
Binary.com Deriv.com API for Python

Apr 04, 2023

Contents

1	Getting Started	1
1.1	Requirements	1
1.2	Installation	1
1.3	Basic Usage	1
1.4	Misc	2
2	Advanced Topics	3
3	API	5
3.1	Active Symbols (active_symbols)	5
3.2	API Token (api_token)	6
3.3	Application: Delete (app_delete)	6
3.4	Application: Get Details (app_get)	7
3.5	Application: List (app_list)	7
3.6	Application: Markup Details (app_markup_details)	8
3.7	Application: Register (app_register)	9
3.8	Application: Update (app_update)	10
3.9	Asset Index (asset_index)	11
3.10	Authorize (authorize)	12
3.11	Balance (balance)	12
3.12	Buy Contract (buy)	13
3.13	Buy Contract for Multiple Accounts (buy_contract_for_multiple_accounts)	14
3.14	Cancel a Contract (cancel)	15
3.15	Cashier Information (cashier)	15
3.16	Contracts For Symbol (contracts_for)	16
3.17	Update Contract (contract_update)	17
3.18	Update Contract History (contract_update_history)	17
3.19	Copy Trading: List (copytrading_list)	18
3.20	Copy Trading: Statistics (copytrading_statistics)	19
3.21	Copy Trading: Start (copy_start)	19
3.22	Copy Trading: Stop (copy_stop)	20
3.23	Cryptocurrency configurations (crypto_config)	20
3.24	Document Upload (document_upload)	21
3.25	Economic Calendar (economic_calendar)	22
3.26	Exchange Rates (exchange_rates)	23
3.27	Forget (forget)	23
3.28	Forget All (forget_all)	24

3.29	Account Status (get_account_status)	24
3.30	Get Financial Assessment (get_financial_assessment)	25
3.31	Account Limits (get_limits)	25
3.32	Get Self-Exclusion (get_self_exclusion)	26
3.33	Get Account Settings (get_settings)	26
3.34	Identity Verification Add Document (identity_verification_document_add)	27
3.35	Landing Company (landing_company)	27
3.36	Landing Company Details (landing_company_details)	28
3.37	Login History (login_history)	29
3.38	Log Out (logout)	29
3.39	MT5: Deposit (mt5_deposit)	30
3.40	MT5: Get Setting (mt5_get_settings)	30
3.41	MT5: Accounts List (mt5_login_list)	31
3.42	MT5: New Account (mt5_new_account)	31
3.43	MT5: Password Change (mt5_password_change)	33
3.44	MT5: Password Check (mt5_password_check)	34
3.45	MT5: Password Reset (mt5_password_reset)	34
3.46	MT5: Withdrawal (mt5_withdrawal)	35
3.47	New Real-Money Account: Deriv Investment (Europe) Ltd (new_account_maltainvest)	36
3.48	New Real-Money Account: Default Landing Company (new_account_real)	39
3.49	New Virtual-Money Account (new_account_virtual)	41
3.50	OAuth Applications (oauth_apps)	43
3.51	P2P Advertiser Adverts (p2p_advertiser_adverts)	44
3.52	P2P Advertiser Create (p2p_advertiser_create)	44
3.53	P2P Advertiser Information (p2p_advertiser_info)	45
3.54	P2P Advertiser Payment Methods (p2p_advertiser_payment_methods)	46
3.55	P2P Advertiser Relations (p2p_advertiser_relations)	46
3.56	P2P Advertiser Update (p2p_advertiser_update)	47
3.57	P2P Advert Create (p2p_advert_create)	48
3.58	P2P Advert Information (p2p_advert_info)	49
3.59	P2P Advert List (p2p_advert_list)	50
3.60	P2P Advert Update (p2p_advert_update)	51
3.61	P2P Chat Create (p2p_chat_create)	52
3.62	P2P Order Cancel (p2p_order_cancel)	53
3.63	P2P Order Confirm (p2p_order_confirm)	53
3.64	P2P Order Create (p2p_order_create)	54
3.65	P2P Order Dispute (p2p_order_dispute)	55
3.66	P2P Order Information (p2p_order_info)	55
3.67	P2P Order List (p2p_order_list)	56
3.68	P2P Order Review (p2p_order_review)	57
3.69	P2P Payment Methods (p2p_payment_methods)	57
3.70	P2P Ping (p2p_ping)	58
3.71	Payment agent create (paymentagent_create)	58
3.72	Payment agent details (paymentagent_details)	60
3.73	Payment Agent: List (paymentagent_list)	60
3.74	Payment Agent: Transfer (paymentagent_transfer)	61
3.75	Payment Agent: Withdraw (paymentagent_withdraw)	61
3.76	Payment Methods (payment_methods)	62
3.77	Payout Currencies (payout_currencies)	63
3.78	Ping (ping)	63
3.79	Portfolio (portfolio)	64
3.80	Profit Table (profit_table)	64
3.81	Price Proposal (proposal)	65
3.82	Price Proposal: Open Contracts (proposal_open_contract)	67

3.83	Reality Check (reality_check)	67
3.84	Countries List (residence_list)	68
3.85	Revoke Oauth Application (revoke_oauth_app)	68
3.86	Sell Contract (sell)	69
3.87	Sell Contracts: Multiple Accounts (sell_contract_for_multiple_accounts)	70
3.88	Sell Expired Contracts (sell_expired)	70
3.89	Set Account Currency (set_account_currency)	71
3.90	Set Financial Assessment (set_financial_assessment)	71
3.91	Set Self-Exclusion (set_self_exclusion)	73
3.92	Set Account Settings (set_settings)	75
3.93	Statement (statement)	77
3.94	States List (states_list)	78
3.95	Ticks Stream (ticks)	78
3.96	Ticks History (ticks_history)	79
3.97	Server Time (time)	80
3.98	Terms and Conditions Approval (tnc_approval)	80
3.99	Top Up Virtual-Money Account (topup_virtual)	81
3.100	Trading Durations (trading_durations)	81
3.101	Trading Platform: Investor Password Reset (trading_platform_investor_password_reset)	82
3.102	Trading Platform: Password Reset (trading_platform_password_reset)	83
3.103	Server list (trading_servers)	84
3.104	Trading Times (trading_times)	84
3.105	Transactions Stream (transaction)	85
3.106	Transfer Between Accounts (transfer_between_accounts)	85
3.107	Verify Email (verify_email)	86
3.108	Server Status (website_status)	87
4	Contributors	89
5	Indices and tables	91
	Index	93

CHAPTER 1

Getting Started

```
# High Level API, This API is based on "binaryapi.api" for easy usage
from binaryapi.stable_api import Binary

# Low Level API
from binaryapi.api import BinaryAPI
```

1.1 Requirements

Requires **Python 3.7.0** or later.

1.2 Installation

```
pip install -U git+https://github.com/mdn522/binaryapi.git
```

1.3 Basic Usage

1.3.1 Import

```
from binaryapi.stable_api import Binary
```

1.3.2 Enabling Debug Logs

```
import logging
logging.basicConfig(level=logging.DEBUG, format='%(asctime)s %(message)s')
```

1.3.3 Authorize/Login

```
binary = Binary(token="YOUR-API-TOKEN-GOES-HERE")
```

1.3.4 Authorize/Login with a message handler/callback function

```
def message_handler(msg):  
    msg_type = msg.get('msg_type')  
  
    print(msg_type, "=>", msg)  
  
binary = Binary(token="YOUR-API-TOKEN-GOES-HERE", message_callback=message_handler)
```

1.4 Misc

Note:

- In any code example: `binary` is instance of `binaryapi.stable_api.Binary`
 - `binary.api` is instance of `binaryapi.api.BinaryAPI`
 - All low level API functions are under `binaryapi.api.BinaryAPI`
 - So you can call it like `binary.api.buy`, `binary.api.ticks_history`, etc. (considering you named the instance `binary`)
-

CHAPTER 2

Advanced Topics

- **TODO**

3.1 Active Symbols (active_symbols)

Retrieve a list of all currently active symbols (underlying markets upon which contracts are available for trading).

active_symbols (*active_symbols: str, landing_company: Optional[str] = None, product_type: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*)
→ int

Parameters

- **active_symbols** (*str*) – If you use *brief*, only a subset of fields will be returned.
- **landing_company** (*Optional[str]*) – [Optional] If you specify this field, only symbols available for trading by that landing company will be returned. If you are logged in, only symbols available for trading by your landing company will be returned regardless of what you specify in this field.
- **product_type** (*Optional[str]*) – [Optional] If you specify this field, only symbols that can be traded through that product type will be returned.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.1.1 Example

```
binary.api.active_symbols(  
    'brief'  
    product_type='basic'  
)
```

See also:

- [Binary API Docs for active_symbols](#)

3.2 API Token (api_token)

This call manages API tokens

Auth Scope(s): admin

api_token (*delete_token: Optional[str] = None, new_token: Optional[str] = None, new_token_scopes: Optional[List] = None, valid_for_current_ip_only: Optional[int] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **delete_token** (*Optional[str]*) – [Optional] The token to remove.
- **new_token** (*Optional[str]*) – [Optional] The name of the created token.
- **new_token_scopes** (*Optional[List]*) – [Optional] List of permission scopes to provide with the token.
- **valid_for_current_ip_only** (*Optional[int]*) – [Optional] If you set this parameter during token creation, then the token created will only work for the IP address that was used to create the token
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.2.1 Example

```
binary.api.api_token(  
    new_token='Token example'  
    new_token_scopes=['admin', 'read', 'trade']  
)
```

See also:

- [Binary API Docs for api_token](#)

3.3 Application: Delete (app_delete)

The request for deleting an application.

Auth Scope(s): admin

app_delete (*app_delete: int, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **app_delete** (*int*) – Application app_id

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.3.1 Example

```
binary.api.app_delete(
    1234
)
```

See also:

- [Binary API Docs for app_delete](#)

3.4 Application: Get Details (app_get)

To get the information of the OAuth application specified by 'app_id'

Auth Scope(s): read

app_get (*app_get: int, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **app_get** (*int*) – Application app_id
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.4.1 Example

```
binary.api.app_get(
    1234
)
```

See also:

- [Binary API Docs for app_get](#)

3.5 Application: List (app_list)

List all of the account's OAuth applications

Auth Scope(s): read

app_list (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.5.1 Example

```
binary.api.app_list()
```

See also:

- [Binary API Docs for app_list](#)

3.6 Application: Markup Details (app_markup_details)

Retrieve details of *app_markup* according to criteria specified.

Auth Scope(s): read

app_markup_details (*date_from: str, date_to: str, app_id: Optional[int] = None, client_loginid: Optional[str] = None, description: Optional[int] = None, limit: Optional[Union[int, float, Decimal]] = None, offset: Optional[Union[int, float, Decimal]] = None, sort: Optional[str] = None, sort_fields: Optional[List] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **date_from** (*str*) – Start date (epoch or YYYY-MM-DD HH:MM:SS). Results are inclusive of this time.
- **date_to** (*str*) – End date (epoch or YYYY-MM-DD HH::MM::SS). Results are inclusive of this time.
- **app_id** (*Optional[int]*) – [Optional] Specific application *app_id* to report on.
- **client_loginid** (*Optional[str]*) – [Optional] Specific client loginid to report on, like CR12345
- **description** (*Optional[int]*) – [Optional] If set to 1, will return *app_markup* transaction details.
- **limit** (*Optional[Union[int, float, Decimal]]*) – [Optional] Apply upper limit to count of transactions received.
- **offset** (*Optional[Union[int, float, Decimal]]*) – [Optional] Number of transactions to skip.
- **sort** (*Optional[str]*) – [Optional] Sort direction on *transaction_time*. Other fields sort order is ASC.
- **sort_fields** (*Optional[List]*) – [Optional] One or more of the specified fields to sort on. Default sort field is by *transaction_time*.

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.6.1 Example

```
binary.api.app_markup_details(
    description=1
    app_id=1234
    client_loginid='CR12345'
    date_from='2017-08-01 00:00:00'
    date_to='2017-08-31 23:59:59'
    limit=100
    offset=0
    sort='ASC'
    sort_fields=['app_id', 'client_loginid', 'transaction_time']
    passthrough={}
    req_id=3
)
```

See also:

- [Binary API Docs for app_markup_details](#)

3.7 Application: Register (app_register)

Register a new OAuth application

Auth Scope(s): admin

app_register (*name: str, scopes: List, app_markup_percentage: Optional[Union[int, float, Decimal]] = None, appstore: Optional[str] = None, github: Optional[str] = None, googleplay: Optional[str] = None, homepage: Optional[str] = None, redirect_uri: Optional[str] = None, verification_uri: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **name** (*str*) – Application name.
- **scopes** (*List*) – List of permission scopes to grant the application.
- **app_markup_percentage** (*Optional[Union[int, float, Decimal]]*) – [Optional] Markup to be added to contract prices (as a percentage of contract payout).
- **appstore** (*Optional[str]*) – [Optional] Application's App Store URL (if applicable).
- **github** (*Optional[str]*) – [Optional] Application's GitHub page (for open-source projects).
- **googleplay** (*Optional[str]*) – [Optional] Application's Google Play URL (if applicable).

- **homepage** (*Optional[str]*) – [Optional] Application’s homepage URL.
- **redirect_uri** (*Optional[str]*) – [Optional] The URL to redirect to after a successful login. Required if charging markup percentage
- **verification_uri** (*Optional[str]*) – [Optional] Used when *verify_email* called. If available, a URL containing the verification token will be sent to the client’s email, otherwise only the token will be sent.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.7.1 Example

```
binary.api.app_register(  
    appstore='https://itunes.apple.com/test_app'  
    github='https://github.com/test_org/app'  
    googleplay='https://play.google.com/store/apps/details?id=test.app'  
    homepage='https://test.example.com/'  
    name='Test Application'  
    redirect_uri='https://test.example.com/redirect'  
    scopes=['read', 'trade']  
    verification_uri='https://test.example.com/verify'  
)
```

See also:

- [Binary API Docs for app_register](#)

3.8 Application: Update (app_update)

Update a new OAuth application

Auth Scope(s): admin

app_update (*app_update: int, name: str, scopes: List, app_markup_percentage: Optional[Union[int, float, Decimal]] = None, appstore: Optional[str] = None, github: Optional[str] = None, googleplay: Optional[str] = None, homepage: Optional[str] = None, redirect_uri: Optional[str] = None, verification_uri: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **app_update** (*int*) – Application app_id.
- **name** (*str*) – Application name.
- **scopes** (*List*) – Change scopes will revoke all user’s grants and log them out.
- **app_markup_percentage** (*Optional[Union[int, float, Decimal]]*) – [Optional] Markup to be added to contract prices (as a percentage of contract payout).
- **appstore** (*Optional[str]*) – [Optional] Application’s App Store URL (if applicable).

- **github** (*Optional[str]*) – [Optional] Application’s GitHub page (for open-source projects).
- **googleplay** (*Optional[str]*) – [Optional] Application’s Google Play URL (if applicable).
- **homepage** (*Optional[str]*) – [Optional] Application’s homepage URL.
- **redirect_uri** (*Optional[str]*) – [Optional] The URL to redirect to after a successful login. Required if charging markup percentage.
- **verification_uri** (*Optional[str]*) – [Optional] Used when *verify_email* called. If available, a URL containing the verification token will send to the client’s email, otherwise only the token will be sent.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.8.1 Example

```
binary.api.app_update(
    999
    appstore='https://itunes.apple.com/test_app'
    github='https://github.com/test_org/app'
    googleplay='https://play.google.com/store/apps/details?id=test.app'
    homepage='https://test.example.com/'
    name='Test Application'
    redirect_uri='https://test.example.com/redirect'
    scopes=['read', 'trade']
    verification_uri='https://test.example.com/verify'
)
```

See also:

- [Binary API Docs for app_update](#)

3.9 Asset Index (asset_index)

Retrieve a list of all available underlyings and the corresponding contract types and duration boundaries. If the user is logged in, only the assets available for that user’s landing company will be returned.

asset_index (*landing_company: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **landing_company** (*Optional[str]*) – [Optional] If specified, will return only the underlyings for the specified landing company.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.9.1 Example

```
binary.api.asset_index()
```

See also:

- [Binary API Docs for asset_index](#)

3.10 Authorize (authorize)

Authorize current WebSocket session to act on behalf of the owner of a given token. Must precede requests that need to access client account, for example purchasing and selling contracts or viewing portfolio.

authorize (*authorize: str, add_to_login_history: Optional[int] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **authorize** (*str*) – Authentication token. May be retrieved from https://www.binary.com/en/user/security/api_tokenws.html
- **add_to_login_history** (*Optional[int]*) – [Optional] Send this when you use api tokens for authorization and want to track activity using *login_history* call.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.10.1 Example

```
binary.api.authorize(  
    'uw2mk7no3oktoRVVsB4Dz7TQncfABuFDgO95dlxfMxRuPUDz '  
)
```

See also:

- [Binary API Docs for authorize](#)

3.11 Balance (balance)

Get user account balance

Auth Scope(s): read, trading_information

balance (*account: Optional[str] = None, subscribe: Optional[Union[bool, int]] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **account** (*Optional[str]*) – [Optional] If set to *all*, return the balances of all accounts one by one; if set to *current*, return the balance of current account; if set as an account id, return the balance of that account.
- **subscribe** (*Optional[Union[bool, int]]*) – [Optional] If set to 1, will send updates whenever the balance changes.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the web-socket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id**Return type** int

3.11.1 Example

```
binary.api.balance(
    subscribe=1
)
```

See also:

- [Binary API Docs for balance](#)

3.12 Buy Contract (buy)

Buy a Contract

Auth Scope(s): trade

buy (*buy: str, price: Union[int, float, Decimal], parameters=None, subscribe: Optional[Union[bool, int]] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **buy** (*str*) – Either the ID received from a Price Proposal (*proposal* call), or 1 if contract buy parameters are passed in the *parameters* field.
- **price** (*Union[int, float, Decimal]*) – Maximum price at which to purchase the contract.
- **parameters** – [Optional] Used to pass the parameters for contract buy.
- **subscribe** (*Optional[Union[bool, int]]*) – [Optional] 1 to stream.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the web-socket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id**Return type** int

3.12.1 Example

```
binary.api.buy(  
    'uw2mk7no3oktoRVVsB4Dz7TQnFfABuFDgO95dlxfMxRuPUsz'  
    price=100  
)
```

See also:

- [Binary API Docs for buy](#)

3.13 Buy Contract for Multiple Accounts (buy_contract_for_multiple_accounts)

Buy a Contract for multiple Accounts specified by the *tokens* parameter. Note, although this is an authorized call, the contract is not bought for the authorized account.

Auth Scope(s): trade

buy_contract_for_multiple_accounts (*buy_contract_for_multiple_accounts*: *str*, *price*: *Union[int, float, Decimal]*, *tokens*: *List*, *parameters*=*None*, *passthrough*: *Optional[Any]* = *None*, *req_id*: *Optional[int]* = *None*) → *int*

Parameters

- **buy_contract_for_multiple_accounts** (*str*) – Either the ID received from a Price Proposal (*proposal* call), or *1* if contract buy parameters are passed in the *parameters* field.
- **price** (*Union[int, float, Decimal]*) – Maximum price at which to purchase the contract.
- **tokens** (*List*) – List of API tokens identifying the accounts for which the contract is bought. Note: If the same token appears multiple times or if multiple tokens designate the same account, the contract is bought multiple times for this account.
- **parameters** – [Optional] Used to pass the parameters for contract buy.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns *req_id*

Return type *int*

3.13.1 Example

```
binary.api.buy_contract_for_multiple_accounts(  
    'AE79667A-3561-11E6-880B-19CE0BCBE464'  
    price=2.57  
    tokens=['EWHdv7feGJRmMf1kqv79lgfPiGjLLGV9GHTZfQ345FzJSfNE', 'ONqj76yAnVKnPtc',  
    ↪ 'oSpp7ohpGf50tP6', 'uz6OSIcFIcPKK5T']  
)
```

See also:

- [Binary API Docs for buy_contract_for_multiple_accounts](#)

3.14 Cancel a Contract (cancel)

Cancel contract with contract id

Auth Scope(s): trade

cancel (*cancel: int, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **cancel** (*int*) – Value should be the *contract_id* which received from the *portfolio* call.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.14.1 Example

```
binary.api.cancel(
    11542203588
)
```

See also:

- [Binary API Docs for cancel](#)

3.15 Cashier Information (cashier)

Request the cashier info for the specified type.

Auth Scope(s): payments

cashier (*cashier: str, address: Optional[str] = None, amount: Optional[Union[int, float, Decimal]] = None, dry_run: Optional[int] = None, provider: Optional[str] = None, type: Optional[str] = None, verification_code: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **cashier** (*str*) – Operation which needs to be requested from cashier
- **address** (*Optional[str]*) – [Optional] Address for crypto withdrawal. Only applicable for *api* type.
- **amount** (*Optional[Union[int, float, Decimal]]*) – [Optional] Amount for crypto withdrawal. Only applicable for *api* type.
- **dry_run** (*Optional[int]*) – [Optional] If set to 1, only validation is performed. Only applicable for *withdraw* using *crypto* provider and *api* type.

- **provider** (*Optional[str]*) – [Optional] Cashier provider. *crypto* will be default option for crypto currency accounts.
- **type** (*Optional[str]*) – [Optional] Data need to be returned from cashier. *api* is supported only for *crypto* provider.
- **verification_code** (*Optional[str]*) – [Optional] Email verification code (received from a *verify_email* call, which must be done first)
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.15.1 Example

```
binary.api.cashier(  
    'deposit'  
    provider='doughflow'  
    verification_code='my_verification_code'  
)
```

See also:

- [Binary API Docs for cashier](#)

3.16 Contracts For Symbol (contracts_for)

For a given symbol, get the list of currently available contracts, and the latest barrier and duration limits for each contract.

contracts_for (*contracts_for: str, currency: Optional[str] = None, landing_company: Optional[str] = None, product_type: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **contracts_for** (*str*) – The short symbol name (obtained from *active_symbols* call).
- **currency** (*Optional[str]*) – [Optional] Currency of the contract's stake and payout (obtained from *payout_currencies* call).
- **landing_company** (*Optional[str]*) – [Optional] Indicates which landing company to get a list of contracts for. If you are logged in, your account's landing company will override this field.
- **product_type** (*Optional[str]*) – [Optional] If you specify this field, only contracts tradable through that contract type will be returned.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.16.1 Example

```
binary.api.contracts_for(
    'R_50'
    currency='USD'
    landing_company='svg'
    product_type='basic'
)
```

See also:

- [Binary API Docs for contracts_for](#)

3.17 Update Contract (contract_update)

Update a contract condition.

Auth Scope(s): trade

contract_update (*contract_id: int, limit_order, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **contract_id** (*int*) – Internal unique contract identifier.
- **limit_order** – Specify limit order to update.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.17.1 Example

```
binary.api.contract_update(
    contract_id=123
    limit_order={'take_profit': 1}
)
```

See also:

- [Binary API Docs for contract_update](#)

3.18 Update Contract History (contract_update_history)

Request for contract update history.

Auth Scope(s): read

contract_update_history (*contract_id: int, limit: Optional[Union[int, float, Decimal]] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **contract_id** (*int*) – Internal unique contract identifier.
- **limit** (*Optional[Union[int, float, Decimal]]*) – [Optional] Maximum number of historical updates to receive.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.18.1 Example

```
binary.api.contract_update_history(  
    contract_id=123  
)
```

See also:

- [Binary API Docs for contract_update_history](#)

3.19 Copy Trading: List (copytrading_list)

Retrieves a list of active copiers and/or traders for Copy Trading

Auth Scope(s): admin

copytrading_list (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.19.1 Example

```
binary.api.copytrading_list()
```

See also:

- [Binary API Docs for copytrading_list](#)

3.20 Copy Trading: Statistics (copytrading_statistics)

Retrieve performance, trading, risk and copiers statistics of trader.

copytrading_statistics (*trader_id: str, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **trader_id** (*str*) – The ID of the target trader.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.20.1 Example

```
binary.api.copytrading_statistics(
    trader_id='CR1234'
)
```

See also:

- [Binary API Docs for copytrading_statistics](#)

3.21 Copy Trading: Start (copy_start)

Start copy trader bets

Auth Scope(s): trade

copy_start (*copy_start: str, assets: Optional[Union[List, str]] = None, max_trade_stake: Optional[Union[int, float, Decimal]] = None, min_trade_stake: Optional[Union[int, float, Decimal]] = None, trade_types: Optional[Union[List, str]] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **copy_start** (*str*) – API tokens identifying the accounts of trader which will be used to copy trades
- **assets** (*Optional[Union[List, str]]*) – [Optional] Used to set assets to be copied. E.x [“frxUSDJPY”, “R_50”]
- **max_trade_stake** (*Optional[Union[int, float, Decimal]]*) – [Optional] Used to set maximum trade stake to be copied.
- **min_trade_stake** (*Optional[Union[int, float, Decimal]]*) – [Optional] Used to set minimal trade stake to be copied.
- **trade_types** (*Optional[Union[List, str]]*) – [Optional] Used to set trade types to be copied. E.x [“CALL”, “PUT”]
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.

- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.21.1 Example

```
binary.api.copy_start(  
    'uw2mk7no3oktoRVVsB4Dz7TQnFgrthg'  
)
```

See also:

- [Binary API Docs for copy_start](#)

3.22 Copy Trading: Stop (copy_stop)

Stop copy trader bets

Auth Scope(s): trade

copy_stop (*copy_stop: str, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **copy_stop** (*str*) – API tokens identifying the accounts which needs not to be copied
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.22.1 Example

```
binary.api.copy_stop(  
    'uw2mk7no3oktoRVVsB4Dz7TQnFgrthg'  
)
```

See also:

- [Binary API Docs for copy_stop](#)

3.23 Cryptocurrency configurations (crypto_config)

The request for cryptocurrencies configuration.

crypto_config (*currency_code: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **currency_code** (*Optional[str]*) – [Optional] Cryptocurrency code. Sending request with `currency_code` provides crypto config for the sent cryptocurrency code only.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns `req_id`

Return type `int`

3.23.1 Example

```
binary.api.crypto_config()
```

See also:

- [Binary API Docs for crypto_config](#)

3.24 Document Upload (document_upload)

Request KYC information from client

Auth Scope(s): admin

document_upload (*document_format: str, document_type: str, expected_checksum: str, file_size: int, document_id: Optional[str] = None, document_issuing_country: Optional[str] = None, expiration_date: Optional[str] = None, lifetime_valid: Optional[int] = None, page_type: Optional[str] = None, proof_of_ownership=None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → `int`

Parameters

- **document_format** (*str*) – Document file format
- **document_type** (*str*) – Document type
- **expected_checksum** (*str*) – The checksum of the file to be uploaded
- **file_size** (*int*) – Document size (should be less than 10MB)
- **document_id** (*Optional[str]*) – [Optional] Document ID (required for Passport, Proof of ID and Driver's License)
- **document_issuing_country** (*Optional[str]*) – [Optional] 2-letter country code
- **expiration_date** (*Optional[str]*) – [Optional] Document expiration date (required for Passport, Proof of ID and Driver's License)
- **lifetime_valid** (*Optional[int]*) – [Optional] Boolean value that indicates whether this document is lifetime valid (only applies to POI document types, cancels out the `expiration_date` given if any)
- **page_type** (*Optional[str]*) – [Optional] To determine document side
- **proof_of_ownership** – [Optional] It contains info about the proof of ownership being uploaded (mandatory for `proof_of_ownership` document type)

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.24.1 Example

```
binary.api.document_upload(  
    document_format='JPG'  
    document_type='bankstatement'  
    expected_checksum='1a79a4d60de6718e8e5b326e338ae533'  
    file_size=12345  
)
```

See also:

- [Binary API Docs for document_upload](#)

3.25 Economic Calendar (economic_calendar)

Specify a currency to receive a list of events related to that specific currency. For example, specifying USD will return a list of USD-related events. If the currency is omitted, you will receive a list for all currencies.

economic_calendar (*currency: Optional[str] = None, end_date: Optional[int] = None, start_date: Optional[int] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **currency** (*Optional[str]*) – [Optional] Currency symbol.
- **end_date** (*Optional[int]*) – [Optional] End date.
- **start_date** (*Optional[int]*) – [Optional] Start date.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.25.1 Example

```
binary.api.economic_calendar(  
    currency='USD'  
    end_date=1561196696  
    start_date=1561096696  
)
```

See also:

- [Binary API Docs for economic_calendar](#)

3.26 Exchange Rates (exchange_rates)

Retrieves the exchange rates from a base currency to all currencies supported by the system.

exchange_rates (*base_currency: str, subscribe: Optional[Union[bool, int]] = None, target_currency: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **base_currency** (*str*) – Base currency (can be obtained from *payout_currencies* call)
- **subscribe** (*Optional[Union[bool, int]]*) – [Optional] 1 - to initiate a realtime stream of exchange rates relative to base currency.
- **target_currency** (*Optional[str]*) – [Optional] Local currency
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.26.1 Example

```
binary.api.exchange_rates(
    base_currency='USD'
)
```

See also:

- [Binary API Docs for exchange_rates](#)

3.27 Forget (forget)

Immediately cancel the real-time stream of messages with a specific ID.

forget (*forget: str, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **forget** (*str*) – ID of the real-time stream of messages to cancel.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.27.1 Example

```
binary.api.forget (
    'd1ee7d0d-3ca9-fbb4-720b-5312d487185b'
)
```

See also:

- [Binary API Docs for forget](#)

3.28 Forget All (forget_all)

Immediately cancel the real-time streams of messages of given type.

forget_all (*forget_all: Union[List], passthrough: Optional[Any] = None, req_id: Optional[int] = None*)
→ int

Parameters

- **forget_all** (*Union[List]*) – Cancel all streams by type. The value can be either a single type e.g. “ticks”, or an array of multiple types e.g. [“candles”, “ticks”].
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the web-socket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.28.1 Example

```
binary.api.forget_all (
    'ticks'
)
```

See also:

- [Binary API Docs for forget_all](#)

3.29 Account Status (get_account_status)

Get Account Status

Auth Scope(s): read

get_account_status (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the web-socket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.29.1 Example

```
binary.api.get_account_status()
```

See also:

- [Binary API Docs for get_account_status](#)

3.30 Get Financial Assessment (get_financial_assessment)

This call gets the financial assessment details. The ‘financial assessment’ is a questionnaire that clients of certain Landing Companies need to complete, due to regulatory and KYC (know your client) requirements.

Auth Scope(s): read

get_financial_assessment (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.30.1 Example

```
binary.api.get_financial_assessment()
```

See also:

- [Binary API Docs for get_financial_assessment](#)

3.31 Account Limits (get_limits)

Trading and Withdrawal Limits for a given user

Auth Scope(s): read

get_limits (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.31.1 Example

```
binary.api.get_limits()
```

See also:

- [Binary API Docs for get_limits](#)

3.32 Get Self-Exclusion (get_self_exclusion)

Allows users to exclude themselves from the website for certain periods of time, or to set limits on their trading activities. This facility is a regulatory requirement for certain Landing Companies.

Auth Scope(s): read

get_self_exclusion (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.32.1 Example

```
binary.api.get_self_exclusion()
```

See also:

- [Binary API Docs for get_self_exclusion](#)

3.33 Get Account Settings (get_settings)

Get User Settings (email, date of birth, address etc)

Auth Scope(s): read

get_settings (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.33.1 Example

```
binary.api.get_settings()
```

See also:

- [Binary API Docs for get_settings](#)

3.34 Identity Verification Add Document (identity_verification_document_add)

Adds document information such as issuing country, id and type for identity verification processes.

Auth Scope(s): admin

```
identity_verification_document_add(document_number: str, document_type: str, issuing_country: str, passthrough: Optional[Any] = None, req_id: Optional[int] = None) → int
```

Parameters

- **document_number** (*str*) – The identification number of the document.
- **document_type** (*str*) – The type of the document based on provided *issuing_country* (can obtained from *residence_list* call).
- **issuing_country** (*str*) – 2-letter country code (can obtained from *residence_list* call).
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.34.1 Example

```
binary.api.identity_verification_document_add(
    document_number='12345678912'
    document_type='nin_slip'
    issuing_country='ng'
)
```

See also:

- [Binary API Docs for identity_verification_document_add](#)

3.35 Landing Company (landing_company)

The company has a number of licensed subsidiaries in various jurisdictions, which are called Landing Companies. This call will return the appropriate Landing Company for clients of a given country. The landing company may differ for Gaming contracts (Synthetic Indices) and Financial contracts (Forex, Stock Indices, Commodities).

landing_company (*landing_company: str, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **landing_company** (*str*) – Client’s 2-letter country code (obtained from *residence_list* call).
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the web-socket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.35.1 Example

```
binary.api.landing_company(  
    'id'  
)
```

See also:

- [Binary API Docs for landing_company](#)

3.36 Landing Company Details (landing_company_details)

The company has a number of licensed subsidiaries in various jurisdictions, which are called Landing Companies (and which are wholly owned subsidiaries of the Deriv Group). This call provides information about each Landing Company.

landing_company_details (*landing_company_details: str, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **landing_company_details** (*str*) – Landing company shortcode.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the web-socket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.36.1 Example

```
binary.api.landing_company_details(  
    'svg'  
)
```

See also:

- [Binary API Docs for landing_company_details](#)

3.37 Login History (login_history)

Retrieve a summary of login history for user.

Auth Scope(s): read

login_history (*limit: Optional[int] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **limit** (*Optional[int]*) – [Optional] Apply limit to count of login history records.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.37.1 Example

```
binary.api.login_history(
    limit=25
)
```

See also:

- [Binary API Docs for login_history](#)

3.38 Log Out (logout)

Logout the session

logout (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.38.1 Example

```
binary.api.logout()
```

See also:

- [Binary API Docs for logout](#)

3.39 MT5: Deposit (mt5_deposit)

This call allows deposit into MT5 account from Binary account.

Auth Scope(s): `payments`

mt5_deposit (*to_mt5: str, amount: Optional[Union[int, float, Decimal]] = None, from_binary: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **to_mt5** (*str*) – MT5 account login to deposit money to
- **amount** (*Optional[Union[int, float, Decimal]]*) – Amount to deposit (in the currency of `from_binary`); min = \$1 or an equivalent amount, max = \$20000 or an equivalent amount
- **from_binary** (*Optional[str]*) – Binary account loginid to transfer money from
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns `req_id`

Return type `int`

3.39.1 Example

```
binary.api.mt5_deposit(  
    amount=1000  
    from_binary='CR100001'  
    to_mt5='MTR1000'  
)
```

See also:

- [Binary API Docs for mt5_deposit](#)

3.40 MT5: Get Setting (mt5_get_settings)

Get MT5 user account settings

Auth Scope(s): `read`

mt5_get_settings (*login: str, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **login** (*str*) – MT5 user login
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns `req_id`

Return type `int`

3.40.1 Example

```
binary.api.mt5_get_settings(
    login='MTR1000'
)
```

See also:

- [Binary API Docs for mt5_get_settings](#)

3.41 MT5: Accounts List (mt5_login_list)

Get list of MT5 accounts for client

Auth Scope(s): read

mt5_login_list (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the web-socket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.41.1 Example

```
binary.api.mt5_login_list()
```

See also:

- [Binary API Docs for mt5_login_list](#)

3.42 MT5: New Account (mt5_new_account)

This call creates new MT5 user, either demo or real money user.

Auth Scope(s): admin

mt5_new_account (*account_type: str, email: str, leverage: Union[int, float, Decimal], mainPassword: str, name: str, address: Optional[str] = None, city: Optional[str] = None, company: Optional[str] = None, country: Optional[str] = None, currency: Optional[str] = None, dry_run: Optional[int] = None, investPassword: Optional[str] = None, mt5_account_category: Optional[str] = None, mt5_account_type: Optional[str] = None, phone: Optional[str] = None, phonePassword: Optional[str] = None, server: Optional[str] = None, state: Optional[str] = None, zipCode: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **account_type** (*str*) – Account type. If set to 'financial', setting 'mt5_account_type' is also required.

- **email** (*str*) – Email address
- **leverage** (*Union[int, float, Decimal]*) – Client leverage (from 1 to 1000).
- **mainPassword** (*str*) – The master password of the account. For validation (Accepts any printable ASCII character. Must be within 8-25 characters, and include numbers, lowercase and uppercase letters. Must not be the same as the user's email address). This field is required.
- **name** (*str*) – Client's name. The maximum length here is 101 characters.
- **address** (*Optional[str]*) – [Optional] The address of the user. The maximum length of this address field is 128 characters.
- **city** (*Optional[str]*) – [Optional] User's city of residence.
- **company** (*Optional[str]*) – [Optional] Name of the client's company. The maximum length of the company name is 64 characters.
- **country** (*Optional[str]*) – [Optional] 2-letter country code (value received from *residence_list* call).
- **currency** (*Optional[str]*) – [Optional] MT5 account currency, the default value will be the qualified account currency.
- **dry_run** (*Optional[int]*) – [Optional] If set to 1, only validation is performed.
- **investPassword** (*Optional[str]*) – [Optional] The investor password of the account. For validation (Accepts any printable ASCII character. Must be within 8-25 characters, and include numbers, lowercase and uppercase letters. Must not be the same as the user's email address).
- **mt5_account_category** (*Optional[str]*) – [Optional] To choose whether account is conventional or not. Unavailable for financial_stp MT5_account_type
- **mt5_account_type** (*Optional[str]*) – [Optional] Financial: Variable spreads, High leverage. Financial STP: Variable spreads, Medium Leverage, more products. If 'account_type' set to 'financial', setting 'mt5_account_type' is also required.
- **phone** (*Optional[str]*) – [Optional] User's phone number.
- **phonePassword** (*Optional[str]*) – [Optional] The user's phone password.
- **server** (*Optional[str]*) – [Optional] Trade server.
- **state** (*Optional[str]*) – [Optional] User's state (region) of residence.
- **zipCode** (*Optional[str]*) – [Optional] User's zip code.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.42.1 Example

```

binary.api.mt5_new_account (
    account_type='demo'
    address='Dummy address'
    city='Valletta'
    company='Deriv Limited'
    country='mt'
    email='test@mailinator.com'
    investPassword='Anoth3r_p4ssword'
    leverage=100
    mainPassword='C0rrect_p4ssword'
    mt5_account_category='conventional'
    mt5_account_type='financial'
    name='Peter Pan'
    phone='+6123456789'
    phonePassword='AbcDv1234'
    state='Valleta'
    zipCode='VLT 1117'
)

```

See also:

- [Binary API Docs for mt5_new_account](#)

3.43 MT5: Password Change (mt5_password_change)

To change passwords of the MT5 account.

Auth Scope(s): admin

mt5_password_change (*login: str, new_password: str, old_password: str, password_type: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **login** (*str*) – MT5 user login
- **new_password** (*str*) – New password of the account. For validation (Accepts any printable ASCII character. Must be within 8-25 characters, and include numbers, lowercase and uppercase letters. Must not be the same as the user's email address).
- **old_password** (*str*) – Old password for validation (non-empty string, accepts any printable ASCII character)
- **password_type** (*Optional[str]*) – [Optional] Type of the password to change.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.43.1 Example

```
binary.api.mt5_password_change(  
    login='MTR1000'  
    new_password='C0rrect_p4ssword'  
    old_password='Abc1234'  
    password_type='main'  
)
```

See also:

- [Binary API Docs for mt5_password_change](#)

3.44 MT5: Password Check (mt5_password_check)

This call validates the main password for the MT5 user

Auth Scope(s): admin

mt5_password_check (*login: str, password: str, password_type: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **login** (*str*) – MT5 user login
- **password** (*str*) – The password of the account.
- **password_type** (*Optional[str]*) – [Optional] Type of the password to check.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.44.1 Example

```
binary.api.mt5_password_check(  
    login='MTR1000'  
    password='abc1234'  
    password_type='main'  
)
```

See also:

- [Binary API Docs for mt5_password_check](#)

3.45 MT5: Password Reset (mt5_password_reset)

To reset the password of MT5 account.

Auth Scope(s): admin

mt5_password_reset (*login: str, new_password: str, verification_code: str, password_type: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **login** (*str*) – MT5 user login
- **new_password** (*str*) – New password of the account. For validation (Accepts any printable ASCII character. Must be within 8-25 characters, and include numbers, lowercase and uppercase letters. Must not be the same as the user's email address).
- **verification_code** (*str*) – Email verification code (received from a *verify_email* call, which must be done first)
- **password_type** (*Optional[str]*) – [Optional] Type of the password to reset.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.45.1 Example

```
binary.api.mt5_password_reset (
    login='MTD1000'
    new_password='C0rrect_p4ssword'
    password_type='main'
    verification_code='08eZ2xMq'
)
```

See also:

- [Binary API Docs for mt5_password_reset](#)

3.46 MT5: Withdrawal (mt5_withdrawal)

This call allows withdrawal from MT5 account to Binary account.

Auth Scope(s): payments

mt5_withdrawal (*amount: Union[int, float, Decimal], from_mt5: str, to_binary: str, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **amount** (*Union[int, float, Decimal]*) – Amount to withdraw (in the currency of the MT5 account); min = \$1 or an equivalent amount, max = \$20000 or an equivalent amount.
- **from_mt5** (*str*) – MT5 account login to withdraw money from
- **to_binary** (*str*) – Binary account loginid to transfer money to
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.

- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.46.1 Example

```
binary.api.mt5_withdrawal(  
    amount=1000  
    from_mt5='MTR1000'  
    to_binary='CR100001'  
)
```

See also:

- [Binary API Docs for mt5_withdrawal](#)

3.47 New Real-Money Account: Deriv Investment (Europe) Ltd (new_account_maltainvest)

This call opens a new real-money account with the *maltainvest* Landing Company. This call can be made from a virtual-money account or real-money account at Deriv (Europe) Limited. If it is the latter, client information fields in this call will be ignored and data from your existing real-money account will be used.

Auth Scope(s): admin

new_account_maltainvest (*accept_risk: int, address_city: str, address_line_1: str, date_of_birth: str, education_level: str, employment_industry: str, estimated_worth: str, first_name: str, income_source: str, last_name: str, net_income: str, occupation: str, residence: str, salutation: str, tax_identification_number: str, tax_residence: str, account_opening_reason: Optional[str] = None, account_turnover: Optional[str] = None, address_line_2: Optional[str] = None, address_postcode: Optional[str] = None, address_state: Optional[str] = None, affiliate_token: Optional[str] = None, binary_options_trading_experience: Optional[str] = None, binary_options_trading_frequency: Optional[str] = None, cfd_trading_experience: Optional[str] = None, cfd_trading_frequency: Optional[str] = None, citizen: Optional[str] = None, client_type: Optional[str] = None, currency: Optional[str] = None, employment_status: Optional[str] = None, forex_trading_experience: Optional[str] = None, forex_trading_frequency: Optional[str] = None, non_pep_declaration: Optional[int] = None, other_instruments_trading_experience: Optional[str] = None, other_instruments_trading_frequency: Optional[str] = None, phone: Optional[str] = None, place_of_birth: Optional[str] = None, secret_answer: Optional[str] = None, secret_question: Optional[str] = None, source_of_wealth: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **accept_risk** (*int*) – Show whether client has accepted risk disclaimer.
- **address_city** (*str*) – Within 100 characters

- **address_line_1** (*str*) – Within 70 characters, with no leading whitespaces and may contain letters/numbers and/or any of following characters ‘.,:;()@#/-
- **date_of_birth** (*str*) – Date of birth format: yyyy-mm-dd.
- **education_level** (*str*) – Level of Education
- **employment_industry** (*str*) – Industry of Employment.
- **estimated_worth** (*str*) – Estimated Net Worth.
- **first_name** (*str*) – Within 2-50 characters, use only letters, spaces, hyphens, full-stops or apostrophes.
- **income_source** (*str*) – Income Source.
- **last_name** (*str*) – Within 2-50 characters, use only letters, spaces, hyphens, full-stops or apostrophes.
- **net_income** (*str*) – Net Annual Income.
- **occupation** (*str*) – Occupation.
- **residence** (*str*) – 2-letter country code, possible value receive from *residence_list* call.
- **salutation** (*str*) – Accept any value in enum list.
- **tax_identification_number** (*str*) – Tax identification number. Only applicable for real money account. Required for *maltainvest* landing company.
- **tax_residence** (*str*) – Residence for tax purpose. Comma separated iso country code if multiple jurisdictions. Only applicable for real money account. Required for *maltainvest* landing company.
- **account_opening_reason** (*Optional[str]*) – [Optional] Purpose and reason for requesting the account opening.
- **account_turnover** (*Optional[str]*) – [Optional] The anticipated account turnover.
- **address_line_2** (*Optional[str]*) – [Optional] Within 70 characters.
- **address_postcode** (*Optional[str]*) – [Optional] Within 20 characters and may not contain ‘+’.
- **address_state** (*Optional[str]*) – [Optional] Possible value receive from *states_list* call.
- **affiliate_token** (*Optional[str]*) – [Optional] Affiliate token, within 32 characters.
- **binary_options_trading_experience** (*Optional[str]*) – [Optional] Binary options trading experience.
- **binary_options_trading_frequency** (*Optional[str]*) – [Optional] Binary options trading frequency.
- **cfd_trading_experience** (*Optional[str]*) – [Optional] CFDs trading experience.
- **cfd_trading_frequency** (*Optional[str]*) – [Optional] CFDs trading frequency.
- **citizen** (*Optional[str]*) – [Optional] Country of legal citizenship, 2-letter country code. Possible value receive from *residence_list* call.

- **client_type** (*Optional[str]*) – [Optional] Indicates whether this is for a client requesting an account with professional status.
- **currency** (*Optional[str]*) – [Optional] To set currency of the account. List of supported currencies can be acquired with *payout_currencies* call.
- **employment_status** (*Optional[str]*) – [Optional] Employment Status.
- **forex_trading_experience** (*Optional[str]*) – [Optional] Forex trading experience.
- **forex_trading_frequency** (*Optional[str]*) – [Optional] Forex trading frequency.
- **non_pep_declaration** (*Optional[int]*) – [Optional] Indicates client's self-declaration of not being a PEP/RCA.
- **other_instruments_trading_experience** (*Optional[str]*) – [Optional] Trading experience in other financial instruments.
- **other_instruments_trading_frequency** (*Optional[str]*) – [Optional] Trading frequency in other financial instruments.
- **phone** (*Optional[str]*) – [Optional] Starting with + followed by 9-35 digits, hyphens or space.
- **place_of_birth** (*Optional[str]*) – [Optional] Place of birth, 2-letter country code.
- **secret_answer** (*Optional[str]*) – [Optional] Answer to secret question, within 4-50 characters.
- **secret_question** (*Optional[str]*) – [Optional] Accept any value in enum list.
- **source_of_wealth** (*Optional[str]*) – [Optional] Source of wealth.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.47.1 Example

```
binary.api.new_account_maltainvest(  
    accept_risk=1  
    account_opening_reason='Speculative'  
    account_turnover='Less than $25,000'  
    address_city='Melbourne'  
    address_line_1='20 Broadway Av'  
    address_line_2='East Melbourne VIC'  
    address_postcode='3002'  
    address_state='Victoria'  
    binary_options_trading_experience='1-2 years'  
    binary_options_trading_frequency='40 transactions or more in the past 12 months'  
    cfd_trading_experience='1-2 years'  
    cfd_trading_frequency='0-5 transactions in the past 12 months'  
    citizen='de'
```

(continues on next page)

(continued from previous page)

```

date_of_birth='1980-01-31'
education_level='Secondary'
employment_industry='Finance'
employment_status='Self-Employed'
estimated_worth='$100,000 - $250,000'
first_name='Peter'
forex_trading_experience='Over 3 years'
forex_trading_frequency='0-5 transactions in the past 12 months'
income_source='Self-Employed'
last_name='Pan'
net_income='$25,000 - $50,000'
non_pep_declaration=1
occupation='Managers'
other_instruments_trading_experience='Over 3 years'
other_instruments_trading_frequency='6-10 transactions in the past 12 months'
phone='+6123456789'
place_of_birth='nl'
residence='de'
salutation='Mr'
secret_answer='Jones'
secret_question="Mother's maiden name"
source_of_wealth='Company Ownership'
tax_identification_number='111-222-333'
tax_residence='de,nl'
)

```

See also:

- [Binary API Docs for new_account_maltainvest](#)

3.48 New Real-Money Account: Default Landing Company (new_account_real)

This call opens a new real-money account. This call can be made from a virtual-money or a real-money account. If it is the latter, client information fields in this call will be ignored and data from your existing real-money account will be used.

Auth Scope(s): admin

new_account_real (*account_opening_reason: Optional[str] = None, account_turnover: Optional[str] = None, address_city: Optional[str] = None, address_line_1: Optional[str] = None, address_line_2: Optional[str] = None, address_postcode: Optional[str] = None, address_state: Optional[str] = None, affiliate_token: Optional[str] = None, citizen: Optional[str] = None, client_type: Optional[str] = None, currency: Optional[str] = None, date_of_birth: Optional[str] = None, first_name: Optional[str] = None, last_name: Optional[str] = None, non_pep_declaration: Optional[int] = None, phone: Optional[str] = None, place_of_birth: Optional[str] = None, residence: Optional[str] = None, salutation: Optional[str] = None, secret_answer: Optional[str] = None, secret_question: Optional[str] = None, tax_identification_number: Optional[str] = None, tax_residence: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **account_opening_reason** (*Optional[str]*) – [Optional] Purpose and reason for requesting the account opening.
- **account_turnover** (*Optional[str]*) – [Optional] The anticipated account turnover.
- **address_city** (*Optional[str]*) – [Optional] Within 100 characters.
- **address_line_1** (*Optional[str]*) – Within 70 characters, with no leading whitespaces and may contain letters/numbers and/or any of following characters ‘.,;()@#/-’.
- **address_line_2** (*Optional[str]*) – [Optional] Within 70 characters.
- **address_postcode** (*Optional[str]*) – [Optional] Within 20 characters and may not contain ‘+’.
- **address_state** (*Optional[str]*) – [Optional] Possible value receive from *states_list* call.
- **affiliate_token** (*Optional[str]*) – [Optional] Affiliate token, within 32 characters.
- **citizen** (*Optional[str]*) – [Optional] Country of legal citizenship, 2-letter country code.
- **client_type** (*Optional[str]*) – [Optional] Indicates whether this is for a client requesting an account with professional status.
- **currency** (*Optional[str]*) – [Optional] To set currency of the account. List of supported currencies can be acquired with *payout_currencies* call.
- **date_of_birth** (*Optional[str]*) – Date of birth format: *yyyy-mm-dd*.
- **first_name** (*Optional[str]*) – Within 2-50 characters, use only letters, spaces, hyphens, full-stops or apostrophes.
- **last_name** (*Optional[str]*) – Within 2-50 characters, use only letters, spaces, hyphens, full-stops or apostrophes.
- **non_pep_declaration** (*Optional[int]*) – [Optional] Indicates client’s self-declaration of not being a PEP/RCA (Politically Exposed Person/Relatives and Close Associates).
- **phone** (*Optional[str]*) – [Optional] Starting with + followed by 9-35 digits, hyphens or space.
- **place_of_birth** (*Optional[str]*) – [Optional] Place of birth, 2-letter country code.
- **residence** (*Optional[str]*) – 2-letter country code, possible value receive from *residence_list* call.
- **salutation** (*Optional[str]*) – [Optional] Accept any value in enum list.
- **secret_answer** (*Optional[str]*) – [Optional] Answer to secret question, within 4-50 characters. Required for new account and existing client details will be used if client open another account.
- **secret_question** (*Optional[str]*) – [Optional] Accept any value in enum list. Required for new account and existing client details will be used if client open another account.

- **tax_identification_number** (*Optional[str]*) – [Optional] Tax identification number. Only applicable for real money account. Required for *maltainvest* landing company.
- **tax_residence** (*Optional[str]*) – [Optional] Residence for tax purpose. Comma separated iso country code if multiple jurisdictions. Only applicable for real money account. Required for *maltainvest* landing company.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.48.1 Example

```
binary.api.new_account_real(
    account_opening_reason='Speculative'
    account_turnover='Less than $25,000'
    address_city='Melbourne'
    address_line_1='20 Broadway Av'
    address_line_2='East Melbourne VIC'
    address_postcode='3002'
    address_state='Victoria'
    date_of_birth='1980-01-31'
    first_name='Peter'
    last_name='Pan'
    non_pep_declaration=1
    phone='+6123456789'
    place_of_birth='id'
    residence='au'
    salutation='Mr'
    secret_answer='Jones'
    secret_question="Mother's maiden name"
    tax_identification_number='012142545'
    tax_residence='ar,sg'
)
```

See also:

- [Binary API Docs for new_account_real](#)

3.49 New Virtual-Money Account (new_account_virtual)

Create a new virtual-money account.

```
new_account_virtual (affiliate_token: Optional[str] = None, client_password: Optional[str] = None, date_first_contact: Optional[str] = None, email_consent: Optional[int] = None, gclid_url: Optional[str] = None, residence: Optional[str] = None, signup_device: Optional[str] = None, type: Optional[str] = None, utm_ad_id=None, utm_adgroup_id=None, utm_adrollclk_id=None, utm_campaign=None, utm_campaign_id=None, utm_content=None, utm_fbcl_id=None, utm_gl_client_id=None, utm_medium=None, utm_msclick_id=None, utm_source=None, utm_term=None, verification_code: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None) → int
```

Parameters

- **affiliate_token** (*Optional[str]*) – [Optional] Affiliate token, within 32 characters.
- **client_password** (*Optional[str]*) – Password (Accepts any printable ASCII character. Must be within 8-25 characters, and include numbers, lowercase and uppercase letters. Must not be the same as the user's email address).
- **date_first_contact** (*Optional[str]*) – [Optional] Date of first contact, format: yyyy-mm-dd in GMT timezone.
- **email_consent** (*Optional[int]*) – [Optional] Boolean value: 1 or 0, indicating whether the client has given consent for marketing emails.
- **gclid_url** (*Optional[str]*) – [Optional] Google Click Identifier to track source.
- **residence** (*Optional[str]*) – 2-letter country code (obtained from *residence_list* call).
- **signup_device** (*Optional[str]*) – [Optional] Show whether user has used mobile or desktop.
- **type** (*Optional[str]*) – Account type
- **utm_ad_id** – [Optional] Identifier of particular ad. Value must match Regex pattern to be recorded
- **utm_adgroup_id** – [Optional] Identifier of ad group in the campaign. Value must match Regex pattern to be recorded
- **utm_adrollclk_id** – [Optional] Unique identifier of click on AdRoll ads platform. Value must match Regex pattern to be recorded
- **utm_campaign** – [Optional] Identifies a specific product promotion or strategic campaign such as a spring sale or other promotions. Value must match Regex pattern to be recorded
- **utm_campaign_id** – [Optional] Identifier of paid ad campaign. Value must match Regex pattern to be recorded
- **utm_content** – [Optional] Used to differentiate similar content, or links within the same ad. Value must match Regex pattern to be recorded
- **utm_fbcl_id** – [Optional] Unique identifier of click on Facebook ads platform. Value must match Regex pattern to be recorded
- **utm_gl_client_id** – [Optional] Unique visitor identifier on Google Ads platform. Value must match Regex pattern to be recorded
- **utm_medium** – [Optional] Identifies the medium the link was used upon such as: email, CPC, or other methods of sharing. Value must match Regex pattern to be recorded

- **utm_msclk_id** – [Optional] Unique click identifier on Microsoft Bing ads platform. Value must match Regex pattern to be recorded
- **utm_source** – [Optional] Identifies the source of traffic such as: search engine, newsletter, or other referral. Value must match Regex pattern to be recorded
- **utm_term** – [Optional] Used to send information related to the campaign term like paid search keywords. Value must match Regex pattern to be recorded
- **verification_code** (*Optional[str]*) – Email verification code (received from a *verify_email* call, which must be done first).
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.49.1 Example

```
binary.api.new_account_virtual(
    type='trading'
    client_password='C0rrect_p4ssword'
    residence='id'
    verification_code='uoJvVuQ6'
)
```

See also:

- [Binary API Docs for new_account_virtual](#)

3.50 OAuth Applications (oauth_apps)

List all my used OAuth applications.

Auth Scope(s): read

oauth_apps (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.50.1 Example

```
binary.api.oauth_apps()
```

See also:

- [Binary API Docs for oauth_apps](#)

3.51 P2P Advertiser Adverts (p2p_advertiser_adverts)

Returns all P2P adverts created by the authorized client. Can only be used by a registered P2P advertiser.

Auth Scope(s): `payments`

p2p_advertiser_adverts (*limit: Optional[int] = None, offset: Optional[int] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **limit** (*Optional[int]*) – [Optional] Used for paging. This value will also apply to subscription responses.
- **offset** (*Optional[int]*) – [Optional] Used for paging. This value will also apply to subscription responses.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns `req_id`

Return type `int`

3.51.1 Example

```
binary.api.p2p_advertiser_adverts()
```

See also:

- [Binary API Docs for p2p_advertiser_adverts](#)

3.52 P2P Advertiser Create (p2p_advertiser_create)

Registers the client as a P2P advertiser.

Auth Scope(s): `payments`

p2p_advertiser_create (*name: str, contact_info: Optional[str] = None, default_advert_description: Optional[str] = None, payment_info: Optional[str] = None, subscribe: Optional[Union[bool, int]] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **name** (*str*) – The advertiser’s displayed name.
- **contact_info** (*Optional[str]*) – [Optional] Advertiser’s contact information, to be used as a default for new sell adverts.

- **default_advert_description** (*Optional[str]*) – [Optional] Default description that can be used every time an advert is created.
- **payment_info** (*Optional[str]*) – [Optional] Advertiser's payment information, to be used as a default for new sell adverts.
- **subscribe** (*Optional[Union[bool, int]]*) – [Optional] If set to 1, will send updates whenever there is an update to advertiser
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.52.1 Example

```
binary.api.p2p_advertiser_create(
    name='your_name'
)
```

See also:

- [Binary API Docs for p2p_advertiser_create](#)

3.53 P2P Advertiser Information (p2p_advertiser_info)

Retrieve information about a P2P advertiser.

Auth Scope(s): payments

p2p_advertiser_info (*id: Optional[str] = None, subscribe: Optional[Union[bool, int]] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **id** (*Optional[str]*) – [Optional] The unique identifier for this advertiser. If not provided, returns advertiser information about the current account.
- **subscribe** (*Optional[Union[bool, int]]*) – [Optional] If set to 1, will send updates whenever there is an update to advertiser
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.53.1 Example

```
binary.api.p2p_advertiser_info()
```

See also:

- [Binary API Docs for p2p_advertiser_info](#)

3.54 P2P Advertiser Payment Methods (p2p_advertiser_payment_methods)

Manage or list P2P advertiser payment methods.

Auth Scope(s): payments

p2p_advertiser_payment_methods (*create: Optional[List] = None, delete: Optional[List] = None, update=None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **create** (*Optional[List]*) – Contains new payment method entries.
- **delete** (*Optional[List]*) – Contains payment methods to delete.
- **update** – Contains payment methods to update.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the web-socket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.54.1 Example

```
binary.api.p2p_advertiser_payment_methods(  
    create=[{'account': '1234', 'bank_name': 'some_bank', 'method': 'bank_transfer'}]  
    delete=[101, 102]  
    update={'103': {'instructions': 'phone first'}}  
)
```

See also:

- [Binary API Docs for p2p_advertiser_payment_methods](#)

3.55 P2P Advertiser Relations (p2p_advertiser_relations)

Updates and returns favourite and blocked advertisers of the current user.

Auth Scope(s): payments

p2p_advertiser_relations (*add_blocked: Optional[List] = None, add_favourites: Optional[List] = None, remove_blocked: Optional[List] = None, remove_favourites: Optional[List] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **add_blocked** (*Optional[List]*) – IDs of advertisers to block.
- **add_favourites** (*Optional[List]*) – IDs of advertisers to add as favourites.
- **remove_blocked** (*Optional[List]*) – IDs of advertisers to remove from blocked.
- **remove_favourites** (*Optional[List]*) – IDs of advertisers to remove from favourites.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.55.1 Example

```
binary.api.p2p_advertiser_relations()
```

See also:

- [Binary API Docs for p2p_advertiser_relations](#)

3.56 P2P Advertiser Update (p2p_advertiser_update)

Update the information of the P2P advertiser for the current account. Can only be used by an approved P2P advertiser.

Auth Scope(s): payments

p2p_advertiser_update (*contact_info: Optional[str] = None, default_advert_description: Optional[str] = None, is_listed: Optional[int] = None, payment_info: Optional[str] = None, show_name: Optional[int] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **contact_info** (*Optional[str]*) – [Optional] Advertiser’s contact information, to be used as a default for new sell adverts.
- **default_advert_description** (*Optional[str]*) – [Optional] Default description that can be used every time an advert is created.
- **is_listed** (*Optional[int]*) – [Optional] Used to set if the advertiser’s adverts could be listed. When 0, adverts won’t be listed regardless of they are active or not. This doesn’t change the *is_active* of each individual advert.
- **payment_info** (*Optional[str]*) – [Optional] Advertiser’s payment information, to be used as a default for new sell adverts.
- **show_name** (*Optional[int]*) – [Optional] When 1, the advertiser’s real name will be displayed on to other users on adverts and orders.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type `int`

3.56.1 Example

```
binary.api.p2p_advertiser_update(  
    is_listed=0  
)
```

See also:

- [Binary API Docs for p2p_advertiser_update](#)

3.57 P2P Advert Create (`p2p_advert_create`)

Creates a P2P (Peer to Peer) advert. Can only be used by an approved P2P advertiser.

Auth Scope(s): `payments`

p2p_advert_create (*amount: Union[int, float, Decimal], max_order_amount: Union[int, float, Decimal], min_order_amount: Union[int, float, Decimal], rate: Union[int, float, Decimal], type: str, contact_info: Optional[str] = None, description: Optional[str] = None, local_currency: Optional[str] = None, payment_info: Optional[str] = None, payment_method: Optional[str] = None, payment_method_ids: Optional[List] = None, payment_method_names: Optional[List] = None, rate_type: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → `int`

Parameters

- **amount** (*Union[int, float, Decimal]*) – The total amount of the advert, in advertiser's account currency.
- **max_order_amount** (*Union[int, float, Decimal]*) – Maximum allowed amount for the orders of this advert, in advertiser's *account_currency*. Should be more than or equal to *min_order_amount*.
- **min_order_amount** (*Union[int, float, Decimal]*) – Minimum allowed amount for the orders of this advert, in advertiser's *account_currency*. Should be less than or equal to *max_order_amount*.
- **rate** (*Union[int, float, Decimal]*) – Conversion rate from advertiser's account currency to *local_currency*. An absolute rate value (fixed), or percentage offset from current market rate (floating).
- **type** (*str*) – The advertisement represents the intention to perform this action on your Deriv account funds.
- **contact_info** (*Optional[str]*) – [Optional] Advertiser contact information.
- **description** (*Optional[str]*) – [Optional] General information about the advert.
- **local_currency** (*Optional[str]*) – [Optional] Local currency for this advert. If not provided, will use the currency of client's residence by default.
- **payment_info** (*Optional[str]*) – [Optional] Payment instructions.
- **payment_method** (*Optional[str]*) – [Optional] Payment method name (deprecated).

- **payment_method_ids** (*Optional[List]*) – IDs of previously saved payment methods as returned from `p2p_advertiser_payment_methods`, only applicable for sell ads.
- **payment_method_names** (*Optional[List]*) – Payment method identifiers as returned from `p2p_payment_methods`, only applicable for buy ads.
- **rate_type** (*Optional[str]*) – Type of rate, fixed or floating.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.57.1 Example

```
binary.api.p2p_advert_create(
    description='Please transfer to account number 1234'
    type='buy'
    amount=100
    max_order_amount=50
    min_order_amount=20
    payment_method='bank_transfer'
    rate=4.25
)
```

See also:

- [Binary API Docs for p2p_advert_create](#)

3.58 P2P Advert Information (p2p_advert_info)

Retrieve information about a P2P advert.

Auth Scope(s): payments

p2p_advert_info (*id: Optional[str] = None, subscribe: Optional[Union[bool, int]] = None, use_client_limits: Optional[int] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **id** (*Optional[str]*) – [Optional] The unique identifier for this advert. Optional when subscribe is 1. If not provided, all advertiser adverts will be subscribed.
- **subscribe** (*Optional[Union[bool, int]]*) – [Optional] If set to 1, will send updates when changes occur. Optional when id is provided.
- **use_client_limits** (*Optional[int]*) – [Optional] If set to 1, the maximum order amount will be adjusted to the current balance and turnover limits of the account.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type `int`

3.58.1 Example

```
binary.api.p2p_advert_info(  
    id='1234'  
)
```

See also:

- [Binary API Docs for p2p_advert_info](#)

3.59 P2P Advert List (`p2p_advert_list`)

Returns available adverts for use with `p2p_order_create`.

Auth Scope(s): `payments`

p2p_advert_list (*advertiser_id: Optional[str] = None, advertiser_name: Optional[str] = None, amount: Optional[Union[int, float, Decimal]] = None, counterparty_type: Optional[str] = None, favourites_only: Optional[int] = None, limit: Optional[int] = None, local_currency: Optional[str] = None, offset: Optional[int] = None, payment_method: Optional[List] = None, sort_by: Optional[str] = None, use_client_limits: Optional[int] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → `int`

Parameters

- **advertiser_id** (*Optional[str]*) – [Optional] ID of the advertiser to list adverts for.
- **advertiser_name** (*Optional[str]*) – [Optional] Search for advertiser by name. Partial matches will be returned.
- **amount** (*Optional[Union[int, float, Decimal]]*) – [Optional] How much to buy or sell, used to calculate prices.
- **counterparty_type** (*Optional[str]*) – [Optional] Filter the adverts by *counterparty_type*.
- **favourites_only** (*Optional[int]*) – [Optional] Only show adverts from favourite advertisers. Default is 0.
- **limit** (*Optional[int]*) – [Optional] Used for paging.
- **local_currency** (*Optional[str]*) – [Optional] Currency to conduct payment transaction in, defaults to the main currency for the client's country.
- **offset** (*Optional[int]*) – [Optional] Used for paging.
- **payment_method** (*Optional[List]*) – [Optional] Search by supported payment methods.
- **sort_by** (*Optional[str]*) – [Optional] How the results are sorted.
- **use_client_limits** (*Optional[int]*) – [Optional] If set to 1, ads that exceed this account's balance or turnover limits will not be shown.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.59.1 Example

```
binary.api.p2p_advert_list (
    counterparty_type='buy'
)
```

See also:

- [Binary API Docs for p2p_advert_list](#)

3.60 P2P Advert Update (p2p_advert_update)

Updates a P2P advert. Can only be used by the advertiser.

Auth Scope(s): payments

p2p_advert_update (*id: str, contact_info: Optional[str] = None, delete: Optional[int] = None, description: Optional[str] = None, is_active: Optional[int] = None, local_currency: Optional[str] = None, max_order_amount: Optional[Union[int, float, Decimal]] = None, min_order_amount: Optional[Union[int, float, Decimal]] = None, payment_info: Optional[str] = None, payment_method_ids: Optional[List] = None, payment_method_names: Optional[List] = None, rate: Optional[Union[int, float, Decimal]] = None, rate_type: Optional[str] = None, remaining_amount: Optional[Union[int, float, Decimal]] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **id** (*str*) – The unique identifier for this advert.
- **contact_info** (*Optional[str]*) – [Optional] Advertiser contact information.
- **delete** (*Optional[int]*) – [Optional] If set to 1, permanently deletes the advert.
- **description** (*Optional[str]*) – [Optional] General information about the advert.
- **is_active** (*Optional[int]*) – [Optional] Activate or deactivate the advert.
- **local_currency** (*Optional[str]*) – [Optional] Local currency for this advert.
- **max_order_amount** (*Optional[Union[int, float, Decimal]]*) – [Optional] Maximum allowed amount for the orders of this advert, in advertiser's *account_currency*. Should be more than or equal to *min_order_amount*.
- **min_order_amount** (*Optional[Union[int, float, Decimal]]*) – [Optional] Minimum allowed amount for the orders of this advert, in advertiser's *account_currency*. Should be less than or equal to *max_order_amount*.
- **payment_info** (*Optional[str]*) – [Optional] Payment instructions.
- **payment_method_ids** (*Optional[List]*) – [Optional] IDs of previously saved payment methods as returned from *p2p_advertiser_payment_methods*, only applicable for sell ads. Existing methods will be replaced.

- **payment_method_names** (*Optional[List]*) – [Optional] Payment method identifiers as returned from `p2p_payment_methods`, only applicable for buy ads. Existing methods will be replaced.
- **rate** (*Optional[Union[int, float, Decimal]]*) – [Optional] Conversion rate from advertiser's account currency to *local_currency*. An absolute rate value (fixed), or percentage offset from current market rate (floating).
- **rate_type** (*Optional[str]*) – [Optional] Type of rate, fixed or floating.
- **remaining_amount** (*Optional[Union[int, float, Decimal]]*) – [Optional] The total available amount of the advert, in advertiser's account currency.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns `req_id`

Return type `int`

3.60.1 Example

```
binary.api.p2p_advert_update(  
    id=1234  
    is_active=0  
)
```

See also:

- [Binary API Docs for p2p_advert_update](#)

3.61 P2P Chat Create (`p2p_chat_create`)

Creates a P2P chat for the specified order.

Auth Scope(s): `payments`

p2p_chat_create (*order_id: str, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → `int`

Parameters

- **order_id** (*str*) – The unique identifier for the order to create the chat for.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns `req_id`

Return type `int`

3.61.1 Example

```
binary.api.p2p_chat_create(
    order_id='1234'
)
```

See also:

- [Binary API Docs for p2p_chat_create](#)

3.62 P2P Order Cancel (p2p_order_cancel)

Cancel a P2P order.

Auth Scope(s): payments

p2p_order_cancel (*id: str, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **id** (*str*) – The unique identifier for this order.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.62.1 Example

```
binary.api.p2p_order_cancel(
    id='1234'
)
```

See also:

- [Binary API Docs for p2p_order_cancel](#)

3.63 P2P Order Confirm (p2p_order_confirm)

Confirm a P2P order.

Auth Scope(s): payments

p2p_order_confirm (*id: str, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **id** (*str*) – The unique identifier for this order.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type `int`

3.63.1 Example

```
binary.api.p2p_order_confirm(  
    id='1234'  
)
```

See also:

- [Binary API Docs for p2p_order_confirm](#)

3.64 P2P Order Create (`p2p_order_create`)

Creates a P2P order for the specified advert.

Auth Scope(s): `payments`

p2p_order_create (*advert_id: str, amount: Union[int, float, Decimal], contact_info: Optional[str] = None, payment_info: Optional[str] = None, payment_method_ids: Optional[List] = None, rate: Optional[Union[int, float, Decimal]] = None, subscribe: Optional[Union[bool, int]] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → `int`

Parameters

- **advert_id** (*str*) – The unique identifier for the advert to create an order against.
- **amount** (*Union[int, float, Decimal]*) – The amount of currency to be bought or sold.
- **contact_info** (*Optional[str]*) – [Optional] Seller contact information. Only applicable for ‘sell orders’.
- **payment_info** (*Optional[str]*) – [Optional] Payment instructions, only applicable for sell orders.
- **payment_method_ids** (*Optional[List]*) – IDs of payment methods, only applicable for sell orders.
- **rate** (*Optional[Union[int, float, Decimal]]*) – [Optional] Conversion rate from account currency to local currency, only applicable for floating rate adverts.
- **subscribe** (*Optional[Union[bool, int]]*) – [Optional] If set to 1, will send updates whenever there is an update to the order.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns `req_id`

Return type `int`

3.64.1 Example

```
binary.api.p2p_order_create(
    advert_id='1234'
    amount=100
)
```

See also:

- [Binary API Docs for p2p_order_create](#)

3.65 P2P Order Dispute (p2p_order_dispute)

Dispute a P2P order.

Auth Scope(s): payments

p2p_order_dispute (*dispute_reason: str, id: str, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **dispute_reason** (*str*) – The predefined dispute reason
- **id** (*str*) – The unique identifier for this order.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.65.1 Example

```
binary.api.p2p_order_dispute(
    dispute_reason='seller_not_released'
    id='1234'
)
```

See also:

- [Binary API Docs for p2p_order_dispute](#)

3.66 P2P Order Information (p2p_order_info)

Retrieves the information about a P2P order.

Auth Scope(s): payments

p2p_order_info (*id: str, subscribe: Optional[Union[bool, int]] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **id** (*str*) – The unique identifier for the order.

- **subscribe** (*Optional[Union[bool, int]]*) – [Optional] If set to 1, will send updates whenever there is an update to order
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.66.1 Example

```
binary.api.p2p_order_info(  
    id='1234'  
)
```

See also:

- [Binary API Docs for p2p_order_info](#)

3.67 P2P Order List (p2p_order_list)

List active orders.

Auth Scope(s): payments

p2p_order_list (*active: Optional[Union[int, float, Decimal]] = None, advert_id: Optional[str] = None, limit: Optional[int] = None, offset: Optional[int] = None, subscribe: Optional[Union[bool, int]] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **active** (*Optional[Union[int, float, Decimal]]*) – [Optional] Should be 1 to list active, 0 to list inactive (historical).
- **advert_id** (*Optional[str]*) – [Optional] If present, lists orders applying to a specific advert.
- **limit** (*Optional[int]*) – [Optional] Used for paging.
- **offset** (*Optional[int]*) – [Optional] Used for paging.
- **subscribe** (*Optional[Union[bool, int]]*) – [Optional] If set to 1, will send updates whenever there is a change to any order belonging to you.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.67.1 Example

```
binary.api.p2p_order_list(
    advert_id='1234'
)
```

See also:

- [Binary API Docs for p2p_order_list](#)

3.68 P2P Order Review (p2p_order_review)

Creates a review for the specified order.

Auth Scope(s): payments

p2p_order_review (*order_id: str, rating: int, recommended: Optional[int] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **order_id** (*str*) – The order identification number.
- **rating** (*int*) – Rating for the transaction, 1 to 5.
- **recommended** (*Optional[int]*) – [Optional] 1 if the counterparty is recommendable to others, otherwise 0.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.68.1 Example

```
binary.api.p2p_order_review(
    order_id='1234'
    rating=4
    recommended=1
)
```

See also:

- [Binary API Docs for p2p_order_review](#)

3.69 P2P Payment Methods (p2p_payment_methods)

List all P2P payment methods.

Auth Scope(s): payments

p2p_payment_methods (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id**Return type** int

3.69.1 Example

```
binary.api.p2p_payment_methods()
```

See also:

- [Binary API Docs for p2p_payment_methods](#)

3.70 P2P Ping (p2p_ping)

Keeps the connection alive and updates the P2P advertiser's online status. The advertiser will be considered offline 60 seconds after a call is made.

Auth Scope(s): payments

p2p_ping (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id**Return type** int

3.70.1 Example

```
binary.api.p2p_ping()
```

See also:

- [Binary API Docs for p2p_ping](#)

3.71 Payment agent create (paymentagent_create)

Saves client's payment agent details.

Auth Scope(s): admin

paymentagent_create (*code_of_conduct_approval: int, commission_deposit: Union[int, float, Decimal], commission_withdrawal: Union[int, float, Decimal], email: str, information: str, payment_agent_name: str, supported_payment_methods: List, urls: List, affiliate_id: Optional[str] = None, phone_numbers: Optional[List] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **code_of_conduct_approval** (*int*) – Indicates client's agreement with the Code of Conduct.
- **commission_deposit** (*Union[int, float, Decimal]*) – Commission (%) the agent wants to take on deposits
- **commission_withdrawal** (*Union[int, float, Decimal]*) – Commission (%) the agent wants to take on withdrawals
- **email** (*str*) – Payment agent's email address.
- **information** (*str*) – [Optional] Information about payment agent and their proposed service.
- **payment_agent_name** (*str*) – The name with which the payment agent is going to be identified.
- **supported_payment_methods** (*List*) – A list of supported payment methods.
- **urls** (*List*) – The URL(s) of payment agent's website(s).
- **affiliate_id** (*Optional[str]*) – [Optional] Client's My Affiliate id, if exists.
- **phone_numbers** (*Optional[List]*) – Payment agent's phone number(s) with country code.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.71.1 Example

```
binary.api.paymentagent_create(
    affiliate_id='1231234'
    code_of_conduct_approval=1
    commission_deposit=2
    commission_withdrawal=3
    email='joe@joy.com'
    information='The best person you can find'
    payment_agent_name='Joe Joy'
    phone_numbers=[{'phone_number': '+923-22-23-13'}]
    supported_payment_methods=[{'payment_method': 'MasterCard'}, {'payment_method':
↪ 'Visa'}]
    urls=[{'url': 'https://abc.com'}]
)
```

See also:

- [Binary API Docs for paymentagent_create](#)

3.72 Payment agent details (paymentagent_details)

Gets client's payment agent details.

Auth Scope(s): admin

paymentagent_details (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.72.1 Example

```
binary.api.paymentagent_details()
```

See also:

- [Binary API Docs for paymentagent_details](#)

3.73 Payment Agent: List (paymentagent_list)

Will return a list of Payment Agents for a given country for a given currency. Payment agents allow users to deposit and withdraw funds using local payment methods that might not be available via the main website's cashier system.

paymentagent_list (*paymentagent_list: str, currency: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **paymentagent_list** (*str*) – Client's 2-letter country code (obtained from *residence_list* call).
- **currency** (*Optional[str]*) – [Optional] If specified, only payment agents that supports that currency will be returned (obtained from *payout_currencies* call).
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.73.1 Example

```
binary.api.paymentagent_list(
    'id'
)
```

See also:

- [Binary API Docs for paymentagent_list](#)

3.74 Payment Agent: Transfer (paymentagent_transfer)

Payment Agent Transfer - this call is available only to accounts that are approved Payment Agents.

Auth Scope(s): payments

paymentagent_transfer (*amount: Union[int, float, Decimal], currency: str, transfer_to: str, description: Optional[str] = None, dry_run: Optional[int] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **amount** (*Union[int, float, Decimal]*) – The amount to transfer.
- **currency** (*str*) – Currency code.
- **transfer_to** (*str*) – The loginid of the recipient account.
- **description** (*Optional[str]*) – [Optional] Remarks about the transfer.
- **dry_run** (*Optional[int]*) – [Optional] If set to 1, just do validation.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.74.1 Example

```
binary.api.paymentagent_transfer(
    amount=1000
    currency='USD'
    transfer_to='CR100001'
)
```

See also:

- [Binary API Docs for paymentagent_transfer](#)

3.75 Payment Agent: Withdraw (paymentagent_withdraw)

Initiate a withdrawal to an approved Payment Agent.

Auth Scope(s): payments

paymentagent_withdraw (*amount: Union[int, float, Decimal], currency: str, paymentagent_loginid: str, verification_code: str, description: Optional[str] = None, dry_run: Optional[int] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **amount** (*Union[int, float, Decimal]*) – The amount to withdraw to the payment agent.
- **currency** (*str*) – The currency code.
- **paymentagent_loginid** (*str*) – The payment agent loginid received from the *paymentagent_list* call.
- **verification_code** (*str*) – Email verification code (received from a *verify_email* call, which must be done first)
- **description** (*Optional[str]*) – [Optional] Remarks about the withdraw. Only letters, numbers, space, period, comma, - ' are allowed.
- **dry_run** (*Optional[int]*) – [Optional] If set to 1, just do validation.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.75.1 Example

```
binary.api.paymentagent_withdraw(  
    amount=1000  
    currency='USD'  
    paymentagent_loginid='CR100001'  
    verification_code='my_verification_code'  
)
```

See also:

- [Binary API Docs for paymentagent_withdraw](#)

3.76 Payment Methods (payment_methods)

Will return a list payment methods available for the given country. If the request is authenticated the client's residence country will be used.

payment_methods (*country: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **country** (*Optional[str]*) – [Optional] 2-letter country code (ISO standard).
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id**Return type** int

3.76.1 Example

```
binary.api.payment_methods (
    country='id'
)
```

See also:

- [Binary API Docs for payment_methods](#)

3.77 Payout Currencies (payout_currencies)

Retrieve a list of available option payout currencies. If a user is logged in, only the currencies available for the account will be returned.

payout_currencies (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id**Return type** int

3.77.1 Example

```
binary.api.payout_currencies()
```

See also:

- [Binary API Docs for payout_currencies](#)

3.78 Ping (ping)

To send the ping request to the server. Mostly used to test the connection or to keep it alive.

ping (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id**Return type** int

3.78.1 Example

```
binary.api.ping()
```

See also:

- [Binary API Docs for ping](#)

3.79 Portfolio (portfolio)

Receive information about my current portfolio of outstanding options

Auth Scope(s): read, trading_information

portfolio (*contract_type: Optional[List] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **contract_type** (*Optional[List]*) – Return only contracts of the specified types
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.79.1 Example

```
binary.api.portfolio()
```

See also:

- [Binary API Docs for portfolio](#)

3.80 Profit Table (profit_table)

Retrieve a summary of account Profit Table, according to given search criteria

Auth Scope(s): read, trading_information

profit_table (*contract_type: Optional[List] = None, date_from: Optional[str] = None, date_to: Optional[str] = None, description: Optional[int] = None, limit: Optional[Union[int, float, Decimal]] = None, offset: Optional[Union[int, float, Decimal]] = None, sort: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **contract_type** (*Optional[List]*) – Return only contracts of the specified types
- **date_from** (*Optional[str]*) – [Optional] Start date (epoch or YYYY-MM-DD)
- **date_to** (*Optional[str]*) – [Optional] End date (epoch or YYYY-MM-DD)

- **description** (*Optional[int]*) – [Optional] If set to 1, will return full contracts description.
- **limit** (*Optional[Union[int, float, Decimal]]*) – [Optional] Apply upper limit to count of transactions received.
- **offset** (*Optional[Union[int, float, Decimal]]*) – [Optional] Number of transactions to skip.
- **sort** (*Optional[str]*) – [Optional] Sort direction.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.80.1 Example

```
binary.api.profit_table(
    description=1
    limit=25
    offset=25
    sort='ASC'
)
```

See also:

- [Binary API Docs for profit_table](#)

3.81 Price Proposal (proposal)

Gets latest price for a specific contract.

proposal (*contract_type: str, currency: str, symbol: str, amount: Optional[Union[int, float, Decimal]] = None, barrier: Optional[str] = None, barrier2: Optional[str] = None, barrier_range: Optional[str] = None, basis: Optional[str] = None, cancellation: Optional[str] = None, date_expiry: Optional[int] = None, date_start: Optional[int] = None, duration: Optional[int] = None, duration_unit: Optional[str] = None, limit_order=None, multiplier: Optional[Union[int, float, Decimal]] = None, product_type: Optional[str] = None, selected_tick: Optional[int] = None, subscribe: Optional[Union[bool, int]] = None, trading_period_start: Optional[int] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **contract_type** (*str*) – The proposed contract type
- **currency** (*str*) – This can only be the account-holder's currency (obtained from *pay-out_currencies* call).
- **symbol** (*str*) – The short symbol name (obtained from *active_symbols* call).
- **amount** (*Optional[Union[int, float, Decimal]]*) – [Optional] Proposed contract payout or stake, or multiplier (for lookbacks).

- **barrier** (*Optional[str]*) – [Optional] Barrier for the contract (or last digit prediction for digit contracts). Contracts less than 24 hours in duration would need a relative barrier (barriers which need +/-), where entry spot would be adjusted accordingly with that amount to define a barrier, except for Synthetic Indices as they support both relative and absolute barriers. Not needed for lookbacks.
- **barrier2** (*Optional[str]*) – [Optional] Low barrier for the contract (for contracts with two barriers). Contracts less than 24 hours in duration would need a relative barrier (barriers which need +/-), where entry spot would be adjusted accordingly with that amount to define a barrier, except for Synthetic Indices as they support both relative and absolute barriers. Not needed for lookbacks.
- **barrier_range** (*Optional[str]*) – [Optional] Barrier range for callputspread.
- **basis** (*Optional[str]*) – [Optional] Indicates type of the *amount*.
- **cancellation** (*Optional[str]*) – Cancellation duration option (only for *MULTUP* and *MULTDOWN* contracts).
- **date_expiry** (*Optional[int]*) – [Optional] Epoch value of the expiry time of the contract. Either *date_expiry* or *duration* is required.
- **date_start** (*Optional[int]*) – [Optional] Indicates epoch value of the starting time of the contract. If left empty, the start time of the contract is now.
- **duration** (*Optional[int]*) – [Optional] Duration quantity. Either *date_expiry* or *duration* is required.
- **duration_unit** (*Optional[str]*) – [Optional] Duration unit - *s*: seconds, *m*: minutes, *h*: hours, *d*: days, *t*: ticks.
- **limit_order** – Add an order to close the contract once the order condition is met (only for *MULTUP* and *MULTDOWN* contracts). Supported orders: *take_profit*, *stop_loss*.
- **multiplier** (*Optional[Union[int, float, Decimal]]*) – [Optional] The multiplier for non-binary options. E.g. lookbacks.
- **product_type** (*Optional[str]*) – [Optional] The product type.
- **selected_tick** (*Optional[int]*) – [Optional] The tick that is predicted to have the highest/lowest value - for *TICKHIGH* and *TICKLOW* contracts.
- **subscribe** (*Optional[Union[bool, int]]*) – [Optional] 1 - to initiate a realtime stream of prices. Note that tick trades (without a user-defined barrier), digit trades and less than 24 hours at-the-money contracts for the following underlying symbols are not streamed: *R_10*, *R_25*, *R_50*, *R_75*, *R_100*, *RDBULL*, *RDBEAR* (this is because their price is constant).
- **trading_period_start** (*Optional[int]*) – [Optional] Required only for multi-barrier trading. Defines the epoch value of the trading period start time.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.81.1 Example


```
binary.api.proposal(
    amount=100
    barrier='+0.1'
    basis='payout'
    contract_type='CALL'
    currency='USD'
    duration=60
    duration_unit='s'
    symbol='R_100'
)
```

See also:

- [Binary API Docs for proposal](#)

3.82 Price Proposal: Open Contracts (proposal_open_contract)

Get latest price (and other information) for a contract in the user's portfolio

Auth Scope(s): read, trading_information

proposal_open_contract (*contract_id: Optional[int] = None, subscribe: Optional[Union[bool, int]] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*)
→ int

Parameters

- **contract_id** (*Optional[int]*) – [Optional] Contract ID received from a *portfolio* request. If not set, you will receive stream of all open contracts.
- **subscribe** (*Optional[Union[bool, int]]*) – [Optional] 1 to stream.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the web-socket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.82.1 Example

```
binary.api.proposal_open_contract(
    contract_id=11111111
    subscribe=1
)
```

See also:

- [Binary API Docs for proposal_open_contract](#)

3.83 Reality Check (reality_check)

Retrieve summary of client's trades and account for the Reality Check facility. A 'reality check' means a display of time elapsed since the session began, and associated client profit/loss. The Reality Check facility is a regulatory

requirement for certain landing companies.

Auth Scope(s): `read, trading_information`

reality_check (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.83.1 Example

```
binary.api.reality_check()
```

See also:

- [Binary API Docs for reality_check](#)

3.84 Countries List (residence_list)

This call returns a list of countries and 2-letter country codes, suitable for populating the account opening form.

residence_list (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.84.1 Example

```
binary.api.residence_list()
```

See also:

- [Binary API Docs for residence_list](#)

3.85 Revoke Oauth Application (revoke_oauth_app)

Used for revoking access of particular app.

Auth Scope(s): `admin`

revoke_oauth_app (*revoke_oauth_app: int, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **revoke_oauth_app** (*int*) – The application ID to revoke.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.85.1 Example

```
binary.api.revoke_oauth_app(
    1234
)
```

See also:

- [Binary API Docs for revoke_oauth_app](#)

3.86 Sell Contract (sell)

Sell a Contract as identified from a previous *portfolio* call.

Auth Scope(s): trade

sell (*sell: int, price: Union[int, float, Decimal], passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **sell** (*int*) – Pass contract_id received from the *portfolio* call.
- **price** (*Union[int, float, Decimal]*) – Minimum price at which to sell the contract, or 0 for ‘sell at market’.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.86.1 Example

```
binary.api.sell(
    11542203588
    price=500
)
```

See also:

- [Binary API Docs for sell](#)

3.87 Sell Contracts: Multiple Accounts (`sell_contract_for_multiple_accounts`)

Sell contracts for multiple accounts simultaneously. Uses the shortcode response from `buy_contract_for_multiple_accounts` to identify the contract, and authorisation tokens to select which accounts to sell those contracts on. Note that only the accounts identified by the tokens will be affected. This will not sell the contract on the currently-authorised account unless you include the token for the current account.

Auth Scope(s): trade

`sell_contract_for_multiple_accounts` (*price: Union[int, float, Decimal], shortcode: str, tokens: List, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **`price`** (*Union[int, float, Decimal]*) – Minimum price at which to sell the contract, or 0 for ‘sell at market’.
- **`shortcode`** (*str*) – An internal ID used to identify the contract which was originally bought. This is returned from the `buy` and `buy_contract_for_multiple_accounts` calls.
- **`tokens`** (*List*) – Authorisation tokens which select the accounts to sell use for the affected accounts.
- **`passthrough`** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the `echo_req` output field.
- **`req_id`** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.87.1 Example

```
binary.api.sell_contract_for_multiple_accounts(  
    price=500  
    shortcode='CALL_R_50_5_1488181433_1488181553_S0P_0'  
    tokens=['FrvservuIFEf1', 'JUBibibkebiuwbeCNEc']  
)
```

See also:

- [Binary API Docs for sell_contract_for_multiple_accounts](#)

3.88 Sell Expired Contracts (`sell_expired`)

This call will try to sell any expired contracts and return the number of sold contracts.

Auth Scope(s): trade

`sell_expired` (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.88.1 Example

```
binary.api.sell_expired()
```

See also:

- [Binary API Docs for sell_expired](#)

3.89 Set Account Currency (set_account_currency)

Set account currency, this will be default currency for your account i.e currency for trading, deposit. Please note that account currency can only be set once, and then can never be changed.

Auth Scope(s): admin

set_account_currency (*set_account_currency: str, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **set_account_currency** (*str*) – Currency of the account. List of supported currencies can be acquired with *payout_currencies* call.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.89.1 Example

```
binary.api.set_account_currency(
    'USD'
)
```

See also:

- [Binary API Docs for set_account_currency](#)

3.90 Set Financial Assessment (set_financial_assessment)

This call sets the financial assessment details based on the client's answers to analyze whether they possess the experience and knowledge to understand the risks involved with binary options trading.

Auth Scope(s): admin

set_financial_assessment (*education_level: str, employment_industry: str, estimated_worth: str, income_source: str, net_income: str, occupation: str, account_turnover: Optional[str] = None, binary_options_trading_experience: Optional[str] = None, binary_options_trading_frequency: Optional[str] = None, cfd_trading_experience: Optional[str] = None, cfd_trading_frequency: Optional[str] = None, employment_status: Optional[str] = None, forex_trading_experience: Optional[str] = None, forex_trading_frequency: Optional[str] = None, other_instruments_trading_experience: Optional[str] = None, other_instruments_trading_frequency: Optional[str] = None, source_of_wealth: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **education_level** (*str*) – Level of Education.
- **employment_industry** (*str*) – Industry of Employment.
- **estimated_worth** (*str*) – Estimated Net Worth.
- **income_source** (*str*) – Income Source.
- **net_income** (*str*) – Net Annual Income.
- **occupation** (*str*) – Occupation.
- **account_turnover** (*Optional[str]*) – [Optional] The anticipated account turnover.
- **binary_options_trading_experience** (*Optional[str]*) – [Optional] Binary options trading experience.
- **binary_options_trading_frequency** (*Optional[str]*) – [Optional] Binary options trading frequency.
- **cfd_trading_experience** (*Optional[str]*) – [Optional] CFDs trading experience.
- **cfd_trading_frequency** (*Optional[str]*) – [Optional] CFDs trading frequency.
- **employment_status** (*Optional[str]*) – [Optional] Employment Status.
- **forex_trading_experience** (*Optional[str]*) – [Optional] Forex trading experience.
- **forex_trading_frequency** (*Optional[str]*) – [Optional] Forex trading frequency.
- **other_instruments_trading_experience** (*Optional[str]*) – [Optional] Trading experience in other financial instruments.
- **other_instruments_trading_frequency** (*Optional[str]*) – [Optional] Trading frequency in other financial instruments.
- **source_of_wealth** (*Optional[str]*) – [Optional] Source of wealth.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type `int`

3.90.1 Example

```
binary.api.set_financial_assessment(
    account_turnover='Less than $25,000'
    binary_options_trading_experience='1-2 years'
    binary_options_trading_frequency='40 transactions or more in the past 12 months'
    cfd_trading_experience='Over 3 years'
    cfd_trading_frequency='6-10 transactions in the past 12 months'
    education_level='Secondary'
    employment_industry='Finance'
    employment_status='Self-Employed'
    estimated_worth='$100,000 - $250,000'
    forex_trading_experience='Over 3 years'
    forex_trading_frequency='0-5 transactions in the past 12 months'
    income_source='Self-Employed'
    net_income='$25,000 - $50,000'
    occupation='Managers'
    other_instruments_trading_experience='Over 3 years'
    other_instruments_trading_frequency='6-10 transactions in the past 12 months'
    source_of_wealth='Company Ownership'
)
```

See also:

- [Binary API Docs for set_financial_assessment](#)

3.91 Set Self-Exclusion (set_self_exclusion)

Set Self-Exclusion (this call should be used in conjunction with `get_self_exclusion`)

Auth Scope(s): admin

set_self_exclusion (*exclude_until: Optional[str] = None, max_30day_deposit: Optional[Union[int, float, Decimal]] = None, max_30day_losses: Optional[Union[int, float, Decimal]] = None, max_30day_turnover: Optional[Union[int, float, Decimal]] = None, max_7day_deposit: Optional[Union[int, float, Decimal]] = None, max_7day_losses: Optional[Union[int, float, Decimal]] = None, max_7day_turnover: Optional[Union[int, float, Decimal]] = None, max_balance: Optional[Union[int, float, Decimal]] = None, max_deposit: Optional[Union[int, float, Decimal]] = None, max_losses: Optional[Union[int, float, Decimal]] = None, max_open_bets: Optional[int] = None, max_turnover: Optional[Union[int, float, Decimal]] = None, session_duration_limit: Optional[int] = None, timeout_until: Optional[int] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **exclude_until** (*Optional[str]*) – [Optional] Exclude me from the website (for a minimum of 6 months, up to a maximum of 5 years). Note: uplifting this self-exclusion may require contacting the company.
- **max_30day_deposit** (*Optional[Union[int, float, Decimal]]*) – [Optional] 7-day limit on deposits.

- **max_30day_losses** (*Optional[Union[int, float, Decimal]]*) – [Optional] 30-day limit on losses.
- **max_30day_turnover** (*Optional[Union[int, float, Decimal]]*) – [Optional] 30-day turnover limit.
- **max_7day_deposit** (*Optional[Union[int, float, Decimal]]*) – [Optional] 7-day limit on deposits.
- **max_7day_losses** (*Optional[Union[int, float, Decimal]]*) – [Optional] 7-day limit on losses.
- **max_7day_turnover** (*Optional[Union[int, float, Decimal]]*) – [Optional] 7-day turnover limit.
- **max_balance** (*Optional[Union[int, float, Decimal]]*) – [Optional] Maximum account cash balance.
- **max_deposit** (*Optional[Union[int, float, Decimal]]*) – [Optional] Daily deposit limit.
- **max_losses** (*Optional[Union[int, float, Decimal]]*) – [Optional] Daily limit on losses.
- **max_open_bets** (*Optional[int]*) – [Optional] Maximum number of open positions.
- **max_turnover** (*Optional[Union[int, float, Decimal]]*) – [Optional] Daily turnover limit.
- **session_duration_limit** (*Optional[int]*) – [Optional] Session duration limit, in minutes.
- **timeout_until** (*Optional[int]*) – [Optional] Exclude me from the website (for up to 6 weeks). Requires time in epoch format. Note: unlike *exclude_until*, this self-exclusion will be lifted automatically at the expiry of the timeout period.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.91.1 Example

```
binary.api.set_self_exclusion(  
    exclude_until='2020-01-01'  
    max_30day_deposit=1000  
    max_30day_losses=100000  
    max_30day_turnover=1000  
    max_7day_deposit=100  
    max_7day_losses=100000  
    max_7day_turnover=1000  
    max_deposit=10  
    max_losses=100000  
    max_turnover=1000  
    session_duration_limit=3600  
    timeout_until=1497357184  
)
```


See also:

- [Binary API Docs for set_self_exclusion](#)

3.92 Set Account Settings (set_settings)

Set User Settings (this call should be used in conjunction with *get_settings*)

Auth Scope(s): admin

set_settings (*account_opening_reason*: *Optional[str]* = *None*, *address_city*: *Optional[str]* = *None*, *address_line_1*: *Optional[str]* = *None*, *address_line_2*: *Optional[str]* = *None*, *address_postcode*: *Optional[str]* = *None*, *address_state*: *Optional[str]* = *None*, *allow_copiers*: *Optional[int]* = *None*, *citizen*: *Optional[str]* = *None*, *date_of_birth*: *Optional[str]* = *None*, *email_consent*: *Optional[int]* = *None*, *feature_flag*: *None*, *first_name*: *Optional[str]* = *None*, *last_name*: *Optional[str]* = *None*, *non_pep_declaration*: *Optional[int]* = *None*, *phone*: *Optional[str]* = *None*, *place_of_birth*: *Optional[str]* = *None*, *preferred_language*: *Optional[str]* = *None*, *request_professional_status*: *Optional[int]* = *None*, *residence*: *Optional[str]* = *None*, *salutation*: *Optional[str]* = *None*, *secret_answer*: *Optional[str]* = *None*, *secret_question*: *Optional[str]* = *None*, *tax_identification_number*: *Optional[str]* = *None*, *tax_residence*: *Optional[str]* = *None*, *passthrough*: *Optional[Any]* = *None*, *req_id*: *Optional[int]* = *None*) → int

Parameters

- **account_opening_reason** (*Optional[str]*) – [Optional] Purpose and reason for requesting the account opening. Only applicable for real money account. Required for clients that have not set it yet. Can only be set once.
- **address_city** (*Optional[str]*) – [Optional] Note: not applicable for virtual account. Required field for real money account.
- **address_line_1** (*Optional[str]*) – [Optional] Note: not applicable for virtual account. Required field for real money account.
- **address_line_2** (*Optional[str]*) – [Optional] Note: not applicable for virtual account. Optional field for real money account.
- **address_postcode** (*Optional[str]*) – [Optional] Note: not applicable for virtual account. Optional field for real money account.
- **address_state** (*Optional[str]*) – [Optional] Note: not applicable for virtual account. Optional field for real money account.
- **allow_copiers** (*Optional[int]*) – [Optional] Boolean value 1 or 0, indicating permission to allow others to follow your trades. Note: not applicable for Virtual account. Only allow for real money account.
- **citizen** (*Optional[str]*) – [Optional] Country of legal citizenship, 2-letter country code.
- **date_of_birth** (*Optional[str]*) – [Optional] Date of birth format: yyyy-mm-dd (can only be changed on unauthenticated svg accounts).
- **email_consent** (*Optional[int]*) – [Optional] Boolean value 1 or 0, indicating permission to use email address for any contact which may include marketing
- **feature_flag** – [Optional] Enable or disable one or multiple features.

- **first_name** (*Optional[str]*) – [Optional] Within 2-50 characters, use only letters, spaces, hyphens, full-stops or apostrophes (can only be changed on unauthenticated svg accounts).
- **last_name** (*Optional[str]*) – [Optional] Within 2-50 characters, use only letters, spaces, hyphens, full-stops or apostrophes (can only be changed on unauthenticated svg accounts).
- **non_pep_declaration** (*Optional[int]*) – [Optional] Indicates client's self-declaration of not being a PEP/RCA (Politically Exposed Person/Relatives and Close Associates). Effective for real accounts only.
- **phone** (*Optional[str]*) – [Optional] Note: not applicable for virtual account. Starting with + followed by 9-35 digits, hyphens or space.
- **place_of_birth** (*Optional[str]*) – [Optional] Place of birth, 2-letter country code.
- **preferred_language** (*Optional[str]*) – [Optional] User's preferred language, ISO standard language code
- **request_professional_status** (*Optional[int]*) – [Optional] Required when client wants to be treated as professional. Applicable for financial accounts only.
- **residence** (*Optional[str]*) – [Optional] 2-letter country code. Note: not applicable for real money account. Only allow for Virtual account without residence set.
- **salutation** (*Optional[str]*) – [Optional] Accept any value in enum list (can only be changed on unauthenticated svg accounts).
- **secret_answer** (*Optional[str]*) – [Optional] Answer to secret question, within 4-50 characters. Required for new account and existing client details will be used if client opens another account.
- **secret_question** (*Optional[str]*) – [Optional] Accept any value in enum list. Required for new account and existing client details will be used if client opens another account.
- **tax_identification_number** (*Optional[str]*) – [Optional] Tax identification number. Only applicable for real money account. Required for maltainvest landing company.
- **tax_residence** (*Optional[str]*) – [Optional] Residence for tax purpose. Comma separated iso country code if multiple jurisdictions. Only applicable for real money account. Required for maltainvest landing company.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.92.1 Example

```
binary.api.set_settings(  
    account_opening_reason='Speculative'  
    address_city='Test City'
```

(continues on next page)

(continued from previous page)

```

address_line_1='Test Address Line 1'
address_line_2='Test Address Line 2'
address_postcode='123456'
allow_copiers=1
email_consent=0
phone='+15417543010'
place_of_birth='ar'
preferred_language='EN'
request_professional_status=1
tax_identification_number='987654321'
tax_residence='hk'
)

```

See also:

- [Binary API Docs for set_settings](#)

3.93 Statement (statement)

Retrieve a summary of account transactions, according to given search criteria

Auth Scope(s): read, trading_information

statement (*action_type: Optional[str] = None, date_from: Optional[int] = None, date_to: Optional[int] = None, description: Optional[int] = None, limit: Optional[Union[int, float, Decimal]] = None, offset: Optional[Union[int, float, Decimal]] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **action_type** (*Optional[str]*) – [Optional] To filter the statement according to the type of transaction.
- **date_from** (*Optional[int]*) – [Optional] Start date (epoch)
- **date_to** (*Optional[int]*) – [Optional] End date (epoch)
- **description** (*Optional[int]*) – [Optional] If set to 1, will return full contracts description.
- **limit** (*Optional[Union[int, float, Decimal]]*) – [Optional] Maximum number of transactions to receive.
- **offset** (*Optional[Union[int, float, Decimal]]*) – [Optional] Number of transactions to skip.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.93.1 Example

```
binary.api.statement(  
    description=1  
    limit=100  
    offset=25  
)
```

See also:

- [Binary API Docs for statement](#)

3.94 States List (states_list)

For a given country, returns a list of States of that country. This is useful to populate the account opening form.

states_list (*states_list: str, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **states_list** (*str*) – Client's 2-letter country code (obtained from *residence_list* call)
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.94.1 Example

```
binary.api.states_list(  
    'id'  
)
```

See also:

- [Binary API Docs for states_list](#)

3.95 Ticks Stream (ticks)

Initiate a continuous stream of spot price updates for a given symbol.

ticks (*ticks: Union[List, str], subscribe: Optional[Union[bool, int]] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **ticks** (*Union[List, str]*) – The short symbol name or array of symbols (obtained from *active_symbols* call).
- **subscribe** (*Optional[Union[bool, int]]*) – [Optional] If set to 1, will send updates whenever a new tick is received.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.

- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.95.1 Example

```
binary.api.ticks(
    'R_50'
    subscribe=1
)
```

See also:

- [Binary API Docs for ticks](#)

3.96 Ticks History (ticks_history)

Get historic tick data for a given symbol.

ticks_history (*ticks_history: str, end: str, adjust_start_time: Optional[int] = None, count: Optional[int] = None, granularity: Optional[int] = None, start: Optional[int] = None, style: Optional[str] = None, subscribe: Optional[Union[bool, int]] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **ticks_history** (*str*) – Short symbol name (obtained from the *active_symbols* call).
- **end** (*str*) – Epoch value representing the latest boundary of the returned ticks. If *latest* is specified, this will be the latest available timestamp.
- **adjust_start_time** (*Optional[int]*) – [Optional] 1 - if the market is closed at the end time, or license limit is before end time, adjust interval backwards to compensate.
- **count** (*Optional[int]*) – [Optional] An upper limit on ticks to receive.
- **granularity** (*Optional[int]*) – [Optional] Only applicable for style: *candles*. Candle time-dimension width setting. (default: 60).
- **start** – [Optional] Epoch value representing the earliest boundary of the returned ticks.
- For “*style*”: “*ticks*”: this will default to 1 day ago.
- For “*style*”: “*candles*”: it will default to 1 day ago if count or granularity is undefined.

Parameters

- **style** (*Optional[str]*) – [Optional] The tick-output style.
- **subscribe** (*Optional[Union[bool, int]]*) – [Optional] 1 - to send updates whenever a new tick is received.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.96.1 Example

```
binary.api.ticks_history(  
    'R_50'  
    adjust_start_time=1  
    count=10  
    end='latest'  
    start=1  
    style='ticks'  
)
```

See also:

- [Binary API Docs for ticks_history](#)

3.97 Server Time (time)

Request back-end server epoch time.

time (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the web-socket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.97.1 Example

```
binary.api.time()
```

See also:

- [Binary API Docs for time](#)

3.98 Terms and Conditions Approval (tnc_approval)

To approve the latest version of terms and conditions.

Auth Scope(s): admin

tnc_approval (*affiliate_coc_agreement: Optional[int] = None, ukgc_funds_protection: Optional[int] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **affiliate_coc_agreement** (*Optional[int]*) – [Optional] For Affiliate's Code of Conduct Agreement.

- **ukgc_funds_protection** (*Optional[int]*) – [Optional] For *ASK_UK_FUNDS_PROTECTION* in *cashier*.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.98.1 Example

```
binary.api.tnc_approval()
```

See also:

- [Binary API Docs for tnc_approval](#)

3.99 Top Up Virtual-Money Account (topup_virtual)

When a virtual-money's account balance becomes low, it can be topped up using this call.

Auth Scope(s): trade

topup_virtual (*passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.99.1 Example

```
binary.api.topup_virtual()
```

See also:

- [Binary API Docs for topup_virtual](#)

3.100 Trading Durations (trading_durations)

Retrieve a list of all available underlyings and the corresponding contract types and trading duration boundaries. If the user is logged in, only the assets available for that user's landing company will be returned.

trading_durations (*landing_company: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **landing_company** (*Optional[str]*) – [Optional] If specified, will return only the underlyings for the specified landing company.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.100.1 Example

```
binary.api.trading_durations()
```

See also:

- [Binary API Docs for trading_durations](#)

3.101 Trading Platform: Investor Password Reset (trading_platform_investor_password_reset)

Reset the investor password of a Trading Platform Account

Auth Scope(s): admin

trading_platform_investor_password_reset (*account_id: str, new_password: str, platform: str, verification_code: str, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **account_id** (*str*) – Trading account ID.
- **new_password** (*str*) – New password of the account. For validation (Accepts any printable ASCII character. Must be within 8-25 characters, and include numbers, lowercase and uppercase letters. Must not be the same as the user's email address).
- **platform** (*str*) – Name of trading platform.
- **verification_code** (*str*) – Email verification code (received from a *verify_email* call, which must be done first)
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.101.1 Example


```
binary.api.trading_platform_investor_password_reset (
    account_id='MTR1000'
    new_password='InvestPwd123@!'
    platform='mt5'
    verification_code='abCD0199'
)
```

See also:

- [Binary API Docs for trading_platform_investor_password_reset](#)

3.102 Trading Platform: Password Reset (trading_platform_password_reset)

Reset the password of a Trading Platform Account

Auth Scope(s): admin

trading_platform_password_reset (*new_password: str, platform: str, verification_code: str, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **new_password** (*str*) – New password of the account. For validation (Accepts any printable ASCII character. Must be within 8-25 characters, and include numbers, lowercase and uppercase letters. Must not be the same as the user's email address).
- **platform** (*str*) – Name of trading platform.
- **verification_code** (*str*) – Email verification code (received from a *verify_email* call, which must be done first)
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.102.1 Example

```
binary.api.trading_platform_password_reset (
    new_password='C0rrect_p4ssword'
    platform='dxtrade'
    verification_code='abCD0199'
)
```

See also:

- [Binary API Docs for trading_platform_password_reset](#)

3.103 Server list (trading_servers)

Get the list of servers for a trading platform.

Auth Scope(s): read

trading_servers (*account_type: Optional[str] = None, environment: Optional[str] = None, market_type: Optional[str] = None, platform: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **account_type** (*Optional[str]*) – [Optional] Trading account type.
- **environment** (*Optional[str]*) – [Optional] Pass the environment (installation) instance. Currently, there are one demo and two real environments. Defaults to 'all'.
- **market_type** (*Optional[str]*) – [Optional] Market type.
- **platform** (*Optional[str]*) – [Optional] Pass the trading platform name, default to mt5
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.103.1 Example

```
binary.api.trading_servers(  
    platform='mt5'  
)
```

See also:

- [Binary API Docs for trading_servers](#)

3.104 Trading Times (trading_times)

Receive a list of market opening times for a given date.

trading_times (*trading_times: str, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **trading_times** (*str*) – Date to receive market opening times for. (yyyy-mm-dd format. *today* can also be specified).
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.104.1 Example

```
binary.api.trading_times(
    '2015-09-14'
)
```

See also:

- [Binary API Docs for trading_times](#)

3.105 Transactions Stream (transaction)

Subscribe to transaction notifications

Auth Scope(s): read, trading_information

transaction (*subscribe: Optional[Union[bool, int]], passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **subscribe** (*Optional[Union[bool, int]]*) – If set to 1, will send updates whenever there is an update to transactions. If not to 1 then it will not return any records.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.105.1 Example

```
binary.api.transaction(
    subscribe=1
)
```

See also:

- [Binary API Docs for transaction](#)

3.106 Transfer Between Accounts (transfer_between_accounts)

This call allows transfers between accounts held by a given user. Transfer funds between your fiat and cryptocurrency accounts (for a fee). Please note that *account_from* should be same as current authorized account.

Auth Scope(s): payments

transfer_between_accounts (*account_from: Optional[str] = None, account_to: Optional[str] = None, accounts: Optional[str] = None, amount: Optional[Union[int, float, Decimal]] = None, currency: Optional[str] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **account_from** (*Optional[str]*) – [Optional] The loginid of the account to transfer funds from.
- **account_to** (*Optional[str]*) – [Optional] The loginid of the account to transfer funds to.
- **accounts** (*Optional[str]*) – [Optional] To control the list of accounts returned when *account_from* or *account_to* is not provided. *brief* (default value) means that accounts with *mt5* *account_type* will be excluded; it will run faster. *all* means that all accounts with any *account_type* (including *mt5*) will be returned.
- **amount** (*Optional[Union[int, float, Decimal]]*) – [Optional] The amount to transfer.
- **currency** (*Optional[str]*) – [Optional] Currency code.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.106.1 Example

```
binary.api.transfer_between_accounts(  
    account_from='MLT100'  
    account_to='MF100'  
    amount=1000  
    currency='EUR'  
)
```

See also:

- [Binary API Docs for transfer_between_accounts](#)

3.107 Verify Email (verify_email)

Verify an email address for various purposes. The system will send an email to the address containing a security code for verification.

verify_email (*verify_email: str*, *type: str*, *url_parameters=None*, *passthrough: Optional[Any] = None*, *req_id: Optional[int] = None*) → int

Parameters

- **verify_email** (*str*) – Email address to be verified.
- **type** (*str*) – Purpose of the email verification call.
- **url_parameters** – [Optional] Extra parameters that can be attached to the verify email link URL.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.107.1 Example

```
binary.api.verify_email(
    'test@mailinator.com'
    type='account_opening'
)
```

See also:

- [Binary API Docs for verify_email](#)

3.108 Server Status (website_status)

Request server status.

website_status (*subscribe: Optional[Union[bool, int]] = None, passthrough: Optional[Any] = None, req_id: Optional[int] = None*) → int

Parameters

- **subscribe** (*Optional[Union[bool, int]]*) – [Optional] 1 to stream the server/website status updates.
- **passthrough** (*Optional[Any]*) – [Optional] Used to pass data through the websocket, which may be retrieved via the *echo_req* output field.
- **req_id** (*Optional[int]*) – [Optional] Used to map request to response.

Returns req_id

Return type int

3.108.1 Example

```
binary.api.website_status()
```

See also:

- [Binary API Docs for website_status](#)

CHAPTER 4

Contributors

- Abdullah Mallik (@mdn522) (maintainer)

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`

A

`active_symbols()`, 5
`api_token()`, 6
`app_delete()`, 6
`app_get()`, 7
`app_list()`, 7
`app_markup_details()`, 8
`app_register()`, 9
`app_update()`, 10
`asset_index()`, 11
`authorize()`, 12

B

`balance()`, 12
`buy()`, 13
`buy_contract_for_multiple_accounts()`, 14

C

`cancel()`, 15
`cashier()`, 15
`contract_update()`, 17
`contract_update_history()`, 17
`contracts_for()`, 16
`copy_start()`, 19
`copy_stop()`, 20
`copytrading_list()`, 18
`copytrading_statistics()`, 19
`crypto_config()`, 20

D

`document_upload()`, 21

E

`economic_calendar()`, 22
`exchange_rates()`, 23

F

`forget()`, 23

`forget_all()`, 24

G

`get_account_status()`, 24
`get_financial_assessment()`, 25
`get_limits()`, 25
`get_self_exclusion()`, 26
`get_settings()`, 26

I

`identity_verification_document_add()`, 27

L

`landing_company()`, 27
`landing_company_details()`, 28
`login_history()`, 29
`logout()`, 29

M

`mt5_deposit()`, 30
`mt5_get_settings()`, 30
`mt5_login_list()`, 31
`mt5_new_account()`, 31
`mt5_password_change()`, 33
`mt5_password_check()`, 34
`mt5_password_reset()`, 34
`mt5_withdrawal()`, 35

N

`new_account_maltainvest()`, 36
`new_account_real()`, 39
`new_account_virtual()`, 41

O

`oauth_apps()`, 43

P

`p2p_advert_create()`, 48

p2p_advert_info(), 49
p2p_advert_list(), 50
p2p_advert_update(), 51
p2p_advertiser_adverts(), 44
p2p_advertiser_create(), 44
p2p_advertiser_info(), 45
p2p_advertiser_payment_methods(), 46
p2p_advertiser_relations(), 46
p2p_advertiser_update(), 47
p2p_chat_create(), 52
p2p_order_cancel(), 53
p2p_order_confirm(), 53
p2p_order_create(), 54
p2p_order_dispute(), 55
p2p_order_info(), 55
p2p_order_list(), 56
p2p_order_review(), 57
p2p_payment_methods(), 57
p2p_ping(), 58
payment_methods(), 62
paymentagent_create(), 58
paymentagent_details(), 60
paymentagent_list(), 60
paymentagent_transfer(), 61
paymentagent_withdraw(), 61
payout_currencies(), 63
ping(), 63
portfolio(), 64
profit_table(), 64
proposal(), 65
proposal_open_contract(), 67

R

reality_check(), 68
residence_list(), 68
revoke_oauth_app(), 68

S

sell(), 69
sell_contract_for_multiple_accounts(),
70
sell_expired(), 70
set_account_currency(), 71
set_financial_assessment(), 72
set_self_exclusion(), 73
set_settings(), 75
statement(), 77
states_list(), 78

T

ticks(), 78
ticks_history(), 79
time(), 80
tnc_approval(), 80

topup_virtual(), 81
trading_durations(), 81
trading_platform_investor_password_reset(),
82
trading_platform_password_reset(), 83
trading_servers(), 84
trading_times(), 84
transaction(), 85
transfer_between_accounts(), 85

V

verify_email(), 86

W

website_status(), 87