Watchlist class method), [8](#)
Portfolio (class in robinhood), [10](#)
Position (class in robinhood), [10](#)
post() (robinhood.Order class method), [4](#)
post_device() (robinhood.Notification class method), [4](#)
post_iav_auth_request() (robinhood.Ach class method), [6](#)
post_iav_mfa_answer() (robinhood.Ach class method), [7](#)
post_queued_deposit() (robinhood.Ach class method), [7](#)
post_transfer() (robinhood.Ach class method), [7](#)

Q

queued_deposit() (robinhood.Ach class method), [7](#)
Quote (class in robinhood), [8](#)

R

recent_day_trades() (robinhood.Account class method), [3](#)
relationships() (robinhood.Ach class method), [7](#)
robinhood (module), [1](#)

T

top_mover() (robinhood.Midland class method), [10](#)
transfers() (robinhood.Ach class method), [7](#)

U

unlink_relationship() (robinhood.Ach class method), [7](#)
update_additional_info() (robinhood.User class method), [3](#)
update_basic_info() (robinhood.User class method), [3](#)
update_employment() (robinhood.User class method), [3](#)
update_investment_profile() (robinhood.User class method), [3](#)
Upload (class in robinhood), [9](#)
User (class in robinhood), [1](#)

V

verify_microdeposits() (robinhood.Ach class method), [7](#)

W

Watchlist (class in robinhood), [8](#)

X

xnys_market_hours() (robinhood.Market class method), [9](#)