Introduction

Welcome to the V3X Core API, or Core API for short. The Core API is a REST API that provides access to the core functionality of the V3X platform. Functionality such as issuing credentials, managing Blockchain verifiable credentials, if you already have an existing issuer system, and revoking credentials. Core API also provides access to the Partner, Job, and Credential objects, which are used to track the status of issuance and revocation.

Core API requests can be sent directly from your server. For security, it is best to route Core API requests through a proxy server.

If you are a developer, you may want to use our Core API client libraries instead of using the Core API directly. These libraries provide a more ergonomic interface to the Core API in your code. Currently, we have libraries for Python and Node.js.

- Python: vdx-helpers
- Node.js: @vdx/helpers

The Core API is available in two environments, Sandbox and Prod. The Sandbox environment is for testing, and Prod is for the "real" environment. The former can be used for free testing, since it is connected to the Testnet version of the Blockchain, and issuance of Blockchain verifiable credentials is free. The Prod environment requires an issuance reward through the Ethereum Mainnet blockchain to validate. The "cost" of issuance is dependent on the Ethereum Mainnet Blockchain, so this cost is limited per the rate-limit on the Prod environment.

The further information on each endpoint, see the Endpoint URLs section. The Resources section is where you can find more information on the Data environment.

Core API is a JSON-RPC 2.0 API. Please see the JSON-RPC 2.0 Specification for further information. For using Core API in our examples, we use the Python client library, Requests.

Endpoint URLs

All endpoints are production and the same between both available environments. Although the endpoints are the same, the environment you are using is determined by the URL used based on the authentication key you are using for the Core API.

The API key used for authentication is different for each environment.

Environment	URL
Sandbox	https://api-sandbox.v3xlabs.com/rpc
Demo	https://api-demo.v3xlabs.com/rpc

Pagination

Some of the provided API requests have support for bulk retrievals, such as the "Get all Partners" endpoint. This is used for these endpoints to prevent overload, and the response object contains the following fields:

All of these parameters are fully optional and default values are applied when they are not provided in the request.

Parameter	Description	Default
start_at	Index to use the results by. Value is specific to the endpoint.	0
order	Order ascending or descending, according to the sort_by parameter.	asc
per_page	Number of results per page. 0 for all results in one page.	20
page	Page that requested for the response	1

Errors:
An unregistered endpoint return a specific error if any given pagination argument is incorrect:

Id	code	Description
wrong_pagination_argument	422	Invalid argument used for pagination

Tags

Tags are a feature that allows you to further help with filtering and searching for credentials, jobs and certificates. These tags can be added to credentials or jobs, and can then be used in the filter parameter of the "Get all Credentials" and "Get all Jobs" endpoints. Credentials can be created or issued with a tag, or added later.

All tags have a format that must be followed:

- They must start with a "-", other characters are allowed.
- Tags are case-sensitive, "-test" and "-Test" are different tags.
- While the name contained a "-", there can be no duplicate tags.

Tags can be directly added to credentials or jobs, but they can also be added through specific tag endpoints. The tag endpoints are useful if you want to add a tag to a set of jobs. The tag can be added afterwards in the same way as credentials.

On issued endpoints, tags can be used as a parameter to filter the results. Only credentials with the specified tag will be returned.

Name	Type	Optional	Description
add_tags	Array<Str>	No	Array object that contains all the tags(-tags). Format is "-tag1,-tag2,-tag3"
rm_tags	Array<Str>	No	Object array that contains all tags that need one of the listed tags. Format is "-rm_tag1,-tag2,-tag3"

Errors:
If the provided argument is not in the correct format, or the tag does not exist, the following error is returned:

Id	code	Description
invalid_tag	422	Tag does not exist

Issuing with Vizidox

With the Vizidox Core API, you can choose to either issue a file, variables in JSON format, or both credentials. A file is also created when you are issuing a credential with variables in JSON format. Furthermore, the credentials that are created in the Core API (which contain files and variables) are also available in the Vizidox Web Application. The Vizidox Web Application is a UI that can help with using the credentials in the Partner, sending the data to the Vizidox API, and creating the Jobs for issuance. The Vizidox Web Application is also available in the Demo environment.

To issue credentials, the Vizidox API requires a connecting issuer. Instead of manually sending the issuing process, which can require a considerable amount of time, you can provide the Vizidox API with the information of your issuer. The Vizidox API will then send the data to your issuer, and send some data about the issuer back to your partner. The Response is changing to either contain the data that your issuer needs to create a credential, or the URL of a Vizidox Web Application depending on your configuration.

To issue data to the Blockchain, there are three important steps:

- Upload the file you want to certify - optional if you are only having metadata.
- Create a credential, with the file uploaded in the previous step, and the data to issue.
- Schedule the credential to be issued on a specified Blockchain.

Each of the endpoints required for these steps are explained in more detail in the Resources section.

Resources

Response

All endpoints in the Core API follow a specific format which contains the results field of the endpoint, as well as a status field and optional errors field. Results is a JSON object, and is always a JSON object.

- Status: Indicates the status of the latest update on the Core API.
- Date: The current date of the request.
- Message: A message with any information regarding breaking changes or specific endpoints.
- Results: Results of the requested endpoint.
- Documentation: A link to the API documentation.

Partner

A Partner represents the client receiving requests to the Core API, all credentials that are issued by you are associated to your partner ID, and your API key will only work with the correct partner ID as well.

File

The File object represents a file uploaded to Vizidox by a Partner. It contains two fields:

Field	Description
file_hash	The hash of the file.
file_name	The WEB type of the file

The File object provided by the Vizidox API is MINE encoded.

Upload File

Upload a file to the Core API, and receive its hash for future issuing. This endpoint will not trigger file issuing process for the provided file.

Parameter	Type	Optional	Description
file	Form File	No	File file to be uploaded
upload_name	Form Text	Yes	Set to true to upload even if file file has already been uploaded in the past. Default is false.

It should be noted that if a file hot used to create a credential, it is automatically deleted from the server after one hour.

Request:
POST https://{{URL}}/api/file/upload

Response:
If successful, the HTTP response is 201 CREATED, along with a file object.

Errors:

Id	code	Description
no_file_or_upload	400	Missing the File file object
file_already_uploaded	409	File has already been uploaded

Get All Files

Retrieve all files uploaded by the Partner. Includes the pagination arguments.

Parameter	Type	Optional	Description
file_hash	Query Str	Yes	Specific file hash to search for

GET Request:
GET https://{{URL}}/api/file/list?file_hash=

Response:
If successful, the HTTP response is 200 OK, along with a file object.

Errors:

Id	code	Description
missing_file_id	400	File file was not found

Credential

Fingerprint of the file including additional data provided by the Partner. Optionally, can only be issued to a specified Partner.

Field	Description
uid	The unique identifier of the credential
file_hash	The hash of the credential
metadata	Additional metadata for the credential
files	Files associated with the metadata
credentials	Other credentials associated with the credential
upload_date	Date of credential upload
tags	Tags to identify the credential
expiry_date	Date the credential expires, if applicable

Create Credential

Create a credential on the Core API, either from a file or from metadata. This endpoint will not trigger the issuing process for the created credential. For the issuance, the requests schedule must be used.

Parameter	Optional	Description
file	Yes	The file to be uploaded.
metadata	Yes	Additional metadata for the credential. Can be empty if file is used.
files	Yes	List of urns or cions file hashes to associate the credential with.
credentials	Yes	List of urns or cions credential IDs to associate the credential to.
tags	Yes	Tags for identifying the credential
expiry_date	Yes	Date of expiry of the credential

GET Request:
POST https://{{URL}}/api/credential/create

Response:
If successful, the HTTP response is 201 CREATED, along with a credential object.

Errors:

Id	code	Description
missing_file_and_metadata	422	Either missing a file or metadata in the request
invalid_uuid	422	Invalid uuid in the request
missing_file	422	Error in one of the given files cannot be found
no_data	422	Missing parameter in the request
missing_field	422	Missing a field in the request body
wrong_blockchain_argument	422	The provided blockchain argument is invalid

Get All Credentials

Obtains all credentials previously created by the Partner. Includes the pagination arguments.

Parameter	Optional	Description
uid	Query	No	Specific Credential UID to search for
upload_date_from	Query	Yes	Obtain Credentials created after this date
upload_date_to	Query	Yes	Obtain Credentials created up to this date
file_hash	Query	Yes	Obtain Credentials that contain a file with the given file hash
tag	Query	Yes	Obtain Credentials that contain at least one of the listed tags. Format is "-tag1,-tag2,-tag3"

GET Request:
GET https://{{URL}}/api/credential/list

Response:
uid_to_parameter can be Filed or with (Default is created_date)

- title: The title of the credential
- created_date: The date the credential was created

Errors:
If successful, the HTTP response is 200 OK, along with a paginated object containing the list of credentials.

Get a Credential

Obtains a specific credential

Parameter	Type	Optional	Description
credential	Path	No	Credential UID to search

GET Request:
GET https://{{URL}}/api/credential/{{CREDENTIAL}}

Response:
If successful, the HTTP response is 200 OK, along with a credential object.

Errors:

Id	code	Description
missing_credential	404	The credential was not found

Add Tags

Add new tags to a credential. The endpoint can be used to update more than one credential, for example, if you want to add a tag to multiple credentials.

GET Request:
POST https://{{URL}}/api/credential/addtags

Response:
If successful, the HTTP response is 200 OK with no additional content

Errors:

Id	code	Description
invalid_tag	422	At least one tag is an incorrect format
duplicate_tags	422	At least one tag is repeated in the request
missing_credential	404	One or more credentials were not found

Replace Tags

Replace tags on a credential. The endpoint can be used to update more than one credential, for example, if you want to fix a common error on the tags of the request.

GET Request:
PUT https://{{URL}}/api/credential/addtags

Response:
If successful, the HTTP response is 200 OK with no additional content

Errors:

Id	code	Description
invalid_tag	422	At least one tags is in an incorrect format
duplicate_tags	422	At least one tags is repeated in the request
missing_job	404	One or more credentials were not found

Delete Credential Tags

Delete the tag from the provided credentials.

Parameter	Type	Optional	Description
all	Bool	No	Left all or a credential to update
tag	Str	No	Tag to be deleted from the credential

GET Request:
DELETE https://{{URL}}/api/credential/addtags

Response:
If successful, the HTTP response is 200 OK with no additional content.

Errors:

Id	code	Description
missing_credential	404	The credential cannot be found
missing_job	404	There is no scheduled job for the given engine
missing_one_scheduled	409	The credential is not scheduled on the given engine
wrong_blockchain_argument	422	Given engine is incorrect or invalid
missing_field	400	Missing a field in the request body

Schedule Credentials

Schedules one or more existing credentials on the Core API. The given credentials will be added to the queue and eventually to the issuer of the given blockchain. Issuing.

If Credential cannot be issued from Blockchain, the Job will be set to Failed. Credentials can be issued to multiple Blockchains.

Parameter	Optional	Description
engine	No	Blockchain engine to schedule the credentials on
uid	No	List of credentials to schedule

GET Request:
PUT https://{{URL}}/api/credential/schedule

Response:
If successful, the HTTP response is 201 CREATED, along with a job object.

Errors:

Id	code	Description
missing_credential	404	One or more of the credentials cannot be found
missing_one_scheduled_or_issued	409	One or more of the credentials have already been scheduled or issued
no_uids	422	Missing the required field
wrong_blockchain_engine	422	Given engine is incorrect or invalid
missing_field	400	Missing a field in the request body

Unschedule Credential

Unschedules an existing and previously scheduled Credential from the given Blockchain. Credential will be removed from the queue and will not be issued to the Blockchain. Fails if it was already issued.

Parameter	Optional	Description
credential	No	Credential UID to unschedule
engine	No	Blockchain engine the credential is scheduled on

GET Request:
DELETE https://{{URL}}/api/credential/schedule/{{credential}}/{{engine}}

Response:
If successful, the HTTP response is 201 CREATED, along with a job object.

Errors:

Id	code	Description
missing_credential	404	There is no such credential
missing_job	404	There is no scheduled job for the given engine
missing_one_scheduled	409	The credential is not scheduled on the given engine
wrong_blockchain_engine	422	Given engine is incorrect or invalid

Job

A Job is a Blockchain transaction containing a bundle of credentials. A job has several states:

- Scheduled - The credentials in the Job have been scheduled for issuance on the Blockchain, but still not issued to the real schedule.
- Issued - The Job has been issued to the Blockchain, but not confirmed it's inclusion.
- Unscheduled - The Blockchain Issuing process has successfully finished. Only the Unscheduled Job contains the Blockchain Transaction ID, which can be used to find the Transaction.
- Finished - The Blockchain Issuing process has successfully finished. All credentials Issued in the Job are now available in the Vizidox Web Application, and can be issued to a partner.
- Failed - Something went wrong with the Blockchain Issuing process, and the credentials could not be issued. They will have to be added again to the next scheduled job. This status is also used for Revoking.

Field	Description
uid	The unique identifier of the job
partner	The partner issuing the JOB
chain	The blockchain engine to issue the job on
tx_hash	Issuance transaction hash
status	Current status of the job
created_date	Date the job was created at
start_date	Date the job started
finish_date	Date the job finished issuing
finished_count	Date the job was confirmed on the Blockchain. It is the finished issuance
failed_date	Date the job failed, if it the failed status
unscheduled_date	Date the job was unscheduled on. It is the unscheduled status

Get All Jobs

Obtains all the Partner's jobs. Includes the pagination arguments and tags to filter by.

Parameter	Type	Optional	Description
uid	Query	No	Specific Credential UID to search for
chain	Query	No	Search for jobs on given chain
start_date_from	Query	No	Obtain Jobs that started issuing after this date
start_date_until	Query	No	Obtain Jobs that started issuing before this date
finished_date_from	Query	No	Obtain Jobs that finished issuing after this date
finished_date_until	Query	No	Obtain Jobs that finished issuing up to this date
created_date_from	Query	No	Obtain Jobs created after this date
created_date_until	Query	No	Obtain Jobs created up to this date

GET Request:
GET https://{{URL}}/api/job/list

Response:
uid_to_parameter can be Filed or with (Default is created_date)

- status: Status of the job
- chain: Blockchain engine the job was issued on
- start_date: Date the job started issuing
- finish_date: Date the job finished issuing
- failed_date: If failed, date the job failed
- finished_count: Date the job was confirmed on the Blockchain. It is the finished status

Get All Jobs

Obtains all the Partner's jobs. Includes the pagination arguments and tags to filter by.

Parameter	Type	Optional	Description
uid	Query	No	Specific Credential UID to search for
chain	Query	No	Search for jobs on given chain
start_date_from	Query	No	Obtain Jobs that started issuing after this date
start_date_until	Query	No	Obtain Jobs that started issuing before this date
finished_date_from	Query	No	Obtain Jobs that finished issuing after this date
finished_date_until	Query	No	Obtain Jobs that finished issuing up to this date
created_date_from	Query	No	Obtain Jobs created after this date
created_date_until	Query	No	Obtain Jobs created up to this date

GET Request:
GET https://{{URL}}/api/job/list

Response:
uid_to_parameter can be Filed or with (Default is created_date)

- status: Status of the job
- chain: Blockchain engine the job was issued on
- start_date: Date the job started issuing
- finish_date: Date the job finished issuing
- failed_date: If failed, date the job failed
- finished_count: Date the job was confirmed on the Blockchain. It is the finished status

Get Job

Obtains a specific job

GET Request:
GET https://{{URL}}/api/job/{{JOB}}

Response:
If successful, the HTTP response is 200 OK, along with a job object.

Errors:

Id	code	Description
missing_job	404	The job was not found

Delete a Job

Deletes a specific job

GET Request:
DELETE https://{{URL}}/api/job/{{JOB}}

Response:
If successful, the HTTP response is 200 OK with no additional content.

Errors:

Id	code	Description
missing_job	404	The job was not found
wrong_job_argument	422	Specified job is not a job
job_not_finished	409	Job is not finished, failed or unscheduled

cURL · Python

```
curl -X POST "https://sandbox.v3xlabs.com/rpc" \
  -H "Content-Type: application/ld+ld" \
  -H "x-API-Key: {{API_KEY}}" \
  -d '{"method": "get_partners", "params": [], "id": 1, "jsonrpc": "2.0"}'
```

JSON Response

```
{
  "status": "ok",
  "date": "2022-06-06",
  "message": "",
  "results": [
    {
      "uid": "f0e4c2f7-b96a-42ca-b0d5-6c78ab1ccf8d",
      "name": "Test Partner",
      "description": "",
      "created_date": "2022-06-06",
      "updated_date": "2022-06-06",
      "deleted": false,
      "blockchain_address": "0x0000000000000000000000000000000000000000",
      "blockchain_private_key": "",
      "blockchain_chain": "",
      "blockchain_rpc": "",
      "blockchain_contract": "",
      "blockchain_gas": "",
      "blockchain_gas_price": "",
      "blockchain_nonce": "",
      "blockchain_chain_id": "",
      "blockchain_block": "",
      "blockchain_block_hash": "",
      "blockchain_block_time": "",
      "blockchain_block_gas_used": "",
      "blockchain_block_gas_limit": "",
      "blockchain_block_miner": "",
      "blockchain_block_difficulty": "",
      "blockchain_block_total_difficulty": "",
      "blockchain_block_size": "",
      "blockchain_block_extra_data": "",
      "blockchain_block_nonce": "",
      "blockchain_block_parent_hash": "",
      "blockchain_block_uncle_hash": "",
      "blockchain_block_transactions": "",
      "blockchain_block_transaction_root": "",
      "blockchain_block_receipts_root": "",
      "blockchain_block_state_root": "",
      "blockchain_block_base_fee_per_gas": "",
      "blockchain_block_mix_hash": "",
      "blockchain_block_seal_fields": "",
      "blockchain_block_uncles": "",
      "blockchain_block_transactions_count": "",
      "blockchain_block_uncles_count
```

Parameter	Type	Optional	Description
job_uid	Path	No	Job UID to obtain

HTTP Request

```
GET https://vizidox.com/api/jobs/<UID>
```

Returns

If successful, the HTTP response is a 200 OK, along with a job object

Errors

id	code	Description
missing_job	404	The job was not found

Add Tags

Add new tags to a job. This endpoint can be used to update more than one job, by providing a list in the correct format on the body of the request.

Old tags will not be deleted.

HTTP Request

```
PATCH https://vizidox.com/api/jobs
```

Returns

If successful, the HTTP response is a 200 OK with no additional content

Errors

id	code	Description
invalid_tag	400	At least one tag is in an incorrect format
duplicated_tags	400	At least one tag is repeated in the request
missing_job	404	One or more jobs were not found

Replace Tags

Replace all the tags in a job. This endpoint can be used to update more than one job, by providing a list in the correct format on the body of the request.

Old tags are all deleted.

HTTP Request

```
PUT https://vizidox.com/api/jobs
```

Returns

If successful, the HTTP response is a 200 OK with no additional content

Errors

id	code	Description
invalid_tag	400	At least one tag is in an incorrect format
duplicated_tags	400	At least one tag is repeated in the request
missing_job	404	One or more jobs were not found

Get Credentials in Job

Obtains all credentials issued in a specific job. Includes the pagination arguments.

Parameter	Type	Optional	Description
job_uid	Path	No	Job UID to obtain the credentials

HTTP Request

```
GET https://vizidox.com/api/jobs/<UID>/credentials
```

Pagination

sort_by parameter can be filled in with (Default is created_date):

- status - The current status of the job
- chain - The blockchain engine the job was issued on
- created_date - The date the job was created
- start_date - The date the issuing process of the job started
- issued_date - the date the issuing process of the job finished
- ready_date - The date the issuing transaction of the job was confirmed on the blockchain
- failed_date - The date the job issuing failed

Returns

If successful, the HTTP response is a 200 OK, along with a paginated object containing the list of credentials.

Get Certificates in Job

Obtains all certificates issued in a specific job. Includes the pagination arguments.

Parameter	Type	Optional	Description
job_uid	Path	No	Job UID to obtain the certificates

HTTP Request

```
GET https://vizidox.com/api/jobs/<UID>/certificates
```

Pagination

sort_by parameter can be filled in with (Default is issued_date):

- issued_date - the date the certificate was issued on
- chain - The Blockchain engine the certificate was issued on

Returns

If successful, the HTTP response is a 200 OK, along with a paginated object containing the list of certificates.

Certificate

A Certificate is a verifiable descriptor for a credential issued over the Blockchain. The Certificate will contain all relevant data to connect the Credential to a Blockchain transaction. There is a one to one relationship between certificates, credentials and Blockchain engines. Each issued Credential has one Certificate per Blockchain engine.

Field	Description
uid	The unique identifier of the certificate
partner	The partner who issued the certificate
credential	The issued credential
issued_date	The date the credential was issued on the Blockchain
signature	Credential hashed and signed by the partner's key, which was issued
revoked_date	Date of revocation, if applicable
last_verification	The status and date of the latest verification of the certificate

Get All Certificates

Obtains all certificates issued by the partner. Includes the pagination arguments.

Parameter	Type	Optional	Description
uid	Query	Yes	Certificate UID to filter by
job_uid	Query	Yes	Obtain certificates issued in job
credential_uid	Query	Yes	Obtain the credential's certificate
issued_date_from	Query	Yes	Obtain certificates that were issued after this date
issued_date_until	Query	Yes	Obtain certificates that were issued up to this date
verification_status	Query	Yes	Obtain certificates with a specific verification status

You can also use either the corresponding Credential or Job tags to search for certificates, even though the Core API does not support the addition of tags per certificates. These work in the same way as all other endpoints which contain tags as a filter option:

- and_credential_tags = or_credential_tags: To filter certificates of credentials that are identified by the given tags.
- and_job_tags = or_job_tags: To filter certificates issued in jobs that are identified by the given tags.

HTTP Request

```
GET https://vizidox.com/api/certificates
```

Pagination

sort_by parameter can be filled in with (Default is issued_date):

- issued_date - the date the certificate was issued on
- chain - The Blockchain engine the certificate was issued on

Returns

If successful, the HTTP response is a 200 OK, along with a paginated object containing the list of certificates.

Download Certificate JSON

Downloads a text file containing all relevant certificate data in a JSON format. Can also be called the proof file, and is usable for verification on the Core API.

Parameter	Type	Optional	Description
cert_uid	Path	No	Certificate UID to download

HTTP Request

```
GET https://vizidox.com/api/certificates/<UID>/download
```

Returns

If successful, the HTTP response is a 200 OK, along with a text file containing a JSON with the certificate data.

Errors

id	code	Description
missing_certificate	404	The certificate was not found

Verification

Verification is the process through which any issued credential can be validated. This only applies to credentials that have been fully issued on the Blockchain, so they cannot be simply scheduled for issuing.

The verification consists of six different steps, and can result in several results:

- **Ok** - All steps fully passed.
- **Pending** - At least one step is still pending some action; for example, if the Blockchain confirmation step has not confirmed, the corresponding step will result in a pending status.
- **Expired** - The credential has expired, per the defined expiry date provided on creation.
- **Revoked** - The credential and its corresponding certificate has been revoked, so it is no longer considered valid.
- **Failed** - At least one verification step failed, meaning that Vizidox cannot guarantee the credential's validity.
- **Error** - An unexpected error occured during the verification. Contact with Vizidox support if this problem persists.

The obtained final result of verification will depend on the result of each individual verification step. If one step fails during the process, then no more steps will be tested since the credential is no longer considered valid by default. The six executed steps are:

Name	Description
Certificate Integrity	Checks if the Certificate has not been tampered with, including the credential metadata
Blockchain Anchoring	Checks if the Credential has been issued and confirmed on the Blockchain
Issuer Authenticity	Checks if the Issuer's identity is valid
File Integrity	Checks if the file has not been tampered with, retaining its original content
Expiry Date	Checks if the credential is still valid as of this date
Revocation Date	Checks if the credential has been revoked

Each of these steps have their own status at the end of the verification, and these are used to determine the final status. The possible verification step status are:

- **Not Started** - The step was skipped due to an earlier step failing.
- **Passed** - The step passed.
- **Pending** - The validation is still pending some other action, such as Blockchain confirmation.
- **Failed** - The step failed.
- **Error** - An error occurred during the execution.

Verify By Certificate UID

Verifies a Certificate via its unique identifier

Parameter	Type	Optional	Description
cert_uid	Path	No	Certificate UID to verify

HTTP Request

```
GET https://vizidox.com/api/verify/<UID>
```

Returns

If successful, the HTTP response is a 200 OK, along with the verification result

Errors

id	code	Description
missing_certificate	404	The certificate was not found

Verify By Credential UID

Verify all certificates corresponding to the credential, via the credential's unique identifier. Includes the pagination arguments.

Parameter	Type	Optional	Description
cred_uid	Path	No	Credential UID to verify

HTTP Request

```
GET https://vizidox.com/api/verify/credential/<UID>
```

Returns

If successful, the HTTP response is a 200 OK, along with the verification result

Errors

id	code	Description
missing_credential	404	The credential was not found

Verify By Certificate File

Verifies a specific certificate, by uploading its certificate proof file

HTTP Request

```
POST https://vizidox.com/api/verify/upload/certificate
```

Returns

If successful, the HTTP response is a 200 OK, along with the verification result

Errors

id	code	Description
no_file_on_payload	400	No file was uploaded
missing_certificate	404	The certificate was not found

Verify By File

Verifies all certificates of the uploaded file includes the pagination arguments.

HTTP Request

```
POST https://vizidox.com/api/verify/upload/file
```

Returns

If successful, the HTTP response is a 200 OK, along with a paginated list of all verification results

Errors

id	code	Description
no_file_on_payload	400	No file was uploaded
missing_certificate	404	The certificate was not found

Revocation

All issued credentials can be revoked if you need to. If for example, there was some error in the issued data and it should not be considered valid, revoking its corresponding certificate will guarantee that all future verifications fail.

Revoke Certificate By UID

Revokes a certificate, identified by its UID

Parameter	Type	Optional	Description
cert_uid	Path	No	Certificate UID to revoke

HTTP Request

```
POST https://vizidox.com/api/certificates/<UID>/revoke
```

Returns

If successful, the HTTP response is a 200 OK, along with the revocation date.

Errors

id	code	Description
missing_certificate	404	The certificate was not found
unconfirmed_certificate_transaction	409	The certificate blockchain issuing transaction has not yet been confirmed
certificate_already_revoked	409	The certificate has already been revoked

Revoke Certificate By Credential UID

Revokes the credential's certificate for the given Blockchain engine.

Parameter	Type	Optional	Description
cred_uid	Path	No	Credential UID to revoke
engine	Path	No	Blockchain Engine of issuing

HTTP Request

```
POST https://vizidox.com/api/credentials/<UID>/revoke/<engine>
```

Returns

If successful, the HTTP response is a 200 OK, along with the revocation date.

Errors

id	code	Description
missing_credential	404	The credential was not found
missing_certificate_for_credential	404	The credential has not been issued on the given engine
unconfirmed_certificate_transaction	409	The certificate blockchain issuing transaction has not yet been confirmed
certificate_already_revoked	409	The certificate has already been revoked

```
curl -X GET "https://vizidox.com/api/jobs/<UID>" -H "accept: application/json" -H "Bearer
```

Tag Addition Request Body

```
{
  "jobs": [
    {
      "job_uid": "5234d5d1-e08b-12d3-a456-426655440000",
      "tags": ["tag0", "tag1", "tag2"]
    },
    {
      "job_uid": "7ce0b0a8-e013-43ca-b775-d950b3b9c42f",
      "tags": ["tag1", "tag2"]
    }
  ]
}
```

```
curl -X PATCH "https://vizidox.com/api/jobs" -H "accept: application/json" -H "Content-Type:
```

Tag Replacement Request Body

```
{
  "jobs": [
    {
      "job_uid": "5234d5d1-e08b-12d3-a456-426655440000",
      "tags": ["tag0", "tag1", "tag2"]
    },
    {
      "job_uid": "7ce0b0a8-e013-43ca-b775-d950b3b9c42f",
      "tags": ["tag1", "tag2"]
    }
  ]
}
```

```
curl -X PUT "https://vizidox.com/api/jobs" -H "accept: application/json" -H "Content-Type:
```

```
curl -X GET "https://vizidox.com/api/jobs/<UID>/credentials" -H "accept: application/j
```

```
curl -X GET "https://vizidox.com/api/jobs/<UID>/certificates" -H "accept: application/j
```

```
curl -X GET "https://vizidox.com/api/certificates" -H "accept: application/json" -H "Bear
```

The Certificate Object

```
{
  "certificate": {
    "uid": "42f51c8b-7429-4667-8017-03009138e1c2",
    "partner": {
      "id": "testdom",
      "name": "testdom"
    },
    "credential": {
      "uid": "1c1b0a7a-a778-4f1c-8d0e-2e1aa4e9581e",
      "issuer": "testdom-issuer",
      "metadata": {"approved_by": "John Smith"},
      "files": [
        {
          "file_hash": "4b4f2ace2c4f4b0d87e3b0bd6f37ef",
          "file_type": "image/png"
        }
      ],
      "expiry_date": [],
      "group_tags": [],
      "tags": [
        "tag0",
        "tag1",
        "tag2"
      ],
      "expiry_date": "2025-02-11T15:34:05.013219+00:00"
    },
    "issued_date": "2020-02-11T15:34:05.013217+00:00",
    "signature": "MewgPhJ9LS3QNMy6g6ttFUfMFVrvMMXUd2Ae8ny5d3DTFfrMmxB6yFuNw7t4TXGEsvqzNNN
```