

Basic	Basic authentication	Basic API Authentication
Trading	Headers	
User	API-Key String Required	Public API Key Examples ASDFGHJKL1234567890
	API-Nonce String Required	A strictly increasing number Examples 4200 4201 4202
	API-Secret String Required	Secret API Key Examples VdSaSgDl82gNHcevBT9zkgajperibsz3XbqmFrXeAF
	API-Signature String Required	A signature of the request Examples ASDFGHJKL1234567890ASDFGHJKL1234567890

Schemas

Blacklist

A Blacklist is a filter used to exclude models from a search result and is used to remove criteria from investment strategies.

▼ Object

id String Required	UUID
obj_id String Required	UUID of model instance targeted for exclusion
parent_id Integer Required	UUID of the object that created this Blacklist
parent_type String Required	Model of the object that created this Blacklist Examples exchanges markets xmarkets currencies
status String Optional	The current state of the Blacklist Examples enabled disabled
type String Required	Model to be excluded from results
user_id String Required	UUID of the User this belongs to

Configuration

A Configuration is a `key:val` pair with additional properties exposed for improved control over Configuration of any Schema. Each Configuration has a `parent_id` and `parent_type` property that define the Instance ID and Schema, respectively, this Configuration is applied to.

 ▼ Object

id String Required	UUID
parent_id Integer Required	UUID of the object this Configuration is applied to.
parent_type String Required	Model of the object that created this Configuration. Examples exchanges markets xmarkets currencies
status String Required	The current state of the Configuration. Examples enabled disabled
var String Required	The name of the setting this Configuration reflects.
val String Required	The value of the setting this Configuration reflects.
user_id String Required	UUID of the User this belongs to
created Time Required	Unix Epoch time stamp of creation expressed in seconds
updated Time Required	Unix Epoch time stamp of last update expressed in seconds

Deposit

A Deposit is an intent to deposit a given Asset into kek.io via sending to `Deposit::address`.

▼ Object

id String Required	UUID
user_id String Required	UUID of the User this belongs to
address String Required	BTC Deposit Address
amount_received Number Required	The amount of <code>Deposit::asset</code> received. Examples 4.2
amount_requested Number Required	The amount of <code>Deposit::asset</code> requested for deposit. Examples 4
asset String Required	The Asset being deposited. Default BTC Examples BTC ETH
status String Required	The current state of the Deposit. Examples waiting pending confirming complete error
created Time Required	Unix Epoch time stamp of creation expressed in seconds
updated Time Required	Unix Epoch time stamp of last update expressed in seconds

Investor

Investor is an automated cryptocurrency trading bot that attempts to make “statistically favorable” trades by buying into markets that match a set of configuration. The Investor also includes Simulation/Backtesting methods that allow users to simulate a set of configuration in the Live Market or by using Historical Market Data.

A “statistically favorable” market is one that meets:

- **Profile** Configuration requirements
 - Mean Reversion Offset
 - Minimum 24 Hour Trade Volume
 - Minimum Profit
 - Maximum Profit
 - Maximum Loss
 - Time To Live (TTL) Settings
- **Blacklist** Configuration requirements
 - Currency Blacklist
 - Exchange Blacklist
 - Market Blacklist
 - XMarket Blacklist
- **Range** Configuration requirements

Each set of configuration is called a **Profile**.

A Profile:

- Has a name
- Can be managed using Kek.io
- Automatically executes a trading strategy based on its own set of Configuration
- Keeps track of and displays Simulation/Trade Data

Each attempted set of trades is called an **Investor**.

An Investor:

- Spawns from a Profile when an XMarket meets all Configuration requirements
- Is a set of two trades:
 - Buy Order - Buy market base_currency with market quote_currency
 - Sell Order - Sell market base_currency for market quote_currency
- Will persist until profit, loss or cancelled
- Will use the Configuration its Profile had at the time of creation and will use this set of configuration for its lifespan, even if the Profile Configuration is changed
- Attempts to start in every market, every minute
- Is isolated from all other Investors, except when Wallets are lacking required funds
- Is considered "dead" (complete) when status is profit, loss or cancelled

▼ Object

id String Required	UUID
parent_id Integer Required	UUID of the object that created this Investor
parent_type String Required	Model of the object that created this Investor Examples profiles users
buy_id String Required	Order::id of the Order this Investor used to buy into its XMarket

debug String Required	Log Debug Level Examples ERROR WARN INFO VERBOSE DISABLED
delta Number Required	The proportional value movement this Investor caused, where a value of 1.0 represents a 100% gain on START_AMOUNT initial investment.
engine String Required	The Investor Engine this Investor is configured to use
enter_price Number Required	The rate at which this Investor bought into its XMarket
enter_volume Number Required	The XMarket::volume_h24 at the time this Investor bought into its XMarket
exchange_id String Required	Exchange::id of the Exchange this Investor resides on.
exit_price Number Required	The rate at which this Investor sold out of its XMarket
exit_volume Number Required	The XMarket::volume_h24 at the time this Investor sold out of its XMarket
heartbeat_count Integer Required	The amount of Investor Heartbeat Invocations this Investor has sustained in its current status . When an Investor changes its status , this counter resets to zero.
heartbeat_total_count Integer Required	The amount of Investor Heartbeat Invocations this Investor has sustained since creation.
market_id String Required	Market::id of the Market this Investor resides in.
sell_id String Required	Order::id of the Order this Investor used to sell out of its XMarket

status String Required	The current state of the Investor Examples <code>initializing</code> <code>sparking</code> <code>waiting</code> <code>success</code> <code>exiting</code> <code>gtfo</code> <code>cancelled</code> <code>loss</code> <code>profit</code> <code>error</code>
user_id String Required	User:: <code>id</code> of the User this Investor belongs to.
xmarket_id String Required	XMarket:: <code>id</code> of the XMarket this Investor resides in.
created Time Required	Unix Epoch time stamp of creation expressed in seconds
updated Time Required	Unix Epoch time stamp of last update expressed in seconds

Job

An Investor Job

 ▼ Object

id String Required	UUID
user_id String Required	UUID of the User that invoked this request
args String Required	Arguments specific to this Investor method (if any)
headers String Required	Encrypted HTTP Headers associated with this Job
method String Required	Investor method that should be invoked
response ▼ Object Required	Job Response Object
status Integer Required	A status code reflecting the result of the request
message String Required	A human-friendly message explaining the result of the request
data ▼ Object Optional	Data that is expected to be in the result of the request (if any)
timestamp Number Required	Unix Epoch time stamp expressed in milliseconds
status String Required	The current status of this job
created Time Required	Unix Epoch time stamp of creation expressed in seconds
elapsed Integer Required	The amount of milliseconds the actual Job Execution required
updated Time Required	Unix Epoch time stamp of last update expressed in seconds

Order

An Order is an intent to buy or sell an Asset in exchange for another Asset and is managed by 3rd party Exchange Partners.

▼ Object

id String Required	UUID
__r String Required	Raw Order HTTP Response Body supplied by XAPI
_mid String Required	UUID of the Market this Order was placed in
_oid String Required	Raw Order UUID supplied by XAPI
_uid String Required	UUID of the User this Order belongs to
_xid String Required	UUID of the Exchange this Order was placed on Examples bitfinex kraken
method String Required	The method that should be used to fill the Order Examples limit market
options ▼ Object Optional	Assorted metadata and advanced options (specific to XAPI).
price Number Required	The rate that should be used in the Order, expressed in terms of 1 Base per X Quote, where X is the price
status String Required	The current state of the Order Examples sparking partial filled cancelled
type String Optional	The direction of this Order Examples buy sell

volume Number Required	The maximum amount that can be executed for this Order
volume_exec Number Required	The amount that has currently been executed for this Order
created Time Required	Unix Epoch time stamp of creation expressed in seconds
updated Time Required	Unix Epoch time stamp of last update expressed in seconds

Profile

A Profile is an automated trading strategy.

 ▼ Object

id String Required	UUID
user_id String Required	UUID of the User this belongs to
profile_name String Required	Display name Examples My First Profile ERC-20 FKnife-01 Blaze It, Boyz
engine String Required	The Investor Engine this Profile is configured to use Examples sim trading
status String Required	The current state of the Profile Examples initializing running stopped error destroyed frozen
created Time Required	Unix Epoch time stamp of creation expressed in seconds
updated Time Required	Unix Epoch time stamp of last update expressed in seconds

Response

All API requests will return a Response Object.

▼ Object

status Integer Required	A status code reflecting the result of the request
message String Required	A human-friendly message explaining the result of the request
data ▼ Object Optional	Data that is expected to be in the result of the request (if any)
timestamp Integer Required	Unix Epoch time stamp expressed in milliseconds

Trade

A Trade is a confirmed exchange of one Asset for another of a certain amount at a given rate.

▼ Object

id String Required	UUID
user_id String Required	UUID of the User this belongs to
order_id String Required	UUID of the Order that created this Trade
exchange_id Integer Required	UUID of the Exchange this Trade occurred on
market_id String Required	UUID of the Market this Trade occurred in Examples shapeshift bitfinex
xmarket_id String Optional	UUID of the XMarket this Trade occurred in Examples kraken(BTC_USD) bitfinex(BTC_ETH)
price Number Required	The rate that was used in the Trade, expressed in terms of 1 Base per X Quote, where X is the
volume Number Required	The amount exchanged in the Trade, expressed in terms of Base
created Time Required	Unix Epoch time stamp of creation expressed in seconds
updated Time Required	Unix Epoch time stamp of last update expressed in seconds

Wallet

A Wallet is a virtual Asset container residing on a 3rd party service and accessed via InvestorAPI Authentication.

 ▼ Object

id String Required	UUID
user_id String Required	UUID of the User this belongs to
balance Number Required	The amount of this <code>Wallet::currency</code> currently available in this Wallet
currency String Required	The <code>Asset::id</code> this Wallet holds Examples BTC ETH
exchange String Required	UUID of the Exchange this Wallet resides on Examples bitfinex kraken
status String Optional	The current state of the Wallet Examples enabled disabled frozen xapi_error
type String Required	The category this Wallet resides on Examples exchange trading
created Time Required	Unix Epoch time stamp of creation expressed in seconds
updated Time Required	Unix Epoch time stamp of last update expressed in seconds

Whitelist

A Whitelist is a filter used to include models in a search result that would otherwise be excluded due to Blacklists. Whitelists override Blacklists and give a detailed level of control over trading strategies.

 ▼ Object

id String Required	UUID
obj_id String Required	UUID of model instance targeted for inclusion
parent_id Integer Required	UUID of the object that created this Whitelist
parent_type String Required	Model of the object that created this Whitelist Examples exchanges markets xmarkets currencies
status String Optional	The current state of the Whitelist Examples enabled disabled
type String Required	Model to be included in results
user_id String Required	UUID of the User this belongs to

Withdrawal

A Withdrawal is an intent to withdraw a given Asset out of kek.io by sending to `Withdrawal::address`.

▼ Object

id String Required	UUID
user_id String Required	UUID of the User this belongs to
address String Required	BTC Deposit Address
amount_received Number Required	The amount of <code>Withdrawal::asset</code> received. Examples 4.2
amount_requested Number Required	The amount of <code>Withdrawal::asset</code> requested for deposit. Examples 4
asset String Required	The Asset being withdrawn. Default BTC Examples BTC ETH
status String Required	The current state of the Withdrawal. Examples waiting pending confirming complete error
created Time Required	Unix Epoch time stamp of creation expressed in seconds
updated Time Required	Unix Epoch time stamp of last update expressed in seconds

Blacklist Methods

Create a Blacklist

POST /blacklists/

A Blacklist is an excluded object in all Investor Actions. This means that you can exclude a certain set of things from your strategy, such as excluding kraken(BTC_USD) or all BTC markets, etc.

Request

Body

application/json

Type

▼ Object

id String Required	UUID of the object that should be excluded. Examples <code>kraken</code> <code>BTC_USD</code> <code>kraken(BTC_USD)</code>
parent_id String Required	UUID of the object that this Configuration will belong to.
parent_type String Optional	Model of the object that this Configuration will belong to. Default <code>profiles</code>
type String Required	Model of the object that should be excluded. Examples <code>exchange</code> <code>market</code> <code>xmarket</code>

Responses

200

Body application/json

Type

Response

Delete a Blacklist

DELETE /blacklists/

Delete an existing Blacklist

Request

Query Parameters

id String Required	<p>UUID of the object that is excluded.</p> <p>Examples kraken BTC_USD kraken(BTC_USD)</p>
parent_id String Required	<p>UUID of the object that whose Configuration should be deleted.</p> <p>Examples kek_investor_gotem</p>
parent_type String Optional	<p>Model of the object whose Configuration should be deleted.</p> <p>Default profiles</p>
type String Required	<p>Model of the object that is excluded.</p> <p>Examples exchange market xmarket</p>

Responses

200

Body application/json

Type

Response

Get a Blacklist

GET /blacklists/

Retrieve an existing Blacklist.

Request

Query Parameters

id String Required	<p>UUID of the object that is excluded.</p> <p>Examples kraken BTC_USD kraken(BTC_USD)</p>
parent_id String Required	<p>UUID of the object that whose Configuration should be retrieved.</p> <p>Examples kek_investor_gotem</p>
parent_type String Optional	<p>Model of the object whose Configuration should be retrieved.</p> <p>Default profiles</p>
type String Required	<p>Model of the object that should be excluded.</p> <p>Examples exchange market xmarket</p>

Responses

200	Body application/json
	Type
	Response

Get All Blacklists

GET /blacklists/get/all

Retrieve all existing Blacklists belonging a Parent.

Security

Basic, Trading

Request

Query Parameters

parent_id String Required	UUID of the object that this Blacklist belongs to.
-------------------------------------	--

parent_type String Optional	Model of the object that this Blacklist belongs to. Default profiles
--------------------------------	---

Responses

200	Body application/json
	Type
	Response

Configuration Methods

Configuration CRUD Operations

Create a Configuration

POST /configure/

A Configuration is a `key:val` pair with additional properties exposed for improved control over Configuration of any Schema. Each Configuration has a `parent_id` and `parent_type` property that define the Instance ID and Schema, respectively, this Configuration is applied to.

Request

Body

application/json

Type

▼ Object

<code>parent_id</code> String Required	UUID of the object that this Configuration will belong to.
--	--

<code>var</code> String Required	Name of the setting that should be created. Examples <code>START_AMOUNT</code> <code>RANGE_h14</code>
--	---

<code>val</code> String Required	Value of the setting. Examples <code>4.20</code> <code>-5:5</code>
--	--

<code>parent_type</code> String Optional	Model of the object that this Configuration will belong to. Default <code>profiles</code>
--	--

Responses

200	Body	application/json
<hr/>		
Type		
<hr/>		
Response		

Delete a Configuration

DELETE /configure/

Delete an existing Configuration

Request

Query Parameters

parent_id String Required	UUID of the object that whose Configuration should be deleted. Examples: kek_investor_gotem
var String Required	Name of the setting that should be deleted. Examples: RANGE_d14
parent_type String Optional	Model of the object whose Configuration should be deleted. Default: profiles

Responses

200	Body	application/json
<hr/>		
Type		
<hr/>		
Response		

Get a Configuration

GET /configure/

Retrieve an existing Configuration.

Request

Query Parameters

parent_id String Required	UUID of the object that whose Configuration should be retrieved. Examples <code>kek_investor_gotem</code>
parent_type String Optional	Model of the object whose Configuration should be retrieved. Default <code>profiles</code>
var String Required	Name of the setting that should be retrieved. Examples <code>START_AMOUNT</code> <code>RANGE_d7</code>

Responses

200

Body `application/json`**Type**

Response

Update a Configuration

PUT `/configure/`

Update an existing Configuration with the given request data.

Request

Query Parameters

parent_id String Required	UUID of the object that whose Configuration should be updated. Examples <code>kek_investor_gotem</code>
--	--

var String Required	Name of the setting that should be updated. Examples <code>START_AMOUNT</code> <code>RANGE_d5</code>
--------------------------------------	--

val String Required	New value of the setting. Examples <code>4.20</code> <code>0.1</code>
--------------------------------------	---

parent_type String Optional	Model of the object whose Configuration should be updated. Default <code>profiles</code>
--	---

Responses

200

Body	<code>application/json</code>
-------------	-------------------------------

Type

Response

Clone ConfigurationGET `/configure/clone`Delete all existing configuration in Target and clone all configuration tied to a given `parent_id` and `parent_type` (Source) into Target.**Request**

Query Parameters

parent_id String Required	UUID of the object that contains the configuration desired to be cloned
parent_type String Required	Model of the object that contains the configuration desired to be cloned
target_id String Required	UUID of the object that the cloned configuration should be applied to. Examples kek_investor_gotem
target_type String Required	Model of the object that the cloned configuration should be applied to.
wipe Boolean Optional	Determines if the Clone should first destroy all Configuration belonging to the Target before applying the cloned configuration. This is disabled by default and can be enabled by setting this to <code>true</code> . Default false

Responses

200	Body	application/json
	Type	
	Response	

Empty Configuration

GET /configure/empty

Destroy all configuration tied to a given parent_id and parent_type .

Request

Query Parameters

parent_id String Required	UUID of the object that contains the configuration desired to be emptied Examples <code>kek_investor_gotem</code>
parent_type String Optional	Model of the object that contains the configuration desired to be emptied Default <code>profiles</code>
result Boolean Optional	Determines if the <code>Response::data</code> should contain an additional key, <code>result</code> , that will contain an array of all destroyed configuration. Default <code>false</code>

Responses

200	Body <code>application/json</code>
	Type
	Response

Get All Configuration

GET `/configure/get/all`

Retrieve all existing Configuration belonging a Parent.

Security

Basic, Trading

Request

Query Parameters

parent_id	UUID of the object that this Configuration will belong to.
String Required	

parent_type	Model of the object that this Configuration will belong to.
String Optional	Default profiles

Responses

200

Body	application/json
------	------------------

Type

Response

Deposit Methods

Create a Deposit

POST /deposits/

A Deposit represents an intent to send a given Asset into your Kek.io Wallet by sending it to a specified address.

Security

Basic

Request

Body

application/json

Type

▼ Object

amount
Number **Required**
The amount being sent.
Examples 4.2

asset
String **Required**
The Asset being sent.
Default BTC
Examples ETH

Responses

200

Body

application/json

Type

Response

Cancel a Deposit

DELETE /deposits/

Delete an existing Deposit

Security

Basic

Request

Query Parameters

id	UUID of the Deposit that should be cancelled.
String Required	

Responses

200

Body application/json

Type

Response

Get a Deposit

GET /deposits/

Retrieve an existing Deposit.

Security

Basic

Request

Query Parameters

id	UUID of the Deposit being retrieved.
String Required	

Responses

200

Body

application/json

Type

Response

Investor Methods

Create an Investor

POST /investors/{id}

Create a new Investor, a one-time automated trading strategy.

Investor States

- **initializing**
 - An initializing Investor has just been created and is creating its initial Buy Order.
- **sparking**
 - A sparking Investor has placed its initial Buy Order but it is not executed yet and is now waiting for one of the following:
 - Buy Order is executed
 - SPARKING_TTL minutes go by
- **waiting**
 - A waiting Investor has executed its initial Buy Order and is now waiting for one of the following:
 - SUCCESS_THRESHOLD - Minimum % Profit
 - WAITING_TTL - # of minutes Investor can be waiting before timing out
 - MAX_LOSS - Maximum % Loss
 - This is the equivalent of the beginning of an "Investment", as the Investor has just bought something with the hope that it will increase in price based on its Configuration
- **success**
 - A success Investor has hit its SUCCESS_THRESHOLD
 - A success Investor is in a state of maximum potential profit and in most cases will always end in the profit state
 - This means the Investor has already met its minimum % profit and will stay in the market until:
 - SUCCESS_STOP_THRESHOLD - Maximum profit decrease after entering success
 - SUCCESS_TTL - # of minutes Investor can be success before timing out
- **exiting**
 - An exiting Investor has completed its waiting state and is now attempting to exit the market using a Limit Order
 - This means a Sell Order has been placed that will sell whatever it bought at the price of the last sale that occurred
 - Both profit and loss Investor go through the exiting state after either SUCCESS_THRESHOLD OR a TTL setting is hit
 - If EXITING_TTL is met, the Profile will switch to a more aggressive exiting strategy called gtfo
- **gtfo**
 - A gtfo Investor needs to exit a market asap to ensure profit or minimize loss
 - This state will exit more aggressively by creating a Market Order
- **cancelled (dead)**
 - A cancelled Investor has been cancelled
 - This can occur in multiple cases:
 - API Errors
 - Insufficient Funds
 - User Interaction (manual withdrawal or cancel of orders, etc.)
- **loss (dead)**
 - A loss Investor that resulted in an overall loss):
- **profit (dead)**
 - A profit Investor that resulted in an overall profit 😊

Security

Basic, Trading

Request

Path Variables

id	Investor UUID
String Required	Examples kek_investor_gotem

Body application/json

Type

▼ Object

config	The configuration that will be applied to the new Investor.
▼ Object Required	

EXITING_TTL	The amount of heartbeat invocations an Investor in state, <code>exiting</code> , can sustain before dying.
Integer Required	

GREED_THRESHOLD	The maximum percentage profit an Investor can observe before entering state <code>gtfo</code> .
Integer Optional	A value of <code>10</code> means "After you have made 10% profit, you need to exit the Market immediately so we don't lose our gains".

GTFO_TTL	The amount of heartbeat invocations an Investor in state, <code>gtfo</code> , can sustain before dying.
Integer Required	

MAX_LOSS	The maximum percentage loss an Investor will sustain before changing state to <code>gtfo</code> and exiting the market asap.
Number Required	If a <code>MAX_LOSS</code> of <code>-1.0</code> is set, all Investors created by this Profile will <code>gtfo</code> when $(Investor::delta * 100) > (START_AMOUNT * (1 - MAX_LOSS))$.
	This means that a value of <code>1.0</code> is "exit after 1% loss".
	Maximum <code>-0.000001</code>

<p>MEAN_REVERSION_OFFSETS Number Optional</p>	<p>Mean Reversion is the concept that all prices will fluctuate over time.</p> <p>But as a given price gets further from its <i>Mean</i> (calculated average over x time), the probability of it reversing its current trend (+ or -) and instead begin trending towards its <i>Mean</i> becomes exponentially more likely.</p> <p>MEAN_REVERSION_OFFSET defines the required minimum percentage difference between a given XMarket Price and its Mean for it to be considered a valid XMarket.</p> <p>A value of 10 means "Not allowed to buy into any XMarket unless its current price is at least 10% LESS THAN its Mean price".</p> <p>A negative value, such as -10, means "Not allowed to buy into any XMarket unless its current price is at least 10% GREATER THAN its Mean price".</p>
<p>MIN_VOLUME_24H Number Required</p>	<p>The minimum 24 hour volume expressed in BTC an XMarket must have to allow Investors to buy in.</p> <p>If a given XMarket::base_currency is not BTC, its 24 Hour Base Volume will be converted to Bitcoin when compared to this setting.</p>
<p>SPARKING_TTL Integer Required</p>	<p>The amount of heartbeat invocations an Investor in state, <code>sparking</code>, can sustain before dying.</p>
<p>START_AMOUNT Number Required</p>	<p>The amount of BTC each Investor will "invest" into an XMarket.</p> <p>A value of 0.1 means "Invest 0.1 BTC worth of XMarket::base_currency into every XMarket that matches your configuration".</p> <p>If a given XMarket::base_currency is not BTC, the amount of BTC defined in this setting will be converted to its base currency.</p> <p>Most Exchanges have a Minimum Order Amount requirement. If START_AMOUNT is below this requirement, the Investor will change to state <code>cancelled</code>.</p> <p>Note: Thousands of Investors can be running at any given time. Each Investor will "lock" its START_AMOUNT funds when entering state, <code>sparking</code>. This is due to having Open Orders on Exchanges, and without this would potentially allow insufficient funds to disturb and ultimately cause failures of pre-existing strategies that rely on those funds to execute actions.</p> <p>Minimum 0.0001</p>
<p>SUCCESS_STOP_THRESHOLD Integer Optional</p>	<p>The minimum percentage profit beyond SUCCESS_THRESHOLD an Investor in state <code>success</code> can observe before entering state <code>gtfo</code>.</p> <p>A value of 10 means "After you have made (SUCCESS_THRESHOLD + 10)% profit, you need to exit the Market immediately so we don't lose our gains".</p>
<p>SUCCESS_THRESHOLD Number Required</p>	<p>The minimum percentage profit an Investor must observe before entering state <code>success</code>.</p> <p>A value of 5 means "After you have made 5% profit, you should get ready to exit the Market so we don't lose our gains".</p>

SUCCESS_TTL Integer Required	The amount of heartbeat invocations an Investor in state, <code>success</code> , can sustain before dying.
WAITING_TTL Integer Required	The amount of heartbeat invocations an Investor in state, <code>waiting</code> , can sustain before dying.
engine String Required	The Investor Engine the Investor will be configured to use. Examples <code>sim</code> <code>trading</code>
exchange_id String Optional	The UUID of the Exchange this Investor will be restricted to.
market_id String Optional	The UUID of the Market this Investor will be restricted to.

Responses

200	Body	application/json
	Type	Response

Delete an Investor

DELETE /investors/{id}

Delete an existing Investor

Security

Basic, Trading

Request

Path Variables

id	Investor UUID
String Required	Examples kek_investor_gotem

Responses

200

Body	application/json
------	------------------

Type

Response

Get an Investor

GET /investors/{id}

Retrieve an existing Investor.

Security

Basic, Trading

Request

Path Variables

id	Investor UUID
String Required	Examples kek_investor_gotem

Query Parameters

engine	Investor::engine that is used as a required filter on the result set.
String Optional	
parent_id	UUID of the Parent object that all child Investors will be retrieved from.
String Optional	
parent_type	Model of the Parent object that all child Investors will be retrieved from.
String Optional	

Responses

200

Body application/json

Type

Response

Update an Investor

PUT /investors/{id}

Update an existing Investor with the given request data.

Security

Basic, Trading

Request

Path Variables

id	Investor UUID
String Required	Examples kek_investor_gotem

Query Parameters

args	Investor Object that will overwrite all pre-existing properties on the Investor.
Investor Required	

Responses

200

Body	application/json
------	------------------

Type

Response

Get All Investors

GET /investors/get/all

Retrieve all existing Investors belonging to this User.

Security

Basic, Trading

Request**Query Parameters**

parent_id	UUID of the object that this Investor belongs to.
String Required	

parent_type	Model of the object that this Investor belongs to.
String Optional	If this field is omitted, it will retrieve all Investors that are not owned by a profile.

Responses

200	Body application/json
	Type
	Response

Order Methods

Order CRUD Operations

Create an Order

POST /orders/

Place an Order on the given XAPI.

Security

Basic, Trading

Request

Body

application/json

Type

▼ Object

exchange_id String Required	UUID of the Exchange this Order should be placed on.
market_id String Required	UUID of the Market this Order should be placed in.
method String Required	The method that should be used to fill the Order. Examples <code>limit</code> <code>market</code>
type String Required	The direction of this Order. Examples <code>buy</code> <code>sell</code>
volume String Required	The maximum amount that can be executed for this Order.
price String Optional	The rate that should be used in the Order, expressed in terms of 1 Base per X Quote, where X is the price. This field is only used for <code>limit</code> Orders. <code>market</code> Orders will fill the specified <code>volume</code> at the current Order Book price until all volume is filled.
options ▼ Object Optional	Assorted metadata and advanced options.

Responses

200	Body	application/json
Type		
Response	Examples	<pre>{ "status": 1, "message": "Order created successfully!", "data": "1506246123_634-vmlfdksLENjRCEzP", "timestamp": 1506246124462 }</pre>

Cancel an Order

DELETE /orders/

Cancel an existing Order

Request

Query Parameters

id	UUID of the Order that should be cancelled.
String Required	

Responses

200

Body application/json

Type

Response

Examples {

```
"status": 1,  
"message": "Order cancelled successfully!",  
"data": "1506245959_7656-XwLYHyiabjKxW0hz",  
"timestamp": 1506246043664  
}
```

Get an Order

GET /orders/

Retrieve an existing Order.

Request

Query Parameters

id	UUID of the Order that should be retrieved.
String Required	

Responses

200

Body application/json

Type

Response	Examples	<pre> { "status": 1, "message": "Order 1506244396_4744-IaNYgocHCchHJEqT retrieved successfully!", "data": { "type": "sell", "method": "limit", "status": "sparking", "price": "4600.000000000000000000000000000000000000", "volume": "0.025000000000000000000000000000000000", "volume_exec": "0.000000000000000000000000000000000000", "volume_exec_percent": "0.000000000000000000000000000000000000", "options": null, "created": "2017-09-24 03:13:16", "updated": "2017-09-24 03:14:44", "id": "1506244396_4744-IaNYgocHCchHJEqT", "_mid": "BTC_USD", "_oid": "0T50VX-DH7BS-WBUB7U", "_xid": "kraken", "_o": null }, "timestamp": 1506246005035 } </pre>
----------	----------	---

Update an Order

PUT /orders/

Update an existing Order with the given request data. If an Order is filled before we can Cancel and Re-create your Order, the Response Object will reflect it. This is required because XAPIs do not support the ability to update pre-existing Orders.

Request

Query Parameters

id	UUID of the Order that should be updated.
String Required	

var	Name of the setting that should be updated.
String Required	Examples price volume

val	New value of the setting.
String Required	Examples 4.20 0.1

Responses

200

Body application/json

Type

Response	Examples { "status": 1, "message": "Order updated successfully!", "data": "1506246123_634~vmlfdksLENjRCEzP", "timestamp": 1506246153932 }
----------	--

Profile Methods

Profiles are a configurable set of configuration that can automatically trade/simulate for you.

Create a Profile

POST /profiles/{id}

Create a new Investor Profile, aka: trading strategy.

Profile Statuses

- **initializing**
 - An initializing Profile has just been created and is pending creation.
- **running**
 - A running Profile is actively fibrillating and invoking heartbeats on Investors.
- **stopped**
 - A stopped Profile is disabled and will not execute any actions.
- **error**
 - An error Profile is disabled due to an unknown error.
- **destroyed**
 - A destroyed Profile is pending permanent deletion as requested by the Owner.
- **frozen**
 - A frozen Profile is disabled from fibrillating new Investors but will continue to invoke heartbeats on Investors that are currently alive and owned by this Profile.

Security

Basic, Trading

Request

Path Variables

id	Profile UUID
String Required	Examples kek_investor_gotem

Body application/json

Type

▼ Object

config	The configuration that will be applied to the new Profile.
▼ Object Required	

EXITING_TTL Integer Required	The amount of heartbeat invocations an Investor in state, <code>exiting</code> , can sustain before dying.
GREED_THRESHOLD Integer Optional	The maximum percentage profit an Investor can observe before entering state <code>gtfo</code> . A value of <code>10</code> means "After you have made 10% profit, you need to exit the Market immediately so we don't lose our gains".
GTFO_TTL Integer Required	The amount of heartbeat invocations an Investor in state, <code>gtfo</code> , can sustain before dying.
MAX_LOSS Number Required	The maximum percentage loss an Investor will sustain before changing state to <code>gtfo</code> and exiting the market asap. If a <code>MAX_LOSS</code> of <code>-1.0</code> is set, all Investors created by this Profile will <code>gtfo</code> when $(\text{Investor}::\text{delta} * 100) > (\text{START_AMOUNT} * (1 - \text{MAX_LOSS}))$. This means that a value of <code>1.0</code> is "exit after 1% loss". Maximum <code>-0.000001</code>
MEAN_REVERSION_OFFSET ET Number Optional	Mean Reversion is the concept that all prices will fluctuate over time. But as a given price gets further from its <i>Mean</i> (calculated average over x time), the probability of it reversing its current trend (+ or -) and instead begin trending towards its <i>Mean</i> becomes exponentially more likely. <code>MEAN_REVERSION_OFFSET</code> defines the required minimum percentage difference between a given XMarket Price and its Mean for it to be considered a valid XMarket. A value of <code>10</code> means "Not allowed to buy into any XMarket unless its current price is at least 10% LESS THAN its Mean price". A negative value, such as <code>-10</code> , means "Not allowed to buy into any XMarket unless its current price is at least 10% GREATER THAN its Mean price".
MIN_VOLUME_24H Number Required	The minimum 24 hour volume expressed in <code>BTC</code> an XMarket must have to allow Investors to buy in. If a given <code>XMarket::base_currency</code> is not <code>BTC</code> , its 24 Hour Base Volume will be converted to Bitcoin when compared to this setting.
SPARKING_TTL Integer Required	The amount of heartbeat invocations an Investor in state, <code>sparking</code> , can sustain before dying.

START_AMOUNT Number Required	<p>The amount of BTC each Investor will “invest” into an XMarket.</p> <p>A value of 0.1 means “Invest 0.1 BTC worth of XMarket::base_currency into every XMarket that matches your configuration”.</p> <p>If a given XMarket::base_currency is not BTC, the amount of BTC defined in this setting will be converted to its base currency.</p> <p>Most Exchanges have a Minimum Order Amount requirement. If START_AMOUNT is below this requirement, the Investor will change to state cancelled.</p> <p>Note: Thousands of Investors can be running at any given time. Each Investor will “lock” its START_AMOUNT funds when entering state, sparking. This is due to having Open Orders on Exchanges, and without this would potentially allow insufficient funds to disturb and ultimately cause failures of pre-existing strategies that rely on those funds to execute actions.</p> <p>Minimum 0.0001</p>
SUCCESS_STOP_THRES HOLD Integer Optional	<p>The minimum percentage profit beyond SUCCESS_THRESHOLD an Investor in state success can observe before entering state gtf.</p> <p>A value of 10 means “After you have made (SUCCESS_THRESHOLD + 10)% profit, you need to exit the Market immediately so we don’t lose our gains”.</p>
SUCCESS_THRESHOLD Number Required	<p>The minimum percentage profit an Investor must observe before entering state success.</p> <p>A value of 5 means “After you have made 5% profit, you should get ready to exit the Market so we don’t lose our gains”.</p>
SUCCESS_TTL Integer Required	<p>The amount of heartbeat invocations an Investor in state, success, can sustain before dying.</p>
WAITING_TTL Integer Required	<p>The amount of heartbeat invocations an Investor in state, waiting, can sustain before dying.</p>
engine String Required	<p>The Investor Engine the Profile will configured to use.</p> <p>Examples sim trading</p>
profile_name String Required	<p>Unique human-friendly display name for this Profile.</p> <p>Examples BlazeItBot420 Get Rekt Noobs</p>

Responses

200	Body	application/json
<hr/>		
Type		
<hr/>		
Response		

Delete a Profile

DELETE /profiles/{id}

Delete an existing Profile

Security

Basic, Trading

Request

Path Variables

id	Profile UUID
String Required	Examples kek_investor_gotem

Responses

200	Body	application/json
<hr/>		
Type		
<hr/>		
Response		

Get a Profile

GET /profiles/{id}

Retrieve an existing Profile.

Security

Basic, Trading

Request**Path Variables**

id	Profile UUID
String Required	Examples kek_investor_gotem

Responses

200

Body application/json

Type

Response

Update a Profile

PUT /profiles/\${id}

Update an existing Profile with the given request data.

Security

Basic, Trading

Request**Path Variables**

id	Profile UUID
String Required	Examples kek_investor_gotem

Query Parameters

args	Profile Object that will overwrite all pre-existing properties on the Profile.
Profile Required	

Responses

200	Body	application/json
	Type	
	Response	

Analyze a Profile

GET /profiles/{id}/analyze

Analyze the performance of a Profile

Request

Path Variables

id	UUID of the Profile to be analyzed
String Required	Examples kek_investor_gotem

Responses

200	Body	application/json
	Type	
	Response	

Clone a Profile

GET /profiles/{id}/clone

Clone a source Profile into a new Profile

Security

Basic, Trading

Request

Path Variables

id String Required	UUID of the target Profile that will cloned. Examples kek_investor_gotem
------------------------------	---

Query Parameters

engine String Required	The Investor Engine the cloned Profile will be configured to use. Examples sim trading
----------------------------------	--

target_id String Optional	<p>UUID that will be assigned to the cloned profile. The assigned UUID will be automatically assigned if this field is not sent.</p> <p>The assigned UUID will always be returned by this method.</p> <p>Examples kek_copy_gotem</p>
-------------------------------------	--

Responses

200

Body	application/json
------	------------------

Type

Response

Rename a Profile

GET /profiles/\${id}/rename

Rename an existing Profile

Security

Basic, Trading

Request

Path Variables

id	UUID of the Profile that should be renamed.
String Required	Examples kek_investor_gotem

Query Parameters

profile_name	The name that the Profile should be renamed to.
String Required	Examples investor_kek

Responses

200

Body application/json

Type

Response

Start a Profile

GET /profiles/\${id}/start

Start an existing Profile by setting its status to `running`

Security

Basic, Trading

Request

Path Variables

id	UUID of the Profile that should be started.
String Required	

Responses

200	Body application/json
	Type
	Response

Stop a Profile

GET /profiles/\${id}/stop

Stop an existing Profile by setting its status to stopped

Security

Basic, Trading

Request

Path Variables

id	UUID of the Profile that will be stopped.
String Required	Examples kek_investor_gotem

Responses

200	Body application/json
	Type
	Response

Get All Profiles

GET /profiles/get/all

Retrieve all existing profiles belonging to this User.

Security

Basic, Trading

Request

Responses

200	Body	application/json
	Type	
	Response	

Get Profile IDs

GET /profiles/get/index

Retrieve an array of all Profile IDs

Security

Basic, Trading

Request

Responses

200	Body	application/json
	Type	
	Response	

Get Profiles by Engine

GET /profiles/get/engine/{engine}

Get Profiles using a given Profile::engine

Security

Basic, Trading

Request

Path Variables

engine	The Investor Engine that is a required filter on the result set.
String Required	Examples <code>sim</code>
	<code>trading</code>

Responses

200

Body `application/json`

Type

Response

Get Profiles by Name

GET `/profiles/get/name/${name}`

Get Profiles using a given Profile::name

Security

Basic, Trading

Request

Path Variables

name	The Profile Name that is a required filter on the result set.
String Required	Examples <code>sim</code>
	<code>trading</code>

Responses

200	Body	application/json
<hr/>		
Type		
<hr/>		
Response		

Get Profiles by Status

GET /profiles/get/status/{status}

Get Profiles using a given Profile::status

Security

Basic, Trading

Request

Path Variables

status	The current state of the Profile
String Required	Examples <code>initializing</code>
	<code>running</code>
	<code>stopped</code>
	<code>error</code>
	<code>destroyed</code>
	<code>frozen</code>

Responses

200	Body	application/json
<hr/>		
Type		
<hr/>		
Response		

Trade Methods

Trade CRUD Operations

Create a Trade

POST /trades/

Create a Trade by placing a market Order on the given XAPI.

Security

Basic, Trading

Request

Body

application/json

Type

▼ Object

exchange_id
String **Required**

UUID of the Exchange this Trade should be placed on.

market_id
String **Required**

UUID of the Market this Trade should be placed in.

type
String **Required**

The direction of this Trade.

Examples buy
sell

volume
String **Required**

The amount that will be executed for this Trade.

options
▼ Object Optional

Assorted metadata and advanced options.

Responses

200	Body	application/json
	Type	
	Response	

Get a Trade

GET /trades/

Retrieve an existing Trade.

Request

Query Parameters

id	UUID of the Trade that should be retrieved.
String Required	

Responses

200	Body	application/json
	Type	
	Response	

User Methods

Kek.io User Account Operations

Create API Key Pair

POST /users/keys/api

Create a new Kek.io User API Key Pair that is used as an authentication step when invoking API Requests. API Requests are defined as all Requests.

Both the Public and Private Key is returned by the API Method strictly one-time. Be sure to save your keys in a very safe place.

Only one Kek.io User API Key Pair per User may be active at any given time.

Security

User

Request

Body

application/json

Type

▼ Object

password
String **Required**

A password that is different than your Account password used to generate your API Key Pair and will be used to authenticate you via the Basic Security Schema.

Responses

200

Body

application/json

Type

Response

Examples {

```
{
  "status": 1,
  "message": "API Key Pair generated successfully!",
  "data": {
    "public": "8e2j6c679fhj22af2196af587fksda87",
    "secret": "f1l73fksac3080defksa1f0bf93kc45e"
  },
  "timestamp": 1506312907912
}
```

Delete API Key Pair

DELETE /users/keys/api

Delete an existing Kek.io User API Key Pair.

To delete an existing User Key Pair, the original `password` used in Key Generation must be provided, along with the Public and Private Key. Before deleting the User API Key Pair, all existing Trading API Key Pairs and XAPI Key Pairs will be securely destroyed and will need to be recreated.

Request

Query Parameters

<code>password</code> String Required	The original <code>password</code> provided during Trading Key Pair Generation.
<code>private_key</code> String Required	The Private Key obtained during Trading Key Pair Generation.
<code>public_key</code> String Required	The Public Key obtained during Trading Key Pair Generation.

Responses

200	Body application/json
	Type
	Response

Get API Key Pair Usage

GET /users/keys/api

Retrieve usage statistics about an existing User API Key Pair.

Security

Basic, Trading, User

Request

Responses

200	Body	application/json
	Type	
	Response	

Create Trading Key Pair

POST /users/keys/trading

Create a new Kek.io Trading API Key Pair that is used as an additional authentication step when invoking Trade Requests. Trade Requests are defined as any request that may in some way affect or interface with Exchange Trading APIs, including but not limited to creating an Order, manipulating an Investor and retrieving Wallet Balances.

The Public Key is returned by the API Method and not stored.

The Private Key is not.

Only one Kek.io Trading API Key Pair per User may be active at any given time.

Security

Basic, User

Request

	Body	application/json
	Type	
	▼ Object	
	password String Required	A one-time password used to generate your Trading Key Pair.

Responses

200

Body application/json

Type

Response

Examples {

```

{
  "status": 1,
  "message": "Trading Key Pair generated successfully!",
  "data": {
    "public": "0896js952ab4c6221c8slf79d08djce4",
    "secret": "bbijas5081l934b2d411jd6c5f85f8b1"
  },
  "timestamp": 1506314417336
}

```

Delete Trading Key Pair

DELETE /users/keys/trading

Delete an existing Kek.io Trading API Key Pair.

To delete an existing Trading Key Pair, the original `password` used in Key Generation must be provided, along with the Public and Private Key.

Before deleting the Trading API Key Pair, all existing XAPI Key Pairs will be securely destroyed and will need to be recreated.

Security

Basic, Trading, User

Request

Query Parameters

password String Required	The original <code>password</code> provided during Trading Key Pair Generation.
private_key String Required	The Private Key obtained during Trading Key Pair Generation.
public_key String Required	The Public Key obtained during Trading Key Pair Generation.

Responses

200	Body	application/json
	Type	
	Response	

Get Trading Key Pair Usage

GET /users/keys/trading

Retrieve usage statistics about an existing Trading Key Pair.

Security

Basic, Trading, User

Request

Responses

200	Body	application/json
	Type	
	Response	

Create an XAPI Key Pair

POST /users/keys/xapi

Create a new XAPI (Exchange API) Key Pair.

XAPI Key Pairs are provided by Exchange Partners and differ in design/format by Exchange. For more information about retrieving your XAPI Key Pairs, visit your Exchange Website and they should provide instructions on obtaining these.

Security

Basic, Trading, User

Request

Body

application/json

Type

▼ Object

exchange_id
String **Required**

The UUID of the Exchange this API Key Pair resides on.

Examples bitfinex
kraken

password
String **Required**

A one-time password used to further protect your API Key Pair.

This password can be up to 8192 Bytes of text. 😊

public_key
String **Required**

Your XAPI Public Key (also referred to as an API Key).

secret_key
String **Required**

Your XAPI Private Key (also referred to as a Private Key).

Responses

200

Body

application/json

Type

Response

Examples {
"status": 1,
"message": "XAPI Key Pair generated successfully!",
"data": [],
"timestamp": 1506321062989
}

Delete an XAPI Key Pair

DELETE /users/keys/xapi

Delete an existing XAPI Key Pair

Security

Basic, Trading, User

Request

Query Parameters

id	Profile UUID
String Optional	Examples kek_investor_gotem

Responses

200

Body application/json

Type

Response

Get XAPI Key Pair Usage

GET /users/keys/xapi

Retrieve usage statistics about an existing XAPI Key Pair.

Security

Basic, Trading, User

Request

Query Parameters

id	Profile UUID
String Optional	Examples kek_investor_gotem

Responses

200

Body

application/json

Type

Response

Wallet Methods

Search Wallets

GET /wallets/

Search existing Wallets.

Security

Basic, Trading

Request

Query Parameters

base String Optional	The Asset:::id that the aggregated currency value should be converted to. Examples ETH EUR Default USD
currency String Required	The currency that is used as a required filter on the result set of Wallets. Examples BTC
exchange String Optional	The exchange that is used as a required filter on the result set of Wallets. Examples kraken
status String Optional	The status that is used as a required filter on the result set of Wallets.

Responses

200

Body application/json

Type

Response

Examples {

```

"status": 1,
"message": "2 Wallets (BTC) retrieved successfully!",
"data": [
  {
    "id": "kek_bitfinex_BTC",
    "exchange": "bitfinex",
    "type": "trading",
    "currency": "BTC",
    "balance": "0.000000000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 04:24:55",
    "updated": "2017-09-24 04:16:02"
  },
  {
    "id": "kek_kraken_BTC",
    "exchange": "kraken",
    "type": "trading",
    "currency": "BTC",
    "balance": "0.892882227000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:32:13",
    "updated": "2017-09-14 10:44:03"
  }
],
"timestamp": 1506248168017
}

```

Update a Wallet

PUT /wallets/

Update an existing Wallet metadata field, such as status .

Security

Basic, Trading

Request

Query Parameters

id	UUID of the Wallet that should be updated.
String Required	Examples kek_wallet_kraken_BTC

var	Name of the setting that should be updated.
String Required	Examples status

val	New value of the setting.
String Optional	Examples disabled enabled no_robots

Responses

200

Body application/json

Type

Response

Get All Wallets

GET /wallets/all

Retrieve all existing Wallets.

Security

Basic, Trading

Request

Responses

200

Body application/json

Type

Response

Examples {

```
"status": 1,
"message": "25 Wallets retrieved successfully!",
"data": [
  {
    "id": "kek_bitfinex_BCH",
    "exchange": "bitfinex",
    "type": "exchange",
    "currency": "BCH",
    "balance": "0.00128077000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 04:24:55",
    "updated": "2017-09-10 04:24:55"
  },
  {
    "id": "kek_bitfinex_BFX",
    "exchange": "bitfinex",
    "type": "exchange",
    "currency": "BFX",
    "balance": "0.000000000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 04:24:55",
    "updated": "2017-09-10 04:24:55"
  },
  {
    "id": "kek_bitfinex_BTC",
    "exchange": "bitfinex",
    "type": "trading",
    "currency": "BTC",
    "balance": "0.000000000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 04:24:55",
    "updated": "2017-09-24 04:23:02"
  },
  {
    "id": "kek_bitfinex_DRK",
    "exchange": "bitfinex",
    "type": "exchange",
    "currency": "DRK",
    "balance": "0.000000000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 04:24:55",
    "updated": "2017-09-10 04:24:55"
  },
  {
    "id": "kek_bitfinex_ETC",
    "exchange": "bitfinex",
    "type": "exchange",
    "currency": "ETC",
    "balance": "0.064725240000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 04:24:55",
    "updated": "2017-09-10 04:24:55"
  },
  {
    "id": "kek_bitfinex_ETH",
    "exchange": "bitfinex",
    "type": "exchange",
```

```
    "currency": "ETH",
    "balance": "0.06459579000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 04:24:55",
    "updated": "2017-09-10 04:24:55"
  },
  {
    "id": "kek_bitfinex_LTC",
    "exchange": "bitfinex",
    "type": "exchange",
    "currency": "LTC",
    "balance": "0.000000000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 04:24:55",
    "updated": "2017-09-10 04:24:55"
  },
  {
    "id": "kek_bitfinex_USD",
    "exchange": "bitfinex",
    "type": "exchange",
    "currency": "USD",
    "balance": "1.012031000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 04:24:55",
    "updated": "2017-09-10 04:24:55"
  },
  {
    "id": "kek_kraken_BCH",
    "exchange": "kraken",
    "type": "trading",
    "currency": "BCH",
    "balance": "0.000008990000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:32:13",
    "updated": "2017-09-10 02:32:13"
  },
  {
    "id": "kek_kraken_BTC",
    "exchange": "kraken",
    "type": "trading",
    "currency": "BTC",
    "balance": "0.892882227000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:32:13",
    "updated": "2017-09-14 10:44:03"
  },
  {
    "id": "kek_kraken_CAD",
    "exchange": "kraken",
    "type": "trading",
    "currency": "CAD",
    "balance": "0.624400000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:32:13",
    "updated": "2017-09-10 02:32:13"
  },
  {
```



```
    "id": "kek_kraken_DAO",
    "exchange": "kraken",
    "type": "trading",
    "currency": "DAO",
    "balance": "0.00000000000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:32:13",
    "updated": "2017-09-10 02:32:13"
  },
  {
    "id": "kek_kraken_DASH",
    "exchange": "kraken",
    "type": "trading",
    "currency": "DASH",
    "balance": "3.00000000000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:32:13",
    "updated": "2017-09-10 02:32:13"
  },
  {
    "id": "kek_kraken_EOS",
    "exchange": "kraken",
    "type": "trading",
    "currency": "EOS",
    "balance": "500.000000000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:32:13",
    "updated": "2017-09-10 02:32:13"
  },
  {
    "id": "kek_kraken_ETC",
    "exchange": "kraken",
    "type": "trading",
    "currency": "ETC",
    "balance": "5.00178343530000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:32:13",
    "updated": "2017-09-10 02:32:13"
  },
  {
    "id": "kek_kraken_ETH",
    "exchange": "kraken",
    "type": "trading",
    "currency": "ETH",
    "balance": "0.71300262060000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:32:13",
    "updated": "2017-09-10 02:32:13"
  },
  {
    "id": "kek_kraken_EUR",
    "exchange": "kraken",
    "type": "trading",
    "currency": "EUR",
    "balance": "0.32120000000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:32:13",
```

```
    "updated": "2017-09-10 02:32:13"
  },
  {
    "id": "kek_kraken_GBP",
    "exchange": "kraken",
    "type": "trading",
    "currency": "GBP",
    "balance": "0.000000000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:32:13",
    "updated": "2017-09-10 02:32:13"
  },
  {
    "id": "kek_kraken_GNO",
    "exchange": "kraken",
    "type": "trading",
    "currency": "GNO",
    "balance": "0.000000000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:32:13",
    "updated": "2017-09-10 02:32:13"
  },
  {
    "id": "kek_kraken_JPY",
    "exchange": "kraken",
    "type": "trading",
    "currency": "JPY",
    "balance": "44.530000000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:32:13",
    "updated": "2017-09-10 02:32:13"
  },
  {
    "id": "kek_kraken_LTC",
    "exchange": "kraken",
    "type": "trading",
    "currency": "LTC",
    "balance": "0.006262800000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:32:13",
    "updated": "2017-09-10 02:32:13"
  },
  {
    "id": "kek_kraken_USD",
    "exchange": "kraken",
    "type": "trading",
    "currency": "USD",
    "balance": "0.000000000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:32:13",
    "updated": "2017-09-14 10:44:03"
  },
  {
    "id": "kek_kraken_XBT",
    "exchange": "kraken",
    "type": "trading",
    "currency": "XBT",
    "balance": "0.756139027000000000000000000000",
```

```
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:31:04",
    "updated": "2017-09-10 02:31:04"
  },
  {
    "id": "kek_kraken_XRP",
    "exchange": "kraken",
    "type": "trading",
    "currency": "XRP",
    "balance": "0.1069800000000000000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:32:13",
    "updated": "2017-09-10 02:32:13"
  },
  {
    "id": "kek_kraken_ZEC",
    "exchange": "kraken",
    "type": "trading",
    "currency": "ZEC",
    "balance": "1.0000030400000000000000000000000000000000",
    "balance_last_change": null,
    "deposit_address": null,
    "created": "2017-09-10 02:32:13",
    "updated": "2017-09-10 02:32:13"
  }
],
"timestamp": 1506248636777
}
```

Whitelist Methods

Create a Whitelist

POST /whitelists/

A Whitelist is an included object in all Investor Actions. This means that you can include a certain set of things in your strategy, regardless of if they are Blacklisted or not, such as including `kraken(BTC_USD)` or all `BTC` markets, etc. for a certain strategy while all other strategies must adhere to the Blacklists.

Whitelists always override blacklists.

Request

Body

application/json

Type

▼ Object

id String Required	<p>UUID of the object that should be excluded.</p> <p>Examples <code>kraken</code> <code>BTC_USD</code> <code>kraken(BTC_USD)</code></p>
parent_id String Required	<p>UUID of the object that this Configuration will belong to.</p>
parent_type String Optional	<p>Model of the object that this Configuration will belong to.</p> <p>Default <code>profiles</code></p>
type String Required	<p>Model of the object that should be excluded.</p> <p>Examples <code>exchange</code> <code>market</code> <code>xmarket</code></p>

Responses

200

Body application/json

Type

Response

Delete a Whitelist

DELETE /whitelists/

Delete an existing Whitelist

Request

Query Parameters

id String Required	<p>UUID of the object that is excluded.</p> <p>Examples kraken BTC_USD kraken(BTC_USD)</p>
parent_id String Required	<p>UUID of the object that whose Configuration should be deleted.</p> <p>Examples kek_investor_gotem</p>
parent_type String Optional	<p>Model of the object whose Configuration should be deleted.</p> <p>Default profiles</p>
type String Required	<p>Model of the object that is excluded.</p> <p>Examples exchange market xmarket</p>

Responses

200

Body application/json

Type

Response

Get a Whitelist

GET /whitelists/

Retrieve an existing Whitelist.

Request

Query Parameters

id String Required	<p>UUID of the object that is excluded.</p> <p>Examples kraken BTC_USD kraken(BTC_USD)</p>
parent_id String Required	<p>UUID of the object that whose Configuration should be retrieved.</p> <p>Examples kek_investor_gotem</p>
parent_type String Optional	<p>Model of the object whose Configuration should be retrieved.</p> <p>Default profiles</p>
type String Required	<p>Model of the object that should be excluded.</p> <p>Examples exchange market xmarket</p>

Responses

200	Body	application/json
<hr/>		
Type		
<hr/>		
Response		

Get All Whitelists

GET /whitelists/get/all

Retrieve all existing Whitelists belonging a Parent.

Security

Basic, Trading

Request

Query Parameters

parent_id String Required	UUID of the object that this Blacklist belongs to.
parent_type String Optional	Model of the object that this Blacklist belongs to. Default profiles

Responses

200	Body	application/json
<hr/>		
Type		
<hr/>		
Response		

Withdrawal Methods

Create a Withdrawal

POST /withdrawals/

A Withdrawal represents an intent to send a given Asset out your Kek.io Wallet by sending it to a specified address.

Security

Basic

Request

Body

application/json

Type

▼ Object

address
String **Required**

The Address where the funds should be sent to.

amount
Number **Required**

The amount being sent.

Examples 4.2

asset
String **Required**

The Asset being sent.

Default BTC

Examples ETH

Responses

200

Body

application/json

Type

Response

Cancel a Withdrawal

DELETE /withdrawals/

Delete an existing Withdrawal

Security

Basic

Request

Query Parameters

id	UUID of the Withdrawal that should be cancelled.
String Required	

Responses

200

Body application/json

Type

Response

Get a Withdrawal

GET /withdrawals/

Retrieve an existing Withdrawal.

Security

Basic

Request

Query Parameters

id	UUID of the Withdrawal being retrieved.
String Required	

Responses

200

Body

application/json

Type

Response