BearerAuth (JWT)

Authenticate using OAuth 2.0

Alternatively, OAuth 2.0 can be used together with the Client ID, Client Secret, and Scope. This information is provided during onboarding.

Using the provided credentials, you will be able to generate a JWT access token that needs to be used in all API calls.

Prerequisites for using APIs

To authenticate with the Verifone APIs, you must obtain an access token. This access token is attached to API requests and inspected for a valid signature and expiration time when performing API calls.

How to obtain the authentication credentials

You will be provided with the following details during onboarding:

- Client ID
- Client Secret (associated to the Client ID)
- Scope

Use the following links for each environment:

US Production	https://us.vam.verifone.cloud/oauth2/realms/root/realms/VerifoneServices/access_token
EMEA Production	https://emea.vam.verifone.cloud/oauth2/realms/root/realms/VerifoneServices/access_token
NZ Production	https://nz.vam.verifone.cloud/oauth2/realms/root/realms/VerifoneServices/access_token
AU Production	https://au.vam.verifone.cloud/oauth2/realms/root/realms/VerifoneServices/access_token
Global Sandbox	https://cst1.test-vam.vfims.com/oauth2/realms/root/realms/VerifoneServices/access_token

With this information combination, you can authenticate/authorize and receive the access token.

How to obtain the access token (JWT)

The access token is formatted as a JWT (Json Web Token).

The OAuth2.0 Client Credential grant flow is used to get the access token. Your application will need to have the Client ID and Client Secret stored securely.

Perform the following call to get your access token:

```
curl -k --request POST \
--data "grant_type=client_credentials" \
--data "client_id=${CLIENT_ID}" \
--data "client_secret=$CLIENT_SECRET}" \
--data "scope=${SCOPE}" \
https://cst1.test-vam.vfims.com/oauth2/realms/root/realms/VerifoneServices/access_token
```

Request parameters

- `client_credentials` - indicating that Client Credentials grant is in use
- `scope` - must be at least one custom scope received from Verifone; in case of multiple scope values, they need to be sent as space-separated values within a single string

Response parameters

- `access_token` : Contains the access token in JWT format RFC 7519.
- `expires_in` : Contains the expiration time in seconds for the access token. Once the access token expires, you should send a new request to the authorization endpoint in order to re-authenticate your application.
- `scope` - Contains the list of all the scopes associated with the "access_token".

As per RFC 6749, a refresh token is not provided. If the request failed client authentication or is invalid, the authorization server will return a HTTP 400 (Bad Request) status code.

Example access token

If the credentials are valid, the application will receive back an access token in JSON Web Token (JWT).

Example access token:

```
eyJ0eXAiOiJKV1QiLCJ6aXAiOiJOT05FIiwia2lkIjoid1UzaWZJSWFMT1VBUmVSQi9GRzZlTTFQMVFNPSIsImFsZyI6IlJTMjU2In0
.eyJzdWIiOiI1OWJlYjAzNy1kNjRhLTQyMjgtODM2NC0wZWQ1NDAyMDVmZDUiLCJjdHMiOiJPQVVUSDJfU1RBVEVMRVNTX0dSQU5UIiwiYXVkaXR
UcmFja2luZ0lkIjoiN2NlYzIzZGItNjc1ZC00ZDJmLWFmMmYtZTkxZjQ4ZGY4NzJkLTQ3NDM2IiwiaXNzIjoiaHR0cHM6Ly9xYXQxLnRlc3QtdmF
tLnZmaW1zLmNvbS9vYXV0aDIvcmVhbG1zL3Jvb3QvcmVhbG1zL1Zlcmlmb25lU2VydmljZXMiLCJ0b2tlbk5hbWUiOiJhY2Nlc3NfdG9rZW4iLCJ
0b2tlbl90eXBlIjoiQmVhcmVyIiwiYXV0aEdyYW50SWQiOiJLOFF1SGFxYnpVTnhIVVVRak04d2FaRmF6bjgiLCJhdWQiOiJNYXJrZXRwbGFjZU9
hdXRoIiwibmJmIjoxNTk2OTcwODc1LCJncmFudF90eXBlIjoiY2xpZW50X2NyZWRlbnRpYWxzIiwic2NvcGUiOlsibWFya2V0cGxhY2UiXSwiYXV
0aF90aW1lIjoxNTk2OTcwODc1LCJyZWFsbSI6Ii9WZXJpZm9uZVNlcnZpY2VzIiwiZXhwIjoxNTk2OTcxMDU1LCJpYXQiOjE1OTY5NzA4NzUsImV
4cGlyZXNfaW4iOjE4MCwianRpIjoidmtRZ01kZW03bm1VYTItT1FZeHRKM1dQMC1BIiwiZW50aXR5X2lkIjoiODEwNDlmZDEtNjEyNi00ZDQxLTg
0MTYtYWEzNTZjNDk4Y2NhIiwicm9sZXMiOiJbVkVSSUZPTkVfQURNSU5dIn0
.YW4MeQAMJgpLM3pOFtnx1KXJfSTTHYLklmxld8NozBuA_lQNrO9kN5Sai3MC4roOavcOYbs_TXNP2jXkOE9tNmgGUGHVtiITTIltBAJ2O7s_QCN
u6v1k3tTK0-GfE5gC4I2TavwBN3nFoXmNQYbzcAh3Qetudh5A-
KlsxeepeyxrLfj0BX2NbK5tVEisI1si7yS89aBJvEvCGcKTw5Ujte8naYZCGUJJUIaE2HXk-DGCNdNpmFD9_hnoDWKnN9n3ABuUW4Z6LoAW0J-
BgKpuO6BroweL8fiCWYHV79f_0hCbm7sUqBmcal9j7pc4J2luy0k7TxOexHfry7w2JbpTBQ
```

To inspect the token, you can use Jwt.io or you can use the jwt command line tool.

```
$ cat Token.txt | xargs jwt decode
Token header
------------
```

```
{
"typ": "JWT",
"alg": "RS256",
"kid": "wU3ifI23aqasB/FG6eM1P1QM="
}
Token claims
------------
{
"aud": "VerifoneOauth",
"auditTrackingId": "7cec23db-555-6666-7777-999999999-47436",
"authGrantId": "K8QuHaqbzUJAQWSM8waZFazn8",
"auth_time": 1596970875,
"cts": "OAUTH2_STATELESS_GRANT",
"entity_id": "81049fd1-6126-4d41-8416-aa356c498cca",
"exp": 1596971055,
"expires_in": 180,
"grant_type": "client_credentials",
"iat": 1596970875,
"iss": "https://cst1.test-vam.vfims.com//oauth2/realms/root/realms/VerifoneServices/access_token",
"jti": "vkQgMdem7nmUa2-OQYxtJ3WP0-A",
"nbf": 1596970875,
"realm": "/VerifoneServices",
"roles": "[VERIFONE_TEST]",
"scope": [
"verifoneScope"
],
"sub": "59beb037-d64a-4228-8364-0ed540205fd5",
"tokenName": "access_token",
"token_type": "Bearer"
}
```


Access Token Format

Obtained Access Token is in JWT format [RFC 7519].

Header

#	Claim	Content	Claim Name	Claim type
1	"alg"	RS256	Hashing algorithm (RS256 - RSASSA-PKCS-v1_5 using SHA-256)	Registered
2	"typ"	JWT	The type of the token	Registered
3	"kid"	Key Identifier ("1ee4d9e7dcfef215d133c7ed7ac87c95f8d8e712")	Key ID (which key was used to secure the JWS)	Registered[RFC7515]

Payload

#	Claim	Content	user ID	Claim type

#	Claim	Content	Claim Name	Claim type
1	"sub"	"5f8a9877-965c-4d25-bc86-45d1cfc1c324"	Subject (User UUID)	Registered
2	"entity_id"	"a4994358-a475-4ee2-aefe-acefd622991c"	User associated Entity_id. The Entity ID can be found in Verifone Central under *Administration → Organisations*. The 'Organisation ID' listed is the Entity ID.	Private
3	"iss"	"https://identity.vfims.com/oauth2/realms/root/realms/MerchantApp"	Issuer	Registered
4	"aud"	"Verifone View"	Audience - recipient for which the JWT is intended	Registered
5	"iat"	1516239022	Issued At Time	Registered
6	"exp"	NumericDate value	Expiration Time	Registered
7	"nbf"	1568783970	(Not Before Time) - Time before which the JWT must not be accepted for processing	Registered
8	"roles"	["MERCHANT_REVIEWER", "MERCHANT_DEVELOPER"]	User associated role(s)	Private
9	"jti"	TO6JCVdqS4hJB3_DzVurB3HOe9s	(JWT ID) - Unique identifier; can be used to prevent the JWT from being replayed	Registered
10	"scope"	Merchant Scope	Scopes (limit the API category that can be accessed)	Registered

11	"auditTrackingId"	cbadf943-c28c-450b-bd53-ef11c2b7d80c-17881178	AM correlation to audit trail	Private
12	"auth_level"	0	AM Authentication level	Private
13	"tokenName"	**access_token**	Token description	Private
14	"realm"	"/MerchantApp"	AM authentication realms	Private

Signature

The result of the following computation:

```
JWT_Hash = SHA256(Header + Payload)
JWT_Signature = RS256(JWT_Hash, Private_key)
```

Using the JWT to authenticate in API calls

Once a access token has been obtained, this must be used in all API requests to any of the Verifone APIs.

This can be done by sending the access token as bearer token in the Authorization HTTP header.

```
curl https://gsc.verifone.cloud/oidc/api/v2/transactions
-H "Accept: application/json"
-H "Authorization: Bearer {token}"
```