Token Auth Method (API)

This is the API documentation for the Vault token auth method. For general information about the usage and operation of the token method, please see the Vault Token method documentation (/docs/auth/token.html).

List Accessors

This endpoint lists token accessor. This requires `sudo` capability, and access to it should be tightly controlled as the accessors can be used to revoke very large numbers of tokens and their associated leases at once.

Method	Path	Produces
`LIST`	`/auth/token/accessors`	`200 application/json`

Sample Request

```
$ curl \
    --header "X-Vault-Token: ..." \
    --request LIST \
    https://vault.rocks/v1/auth/token/accessors
```

Sample Response

```
{
  "auth": null,
  "warnings": null,
  "wrap_info": null,
  "data": {
    "keys": [
      "476ea048-ded5-4d07-eeea-938c6b4e43ec",
      "bb00c093-b7d3-b0e9-69cc-c4d85081165b"
    ]
  },
  "lease_duration": 0,
  "renewable": false,
  "lease_id": ""
}
```

Create Token

Creates a new token. Certain options are only available when called by a root token. If used via the `/auth/token/create-orphan` endpoint, a root token is not required to create an orphan token (otherwise set with the `no_parent` option). If used with a role name in the path, the token will be created against the specified role name; this may override options set during this call.

Method	Path	Produces
`POST`	`/auth/token/create`	`200 application/json`
`POST`	`/auth/token/create-orphan`	`200 application/json`
`POST`	`/auth/token/create/:role_name`	`200 application/json`

Parameters

- `id` `(string: "")` – The ID of the client token. Can only be specified by a root token. Otherwise, the token ID is a randomly generated UUID.

- `role_name` `(string: "")` – The name of the token role.

- `policies` `(array: "")` – A list of policies for the token. This must be a subset of the policies belonging to the token making the request, unless root. If not specified, defaults to all the policies of the calling token.

- `meta` `(map: {})` – A map of string to string valued metadata. This is passed through to the audit devices.

- `no_parent` `(bool: false)` - If true and set by a root caller, the token will not have the parent token of the caller. This creates a token with no parent.

- `no_default_policy` `(bool: false)` - If true the `default` policy will not be contained in this token's policy set.

- `renewable` `(bool: true)` - Set to `false` to disable the ability of the token to be renewed past its initial TTL. Setting the value to `true` will allow the token to be renewable up to the system/mount maximum TTL.

- `lease` `(string: "")` - DEPRECATED; use `ttl` instead

- `ttl` `(string: "")` -The TTL period of the token, provided as "1h", where hour is the largest suffix. If not provided, the token is valid for the default lease TTL (/docs/configuration/index.html), or indefinitely if the root policy is used.

- `explicit_max_ttl` `(string: "")` - If set, the token will have an explicit max TTL set upon it. This maximum token TTL *cannot* be changed later, and unlike with normal tokens, updates to the system/mount max TTL value will have no effect at renewal time -- the token will never be able to be renewed or used past the value set at issue time.

- `display_name` `(string: "token")` - The display name of the token.

- `num_uses` `(integer: 0)` - The maximum uses for the given token. This can be used to create a one-time-token or limited use token. The value of 0 has no limit to the number of uses.

- `period` `(string: "")` - If specified, the token will be periodic; it will have no maximum TTL (unless an "explicit-max-ttl" is also set) but every renewal will use the given period. Requires a root/sudo token to use.

Sample Payload

```
{
  "policies": [
    "web",
    "stage"
  ],
  "metadata": {
    "user": "armon"
  },
  "ttl": "1h",
  "renewable": true
}
```

Sample Request

```
$ curl \
    --header "X-Vault-Token: ..." \
    --request POST \
    --data @payload.json \
    https://vault.rocks/v1/auth/token/create
```

Sample Response

```
{
  "auth": {
    "client_token": "ABCD",
    "policies": [
      "web",
      "stage"
    ],
    "metadata": {
      "user": "armon"
    },
    "lease_duration": 3600,
    "renewable": true,
  }
}
```

Lookup a Token

Returns information about the client token.

Method	Path	Produces
`POST`	`/auth/token/lookup`	`200 application/json`
`GET`	`/auth/token/lookup/:token`	`200 application/json`

Parameters

- `token (string: <required>)` - Token to lookup.

Sample Payload

```
{
  "token": "ClientToken"
}
```

Sample Request

```
$ curl \
    --header "X-Vault-Token: ..." \
    --request POST \
    --data @payload.json \
    https://vault.rocks/v1/auth/token/lookup
```

Sample Response

```
{
  "data": {
    "id": "ClientToken",
    "policies": [
      "web",
      "stage"
     ],
    "path": "auth/github/login",
    "meta": {
      "user": "armon",
      "organization": "hashicorp"
    },
    "display_name": "github-armon",
    "num_uses": 0,
  }
}
```

Lookup a Token (Self)

Returns information about the current client token.

Method	Path	Produces
`GET`	`/auth/token/lookup-self`	`200 application/json`

Sample Request

```
$ curl \
    --header "X-Vault-Token: ..." \
    https://vault.rocks/v1/auth/token/lookup-self
```

Sample Response

```
{
  "data": {
    "id": "ClientToken",
    "policies": [
      "web",
      "stage"
     ],
    "path": "auth/github/login",
    "meta": {
      "user": "armon",
      "organization": "hashicorp"
    },
    "display_name": "github-armon",
    "num_uses": 0,
  }
}
```

Lookup a Token Accessor

Returns information about the client token from the accessor.

Method	Path	Produces
`POST`	`/auth/token/lookup-accessor`	`200 application/json`
`GET`	`/auth/token/lookup-accessor/:accessor`	`200 application/json`

Parameters

- `accessor` `(string: <required>)` - Token accessor to lookup.

Sample Payload

```
{
  "accessor": "2c84f488-2133-4ced-87b0-570f93a76830"
}
```

Sample Request

```
$ curl \
    --header "X-Vault-Token: ..." \
    --request POST \
    --data @payload.json \
    https://vault.rocks/v1/auth/token/lookup-accessor
```

Sample Response

```
{
  "lease_id": "",
  "renewable": false,
  "lease_duration": 0,
  "data": {
    "creation_time": 1457533232,
    "creation_ttl": 2764800,
    "display_name": "token",
    "meta": null,
    "num_uses": 0,
    "orphan": false,
    "path": "auth/token/create",
    "policies": [
      "default",
      "web"
    ],
    "ttl": 2591976
  },
  "warnings": null,
  "auth": null
}
```

Renew a Token

Renews a lease associated with a token. This is used to prevent the expiration of a token, and the automatic revocation of it. Token renewal is possible only if there is a lease associated with it.

Method	Path	Produces
`POST`	`/auth/token/renew`	`200 application/json`
`POST`	`/auth/token/renew/:token`	`200 application/json`

Parameters

- `token` `(string: <required>)` - Token to renew. This can be part of the URL or the body.
- `increment` `(string: "")` - An optional requested lease increment can be provided. This increment may be ignored.

Sample Payload

```
{
  "token": "ClientToken"
}
```

Sample Request

```
$ curl \
    --header "X-Vault-Token: ..." \
    --request POST \
    --data @payload.json \
    https://vault.rocks/v1/auth/token/renew
```

Sample Response

```
{
  "auth": {
    "client_token": "ABCD",
    "policies": [
      "web",
      "stage"
    ],
    "metadata": {
      "user": "armon"
    },
    "lease_duration": 3600,
    "renewable": true,
  }
}
```

Renew a Token (Self)

Renews a lease associated with the calling token. This is used to prevent the expiration of a token, and the automatic revocation of it. Token renewal is possible only if there is a lease associated with it.

Method	Path	Produces
`POST`	`/auth/token/renew-self`	`200 application/json`

Parameters

- `increment` `(string: "")` - An optional requested lease increment can be provided. This increment may be ignored.

Sample Payload

```
{
  "increment": "1h"
}
```

Sample Request

```
$ curl \
    --header "X-Vault-Token: ..." \
    --request POST \
    --data @payload.json \
    https://vault.rocks/v1/auth/token/renew-self
```

Sample Response

```
{
  "auth": {
    "client_token": "ABCD",
    "policies": [
      "web",
      "stage"
    ],
    "metadata": {
      "user": "armon"
    },
    "lease_duration": 3600,
    "renewable": true,
  }
}
```

Revoke a Token

Revokes a token and all child tokens. When the token is revoked, all secrets generated with it are also revoked.

Method	Path	Produces
`POST`	`/auth/token/revoke`	`204 (empty body)`

Parameters

- `token` `(string: <required>)` - Token to revoke.

Sample Payload

```
{
  "token": "ClientToken"
}
```

Sample Request

```
$ curl \
    --header "X-Vault-Token: ..." \
    --request POST \
    --data @payload.json \
    https://vault.rocks/v1/auth/token/revoke
```

Revoke a Token (Self)

Revokes the token used to call it and all child tokens. When the token is revoked, all dynamic secrets generated with it are also revoked.

Method	Path	Produces
`POST`	`/auth/token/revoke-self`	`204 (empty body)`

Sample Request

```
$ curl \
    --header "X-Vault-Token: ..." \
    --request POST \
    https://vault.rocks/v1/auth/token/revoke-self
```

Revoke a Token Accessor

Revoke the token associated with the accessor and all the child tokens. This is meant for purposes where there is no access to token ID but there is need to revoke a token and its children.

Method	Path	Produces
`POST`	`/auth/token/revoke-accessor`	`204 (empty body)`

Parameters

- `accessor` `(string: <required>)` - Accessor of the token.

Sample Payload

```
{
  "accessor": "2c84f488-2133-4ced-87b0-570f93a76830"
}
```

Sample Request

```
$ curl \
    --header "X-Vault-Token: ..." \
    --request POST \
    --data @payload.json \
    https://vault.rocks/v1/auth/token/revoke-accessor
```

Revoke Token and Orphan Children

Revokes a token but not its child tokens. When the token is revoked, all secrets generated with it are also revoked. All child tokens are orphaned, but can be revoked sub-sequently using `/auth/token/revoke/`. This is a root-protected endpoint.

Method	Path	Produces
`POST`	`/auth/token/revoke-orphan`	`204 (empty body)`
`POST`	`/auth/token/revoke-orphan/:token`	`204 (empty body)`

Parameters

- `token` `(string: <required>)` - Token to revoke. This can be part of the URL or the body.

Sample Payload

```
{
  "token": "ClientToken"
}
```

Sample Request

```
$ curl \
    --header "X-Vault-Token: ..." \
    --request POST \
    --data @payload.json \
    https://vault.rocks/v1/auth/token/revoke-orphan
```

Read Token Role

Fetches the named role configuration.

Method	Path	Produces
`GET`	`/auth/token/roles/:role_name`	`200 application/json`

Parameters

- `role_name` `(string: <required>)` - The name of the token role.

Sample Request

```
$ curl \
    --header "X-Vault-Token: ..." \
    https://vault.rocks/v1/auth/token/roles/nomad
```

Sample Response

```
{
  "request_id": "075a19cd-4e56-a3ca-d956-7609819831ec",
  "lease_id": "",
  "lease_duration": 0,
  "renewable": false,
  "data": {
    "allowed_policies": [
      "dev"
    ],
    "disallowed_policies": [],
    "explicit_max_ttl": 0,
    "name": "nomad",
    "orphan": false,
    "path_suffix": "",
    "period": 0,
    "renewable": true
  },
  "warnings": null
}
```

List Token Roles

List available token roles.

Method	Path	Produces
`LIST`	`/auth/token/roles`	`200 application/json`

Sample Request

```
$ curl \
    --header "X-Vault-Token: ..." \
    --request LIST
    https://vault.rocks/v1/auth/token/roles
```

Sample Response

```
{
  "data": {
    "keys": [
      "role1",
      "role2"
    ]
  }
}
```

Create/Update Token Role

Creates (or replaces) the named role. Roles enforce specific behavior when creating tokens that allow token functionality that is otherwise not available or would require `sudo`/root privileges to access. Role parameters, when set, override any provided options to the `create` endpoints. The role name is also included in the token path, allowing all tokens created against a role to be revoked using the `/sys/leases/revoke-prefix` endpoint.

Method	Path	Produces
`POST`	`/auth/token/roles/:role_name`	`204 (empty body)`

Parameters

- `role_name` `(string: <required>)` – The name of the token role.

- `allowed_policies` `(list: [])` – If set, tokens can be created with any subset of the policies in this list, rather than the normal semantics of tokens being a subset of the calling token's policies. The parameter is a comma-delimited string of policy names. If at creation time `no_default_policy` is not set and `"default"` is not contained in `disallowed_policies`, the `"default"` policy will be added to the created token automatically.

- `disallowed_policies` `(list: [])` – If set, successful token creation via this role will require that no policies in the given list are requested. The parameter is a comma-delimited string of policy names. Adding `"default"` to this list will prevent `"default"` from being added automatically to created tokens.

- `orphan` `(bool: false)` - If `true`, tokens created against this policy will be orphan tokens (they will have no parent). As such, they will not be automatically revoked by the revocation of any other token.

- `period` `(string: "")` - If specified, the token will be periodic; it will have no maximum TTL (unless an "explicit-max-ttl" is also set) but every renewal will use the given period. Requires a root/sudo token to use.

- `renewable` `(bool: true)` - Set to `false` to disable the ability of the token to be renewed past its initial TTL. Setting the value to `true` will allow the token to be renewable up to the system/mount maximum TTL.

- `explicit_max_ttl` `(string: "")` - If set, the token will have an explicit max TTL set upon it. This maximum token TTL *cannot* be changed later, and unlike with normal tokens, updates to the system/mount max TTL value will have no effect at renewal time -- the token will never be able to be renewed or used past the value set at issue time.

- `path_suffix` `(string: "")` - If set, tokens created against this role will have the given suffix as part of their path in addition to the role name. This can be useful in certain scenarios, such as keeping the same role name in the future but revoking all tokens created against it before some point in time. The suffix can be changed, allowing new callers to have the new suffix as part of their path, and then tokens with the old suffix can be revoked via `/sys/leases/revoke-prefix`.

Sample Payload

```
  "allowed_policies": [
    "dev"
  ],
  "name": "nomad",
  "orphan": false,
  "renewable": true
```

Sample Request

```
$ curl \
    --header "X-Vault-Token: ..." \
    --request POST
    --data @payload.json
    https://vault.rocks/v1/auth/token/roles/nomad
```

Delete Token Role

This endpoint deletes the named token role.

Method	Path	Produces
`DELETE`	`/auth/token/roles/:role_name`	`204 (empty body)`

Parameters

- `role_name` `(string: <required>)` - The name of the token role.

Sample Request

```
$ curl \
    --header "X-Vault-Token: ..." \
    --request DELETE \
    https://vault.rocks/v1/auth/token/roles/admins
```

Tidy Tokens

Performs some maintenance tasks to clean up invalid entries that may remain in the token store. Generally, running this is not needed unless upgrade notes or support personnel suggest it. This may perform a lot of I/O to the storage method so should be used sparingly.

Method	Path	Produces
POST	/auth/token/tidy	204 (empty body)

Sample Request

```
$ curl \
    --header "X-Vault-Token: ..." \
    --request POST \
    https://vault.rocks/v1/auth/token/tidy
```