Quickstart: Verify Verite Credentials

Suggest Edits

Verification is a chain of three client-server requests:

1. Initialization call
2. Manifest call
3. Verification call

A sequential API request is submitted based on the response of the previous request.

Verite verification flow

Initialization call

The verification flow starts with the initialization call.

Endpoint

POST: /verifications

Query

Its query contains the network and the chainId to help the verifier server decide which blockchain to use. Subject is the DeFi application's address on that blockchain.

Initialization Query Payload { "network": "ethereum", "chainId": 1337, "subject": "0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266" }

Response

The response contains two fields: challengeTokenUrl and statusUrl . challengeTokenUrl is for the manifest call and statusUrl is for asynchronous status checks.

Initialization Response { 'challengeTokenUrl': 'https://verifier-sandbox.circle.com/api/v1/verifications/14b2ade1-389c-4acc-87c6-89d96c95af28', 'statusUrl': 'https://verifier-sandbox.circle.com/api/v1/verifications/14b2ade1-389c-4acc-87c6-89d96c95af28/status' }

Manifest call

Endpoint

GET: /verifications/{id}

The Manifest step calls the challengeTokenUrl from the initialization call's response. It is a GET call without additional parameters.

Response

The response is a manifest list that the server accepts. It contains many meta fields as well as fields that should be carried to the verification call.

Manifest call response { "id": "14b2ade1-389c-4acc-87c6-89d96c95af28", "type": "https://circle.com/types/VerificationRequest", "from": "did:web:circle.com", "created_time": "2022-09-26T18:42:17.687Z", "expires_time": "2022-10-26T18:42:17.687Z", "reply_url": "https://verifier-sandbox.circle.com/api/v1/verifications/14b2ade1-389c-4acc-87c6-89d96c95af28", "body": { "status_url": "https://verifier-sandbox.circle.com/api/v1/verifications/14b2ade1-389c-4acc-87c6-89d96c95af28/status", "challenge": "126c210d-f458-4ec7-a0d2-0e0a2b8aac42", "presentation_definition": { "id": "14b2ade1-389c-4acc-87c6-89d96c95af28", "format": { "jwt": { "alg": ["EdDSA", "ES256K"] }, "jwt_vc": { "alg": ["EdDSA", "ES256K"] }, "jwt_vp": { "alg": ["EdDSA", "ES256K"] } }, "input_descriptors": [{ "id": "kybpaml_input", "name": "Proof of KYBP", "schema": [{ "uri": "https://verite.id/definitions/processes/kycaml/0.0.1/generic--usa-legal_person", "required": true }], "purpose": "Please provide a valid credential from a KYBP/AML issuer", "constraints": { "fields": [{ "path": [".issuer.id", ".issuer", ".vc.issuer", ".iss"], "filter": { "type": "string", "pattern": "^did:web:issuer-sandbox.circle.com|^did:web:assets.circle.com" }, "purpose": "The issuer of the credential must be trusted", "predicate": "required" }, { "path": [".credentialSubject.KYBPAMLAttestation.process", ".vc.credentialSubject.KYBPAMLAttestation.process", ".KYBPAMLAttestation.process"], "filter": { "type": "string" }, "purpose": "The process used for KYBP/AML.", "predicate": "required" }, { "path": [".credentialSubject.KYBPAMLAttestation.approvalDate", ".vc.credentialSubject.KYBPAMLAttestation.approvalDate", ".KYBPAMLAttestation.approvalDate"], "filter": { "type": "string", "pattern": "^[0-9]{4}-[0-9]{2}-[0-9]{2}T[0-9]{2}:[0-9]{2}:[0-9]{2}.[0-9]{3}Z" }, "purpose": "The date upon which this KYBP/AML Attestation was issued.", "predicate": "required" }], "statuses": { "active": { "directive": "required" }, "revoked": { "directive": "disallowed" } }, "is_holder": [{ "field_id": ["subjectId"], "directive": "required" }] } }] } } } reply_url field is the URL that the client should call in the verification step.

Verification call

Endpoint

POST: /verifications/{id}

The URL that the client should call at this step is the reply_url field in the response of the previous call.

Request

JSON { "credential_fulfillment": { "descriptor_map": [{ "format": "jwt_vc", "id": "proofOfIdentifierControlVP", "path": ".presentation.credential[0]" }], "id": "e921d5b2-5293-4297-a467-907f9d565e4e", "manifest_id": "KYBPAMLAttestation" }, "presentation_submission": { "id": "b68fda51-21aa-4cdf-84b7-d452b1c9c3cc", "descriptor_map": [{ "format": "jwt_vc", "id": "kybpaml_input", "path": ".verifiableCredential[0]" }], "definition_id": "14b2ade1-389c-4acc-87c6-89d96c95af28" }, "vp": { "@context": ["https://www.w3.org/2018/credentials/v1"], "type": ["VerifiablePresentation", "CredentialFulfillment"], "verifiableCredential": [
"eyJ0eXAiOiJKV1QiLCJhbGciOiJFUzI1NksifQ.eyJzdWIiOiJkaWQ6a2V5OnpRM3NodjM3OFB2a011UnJZTUdGVjlhM010S3BKa3RlcWIyZFViUU1FTXZ0V2MydEUiLCJuYmYiOjE2NTg5NTM4MjIsImlzcyI
-PQyKfEz8HUO3sYoNIuZgRILZTPOhrj10wPyjw"], "holder": "did:key:zQ3shv378PvkMuRrYMGFV9a3MtKpJkteqb2dUbQMEMvtWc2tE" }, "nonce": "126c210d-f458-4ec7-a0d2-0e0a2b8aac42", "sub": "0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266", "iss": "did:key:zQ3shv378PvkMuRrYMGFV9a3MtKpJkteqb2dUbQMEMvtWc2tE" } There are several fields in the query that the clients need to carry from either the response of the previous query or the verifiable credential (VC) fetched from the wallet.

field in the query field in the response of the previous call iss VC's DID sub wallet's address nonce body.challenge presentation_submission.definition_id body.presentation_definition.id vp.verifiableCredential VC that is from wallet. vp.holder wallet's address

Response

A successful verification returns a JSON blob, as illustrated below.

Verification call response { "status": "success", "verificationResult": { "schema": "verite.id/definitions/processes/kycaml/0.0.1/generic--usa-legal_person", "subject": "0xb4d00788f75f27b85285b3af21bdc16b3336d91e", "expiration": 1663906402, "verifier_verification_id": "2912ef9f-10af-43a8-82a2-b992a8cee111" }, "signature": "0x52e5b656006abdc0b741e1671ab687d71d1369455d4d6ee9ede35262b867dc9e49fc2f59aadbe4829b0ce53c7c52b9a88509e28a888025ec5e503f1cc14da8fb1c" } field name meaning status The status of the verification. verificationResult The result of the verification. signature The signed string, by verifier, of the verificationResult Updated5 months ago * Table of Contents * * Initialization call * * Manifest call * * Verification call