Scan Open API Reference

external

GET /v0/dso

Example request:

```
GET /v0/dso HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –
  ok
  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "sv_user": "string",
    "sv_party_id": "string",
    "dso_party_id": "string",
    "voting_threshold": 1,
    "latest_mining_round": {
        "contract": {
            "template_id": "string",
            "contract_id": "string",
            "payload": {},
            "created_event_blob": "string",
            "created_at": "string"
        },
        "domain_id": "string"
    },
    "amulet_rules": {
        "contract": {
            "template_id": "string",
            "contract_id": "string",
            "payload": {},
            "created_event_blob": "string",
            "created_at": "string"
        },
        "domain_id": "string"
    },
    "dso_rules": {
        "contract": {
            "template_id": "string",
            "contract_id": "string",
            "payload": {},
            "created_event_blob": "string",
            "created_at": "string"
        },
        "domain_id": "string"
    },
    "sv_node_states": [
        {
            "contract": {
                "template_id": "string",
                "contract_id": "string",
                "payload": {},
                "created_event_blob": "string",
                "created_at": "string"
            },
            "domain_id": "string"
        }
    ],
    "initial_round": "string"
}
```

GET /v0/validators/validator-faucets

For every argument that is a valid onboarded validator, return statistics on its liveness activity, according to on-ledger state at the time of the request.

Query Parameters:
- **validator_ids** (*array*) – A list of validator party IDs, one per specification of the parameter. Any party IDs not matching onboarded validators will be ignored (Required)

Example request:

```
GET /v0/validators/validator-faucets?validator_ids=%5B%27string%27%5D HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –

  ok

  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "validatorsReceivedFaucets": [
          {
              "validator": "string",
              "numRoundsCollected": 1,
              "numRoundsMissed": 1,
              "firstCollectedInRound": 1,
              "lastCollectedInRound": 1
          }
      ]
  }
  ```

- 400 Bad Request –

  bad request

  Example response:

  ```
  HTTP/1.1 400 Bad Request
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

- 404 Not Found –

  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/scans

Retrieve Canton scan configuration for all SVs, grouped by connected synchronizer ID

Example request:

```
GET /v0/scans HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –
  ok

  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "scans": [
          {
              "domainId": "string",
              "scans": [
                  {
                      "publicUrl": "string",
                      "svName": "string"
                  }
              ]
          }
      ]
  }
  ```

GET /v0/admin/validator/licenses

List all validators currently approved by members of the DSO, paginated, sorted newest-first.

Query Parameters:
- **after** (*integer*) – A `next_page_token` from a prior response; if absent, return the first page.
- **limit** (*integer*) – Maximum number of elements to return, 1000 by default.

Example request:

```
GET /v0/admin/validator/licenses HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –

  ok

  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "validator_licenses": [
          {
              "template_id": "string",
              "contract_id": "string",
              "payload": {},
              "created_event_blob": "string",
              "created_at": "string"
          }
      ],
      "next_page_token": 1
  }
  ```

GET /v0/dso-sequencers

Retrieve Canton sequencer configuration for all SVs, grouped by connected synchronizer ID

Example request:

```
GET /v0/dso-sequencers HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –

  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "domainSequencers": [
        {
            "domainId": "string",
            "sequencers": [
                {
                    "migrationId": 1,
                    "id": "string",
                    "url": "string",
                    "svName": "string",
                    "availableAfter": "2026-01-
16T12:26:57.979425"
                }
            ]
        }
    ]
}
```

GET /v0/domains/{domain_id}/parties/{party_id}/participant-id

Get the ID of the participant hosting a given party. This will fail if there are multiple party-to-participant mappings for the given synchronizer and party, which is not currently supported.

Parameters:
- **domain_id** (*string*) – The synchronizer ID to look up a mapping for.
- **party_id** (*string*) – The party ID to lookup a participant ID for.

Example request:

```
GET /v0/domains/{domain_id}/parties/{party_id}/participant-id HTTP/1.1
Host: example.com
```

Status Codes:
- **200 OK** –
  ok
  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "participant_id": "string"
  }
  ```

- **404 Not Found** –
  not found

Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

- 500 Internal Server Error –

  internal server error

  Example response:

```
HTTP/1.1 500 Internal Server Error
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/domains/{domain_id}/members/{member_id}/traffic-status

Get a member's traffic status as reported by the sequencer, according to ledger state at the time of the request.

Parameters:
- **domain_id** (*string*) – The synchronizer ID to look up traffic for.
- **member_id** (*string*) – The participant or mediator whose traffic to look up, in the format `code::id::fingerprint` where `code` is `PAR` or `MED`.

Example request:

```
GET /v0/domains/{domain_id}/members/{member_id}/traffic-status HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –

  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "traffic_status": {
        "actual": {
            "total_consumed": 1,
            "total_limit": 1
        },
        "target": {
            "total_purchased": 1
        }
    }
}
```

- **400 Bad Request** –

  bad request

  Example response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
    "error": "string"
}
```

- **404 Not Found** –

  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

- **500 Internal Server Error** –

  internal server error

  Example response:

```
HTTP/1.1 500 Internal Server Error
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/closed-rounds

Every closed mining round on the ledger still in post-close process for the connected Splice network, in round number order, earliest-first.

Example request:

```
GET /v0/closed-rounds HTTP/1.1
Host: example.com
```

Status Codes:
- **200 OK** –
  ok

  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "rounds": [
          {
              "template_id": "string",
              "contract_id": "string",
              "payload": {},
              "created_event_blob": "string",
              "created_at": "string"
          }
      ]
  }
  ```

POST /v0/open-and-issuing-mining-rounds

All current open and issuing mining rounds, if the request is empty; passing contract IDs in the request can reduce the response data for polling/client-cache-update efficiency.

Example request:

```
POST /v0/open-and-issuing-mining-rounds HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "cached_open_mining_round_contract_ids": [
        "string"
    ],
    "cached_issuing_round_contract_ids": [
        "string"
    ]
}
```

Status Codes:

- 200 OK –

  ok

  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "time_to_live_in_microseconds": 1,
      "open_mining_rounds": {},
      "issuing_mining_rounds": {}
  }
  ```

POST /v2/updates

Returns the update history in ascending order, paged, from ledger begin or optionally starting after a record time. Compared to `/v1/updates`, the `/v2/updates` removes the `offset` field in responses, which was hardcoded to 1 in `/v1/updates` for compatibility, and is now removed. `/v2/updates` sorts events lexicographically in `events_by_id` by `ID` for convenience, which should not be confused with the order of events in the transaction, for this you should rely on the order of `root_event_ids` and `child_event_ids`. Updates are ordered lexicographically by `(migration id, record time)`. For a given migration id, each update has a unique record time. The record time ranges of different migrations may overlap, i.e., it is not guaranteed that the maximum record time of one migration is smaller than the minimum record time of the next migration, and there may be two updates with the same record time but different migration ids.

Example request:

```
POST /v2/updates HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "after": {
        "after_migration_id": 1,
        "after_record_time": "string"
    },
    "page_size": 1,
    "daml_value_encoding": "compact_json"
}
```

Status Codes:

- 200 OK –

ok

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "transactions": [
        {
            "update_id": "string",
            "migration_id": 1,
            "workflow_id": "string",
            "record_time": "string",
            "synchronizer_id": "string",
            "effective_at": "string",
            "root_event_ids": [
                "string"
            ],
            "events_by_id": {}
        },
        {
            "update_id": "string",
            "offset": "string",
            "record_time": "string",
            "event": {
                "submitter": "string",
                "source_synchronizer": "string",
                "target_synchronizer": "string",
                "migration_id": 1,
                "unassign_id": "string",
                "created_event": {
                    "event_type": "string",
                    "event_id": "string",
                    "contract_id": "string",
                    "template_id": "string",
                    "package_name": "string",
                    "create_arguments": {},
                    "created_at": "2026-01-16T12:26:57.979425",
                    "signatories": [
                        "string"
                    ],
                    "observers": [
                        "string"
                    ]
                },
                "reassignment_counter": 1
            }
        }
    ]
}
```

- 400 Bad Request –

bad request

Example response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
    "error": "string"
}
```

- 500 Internal Server Error –

  internal server error

  Example response:

```
HTTP/1.1 500 Internal Server Error
Content-Type: application/json

{
    "error": "string"
}
```

GET /v2/updates/{update_id}

Returns the update with the given update_id. Compared to `/v1/updates/{update_id}`, the `/v2/updates/{update_id}` removes the `offset` field in responses, which was hardcoded to 1 in `/v1/updates/{update_id}` for compatibility, and is now removed. `/v2/updates/{update_id}` sorts events lexicographically in `events_by_id` by `ID` for convenience, which should not be confused with the order of events in the transaction, for this you should rely on the order of `root_event_ids` and `child_event_ids`.

Parameters:	• **update_id** (*string*)
Query Parameters:	• **daml_value_encoding** (*string*)

Example request:

```
GET /v2/updates/{update_id} HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –

  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "update_id": "string",
    "migration_id": 1,
    "workflow_id": "string",
    "record_time": "string",
    "synchronizer_id": "string",
    "effective_at": "string",
    "root_event_ids": [
        "string"
    ],
    "events_by_id": {}
}
```

- 400 Bad Request –
  bad request
  Example response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
    "error": "string"
}
```

- 404 Not Found –
  not found
  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

- 500 Internal Server Error –
  internal server error
  Example response:

```
HTTP/1.1 500 Internal Server Error
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/state/acs/snapshot-timestamp

Returns the timestamp of the most recent snapshot before the given date, for the given migration_id. This corresponds to the record time of the last transaction in the snapshot.

Query Parameters:
- **before** (*string*) – (Required)
- **migration_id** (*integer*) – (Required)

Example request:

```
GET /v0/state/acs/snapshot-timestamp?before=2026-01-
16T12%3A26%3A57.979425&migration_id=1 HTTP/1.1
Host: example.com
```

Status Codes:
- **200 OK** –
  ok

  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "record_time": "2026-01-16T12:26:57.979425"
  }
  ```

- **400 Bad Request** –
  bad request

  Example response:

  ```
  HTTP/1.1 400 Bad Request
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

- **404 Not Found** –
  not found

  Example response:

  ```
  HTTP/1.1 404 Not Found
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

- **500 Internal Server Error** –
  internal server error

  Example response:

```
HTTP/1.1 500 Internal Server Error
Content-Type: application/json


{
    "error": "string"
}
```

POST /v0/state/acs

Returns the ACS in creation date ascending order, paged, for a given migration id and record time.

Example request:

```
POST /v0/state/acs HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "migration_id": 1,
    "record_time": "2026-01-16T12:26:57.979425",
    "record_time_match": "exact",
    "after": 1,
    "page_size": 1,
    "party_ids": [
        "string"
    ],
    "templates": [
        "string"
    ]
}
```

Status Codes:

- **200 OK** –
  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "record_time": "2026-01-16T12:26:57.979425",
    "migration_id": 1,
    "created_events": [
        {
            "event_type": "string",
            "event_id": "string",
            "contract_id": "string",
            "template_id": "string",
            "package_name": "string",
            "create_arguments": {},
            "created_at": "2026-01-16T12:26:57.979425",
            "signatories": [
                "string"
            ],
            "observers": [
                "string"
            ]
        }
    ],
    "next_page_token": 1
}
```

- **400 Bad Request** –
  bad request
  Example response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
    "error": "string"
}
```

- **404 Not Found** –
  not found
  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

- **500 Internal Server Error** –
  internal server error
  Example response:

```
HTTP/1.1 500 Internal Server Error
Content-Type: application/json

{
    "error": "string"
}
```

POST /v0/state/acs/force

Takes a snapshot of the ACS at the current time. The responses can be used as parameters to `/v0/state/acs` to retrieve the snapshot. Disabled in production environments due to its persistent alteration of the behavior of future invocations of `/v0/state/acs`, as it causes an immediate internal snapshot and delay in the next automatic snapshot.

Status Codes:
- **200 OK** –
  ok
  Example response:
  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "record_time": "2026-01-16T12:26:57.979425",
      "migration_id": 1
  }
  ```

- **400 Bad Request** –
  bad request
  Example response:
  ```
  HTTP/1.1 400 Bad Request
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

- **500 Internal Server Error** –
  internal server error
  Example response:
  ```
  HTTP/1.1 500 Internal Server Error
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

POST /v0/holdings/state

Returns the active amulet contracts for a given migration id and record time, in creation date ascending order, paged.

Example request:

```
POST /v0/holdings/state HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "migration_id": 1,
    "record_time": "2026-01-16T12:26:57.979425",
    "record_time_match": "exact",
    "after": 1,
    "page_size": 1,
    "owner_party_ids": [
        "string"
    ]
}
```

Status Codes:
- 200 OK –

  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "record_time": "2026-01-16T12:26:57.979425",
    "migration_id": 1,
    "created_events": [
        {
            "event_type": "string",
            "event_id": "string",
            "contract_id": "string",
            "template_id": "string",
            "package_name": "string",
            "create_arguments": {},
            "created_at": "2026-01-16T12:26:57.979425",
            "signatories": [
                "string"
            ],
            "observers": [
                "string"
            ]
        }
    ],
    "next_page_token": 1
}
```

- **400 Bad Request** –
  bad request
  Example response:

  ```
  HTTP/1.1 400 Bad Request
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

- **404 Not Found** –
  not found
  Example response:

  ```
  HTTP/1.1 404 Not Found
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

- **500 Internal Server Error** –
  internal server error
  Example response:

  ```
  HTTP/1.1 500 Internal Server Error
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

POST /v0/holdings/summary

Returns the summary of active amulet contracts for a given migration id and record time, for the given parties. This is an aggregate of `/v0/holdings/state` by owner party ID with better performance than client-side computation.

Example request:

```
POST /v0/holdings/summary HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "migration_id": 1,
    "record_time": "2026-01-16T12:26:57.979425",
    "record_time_match": "exact",
    "owner_party_ids": [
        "string"
    ],
    "as_of_round": 1
}
```

Status Codes:

- 200 OK –

  ok

  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "record_time": "2026-01-16T12:26:57.979425",
      "migration_id": 1,
      "computed_as_of_round": 1,
      "summaries": [
          {
              "party_id": "string",
              "total_unlocked_coin": "string",
              "total_locked_coin": "string",
              "total_coin_holdings": "string",
              "accumulated_holding_fees_unlocked": "string",
              "accumulated_holding_fees_locked": "string",
              "accumulated_holding_fees_total": "string",
              "total_available_coin": "string"
          }
      ]
  }
  ```

- 400 Bad Request –

  bad request

  Example response:

  ```
  HTTP/1.1 400 Bad Request
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

- **404 Not Found** –
  not found
  Example response:

  ```
  HTTP/1.1 404 Not Found
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

- **500 Internal Server Error** –
  internal server error
  Example response:

  ```
  HTTP/1.1 500 Internal Server Error
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

GET /v0/ans-entries

Lists all non-expired ANS entries whose names are prefixed with the `name_prefix`, up to `page_size` entries.

Query Parameters:
- **name_prefix** (*string*) – Every result's name will start with this substring; if empty or absent, all entries will be listed. Does not have to be a whole word or segment; any substring will be accepted.
- **page_size** (*integer*) – The maximum number of results returned. Older (but still non-expired) results are listed first. (Required)

Example request:

```
GET /v0/ans-entries?page_size=1 HTTP/1.1
Host: example.com
```

Status Codes:
- **200 OK** –
  ok
  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "entries": [
        {
            "contract_id": "string",
            "user": "string",
            "name": "string",
            "url": "string",
            "description": "string",
            "expires_at": "2026-01-16T12:26:57.979425"
        }
    ]
}
```

GET /v0/ans-entries/by-party/{party}

If present, the first ANS entry for user `party` according to `name` lexicographic order.

Parameters:
- **party** (*string*) – The user party ID that holds the ANS entry.

Example request:

```
GET /v0/ans-entries/by-party/{party} HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –

  ok

  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "entry": {
          "contract_id": "string",
          "user": "string",
          "name": "string",
          "url": "string",
          "description": "string",
          "expires_at": "2026-01-16T12:26:57.979425"
      }
  }
  ```

- 404 Not Found –

  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/ans-entries/by-name/{name}

If present, the ANS entry named exactly name .

Parameters: • **name** (*string*)

Example request:

```
GET /v0/ans-entries/by-name/{name} HTTP/1.1
Host: example.com
```

Status Codes: • 200 OK –

ok

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "entry": {
        "contract_id": "string",
        "user": "string",
        "name": "string",
        "url": "string",
        "description": "string",
        "expires_at": "2026-01-16T12:26:57.979425"
    }
}
```

• 404 Not Found –

not found

Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/dso-party-id

The party ID of the DSO for the Splice network connected by this Scan app.

Example request:

```
GET /v0/dso-party-id HTTP/1.1
Host: example.com
```

Status Codes:

- **200 OK** –

  ok

  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "dso_party_id": "string"
  }
  ```

POST /v0/events

Returns the event history in ascending order, paged, from ledger begin or optionally starting after a record time. An event bears some combination of a transaction, a contract reassignment, and a verdict. Events are ordered lexicographically by `(migration id, record time)`. For a given migration id, each event has a unique record time. The record time ranges of different migrations may overlap, i.e., it is not guaranteed that the maximum record time of one migration is smaller than the minimum record time of the next migration, and there may be two updates with the same record time but different migration ids.

Example request:

```
POST /v0/events HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "after": {
        "after_migration_id": 1,
        "after_record_time": "string"
    },
    "page_size": 1,
    "daml_value_encoding": "compact_json"
}
```

Status Codes:
- **200 OK** –
  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "events": [
        {
            "update": {
                "update_id": "string",
                "migration_id": 1,
                "workflow_id": "string",
                "record_time": "string",
                "synchronizer_id": "string",
                "effective_at": "string",
                "root_event_ids": [
                    "string"
                ],
                "events_by_id": {}
            },
            "verdict": {
                "update_id": "string",
                "migration_id": 1,
                "domain_id": "string",
                "record_time": "string",
                "finalization_time": "string",
                "submitting_parties": [
                    "string"
                ],
                "submitting_participant_uid": "string",
                "verdict_result": "VERDICT_RESULT_UNSPECIFIED",
                "mediator_group": 1,
                "transaction_views": {
                    "views": [
                        {
                            "view_id": 1,
                            "informees": [
                                "string"
                            ],
                            "confirming_parties": [
                                {
                                    "parties": [
                                        "string"
                                    ],
                                    "threshold": 1
                                }
                            ],
                            "sub_views": [
                                1
                            ]
                        }
                    ],
                    "root_views": [
                        1
                    ]
                }
            }
        }
    ]
}
```

- **400 Bad Request** –

  bad request

  Example response:

  ```
  HTTP/1.1 400 Bad Request
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

- **500 Internal Server Error** –

  internal server error

  Example response:

  ```
  HTTP/1.1 500 Internal Server Error
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

GET /v0/events/{update_id}

Returns the event with the given update_id. An event bears some combination of a transaction, a contract reassignment, and a verdict.

Parameters:	• **update_id** (*string*)
Query Parameters:	• **daml_value_encoding** (*string*)

Example request:

```
GET /v0/events/{update_id} HTTP/1.1
Host: example.com
```

Status Codes:
- **200 OK** –

  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "update": {
        "update_id": "string",
        "migration_id": 1,
        "workflow_id": "string",
        "record_time": "string",
        "synchronizer_id": "string",
        "effective_at": "string",
        "root_event_ids": [
            "string"
        ],
        "events_by_id": {}
    },
    "verdict": {
        "update_id": "string",
        "migration_id": 1,
        "domain_id": "string",
        "record_time": "string",
        "finalization_time": "string",
        "submitting_parties": [
            "string"
        ],
        "submitting_participant_uid": "string",
        "verdict_result": "VERDICT_RESULT_UNSPECIFIED",
        "mediator_group": 1,
        "transaction_views": {
            "views": [
                {
                    "view_id": 1,
                    "informees": [
                        "string"
                    ],
                    "confirming_parties": [
                        {
                            "parties": [
                                "string"
                            ],
                            "threshold": 1
                        }
                    ],
                    "sub_views": [
                        1
                    ]
                }
            ],
            "root_views": [
                1
            ]
        }
    }
}
```

- 400 Bad Request –

bad request

Example response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
    "error": "string"
}
```

- **404 Not Found** –

  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

- **500 Internal Server Error** –

  internal server error

  Example response:

```
HTTP/1.1 500 Internal Server Error
Content-Type: application/json

{
    "error": "string"
}
```

internal

GET /v0/sv-bft-sequencers

Retrieve Canton BFT sequencer configuration for this SV, for each configured Synchronizer

Example request:

```
GET /v0/sv-bft-sequencers HTTP/1.1
Host: example.com
```

Status Codes:
- **200 OK** –

  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "bftSequencers": [
        {
            "migrationId": 1,
            "id": "string",
            "p2pUrl": "string"
        }
    ]
}
```

POST /v0/amulet-rules

Example request:

```
POST /v0/amulet-rules HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "cached_amulet_rules_contract_id": "string",
    "cached_amulet_rules_domain_id": "string"
}
```

Status Codes:
- 200 OK –
  ok

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "amulet_rules_update": {
        "contract": {
            "template_id": "string",
            "contract_id": "string",
            "payload": {},
            "created_event_blob": "string",
            "created_at": "string"
        },
        "domain_id": "string"
    }
}
```

POST /v0/external-party-amulet-rules

Example request:

```
POST /v0/external-party-amulet-rules HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "cached_external_party_amulet_rules_contract_id": "string",
    "cached_external_party_amulet_rules_domain_id": "string"
}
```

Status Codes:
- **200 OK** –

  ok

  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "external_party_amulet_rules_update": {
          "contract": {
              "template_id": "string",
              "contract_id": "string",
              "payload": {},
              "created_event_blob": "string",
              "created_at": "string"
          },
          "domain_id": "string"
      }
  }
  ```

POST /v0/ans-rules

Example request:

```
POST /v0/ans-rules HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "cached_ans_rules_contract_id": "string",
    "cached_ans_rules_domain_id": "string"
}
```

Status Codes:
- **200 OK** –

  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "ans_rules_update": {
        "contract": {
            "template_id": "string",
            "contract_id": "string",
            "payload": {},
            "created_event_blob": "string",
            "created_at": "string"
        },
        "domain_id": "string"
    }
}
```

GET /v0/featured-apps

List every FeaturedAppRight registered with the DSO on the ledger.

Example request:

```
GET /v0/featured-apps HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –
  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "featured_apps": [
        {
            "template_id": "string",
            "contract_id": "string",
            "payload": {},
            "created_event_blob": "string",
            "created_at": "string"
        }
    ]
}
```

GET /v0/featured-apps/{provider_party_id}

If `provider_party_id` has a `FeaturedAppRight` registered with the DSO, return it; `featured_app_right` will be empty otherwise.

Parameters: • provider_party_id (*string*)

Example request:

```
GET /v0/featured-apps/{provider_party_id} HTTP/1.1
Host: example.com
```

Status Codes: • 200 OK –

ok

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "featured_app_right": {
        "template_id": "string",
        "contract_id": "string",
        "payload": {},
        "created_event_blob": "string",
        "created_at": "string"
    }
}
```

GET /v0/top-validators-by-validator-faucets

Get a list of top validators by number of rounds in which they collected faucets, and basis statistics on their round collection history

Query Parameters: • **limit** (*integer*) – Maximum number of validator records that may be returned in the response (Required)

Example request:

```
GET /v0/top-validators-by-validator-faucets?limit=1 HTTP/1.1
Host: example.com
```

Status Codes: • 200 OK –

ok

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "validatorsByReceivedFaucets": [
        {
            "validator": "string",
            "numRoundsCollected": 1,
            "numRoundsMissed": 1,
            "firstCollectedInRound": 1,
            "lastCollectedInRound": 1
        }
    ]
}
```

- 400 Bad Request –
  bad request

  Example response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
    "error": "string"
}
```

- 404 Not Found –
  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/transfer-preapprovals/by-party/{party}

Lookup a TransferPreapproval by the receiver party.

Parameters:
- **party** (*string*)

Example request:

```
GET /v0/transfer-preapprovals/by-party/{party} HTTP/1.1
Host: example.com
```

Status Codes:

- **200 OK** –
  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "transfer_preapproval": {
        "contract": {
            "template_id": "string",
            "contract_id": "string",
            "payload": {},
            "created_event_blob": "string",
            "created_at": "string"
        },
        "domain_id": "string"
    }
}
```

- **404 Not Found** –
  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/transfer-command-counter/{party}

Lookup a TransferCommandCounter by the receiver party.

Parameters:

- **party** (*string*)

Example request:

```
GET /v0/transfer-command-counter/{party} HTTP/1.1
Host: example.com
```

Status Codes:

- **200 OK** –
  ok
  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "transfer_command_counter": {
        "contract": {
            "template_id": "string",
            "contract_id": "string",
            "payload": {},
            "created_event_blob": "string",
            "created_at": "string"
        },
        "domain_id": "string"
    }
}
```

- **404 Not Found** –

  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/transfer-command/status

Retrieve the status of all transfer commands (up to a limit of 100) of the given sender for the specified nonce.

| Query Parameters: | • **sender** (*string*) – (Required) |
| | • **nonce** (*integer*) – (Required) |

Example request:

```
GET /v0/transfer-command/status?sender=string&nonce=1 HTTP/1.1
Host: example.com
```

Status Codes:	• **200 OK** –
	ok
	Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "transfer_commands_by_contract_id": {}
}
```

- 404 Not Found –

  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/migrations/schedule

If the DSO has scheduled a synchronizer upgrade, return its planned time and the new migration ID.

Example request:

```
GET /v0/migrations/schedule HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –

  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "time": "2026-01-16T12:26:57.979425",
    "migration_id": 1
}
```

- 404 Not Found – No migration scheduled

GET /v0/synchronizer-identities/{domain_id_prefix}

Parameters:
- **domain_id_prefix** (*string*)
```

**Example request:**

```
GET /v0/synchronizer-identities/{domain_id_prefix} HTTP/1.1
Host: example.com
```

Status Codes:
- **200 OK** –

  ok

  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "sequencer_id": "string",
      "sequencer_identity_transactions": [
          "string"
      ],
      "mediator_id": "string",
      "mediator_identity_transactions": [
          "string"
      ]
  }
  ```

- **404 Not Found** –

  not found

  Example response:

  ```
  HTTP/1.1 404 Not Found
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

GET /v0/synchronizer-bootstrapping-transactions/{domain_id_prefix}

Parameters:
- **domain_id_prefix** (*string*)

Example request:

```
GET /v0/synchronizer-bootstrapping-transactions/{domain_id_prefix} HTTP/1.1
Host: example.com
```

Status Codes:
- **200 OK** –

ok

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "domain_parameters": "string",
    "sequencer_domain_state": "string",
    "mediator_domain_state": "string"
}
```

- **404 Not Found** –

  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/splice-instance-names

Retrieve the UI names of various elements of this Splice network.

Example request:

```
GET /v0/splice-instance-names HTTP/1.1
Host: example.com
```

Status Codes:
- **200 OK** –

  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "network_name": "string",
    "network_favicon_url": "string",
    "amulet_name": "string",
    "amulet_name_acronym": "string",
    "name_service_name": "string",
    "name_service_name_acronym": "string"
}
```

- 404 Not Found –

  not found

  Example response:

  ```
  HTTP/1.1 404 Not Found
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

POST /v0/voterequest

Look up several `VoteRequest` s at once by their contract IDs.

Example request:

```
POST /v0/voterequest HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "vote_request_contract_ids": [
        "string"
    ]
}
```

Status Codes:
- 200 OK –

  ok

  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "vote_requests": [
          {
              "template_id": "string",
              "contract_id": "string",
              "payload": {},
              "created_event_blob": "string",
              "created_at": "string"
          }
      ]
  }
  ```

GET /v0/voterequests/{vote_request_contract_id}

Look up a `VoteRequest` by contract ID.

Parameters:
- vote_request_contract_id (*string*)

Example request:

```
GET /v0/voterequests/{vote_request_contract_id} HTTP/1.1
Host: example.com
```

Status Codes:
- 404 Not Found –
  VoteRequest contract not found.
  Example response:

  ```
  HTTP/1.1 404 Not Found
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

- 200 OK –
  ok
  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "dso_rules_vote_request": {
          "template_id": "string",
          "contract_id": "string",
          "payload": {},
          "created_event_blob": "string",
          "created_at": "string"
      }
  }
  ```

GET /v0/admin/sv/voterequests

List all active `VoteRequest`s.

Example request:

```
GET /v0/admin/sv/voterequests HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –
  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "dso_rules_vote_requests": [
        {
            "template_id": "string",
            "contract_id": "string",
            "payload": {},
            "created_event_blob": "string",
            "created_at": "string"
        }
    ]
}
```

POST /v0/admin/sv/voteresults

Example request:

```
POST /v0/admin/sv/voteresults HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "actionName": "string",
    "accepted": true,
    "requester": "string",
    "effectiveFrom": "string",
    "effectiveTo": "string",
    "limit": 1
}
```

Status Codes:
- 200 OK –
  ok
  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "dso_rules_vote_results": [
        {}
    ]
}
```

POST /v0/backfilling/migration-info

List all previous synchronizer migrations in this Splice network's history.

Example request:

```
POST /v0/backfilling/migration-info HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "migration_id": 1
}
```

Status Codes:
- 200 OK –

  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "previous_migration_id": 1,
    "record_time_range": [
        {
            "synchronizer_id": "string",
            "min": "2026-01-16T12:26:57.979425",
            "max": "2026-01-16T12:26:57.979425"
        }
    ],
    "last_import_update_id": "string",
    "complete": true,
    "import_updates_complete": true
}
```

- 404 Not Found –

  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json


{
    "error": "string"
}
```

POST /v0/backfilling/updates-before

Retrieve transactions and synchronizer reassignments prior to the request's specification.

Example request:

```
POST /v0/backfilling/updates-before HTTP/1.1
Host: example.com
Content-Type: application/json


{
    "migration_id": 1,
    "synchronizer_id": "string",
    "before": "2026-01-16T12:26:57.979425",
    "at_or_after": "2026-01-16T12:26:57.979425",
    "count": 1
}
```

Status Codes:
- 200 OK –
  ok
  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "transactions": [
        {
            "update_id": "string",
            "migration_id": 1,
            "workflow_id": "string",
            "record_time": "string",
            "synchronizer_id": "string",
            "effective_at": "string",
            "offset": "string",
            "root_event_ids": [
                "string"
            ],
            "events_by_id": {}
        },
        {
            "update_id": "string",
            "offset": "string",
            "record_time": "string",
            "event": {
                "submitter": "string",
                "source_synchronizer": "string",
                "target_synchronizer": "string",
                "migration_id": 1,
                "unassign_id": "string",
                "created_event": {
                    "event_type": "string",
                    "event_id": "string",
                    "contract_id": "string",
                    "template_id": "string",
                    "package_name": "string",
                    "create_arguments": {},
                    "created_at": "2026-01-16T12:26:57.979425",
                    "signatories": [
                        "string"
                    ],
                    "observers": [
                        "string"
                    ]
                },
                "reassignment_counter": 1
            }
        }
    ]
}
```

- 404 Not Found –
not found

Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json


{
    "error": "string"
}
```

GET /v0/backfilling/status

Retrieve the status of the backfilling process.

Example request:

```
GET /v0/backfilling/status HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –
  ok
  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json


  {
      "complete": true
  }
  ```

GET /v0/feature-support

Example request:

```
GET /v0/feature-support HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –
  ok
  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "no_holding_fees_on_transfers": true
}
```

- 500 Internal Server Error –
  internal server error

Example response:

```
HTTP/1.1 500 Internal Server Error
Content-Type: application/json

{
    "error": "string"
}
```

deprecated

POST /v1/updates

Returns the update history in ascending order, paged, from ledger begin or optionally starting after a record time. Unlike /v0/updates, this endpoint returns responses that are consistent across different scan instances. Event ids returned by this endpoint are not comparable to event ids returned by /v0/updates.

Updates are ordered lexicographically by `(migration id, record time)`. For a given migration id, each update has a unique record time. The record time ranges of different migrations may overlap, i.e., it is not guaranteed that the maximum record time of one migration is smaller than the minimum record time of the next migration, and there may be two updates with the same record time but different migration ids. The order of items in events_by_id is not defined.

Example request:

```
POST /v1/updates HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "after": {
        "after_migration_id": 1,
        "after_record_time": "string"
    },
    "page_size": 1,
    "daml_value_encoding": "compact_json"
}
```

Status Codes:

- 200 OK –
  ok

  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "transactions": [
          {
              "update_id": "string",
              "migration_id": 1,
              "workflow_id": "string",
              "record_time": "string",
              "synchronizer_id": "string",
              "effective_at": "string",
              "offset": "string",
              "root_event_ids": [
                  "string"
              ],
              "events_by_id": {}
          },
          {
              "update_id": "string",
              "offset": "string",
              "record_time": "string",
              "event": {
                  "submitter": "string",
                  "source_synchronizer": "string",
                  "target_synchronizer": "string",
                  "migration_id": 1,
                  "unassign_id": "string",
                  "created_event": {
                      "event_type": "string",
                      "event_id": "string",
                      "contract_id": "string",
                      "template_id": "string",
                      "package_name": "string",
                      "create_arguments": {},
                      "created_at": "2026-01-16T12:26:57.979425",
                      "signatories": [
                          "string"
                      ],
                      "observers": [
                          "string"
                      ]
                  },
                  "reassignment_counter": 1
              }
          }
      ]
  }
  ```

- 400 Bad Request –

bad request

Example response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
    "error": "string"
}
```

- 500 Internal Server Error –
  internal server error

  Example response:

```
HTTP/1.1 500 Internal Server Error
Content-Type: application/json

{
    "error": "string"
}
```

GET /v1/updates/{update_id}

Returns the update with the given update_id. Unlike /v0/updates/{update_id}, this endpoint returns responses that are consistent across different scan instances. Event ids returned by this endpoint are not comparable to event ids returned by /v0/updates. The order of items in events_by_id is not defined.

Parameters:	- **update_id** (*string*)
Query Parameters:	- **daml_value_encoding** (*string*)

Example request:

```
GET /v1/updates/{update_id} HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –
  ok
  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "update_id": "string",
    "migration_id": 1,
    "workflow_id": "string",
    "record_time": "string",
    "synchronizer_id": "string",
    "effective_at": "string",
    "offset": "string",
    "root_event_ids": [
        "string"
    ],
    "events_by_id": {}
}
```

- **400 Bad Request** –
  bad request

  Example response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
    "error": "string"
}
```

- **404 Not Found** –
  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

- **500 Internal Server Error** –
  internal server error

  Example response:

```
HTTP/1.1 500 Internal Server Error
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/acs/{party}

Deprecated. Fetch the current SV participant ACS snapshot for the DSO and `party`.

Parameters:
- **party** (*string*)

Query Parameters:
- **record_time** (*string*)

Example request:

```
GET /v0/acs/{party} HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –

  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "acs_snapshot": "string"
}
```

GET /v0/aggregated-rounds

Deprecated. Retrieve the current earliest and latest rounds aggregated for this Scan.

Example request:

```
GET /v0/aggregated-rounds HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –

  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "start": 1,
    "end": 1
}
```

- 404 Not Found –

not found

Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

POST /v0/round-totals

Deprecated. List Amulet statistics for up to 200 closed rounds.

Example request:

```
POST /v0/round-totals HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "start_round": 1,
    "end_round": 1
}
```

Status Codes:
- 200 OK –

ok

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "entries": [
        {
            "closed_round": 1,
            "closed_round_effective_at": "2026-01-
16T12:26:57.979425",
            "app_rewards": "string",
            "validator_rewards": "string",
            "change_to_initial_amount_as_of_round_zero":
"string",
            "change_to_holding_fees_rate": "string",
            "cumulative_app_rewards": "string",
            "cumulative_validator_rewards": "string",

    "cumulative_change_to_initial_amount_as_of_round_zero":
"string",
            "cumulative_change_to_holding_fees_rate": "string",
            "total_amulet_balance": "string"
        }
    ]
}
```

- **400 Bad Request** –

  bad request

  Example response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
    "error": "string"
}
```

- **404 Not Found** –

  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

- **500 Internal Server Error** –

  internal server error

  Example response:

```
HTTP/1.1 500 Internal Server Error
Content-Type: application/json

{
    "error": "string"
}
```

POST /v0/round-party-totals

Deprecated. Retrieve per-party Amulet statistics for up to 50 closed rounds.

Example request:

```
POST /v0/round-party-totals HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "start_round": 1,
    "end_round": 1
}
```

Status Codes:

- 200 OK –

  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "entries": [
        {
            "closed_round": 1,
            "party": "string",
            "app_rewards": "string",
            "validator_rewards": "string",
            "traffic_purchased": 1,
            "traffic_purchased_cc_spent": "string",
            "traffic_num_purchases": 1,
            "cumulative_app_rewards": "string",
            "cumulative_validator_rewards": "string",

"cumulative_change_to_initial_amount_as_of_round_zero":
"string",
            "cumulative_change_to_holding_fees_rate": "string",
            "cumulative_traffic_purchased": 1,
            "cumulative_traffic_purchased_cc_spent": "string",
            "cumulative_traffic_num_purchases": 1
        }
    ]
}
```

- **400 Bad Request** –
  bad request

  Example response:

  ```
  HTTP/1.1 400 Bad Request
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

- **404 Not Found** –
  not found

  Example response:

  ```
  HTTP/1.1 404 Not Found
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

- **500 Internal Server Error** –
  internal server error

  Example response:

  ```
  HTTP/1.1 500 Internal Server Error
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

`GET /v0/total-amulet-balance`

Deprecated, use /registry/metadata/v1/instruments/Amulet token standard metadata API endpoint, see
https://docs.sync.global/app_dev/token_standard/openapi/token_metadata.html.

This endpoint will be removed in a future release

Get the total balance of Amulet in the network.

Query Parameters:
- **asOfEndOfRound** (*integer*) – (Required)

Example request:

```
GET /v0/total-amulet-balance?asOfEndOfRound=1 HTTP/1.1
Host: example.com
```

Status Codes:

- **200 OK** –

  ok

  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "total_balance": "string"
  }
  ```

- **404 Not Found** –

  not found

  Example response:

  ```
  HTTP/1.1 404 Not Found
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

GET /v0/wallet-balance

Deprecated, use /v0/holdings/summary with /v0/state/acs/snapshot-timestamp instead.

This endpoint will be removed in a future release

Get the Amulet balance for a specific party at the end of a closed round.

Query Parameters:

- **party_id** (*string*) – (Required)
- **asOfEndOfRound** (*integer*) – (Required)

Example request:

```
GET /v0/wallet-balance?party_id=string&asOfEndOfRound=1 HTTP/1.1
Host: example.com
```

Status Codes:
- **200 OK** –
  ok

  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "wallet_balance": "string"
  }
  ```

- **404 Not Found** –
  not found

  Example response:

  ```
  HTTP/1.1 404 Not Found
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

GET /v0/amulet-config-for-round

Deprecated. Retrieve some information from the `AmuletRules` selected for the given round

Query Parameters:
- **round** (*integer*) – (Required)

Example request:

```
GET /v0/amulet-config-for-round?round=1 HTTP/1.1
Host: example.com
```

Status Codes:
- **200 OK** –
  ok
  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "amulet_create_fee": "string",
    "holding_fee": "string",
    "lock_holder_fee": "string",
    "transfer_fee": {
        "initial": "string",
        "steps": [
            {
                "amount": "string",
                "rate": "string"
            }
        ]
    }
}
```

- **404 Not Found** –

  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/round-of-latest-data

Deprecated. Get the latest round number for which aggregated data is available and the ledger effective time at which the round was closed.

Example request:

```
GET /v0/round-of-latest-data HTTP/1.1
Host: example.com
```

Status Codes:
- **200 OK** –

  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "round": 1,
    "effectiveAt": "2026-01-16T12:26:57.979425"
}
```

- 404 Not Found –

  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/rewards-collected

Deprecated. Get the total rewards collected ever

Query Parameters:
- **round** (*integer*)

Example request:

```
GET /v0/rewards-collected HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –

  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "amount": "string"
}
```

- 404 Not Found –

  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/top-providers-by-app-rewards

Deprecated. Get a list of top-earning app providers, and the total earned app rewards for each

Query Parameters:
- **round** (*integer*) – (Required)
- **limit** (*integer*) – (Required)

Example request:

```
GET /v0/top-providers-by-app-rewards?round=1&limit=1 HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –
  ok
  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "providersAndRewards": [
          {
              "provider": "string",
              "rewards": "string"
          }
      ]
  }
  ```

- 400 Bad Request –
  bad request
  Example response:

  ```
  HTTP/1.1 400 Bad Request
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

- **404 Not Found** – not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/top-validators-by-validator-rewards

Deprecated. Get a list of top-earning validators, and the total earned validator rewards for each

Query Parameters:	• **round** (*integer*) – (Required)
	• **limit** (*integer*) – (Required)

Example request:

```
GET /v0/top-validators-by-validator-rewards?round=1&limit=1 HTTP/1.1
Host: example.com
```

Status Codes:	• 200 OK –

  ok

  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "validatorsAndRewards": [
        {
            "provider": "string",
            "rewards": "string"
        }
    ]
}
```

- **400 Bad Request** – bad request

  Example response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
    "error": "string"
}
```

- 404 Not Found –
  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/top-validators-by-purchased-traffic

Deprecated. Get a list of validators and their domain fees spends, sorted by the amount of extra traffic purchased

Query Parameters:
- **round** (*integer*) – (Required)
- **limit** (*integer*) – (Required)

Example request:

```
GET /v0/top-validators-by-purchased-traffic?round=1&limit=1 HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –
  ok
  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "validatorsByPurchasedTraffic": [
        {
            "validator": "string",
            "numPurchases": 1,
            "totalTrafficPurchased": 1,
            "totalCcSpent": "string",
            "lastPurchasedInRound": 1
        }
    ]
}
```

- 400 Bad Request –

  bad request

  Example response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
    "error": "string"
}
```

- 404 Not Found –

  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

POST /v0/activities

Deprecated. Lists activities in descending order, paged, optionally starting after a provided event id.

Example request:

```
POST /v0/activities HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "begin_after_id": "string",
    "page_size": 1
}
```

Status Codes:

- [200 OK](#) –
  ok

  Example response:

```
POST /v0/activities HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "begin_after_id": "string",
```

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "activities": [
        {
            "activity_type": "transfer",
            "event_id": "string",
            "offset": "string",
            "date": "2026-01-16T12:26:57.979425",
            "domain_id": "string",
            "round": 1,
            "amulet_price": "string",
            "transfer": {
                "provider": "string",
                "sender": {
                    "party": "string",
                    "input_amulet_amount": "string",
                    "input_app_reward_amount": "string",
                    "input_validator_reward_amount": "string",
                    "input_sv_reward_amount": "string",
                    "input_validator_faucet_amount": "string",
                    "sender_change_fee": "string",
                    "sender_change_amount": "string",
                    "sender_fee": "string",
                    "holding_fees": "string"
                },
                "receivers": [
                    {
                        "party": "string",
                        "amount": "string",
                        "receiver_fee": "string"
                    }
                ],
                "balance_changes": [
                    {
                        "party": "string",

"change_to_initial_amount_as_of_round_zero": "string",
                        "change_to_holding_fees_rate": "string"
                    }
                ],
                "description": "string",
                "transferInstructionReceiver": "string",
                "transferInstructionAmount": "string",
                "transferInstructionCid": "string",
                "transfer_kind": "create_transfer_instruction"
            },
            "mint": {
                "amulet_owner": "string",
                "amulet_amount": "string"
            },
            "tap": {
                "amulet_owner": "string",
                "amulet_amount": "string"
            },
            "abort_transfer_instruction": {
                "abort_kind": "withdraw",
                "transfer_instruction_cid": "string"
```

```
                    }
                }
            ]
        }
```

- **404 Not Found** –

  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

- **500 Internal Server Error** –

  internal server error

  Example response:

```
HTTP/1.1 500 Internal Server Error
Content-Type: application/json

{
    "error": "string"
}
```

POST /v0/transactions

Deprecated with known bugs that will not be fixed, use /v2/updates instead.

Lists transactions, by default in ascending order, paged, from ledger begin or optionally starting after a provided event id.

Example request:

```
POST /v0/transactions HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "page_end_event_id": "string",
    "sort_order": "asc",
    "page_size": 1
}
```

Status Codes:
- **200 OK** –

ok

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "transactions": [
        {
            "transaction_type": "transfer",
            "event_id": "string",
            "offset": "string",
            "date": "2026-01-16T12:26:57.979425",
            "domain_id": "string",
            "round": 1,
            "amulet_price": "string",
            "transfer": {
                "provider": "string",
                "sender": {
                    "party": "string",
                    "input_amulet_amount": "string",
                    "input_app_reward_amount": "string",
                    "input_validator_reward_amount": "string",
                    "input_sv_reward_amount": "string",
                    "input_validator_faucet_amount": "string",
                    "sender_change_fee": "string",
                    "sender_change_amount": "string",
                    "sender_fee": "string",
                    "holding_fees": "string"
                },
                "receivers": [
                    {
                        "party": "string",
                        "amount": "string",
                        "receiver_fee": "string"
                    }
                ],
                "balance_changes": [
                    {
                        "party": "string",

"change_to_initial_amount_as_of_round_zero": "string",
                        "change_to_holding_fees_rate": "string"
                    }
                ],
                "description": "string",
                "transferInstructionReceiver": "string",
                "transferInstructionAmount": "string",
                "transferInstructionCid": "string",
                "transfer_kind": "create_transfer_instruction"
            },
            "mint": {
                "amulet_owner": "string",
                "amulet_amount": "string"
            },
            "tap": {
                "amulet_owner": "string",
                "amulet_amount": "string"
            },
            "abort_transfer_instruction": {
                "abort_kind": "withdraw",
                "transfer_instruction_cid": "string"
```

```
                    }
                }
            ]
        }
```

- **400 Bad Request** –

  bad request

  Example response:

  ```
  HTTP/1.1 400 Bad Request
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

- **404 Not Found** –

  not found

  Example response:

  ```
  HTTP/1.1 404 Not Found
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

- **500 Internal Server Error** –

  internal server error

  Example response:

  ```
  HTTP/1.1 500 Internal Server Error
  Content-Type: application/json

  {
      "error": "string"
  }
  ```

POST /v0/updates

Deprecated, use /v2/updates instead. Returns the update history in ascending order, paged, from ledger begin or optionally starting after a record time.

Example request:

```
POST /v0/updates HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "after": {
        "after_migration_id": 1,
        "after_record_time": "string"
    },
    "page_size": 1,
    "lossless": true
}
```

Status Codes:
- 200 OK –
  ok
  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "transactions": [
        {
            "update_id": "string",
            "migration_id": 1,
            "workflow_id": "string",
            "record_time": "string",
            "synchronizer_id": "string",
            "effective_at": "string",
            "offset": "string",
            "root_event_ids": [
                "string"
            ],
            "events_by_id": {}
        },
        {
            "update_id": "string",
            "offset": "string",
            "record_time": "string",
            "event": {
                "submitter": "string",
                "source_synchronizer": "string",
                "target_synchronizer": "string",
                "migration_id": 1,
                "unassign_id": "string",
                "created_event": {
                    "event_type": "string",
                    "event_id": "string",
                    "contract_id": "string",
                    "template_id": "string",
                    "package_name": "string",
                    "create_arguments": {},
                    "created_at": "2026-01-16T12:26:57.979425",
                    "signatories": [
                        "string"
                    ],
                    "observers": [
                        "string"
                    ]
                },
                "reassignment_counter": 1
            }
        }
    ]
}
```

- 400 Bad Request –

bad request

Example response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
    "error": "string"
}
```

- 500 Internal Server Error –

  internal server error

  Example response:

```
HTTP/1.1 500 Internal Server Error
Content-Type: application/json

{
    "error": "string"
}
```

GET /v0/updates/{update_id}

Deprecated, use /v2/updates/{update_id} instead.

Parameters:	• **update_id** (*string*)
Query Parameters:	• **lossless** (*boolean*) – Whether contract payload should be encoded into json using a lossless, but much harder to process, encoding. This is mostly used for backend calls, and is not recommended for external users. Optional and defaults to false.

Example request:

```
GET /v0/updates/{update_id} HTTP/1.1
Host: example.com
```

Status Codes:	• 200 OK – ok **Example response:**

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "update_id": "string",
    "migration_id": 1,
    "workflow_id": "string",
    "record_time": "string",
    "synchronizer_id": "string",
    "effective_at": "string",
    "offset": "string",
    "root_event_ids": [
        "string"
    ],
    "events_by_id": {}
}
```

- **400 Bad Request** –

  bad request

  Example response:

```
HTTP/1.1 400 Bad Request
Content-Type: application/json

{
    "error": "string"
}
```

- **404 Not Found** –

  not found

  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

- **500 Internal Server Error** –

  internal server error

  Example response:

```
HTTP/1.1 500 Internal Server Error
Content-Type: application/json

{
    "error": "string"
}
```

scan

GET /v0/amulet-price/votes

Retrieve a list of the latest amulet price votes

Example request:

```
GET /v0/amulet-price/votes HTTP/1.1
Host: example.com
```

Status Codes:
- 200 OK –
  ok

  Example response:

  ```
  HTTP/1.1 200 OK
  Content-Type: application/json

  {
      "amulet_price_votes": [
          {
              "template_id": "string",
              "contract_id": "string",
              "payload": {},
              "created_event_blob": "string",
              "created_at": "string"
          }
      ]
  }
  ```

POST /v0/backfilling/import-updates

Example request:

```
POST /v0/backfilling/import-updates HTTP/1.1
Host: example.com
Content-Type: application/json

{
    "migration_id": 1,
    "after_update_id": "string",
    "limit": 1
}
```

Status Codes:
- 200 OK –
  ok

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "transactions": [
        {
            "update_id": "string",
            "migration_id": 1,
            "workflow_id": "string",
            "record_time": "string",
            "synchronizer_id": "string",
            "effective_at": "string",
            "offset": "string",
            "root_event_ids": [
                "string"
            ],
            "events_by_id": {}
        },
        {
            "update_id": "string",
            "offset": "string",
            "record_time": "string",
            "event": {
                "submitter": "string",
                "source_synchronizer": "string",
                "target_synchronizer": "string",
                "migration_id": 1,
                "unassign_id": "string",
                "created_event": {
                    "event_type": "string",
                    "event_id": "string",
                    "contract_id": "string",
                    "template_id": "string",
                    "package_name": "string",
                    "create_arguments": {},
                    "created_at": "2026-01-16T12:26:57.979425",
                    "signatories": [
                        "string"
                    ],
                    "observers": [
                        "string"
                    ]
                },
                "reassignment_counter": 1
            }
        }
    ]
}
```

- 404 Not Found –
  not found
  Example response:

```
HTTP/1.1 404 Not Found
Content-Type: application/json

{
    "error": "string"
}
```

common

GET /readyz

Example request:

```
GET /readyz HTTP/1.1
Host: example.com
```

Status Codes:
- **200 OK** – ok
- **503 Service Unavailable** – service_unavailable

GET /livez

Example request:

```
GET /livez HTTP/1.1
Host: example.com
```

Status Codes:
- **200 OK** – ok
- **503 Service Unavailable** – service_unavailable

GET /status

Example request:

```
GET /status HTTP/1.1
Host: example.com
```

Status Codes:
- **200 OK** –
  ok
  Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "success": {
        "id": "string",
        "uptime": "string",
        "ports": {},
        "extra": "01010101",
        "active": true
    }
}
```

GET /version

Example request:

```
GET /version HTTP/1.1
Host: example.com
```

Status Codes:
- **200 OK** –
  ok

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "version": "string",
    "commit_ts": "2026-01-16T12:26:57.979425"
}
```

Comments

An error occurred: giscus is not installed on this repository