Welcome to KYT Office by BitOK

KYT Office is the service where you can register all transfers in crypto and then control all its properties and risks.

API Guide

Transfers

Transfers are core objects of KYT Service.

Types of transfers

There are two types of transfers:

- **Full transfer** - a transfer that has already occurred in a blockchain. The network, the transaction hash, the input and output addresses, the amount and the accurate date and time are defined.

- **Transfer attempt** - a transfer not yet occurred in a blockchain. Such types of transfers are used to pre-check a counterparty wallet for potential risks.

There are 2 subtypes of the full transfers:

- **Deposit** - incoming transfer bound to a transaction.

- **Withdrawal** - outgoing transfer bound to a transaction.

There are 2 subtypes of the transfer attempts:

- **Deposit attempt** - incoming transfer not bound to a transaction.

- **Withdrawal attempt** - outgoing transfer not bound to a transaction.

Registering transfers

How to register a full transfer

To register a full transfer using **/transfers/register/** endpoint you must define the following fields about the transfer:

- **client_id** (optional) - an external ID of the client the transfer will be associated with.
- **direction** - the direction of the transfer.

- **network -** the code of the network where the transfer occurred.
- **tx_hash** - the hash of the transaction the transfer belongs to.
- **token_id** - the identifier of the token within its network.
- **output_address** - the address of a recipient of the transfer.

Request:

```
curl -X POST "https://kyt-api.bitok.org/v1/transfers/register/" \
--header  "Content-Type: application/json" \
--header  "Accept: application/json" \
--header  "API-KEY-ID:{KEY_ID}" \
--header  "API-TIMESTAMP:{TIMESTAMP}" \
--header  "API-SIGNATURE:{SIGNATURE}" \
--data ‘{
  "client_id": "id0001",
  "direction": "incoming",
  "network": "ETH",
  "tx_hash": "0x46bf4313a1f7f22cf97859d119c609fedad81541330de661f967795cc4f46e89",
  "token_id": "0xdac17f958d2ee523a2206206994597c13d831ec7",
  "output_address": "0x98Cb5718876AaB18e3A8429a18Ad543f6369A6f3"
}’
```

Response:

```
{
  "id": "cdc3fd93-c975-4b79-beb7-4ad058078b48",
  "client_id": "id0001",
  "registered_at": "2023-12-18T13:47:25.197606+03:00",
  "occurred_at": null,
  "direction": "incoming",
  "risk_level": "undefined",
  "network": "ETH",
  "token_id": "0xdac17f958d2ee523a2206206994597c13d831ec7",
  "token_symbol": "USDT",
  "tx_hash": "0x46bf4313a1f7f22cf97859d119c609fedad81541330de661f967795cc4f46e89",
  "tx_status": "binding",
  "input_address": null,
  "output_address": "0x98cb5718876aab18e3a8429a18ad543f6369a6f3",
  "amount": null,
  "fiat_currency": "USD",
  "value_in_fiat": null,
  "check_state": {
          "exposure": "queued",
          "exposure_checked_at": null,
          "counterparty": "none",
          "counterparty_checked_at": null,
          "sanctions": "none",
          "sanctions_checked_at": null
  }
}
```

When the transaction is bound the state of the transfer state will be enriched with all necessary properties.

Request:

```
curl -X GET "https://kyt-api.bitok.org/v1/transfers/cdc3fd93-c975-4b79-beb7-4ad058078b48/" \
--header  "Accept: application/json" \
--header  "API-KEY-ID:{KEY_ID}" \
--header  "API-TIMESTAMP:{TIMESTAMP}" \
--header  "API-SIGNATURE:{SIGNATURE}"
```

Response:

```
{
  "id": "cdc3fd93-c975-4b79-beb7-4ad058078b48",
  "client_id": "id0001",
  "registered_at": "2023-12-18T13:47:25.197606+03:00",
  "occurred_at": "2023-12-18T13:46:23+03:00",
  "direction": "incoming",
  "risk_level": "medium",
  "network": "ETH",
  "token_id": "0xdac17f958d2ee523a2206206994597c13d831ec7",
  "token_symbol": "USDT",
  "tx_hash": "0x46bf4313a1f7f22cf97859d119c609fedad81541330de661f967795cc4f46e89",
  "tx_status": "bound",
  "input_address": "0x56eddb7aa87536c09ccc2793473599fd21a8b17f",
  "output_address": "0x98cb5718876aab18e3a8429a18ad543f6369a6f3",
  "amount": 1206,
  "fiat_currency": "USD",
  "value_in_fiat": 1205.2,
  "check_state": {
    "exposure": "checked",
    "exposure_checked_at": "2023-12-18T10:47:27.045732Z",
    "counterparty": "none",
    "counterparty_checked_at": null,
    "sanctions": "checked",
    "sanctions_checked_at": "2023-12-18T10:47:27.045732Z"
  }
}
```

How to register a transfer attempt

To register a transfer attempt using **/transfers/register-attempt/** endpoint you must define the following fields:

- **client_id** (optional) - an external ID of the client the transfer will be associated with.
- **attempt_id** (optional) - a unique external ID of the attempt used while registering the transfer.
- **direction** - the direction of the transfer.
- **network -** the code of the network where the transfer occurred.
- **input_address** (when **direction** is “**incoming**”) - the address of a sender of the transfer.
- **output_address** (when **direction** is “**outgoing**”) - the address of a recipient of the transfer.
- **token_id** (optional)
- **amount** (optional)

Request:

```
curl -X POST "https://kyt-api.bitok.org/v1/transfers/register-attempt/" \
--header  "Content-Type: application/json" \
--header  "Accept: application/json" \
--header  "API-KEY-ID:{KEY_ID}" \
--header  "API-TIMESTAMP:{TIMESTAMP}" \
--header  "API-SIGNATURE:{SIGNATURE}" \
--data ‘{
  "client_id": "id0001",
  "attempt_id": "0a805206bab649a68b3408032a7352e6",
  "direction": "outgoing",
  "network": "ETH",
  "token_id": "0xdac17f958d2ee523a2206206994597c13d831ec7",
  "output_address": "0x92a5B444907902dAa39dE28A82EF66AF12e7f170",
  "amount": 500
}’
```

Response:

```
{
  "id": "3c6b874e-f76c-42b4-8a08-e13fc50fa6a5",
  "client_id": "id0001",
  "attempt_id": "0a805206bab649a68b3408032a7352e6",
  "registered_at": "2023-12-18T14:15:02.266520+03:00",
  "occurred_at": "2023-12-18T14:15:02.258132+03:00",
  "direction": "outgoing",
  "risk_level": "undefined",
  "network": "ETH",
  "token_id": "0xdac17f958d2ee523a2206206994597c13d831ec7",
  "token_symbol": "USDT",
  "tx_hash": null,
  "tx_status": "none",
  "input_address": null,
  "output_address": "0x92a5b444907902daa39de28a82ef66af12e7f170",
  "amount": 500,
  "fiat_currency": "USD",
  "value_in_fiat": 499.66889865320763,
  "check_state": {
    "exposure": "none",
    "exposure_checked_at": null,
    "counterparty": "checking",
    "counterparty_checked_at": null,
    "sanctions": "none",
    "sanctions_checked_at": null
  }
}
```

How to bind a transaction to a transfer attempt

Each transfer attempt could be upgraded to a full transfer binding a transaction by a hash.

To bind a transaction to a transfer attempt using **/transfers/{id}/bind-transaction/** endpoint you must define the following fields:

- **tx_hash** - the hash of the transaction the transfer must belong to.
- **token_id** (required if not defined before) - ID of the token.
- **output_address** (required if not defined before) - the address of a recipient of the transfer. Required for deposits attempts only.

Request:

```
curl -X POST
"https://kyt-api.bitok.org/v1/transfers/3c6b874e-f76c-42b4-8a08-e13fc50fa6a5/bind-transaction/" \
--header  "Content-Type: application/json" \
--header  "Accept: application/json" \
--header  "API-KEY-ID:{KEY_ID}" \
--header  "API-TIMESTAMP:{TIMESTAMP}" \
--header  "API-SIGNATURE:{SIGNATURE}" \
--data ‘{
  "tx_hash": "0xc9ebe3254e683705c2553e268b915bf310995bf7540285146901b17fc3b437e6"
}’
```

Response:

```
{
  "id": "3c6b874e-f76c-42b4-8a08-e13fc50fa6a5",
  "client_id": "id0001",
  "attempt_id": "0a805206bab649a68b3408032a7352e6",
  "registered_at": "2023-12-18T14:15:02.266520+03:00",
  "occurred_at": "2023-12-18T14:15:02.258132+03:00",
  "direction": "outgoing",
  "risk_level": "medium",
  "network": "ETH",
  "token_id": "0xdac17f958d2ee523a2206206994597c13d831ec7",
  "token_symbol": "USDT",
  "tx_hash": "0xc9ebe3254e683705c2553e268b915bf310995bf7540285146901b17fc3b437e6",
  "tx_status": "binding",
  "input_address": null,
  "output_address": "0x92a5b444907902daa39de28a82ef66af12e7f170",
  "amount": 500,
  "fiat_currency": "USD",
  "value_in_fiat": 499.67,
  "check_state": {
    "exposure": "queued",
    "exposure_checked_at": null,
    "counterparty": "checked",
    "counterparty_checked_at": "2023-12-18T11:15:04.733475Z",
    "sanctions": "checked",
    "sanctions_checked_at": "2023-12-18T11:15:04.733475Z"
  }
}
```

When the transaction is bound the state of the transfer state will be enriched with all necessary properties.

Request:

```
curl -X GET "https://kyt-api.bitok.org/v1/transfers/3c6b874e-f76c-42b4-8a08-e13fc50fa6a5/" \
--header  "Accept: application/json" \
--header  "API-KEY-ID:{KEY_ID}" \
--header  "API-TIMESTAMP:{TIMESTAMP}" \
--header  "API-SIGNATURE:{SIGNATURE}"
```

Response:

```
{
  "id": "3c6b874e-f76c-42b4-8a08-e13fc50fa6a5",
  "client_id": "id0001",
  "attempt_id": "0a805206bab649a68b3408032a7352e6",
  "registered_at": "2023-12-18T14:15:02.266520+03:00",
  "occurred_at": "2023-12-18T13:52:35+03:00",
  "direction": "outgoing",
  "risk_level": "medium",
  "network": "ETH",
  "token_id": "0xdac17f958d2ee523a2206206994597c13d831ec7",
  "token_symbol": "USDT",
  "tx_hash": "0xc9ebe3254e683705c2553e268b915bf310995bf7540285146901b17fc3b437e6",
  "tx_status": "bound",
  "input_address": "0x3a2c752d3a78a2234b0caf8d6bcc2ec4c9dedfa8",
  "output_address": "0x92a5b444907902daa39de28a82ef66af12e7f170",
  "amount": 500,
  "fiat_currency": "USD",
  "value_in_fiat": 499.67,
  "check_state": {
        "exposure": "checked",
        "exposure_checked_at": "2023-12-18T11:37:54.204528Z",
        "counterparty": "checked",
        "counterparty_checked_at": "2023-12-18T11:37:54.204528Z",
        "sanctions": "checked",
        "sanctions_checked_at": "2023-12-18T11:37:54.204528Z"
  }
}
```

Transfer exposure and counterparty

Transfer exposure

One of the most important properties of a transfer is its exposure. The exposure defined the origin of funds for incoming transfers and the destination of funds for outgoing transfers.

The exposure is a property of a full transfer only,

Usually the exposure is automatically checked when a transaction is bound.
The exposure also may be rechecked using the API.

Transfer counterparty

Another important property of a transfer is its counterparty. The counterparty is represented by an address exposure and defines a sender of incoming transfers and a recipient for outgoing transfers.

Firstly the counterparty is a property of a transfer attempt but it is also used for full transfers.

Risks and alerts

Risks

Risks indicate different kinds of interaction with risky entities.

Levels of risks

There are the following levels of risks:

Level	**Description**
low	
medium	
high	
severe	

Additional levels of risks:

Level	**Description**
none	No risk detected
undefined	The risk is not yet calculated.

Types of risks

Type	**Description**
sender_entity	The risk is associated with an entity of the sender.
recipient_entity	The risk is associated with an entity of the recipient.
origin_of_funds	The risk is associated with the entity owned the assets that are a part of the transfer now.
destination_of_funds	The risk is associated with the entity received the assets that was a part of the transfer.
sender_exposure	The risk is associated with a risky entity at the sender’s exposure.

recipient_exposure	The risk is associated with a risky entity at the sender’s exposure.
attempt_sender_entity	The risk is associated with an entity of the attempt’s sender.
attempt_recipient_entity	The risk is associated with an entity of the attempt’s recipient.
attempt_sender_exposure	The risk is associated with a risky entity at the exposure of the attempt’s sender.
attempt_recipient_exposure	The risk is associated with a risky entity at the exposure of the attempt’s recipient.

Alerts

Alerts are signals about risks.

Manual checks

How to create a manual transfer check

To create a manual transfer check you have to use **/manual-check/check-transfer/** endpoint and define the following fields about the transfer:

- **network**
- **token_id**
- **tx_hash**
- **output_address**
- **direction**

Request:

```
curl -X POST "https://kyt-api.bitok.org/v1/manual-checks/check-transfer/" \
--header  "Content-Type: application/json" \
--header  "Accept: application/json" \
--header  "API-KEY-ID:{KEY_ID}" \
--header  "API-TIMESTAMP:{TIMESTAMP}" \
--header  "API-SIGNATURE:{SIGNATURE}" \
--data ‘{
  "network": "ETH",
  "token_id": "native",
  "tx_hash": "0xd74f7e2a5081eb82c1d0a4fbd1859f23bed5fab8280f0aaf9e987019acc973a1",
  "output_address": "0x2A6Ced4B10769147824A36e3D646eDA222E50f2A",
  "direction": "incoming"
}’
```

Response:

```
{
  "id": "4976989b-c116-47fd-9a77-03502f578bc7",
  "created_at": "2024-02-29T19:30:17.278452+03:00",
  "check_type": "deposit",
  "check_status": "checking",
  "checked_at": null,
  "transfer": {
    "network": "ETH",
    "token_id": "native",
    "token_symbol": "ETH",
    "tx_hash": "0xd74f7e2a5081eb82c1d0a4fbd1859f23bed5fab8280f0aaf9e987019acc973a1",
    "input_address": "0x98f79674d5f2f777d44e253bfaf905d7491e8cef",
    "output_address": "0x2a6ced4b10769147824a36e3d646eda222e50f2a",
    "direction": "incoming",
    "occurred_at": "2024-02-29T19:24:59+03:00",
    "amount": 7.2181350653139,
    "value_in_fiat": 16348.434545956596
  },
  "address": null,
  "risk_level": "undefined",
  "fiat_currency": "USD"
}
```

How to create a manual address check

To create a manual transfer check you have to use **/manual-check/check-address/** endpoint and define the following fields about the transfer:

- **network**
- **token_id** (optional)
- **address**

Request:

```
curl -X POST "https://kyt-api.bitok.org/v1/manual-checks/check-address/" \
--header  "Content-Type: application/json" \
--header  "Accept: application/json" \
--header  "API-KEY-ID:{KEY_ID}" \
--header  "API-TIMESTAMP:{TIMESTAMP}" \
--header  "API-SIGNATURE:{SIGNATURE}" \
--data ‘{
  "network": "ETH",
  "token_id": "0xdac17f958d2ee523a2206206994597c13d831ec7",
  "address": "0x98f79674D5F2f777d44e253BfAf905D7491E8cEF"
}’
```

Response:

```
{
  "id": "15b9bf78-a814-4b8e-9dbd-09f2596a6b00",
  "created_at": "2024-02-29T19:40:12.287316+03:00",
  "check_type": "single_address",
  "check_status": "checking",
  "checked_at": null,
  "transfer": null,
  "address": {
    "network": "ETH",
    "address": "0x98f79674d5f2f777d44e253bfaf905d7491e8cef"
  },
  "risk_level": "undefined",
  "fiat_currency": "USD"
}
```

API Reference

Authorization

KYT API uses a custom HTTP-scheme based on a keyed-HMAC (Hash Message Authentication Code) for authentication.

To authenticate a request, you first concatenate selected elements of the request to form a string. You then use your KYT secret access key to calculate the HMAC of that string. Informally, we call this process "signing the request," and we call the output of the HMAC algorithm the signature, because it simulates the security properties of a real signature. Finally, you add this signature as a parameter of the request by using the syntax described in this section.

Header parameters:

Parameter	**Description**
API-KEY-ID	API Key ID.
API-TIMESTAMP	Current timestamp in milliseconds.

API-SIGNATURE	HMAC-256 signature encoded in Base-64.

Building a HMAC-256 signature

The following Python script builds the preceding HMAC-256 signature, using the provided parameters.
You can use this script to construct your own signatures, replacing the keys and other input parameters.

Parameter	**Description**
http_method	HTTP-method of the request. Example: GET, POST.
endpoint_with_query_params	The endpoint path supplemented by query parameters.
timestamp	The timestamp included to the request header.
json_payload	The payload of the request as a dict.
api_secret	The secret part of the key.

```
import json
import hmac
import hashlib
import base64

str_to_sign = (
    http_method + '\n' +
    endpoint_with_query_params + '\n' +
    timestamp
)

if json_payload:
    string_to_sign += '\n' + json.dumps(json_payload, separators=(',', ':'))

built_signature = hmac.new(
    api_secret.encode('utf-8'),
    msg=str_to_sign.encode('utf-8'),
    digestmod=hashlib.sha256
).digest()

signature = base64.b64encode(built_signature).decode()
```

Example of the string to sign:

```
POST
/v1/transfers/register/
1713449845309
{"client_id":null,"direction":"incoming","network":"ETH","tx_hash":"0x28138cd586826bbad08d1d0
e64b566795b5907790ad30ebb0722948c2ba21d09","token_id":"usdt","output_address":"0x0166
06acc6b0cfe537acc221e3bf1bb44b4049ee"}
```

API Key:

```
qgbtA4OrsHIx67APkTFGfUSctuEEwOYm
```

API Secret:

```
CXOlYKZgeSM3TpIyPwjSM84Ews2hARKi2m1MlLpnbI7UrF5bqtB2WQ3nW6Qh4vSJ
```

HMAC-256 signature:

```
2dJYm8qkR8fCO3s7ZsSVBo1xKpLgx/eYAkewE82pyIs=
```

HTTP-header:

```
POST /v1/transfers/register/ HTTP/1.1
…
Content-Type: application/json
API-KEY-ID: qgbtA4OrsHIx67APkTFGfUSctuEEwOYm
API-TIMESTAMP: 1713449845309
API-SIGNATURE: 2dJYm8qkR8fCO3s7ZsSVBo1xKpLgx/eYAkewE82pyIs=
```

Pagination

Some endpoints support pagination with the following parameters:

- **page -** a page number within the paginated result set.
- **page_size -** number of results to return per page

Basics

Endpoint	Method	Query params	Description
/basics/networks/	GET	Yes	*Pagination enabled.*
/basics/tokens/	GET	Yes	*Pagination enabled.*
/basics/entity-categories/	GET	None	*Pagination enabled.*

Transfers

Endpoint	Method	Query params	Description
/transfers/	GET	Yes	Retrieve transfers based on certain criteria. Returns a list of REGISTERED_TRANSFER_EXTRA. *Pagination enabled.*
/transfers/{id}/	GET		Retrieve information about a specific transfer by its ID Returns REGISTERED_TRANSFER_EXTRA.
/transfers/register/	POST		Register a full transfer. Returns REGISTERED_TRANSFER_EXTRA.
/transfers/register-attempt/	POST		Register a transfer attempt. Returns REGISTERED_TRANSFER_EXTRA.
/transfers/{id}/bind-transaction/	POST		Bind a transaction to a transfer. Returns REGISTERED_TRANSFER_EXTRA.
transfers/{id}/exposure/	GET		Get an exposure of a transfer. Return TRANSFER_EXPOSURE.
/transfers/{id}/recheck-exposure/	POST		Recheck an exposure of a transfer.

/transfers/{id}/counterparty/	GET		Get information about the counterparty info of a transfer. Return ADDRESS_EXPOSURE.
/transfers/{id}/recheck-counterparty/	POST		Recheck the counterparty of a transfer.
/transfers/{id}/risks/	GET		Get all risks of the transfer.

Alerts

Endpoint	**Method**	**Query params**	**Description**
/alerts/	GET	Yes	Retrieve alerts based on certain criteria. *Pagination enabled.*
/alerts/id}/	GET		Retrieve information about a specific alert by its ID

Manual checks

Endpoint	**Method**	**Query params**	**Description**
/manual-checks/check-transfer/	POST		Create manual transfer check. Returns MANUAL_CHECK.
/manual-checks/check-address/	POST		Create manual address check. Returns MANUAL_CHECK.
/manual-checks/	GET		Retrieve all manual checks. Returns MANUAL_CHECK objects.
/manual-checks/{id}/	GET		Retrieve manual check details, Returns MANUAL_CHECK.
/manual-checks/{id}/risks/	GET		Retrieve risks of a manual check. Returns RISK objects

/manual-checks/{id}/transfer-exposure/	GET		Retrieve a transfer exposure of the transfer being checked. Returns TRANSFER_EXPOSURE.
/manual-checks/{id}/address-exposure/	GET		Retrieve an address exposure of the address being checked. Returns ADDRESS_EXPOSURE.

API Structures

Transfers

Transfer (TRANSFER)

There are base properties of any kind of a transfer (registered transfer, registered attempt, transfer of a manual check):

Property	**Type**	**Allow null**	**Description**
network	String		The code of the network where the transfer occurred. Examples: **ETH**, **BTC**, **TRX**.
token_id	String		The identifier of the token within its network. This is a contract address for ERC-20/TRC-20/BEP-20 tokens and **“native”** for a native token of the network. Examples: - Native ETH token in Ethereum has token_id as “**native**”. - ERC-20 token USDT in Ethereum has **token_id** as **“0xdac17f958d2ee523a2206206994597c13d831ec7”**.
token_symbol	String		The symbol of the token.

tx_status	String		The status of the bound transaction. Possible values: - **none** - the transaction is not bound. - **bound** - the transaction is already bound. - **binding** - the transaction is in the binding process. - **not_found** - the transaction is not bound due to an incorrect transaction hash. - **error** - error occurred while binding the transaction.
tx_hash	String	Yes	The transaction hash of the bound transaction in the blockchain. *Not defined for the transfer attempts.*
occurred_at	Timestamp	Yes	The date and time when the transfer occurred. *Not defined for the transfer attempts.*
input_address	String	Yes	The address of the sender. *Not defined for the withdrawal attempts (the outgoing transfer attempts).*
output_address	String	Yes	The address of the recipient. *Not defined for the deposit attempts (the incoming transfer attempts).*
direction	String		The direction of the transfer. Possible values: **incoming**, **outgoing**.
amount	Float	Yes	The amount (in token asset) of the transfer.
value_in_fiat	Float	Yes	May be not defined if the fiat price is unknown (for some tokens).

Registered transfer (REGISTERED_TRANSFER)

The properties of a registered transfer described in the table below:

Property	Type	Allow null	Description
id	String		The identifier of the transfer.
registered_at	Timestamp		The date and time when the transfer was registered
client_id	String	Yes	External ID of the client the transfer is associated with. If client_id is null the transfer isn't associated with any client.
attempt_id	String	Yes	Unique external ID of the attempt used while registering the transfer.
risk_level	String		The risk associated with the transfer. Possible values: - **none** - no risk detected; - **low** - the transfer has a low risk; - **medium** - the transfer has a medium risk; - **high** - the transfer has a high risk; - **severe** - the transfer has a severe (maximum) risk; - **undefined** - the transfer was never checked. **See the details about risks below.**
risk_score	Float	Yes	The risk score from 0.0 to 1.0 if the risk level is defined.
All fields of TRANSFER.			

Registered transfer with extra properties (REGISTERED_TRANSFER_EXTRA)

Transfers have some extra properties that are used at **/transfers/*** endpoints only. Its described in the table below:

Property	Type	Allow null	Description

All fields of REGISTERED_TRANSFER.			
check_state	Object		The state of checks of the transfer exposure and counterparty.
check_state.exposure	String		The check status of the exposure. Possible values: - **none** - the exposure was never checked. - **queued** - a check will be started after binding a transaction. - **checked** - the check is completed successfully while the last checking. - **checking** - the check is being checked. - **error** - an error occurred while the last checking.
check_state.exposure_checked_at	Timestamp	Yes	The date and time when the exposure is checked. *Null if not checked.*
check_state.counterparty	String		The check status of the counterparty. Possible values: - **none** - the exposure was never checked. - **checked** - the check is completed successfully while the last checking. - **checking** - the check is being checked. - **error** - an error occurred while the last checking.
check_state.counterparty_checked_at	Timestamp	Yes	The date and time when the counterparty is checked. *Null if not checked.*
fiat_currency	String		The symbol of the fiat currency used to calculate the amount in fiat.

Transfer exposure

Transfer exposure (TRANSFER_EXPOSURE)

The base properties described in the table below:

Property	**Type**	**Allow null**	**Description**
direct_interaction	Object	Yes	Information about the counterparty if the exposure has direct interaction. *Null if not checked or indirectly interacted.*
direct_interaction .entity_category	String		The entity category of the counterparty.
direct_interaction .entity_name	String		The name of the counterparty.
direct_interaction .value_in_fiat	Float		The value in fiat.
indirect_interaction	Array	Yes	An array of all indirect connections. *Null if not checked or directly interacted.*
indirect_interaction [ ].entity_category	String		The entity category of assets.
indirect_interaction [ ].value_share	Float		The share of assets of a specific category in the total exposure.
indirect_interaction [ ].value_in_fiat	Float		The value of assets of a specific category in fiat.
fiat_currency	String		The symbol of the fiat currency used to calculate the amount in fiat.

Transfer exposure of a registered transfer (REGISTERED_TRANSFER_EXPOSURE)

There are some extra fields used for registered transfers:

Property	Type	Allow null	Description
All fields of TRANSFER_EXPOSURE.			
check_status	String		The check status of the exposure. *See “Checks” below.*
checked_at	Timestamp	Yes	The date and time when the exposure is checked. *Null if not checked.*

Address exposure

Address exposure (ADDRESS_EXPOSURE)

The counterparty properties described in the table below:

Property	Type	Allow null	Description
entity_category	String	Yes	The entity category of the counterparty. *Null if the counterparty isn’t identified.*
entity_name	String	Yes	The name of the counterparty. *Null if the counterparty isn’t identified.*
exposure	Array	Yes	Information about the counterparty exposure (incoming and outgoing connections). *Null if the counterparty is already identified.*
exposure [].entity_category	String		The entity category of assets.
exposure [].value_share	Float		The share of assets of a specific category in the total exposure of the counterparty.

Address exposure of a registered transfer counterparty (REGISTERED_COUNTERPARTY_EXPOSURE)

There are some extra fields used for registered transfers:

Property	Type	Allow null	Description
All fields of ADDRESS_EXPOSURE.			
check_status	String		The check status of the counterparty. *See "Checks" below.*
checked_at	Timestamp	Yes	The date and time when the counterparty is checked. *Null if not checked.*

Risks and alerts

Risks (RISK)

The risks associated with transfer are about interacting with different risky entities. Each risk have a number of the properties described below:

Property	Type	Allow null	Description
risk_level	String		The level of risk.
occurred_at	Timestamp		The date and time when the risk occurred.
detected_at	Timestamp		The date and time when the risk was detected.
risk_type	String		The type of risk.
entity_category	String		The entity category associated with the risk.
proximity	String		The proximity of a risky entity.

			Possible values: - **direct** - **indirect**
value_in_fiat	Float	Yes	The amount of risky value in fiat.
value_share	Float	Yes	The share of the risky value relative to the transfer value.
rule	Object		The rule created the risk. *See RISK_RULE below.*

Risk rule (RISK_RULE)

Property	**Type**	**Allow null**	**Description**
rule_type	String	Yes	*Note: Null for old risks only.*
rule_sub_type	String	Yes	*Note: Null for old risks only.*
entity_category	String		The entity category defined in the rule.
min_value_in_fiat	Float	Yes	The minimum risky value in fiat in the rule.
min_value_share	Float	Yes	The minimum share of a risky value in the rule.

Alerts (ALERT)

Property	**Type**	**Allow null**	**Description**
id	String		The identifier of the alert.
created_at	Timestamp (ISO)		The date and time when the alert was created.

updated_at	Timestamp (ISO)		The date and time when the alert was updated.
risk_level	String		The risk level of the associated risk.
risk_type	String		The type of the associated risk.
risk_occurred_at	Timestamp (ISO)		The date and time when the risk occurred.
alert_status (ex. status)	String		The status of the alert. Possible values: - **open** - **in_progress** - **awaiting_response** - **done**
entity_interaction	Object	Yes	Details of the risks connected with a risky entity. *See RISK_ENTITY_INTERACTION below.*
transfer	Object	Yes	The transfer associated with the alert. *See REGISTERED_TRANSFER.*
fiat_currency	String		The symbol of the fiat currency used to calculate values in fiat.
type	String		Deprecated. Will be removed soon.
status	String		Deprecated. Will be removed soon.

Entity interaction of a risk (RISK_ENTITY_INTERACTION)

The structure of the entity-interaction details:

Property	**Type**	**Allow null**	**Description**
entity_category	String		The entity category associated with the risk.

proximity (ex. interaction)	String		The type of interaction with a risky entity. Possible values: - **direct** - **indirect**
value_in_fiat	Float	Yes	The amount of risky value in fiat.
value_share	Float	Yes	The share of the risky value relative to the transfer value.
rule	Object		The rule created the risk. *See RISK_RULE above.*
interaction	String		Deprecated. Will be removed soon.

Manual checks

Manual check (MANUAL_CHECK)

Property	**Type**	**Allow null**	**Description**
id	String		The identifier of the check.
created_at	Timestamp (ISO)		The date and time when the check was created.
check_type	String		The type of the check. Possible values: - **deposit** - **withdrawal** - **single_address**
check_status	String		The current status of the check: Possible values: - **checked** - the check is already checked. - **checking** - the check is being checked.

			- **error** - an error occurred while checking.
checked_at	Timestamp (ISO)	Yes	The date and time when the check was completed.
risk_level	String		The level of risk.
risk_score	Float	Yes	The risk score from 0.0 to 1.0 if the risk level is defined.
transfer	Object	Yes	The transfer was checked by the manual check. *See TRANSFER.*
address	Object	Yes	The transfer associated with the alert. *See ADDRESS.*
fiat_currency	String		The symbol of the fiat currency used to calculate values in fiat.

Address (ADDRESS)

Property	**Type**	**Allow null**	**Description**
network	String		The network where the transfer occurred
address	String		The address in the network.