Discover Stored Token Services

Replace your customer's Primary Account Number (PAN) with a unique, secure token at checkout.

Introduction

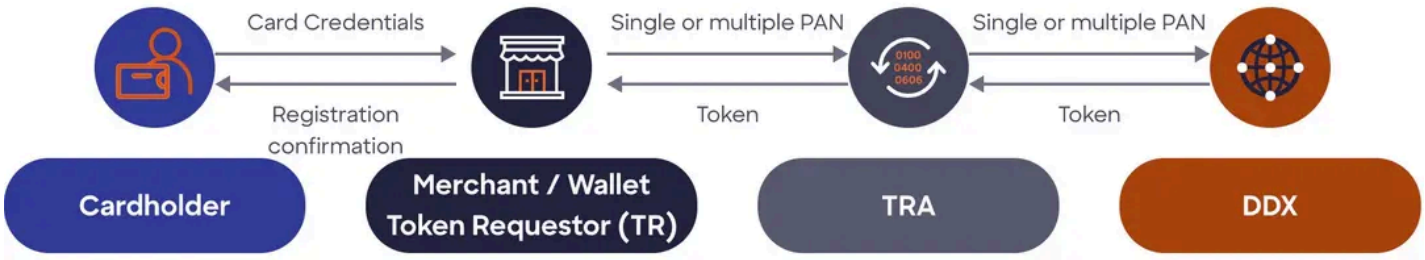
DISCOVER Global Network

Process Flows

Provision Flow

Merchants and Wallets (Token Requestors) can provide PANs directly or via a Token Request Aggregator (TRA), obtaining payment tokens from the Discover Digital Exchange (DDX) platform. A bulk method to provide multiple PANs at the same time is also possible

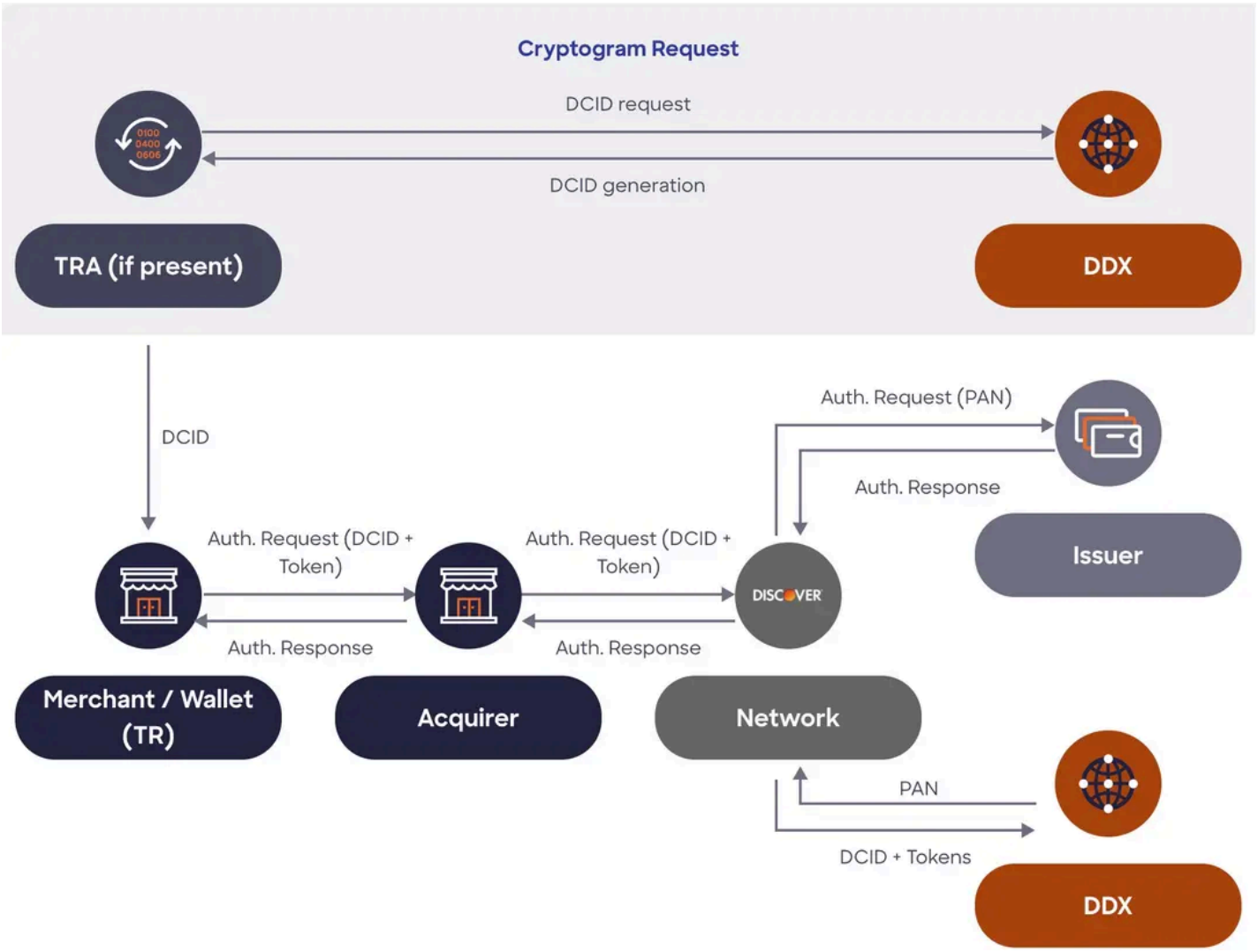
Token provisioning with Token Request Aggregator (TRA)



Token provisioning without Token Request Aggregator (TRA)

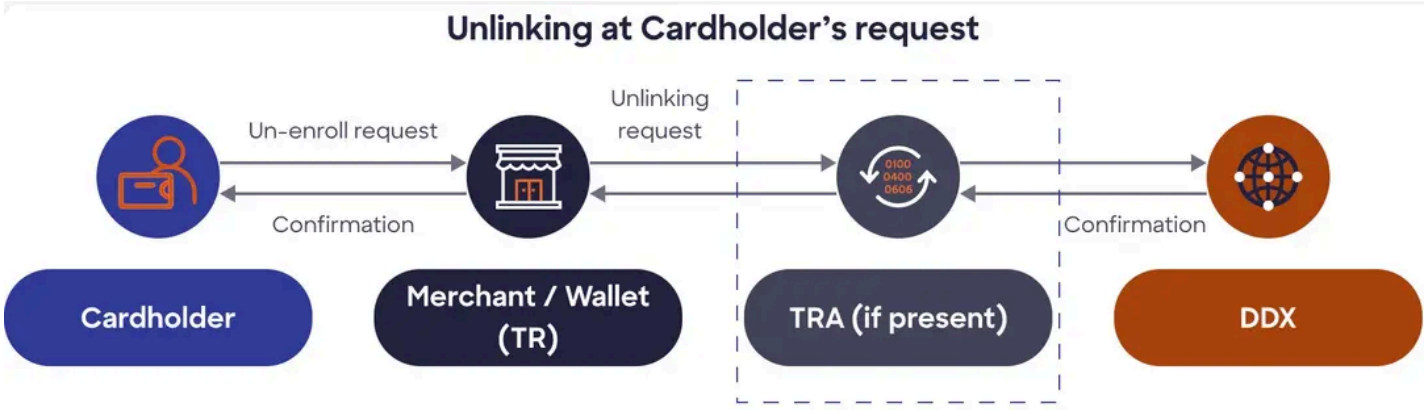
Transaction Processing Flow

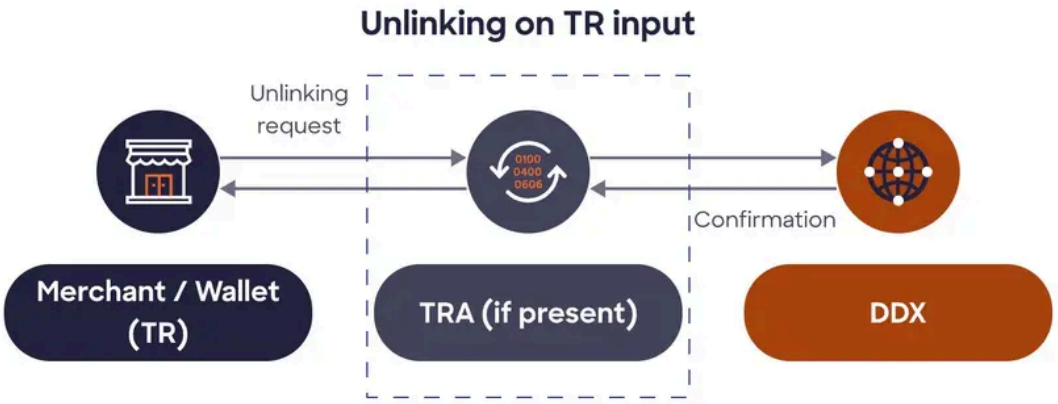
Tokenized payments require a cryptogram, which is generated by DDX at the request of the Token Requestor Aggregator and needs to be included in the Authorization Request



Lifecycle Management Flow

Unlinking requests can come to DDX from Token Requestors as well, either at the Cardholder's request or for fraud event/accounts impairment (suspension and resumption are also possible)

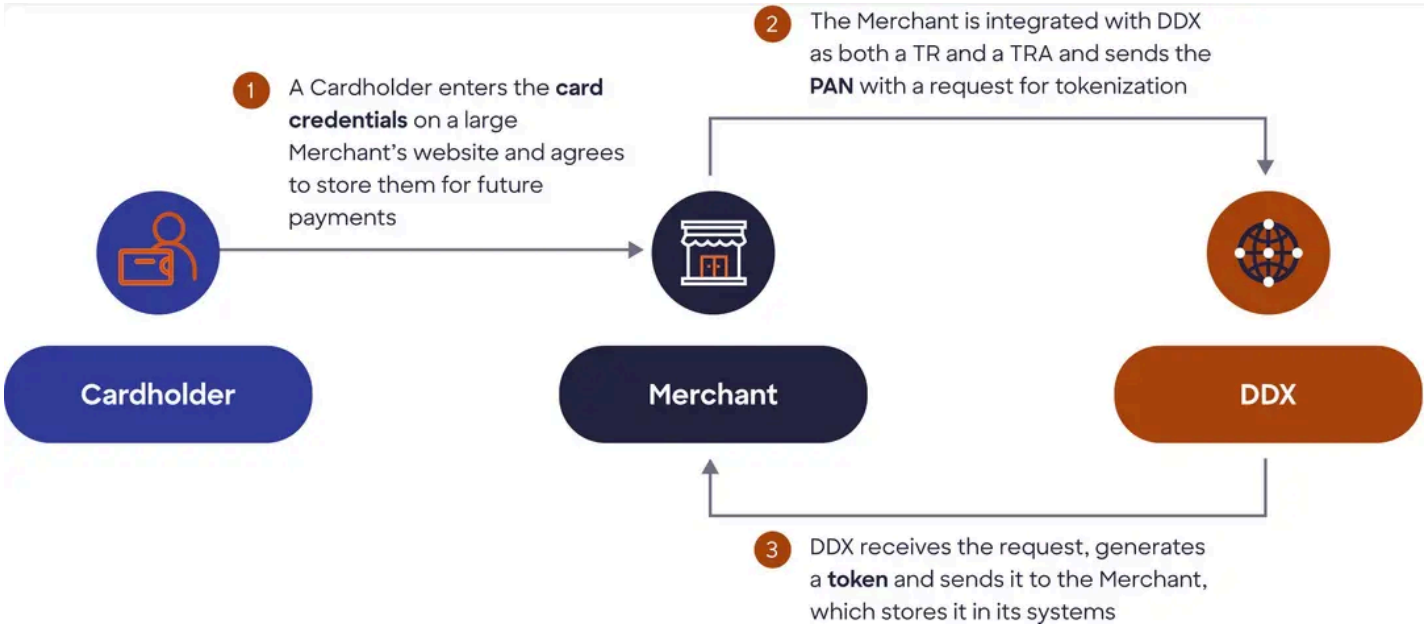




Use Cases

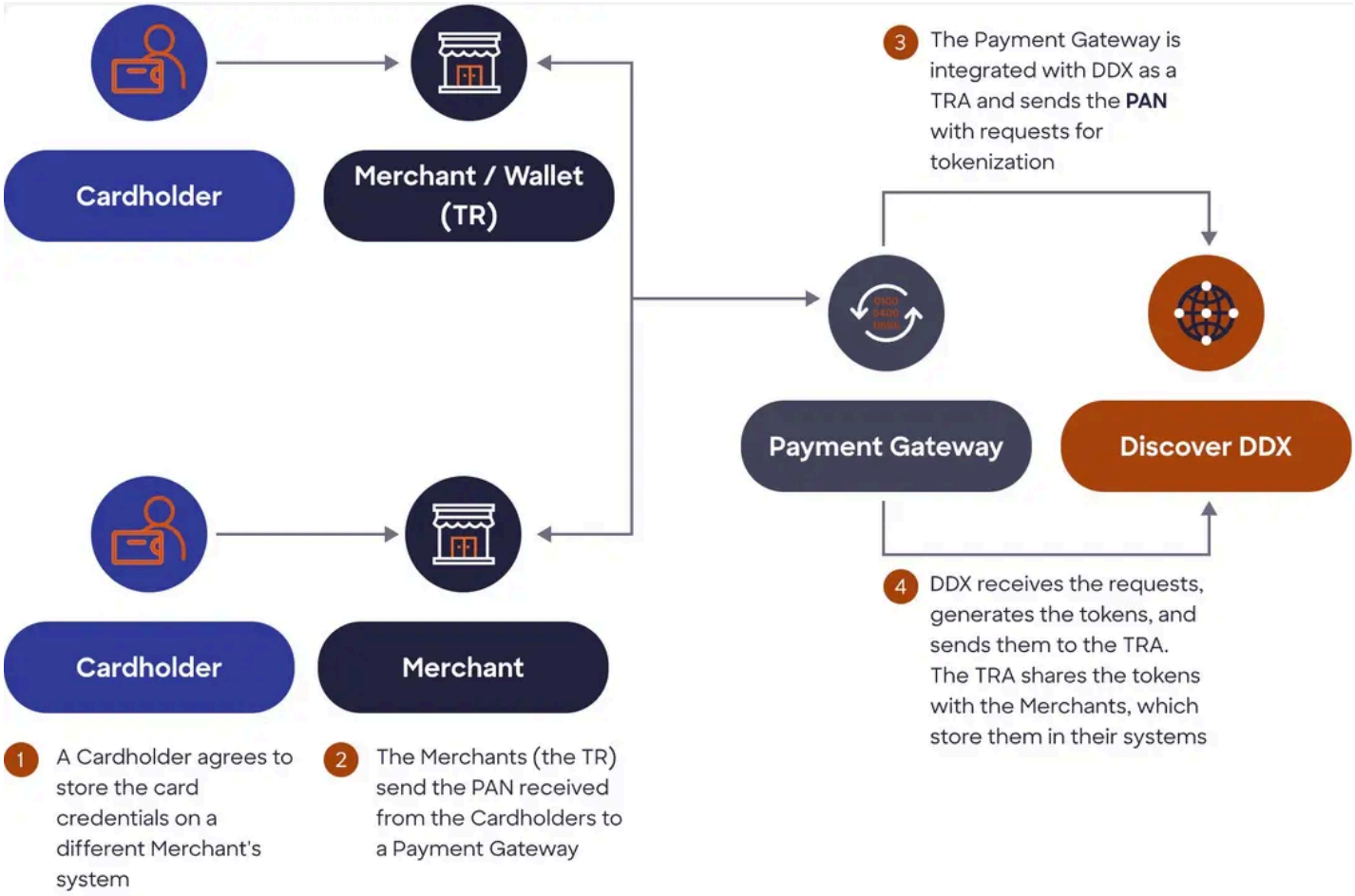
Use Case 1: Large Merchant Tokenizing Customers Card Credentials

Description	A Larger Merchant (Token Requestor) choose to integrate directly with Discover DDX and to implement the role of the Token Requestor Aggregator
Relationship structure	Cardholder - Merchant - DDX
Product category	Stored Payment Tokens/Cloud Wallet



Use Case 2: Third Party Providing Tokenization Services to Multiple Token Requestors

Description	A third-party (e.g., a Payment Gateway) assumes the role of Token Requestor Aggregator and integrates with Discover DDX, acting on behalf of its Merchants and/or Wallets towards Discover.
Relationship structure	Cardholder - Merchant/Wallet - TRA - DDX
Product category	Stored Payment Tokens/Cloud Wallet



Use Case 3: Cloud Wallet Tokenizing Customer's Card Credentials

Description	A Cloud Wallet (Token Requestor) chooses to integrate directly with Discover DDX and assumes the role of Token Requestor.
Relationship structure	Cardholder - Cloud Wallet - DDX
Product category	Cloud Wallet

Getting Started

The Discover Stored Payment Token Services API saves and retrieves payment tokens for recurring billing and online payments using the Discover Digital Exchange (DDX).
Go here to create your DFS Services LLC account.

Partner Registration

To use this API, your organization must first complete the following steps:

Step	Requirement
Project Set-Up	Sign NDA and contract, process exchange certificates & whitelist API endpoint(s)
API Development	System Testing, Integrate API calls and test against DDX systems
Certification Boarding	Onboard to the certification environment, Connectivity test, Receive API test cases, Execute API test cases, Submit test cases to DXX, DDX validates test results & provides certification letter
Production Boarding	Process exchange certificates, Onboard to production environment, Test Production E2E
Production E2E testing	Deploy & validate configurations, Complete connectivity with DDX production environment, Perform successful Production E2E testing
Launch	Prepare launch activity plan, Remove API whitelisting, Go live

Once your registration is approved, your DFS Services LLC API credentials are issued and stored in your account profile. Remember to keep your credentials safe and secure. Never publish online.

Below are credentials assigned to your account:

Name	Description
API Plan	Environment routing and service agreement validation
API Scope	Granular authorization of API resources
Application	TEST environment applications, PROD for production applications
API key	Your API key for access tokens
Client ID	Client ID
Client password	Client password
API secret	Your API secret for access tokens
Consumer Application Certificates	Provides additional authentication
Discover JWS Public Keys	Validate responses with the DFS Services LLC signed JWTs
Discover JWS Public KID	Key Identifier (KID) used to validate responses with DFS Services LLC signed JWTs
Organization	DEV for the test environment applications, PROD for production applications

Partner JWE Public Key	Used by DFS Services LLC to validate your signed data
Partner JWS Public Key	Used by DFS Services LLC to validate your signed data
Partner JWE Public KID	Used by DFS Services LLC to validate your signed data

Supported Entities

Entity	Description
Acquirer	The financial institution that acts as an intermediary between merchants, card payment networks, and the Cardholder bank (issuer).
Cloud wallet provider	A service allowing users to store and manage their digital assets, like cryptocurrencies and NFTs, on a remote server.
Token requestor (TR)	A Merchant or digital wallet that initiates tokenization to replace sensitive payment card data with a surrogate token.
Token requestor aggregator (TRA)	An entity that works with token service providers (TSPs) to request payment tokens on behalf of multiple token requestors.
Payment facilitators (PayFacs)	Act as intermediaries managing payment processing and compliance on behalf of their clients.
Payment service provider (PSP)	Connect businesses to payment networks, including credit card companies, digital wallets, and other payment methods.

Use Cases

Use Case	Description

Tokenize card data	Converts card data into a token to process transactions.
Detokenize card data	Retrieves the original card data from a token.
Update token details	Modifies existing token data such as expiration dates.
Delete token	Deletes a token when it's no longer needed.
Process transaction	Initiates a token to authorize and process payments.

Authentication

Access Control

The following authentication methods are required for access control, payload integrity, and non-repudiation of API requests:

Method	Description
OAuth 2 client credentials	Used to get an access token
Consumer Application Certificate	The X-DFS-C-APP-CERT is used to confirm compliance
Second factor JWT	The X-DFS-C-APP-JWT is used with two factor authentication (2FA).
Mutual TLS authentication (mTLS)	An authentication standard used by two parties to authenticate each other using a crypto-graphic transport layer security (TLS) protocol.

Requests & Responses

For payload and field encryption for requests and responses the following are required:

- JWE payload encryption
- JWS+JWE payload signature and encryption
- API-specific field-level encryption mechanisms

> **Note:** You must set your TLS version to 1.2 or 1.3 to connect to the APIs.

Keys

The required API security methods needed to access this API rely on public key cryptography and require an exchange of certificates between DFS Services LLC and your organization.

> **Note:** No public key certificates are used.

Public Keys

DFS Services LLC public keys are available at the following JSON Web Key Set (JWKS) endpoint:
`https://apis.discover.com/dfs/jwk/v1/public-keys`

- The JWKS endpoint keys expire every 90 days
- Keys are used for encryption and signing
- Your `client_id` (API key located in your dashboard) must be used in the request

JWKS Endpoint

```
https://apis.discover.com/dfs/certs/v1/jwks.json?client_id=<your_client_
```

Example Request


```
{
  "keys": [
    {
      "alg": "RS256",
      "e": "AQAB",
      "kid": "AQA8",
      "kty": "RSA",
      "use": "sig",
      "n": "<modulus of the RSA public key in the DISCOVER_SIGNATURE_CERT
      "x5c": "<base64-encoding of DISCOVER_SIGNATURE_CERT used with Partne
      "x5t#S256": "<SHA-256 thumbprint of DISCOVER-SIGNATURE-CERT used wit
    },
    {
      "alg": "RS256",
      "e": "AQAB",
```

```
      "kid": "AQA9",
      "kty": "RSA",
      "use": "enc",
      "n": "<modulus of the RSA public key in the DISCOVER_ENCRYPTION_CERT
      "x5c": "<base64-encoding of DISCOVER_ENCRYPTION_CERT used with Partn
    }
  ],
  "x5t#S256": "<SHA-256 thumbprint of DISCOVER_ENCRYPTION_CERT used with P
}
```

Parameter Definitions

Parameter	Description
alg	The cryptographic algorithm used with the key
e	The RSA public exponent
kid	A unique identifier for the key used to match keys during signature verification and encryption
kty	The key type
n	The RSA modulus of the public key used to verify signatures and encrypted data
use	The intended use of the key (**sig** = digital signature | enc = encryption)
x5c	A base64-encoded array of certificates
x5t#S256	A SHA-256 thumbprint of the X.509 certificate used for certificate verification

Certificates

Consumer Application Certificate

We use Consumer Application Certificates. They're issued to your organization as a second authentication factor.

1. Download your Consumer Application Certificate from your profile dashboard.
2. The certificate must be in the X-DFS-C-APP-CERT API request header.
3. You can only download the Consumer Application Certificate once.
4. Additional certificates must be requested using your account portal.

Mutual TLS

Mutual TLS authentication (mTLS) uses a separate set of X.509 certificates. For payload and field specific encryption for requests and responses, the JWE payload encryption relies on these certificates:

- `PARTNER_ENCRYPTION_CERT`
- `DISCOVER_ENCRYPTION_CERT`

The JWS and JWE payload signature and encryption uses all four of the following certificates:

- `PARTNER_SIGNATURE_CERT`
- `DISCOVER_SIGNATURE_CERT`
- `PARTNER_ENCRYPTION_CERT`
- `DISCOVER_ENCRYPTION_CERT`

Your organization is required to supply certificates per environment and per API.

Partner Certificate for Signing

The partner certificate for signing includes the public key from the public-private key pair generated by your organization for creating signatures of outgoing messages. The certificate is referred to as `PARTNER_SIGNATURE_CERT`. It must use the following properties:

- Certificate Type: RSA 2048
- Format: PKCS#12
- Signed by a 3rd-party public Certificate Authority (CA)
- The certificate must include both root and intermediate certificates
- It must include the Subject Alternative Name (SAN) extension with the DNS name

The unique `kid` key pair used for signing is calculated as BASE64URL and SHA-256 public key bytes using the SHA algorithm.

Steps

1. Upload your partner certificate into your account portal using the x.509 format to make it available on the JWKS endpoint.
2. Or send your partner certificate to us using secure email.

Certificate for Encryption

The `PARTNER_ENCRYPTION_CERT` includes the public key from the public-private key pair generated by your organization for creating signatures of outgoing API messages. It must conform to the following properties:

- Certificate Type: RSA 2048
- Format: PKCS#12
- Signed by a 3rd-party public CA
- The certificate must include both root and intermediate certificates
- It must include the Subject Alternative Name (SAN) extension with the DNS name

Discover Certificate for Signing

This certificate includes the public key from the public-private key pair generated by our APIs for creating signatures of outgoing messages.

- The certificate is referred to as `DISCOVER_SIGNATURE_CERT`
- Each key pair must have a `kid` formatted as the Universally Unique Identifier (UUID)
- JWKS endpoint: `https://apis.discover.com/dfs/jwk/v1/public-keys`

Consumer Application Certificate

The API uses a Consumer Application Certificate which is available to your organization as a second authentication factor to help ensure a request's authenticity.

1. The certificate must be included in the `X-DFS-C-APP-CERT API` request header.
2. Download your Consumer Application Certificate from your Discover dashboard.
3. It can only be downloaded once. Contact us to get another one using your account portal.

Register Partner Certificates

The registration endpoint is protected using HTTP basic access authentication.

Endpoint

```
https://apis.discover.com/dfs/jwk/v1/public-keys/partner/registration
```

Register your certificates using one of the following two methods:

Method	Description
Preferred	Register your partner JWKS endpoint URL in your account portal for us to obtain your organization's public key certificates.
Alternative	Push your partner certificate value and its **`kid`** to us.

OAuth 2.0 Token Request

You need the following to use OAuth 2 to request an access token to authenticate your application:

- `client_id`
- `client_secret`
- `token_endpoint`

Make a POST request to the token endpoint:
`https://apis.discover.com/auth/OAuth 2/v2/token`

The HTTP POST request must include the following in the body:

```
curl --request POST "https://apis.discover.com/auth/OAuth 2/v2/token" \
    --header "Content-Type: application/x-www-form-urlencoded" \
    --header "Authorization: Basic $(echo -n 'client_id:client_secret' | b
    --header "X-DFS-API-PLAN: Discover-API-Plan-Name" \
    --data "grant_type=client_credentials&scope=your_scope"
```

A successful response returns the bearer token:

```
{
  "access_token": "XXXX",
  "token_type": "Bearer",
  "expires_in": 3600,
  "scope": "your_scope"
}
```

The JSON payload contains the access token which is used by the application to access the API. You can use a single OAuth 2 token or multiple OAuth 2 tokens to facilitate multiple consumer application instances.

> **Note:** By default, the token expires in one hour (3600 seconds). When the access token expires, make another token request.

Second-Factor JWT

We use a second-factor JSON Web Token (JWT) called a JSON Web Signature (JWS) as an additional layer of authentication.

- The JWS token is in the `X-DFS-C-APP-JWT` request header
- The response contains `X-DFS-C-APP-JWT` in the header
- The JWS token contains a header, claims, and a signature, each concatenated together

JWS Elements

Element	Description
Header	The header contains the algo and typ parameters. The algo parameter value is used to sign the token and typ identifies the token (JWS).
Claims (payload)	Claims data is passed to DFS Services LLC for authentication. Claims are a hash of the body content (encrypted) and the metadata.
Signature	The signature is generated by signing the encoded header and payload with your private JWS (JWT Signing) key. This confirms the integrity and authenticity of the token.

Note: You must have the `PARTNER_SIGNATURE_CERT` created and the public key uploaded to the DFS Services LLC endpoint. Or, make your `PARTNER_SIGNATURE_CERT` available on the JWKS endpoint so the signature can be verified by our team

Building the JWS Token

The client application generates a JWT token by using its own signing private key with the JWT algorithm RS256. The JWT is composed of three base64url-encoded JSON structures that are separated by a period. The following sections are part of the token:

1. JWT header (protected) is a JSON object:

```
{
"typ": "JWT",
```

```
"alg": "RS256",
"kid": "xxx"
}
```

Field	Type	Description
typ	String	An optional type defining the claims object (default is JWT)
alg	String	The algorithm used to sign the JWT claims
kid	String	Key identifier used to sign the PARTNER_SIGNATURE_CERT.

2. JWT claims (payload) is a JSON object:

```
{
"typ": "JWT",
"alg": "RS256",
"kid": "xxx"
}
```

Field	Type	Description
typ	String	An optional type defining the claims object (default is JWT)
jti	String	The **requestBody.requestId** (preferred) or a random UUID
iat	String	The time the JWT was issued
exp	String	The token expiration in seconds (choose a low value: 300 seconds or less)
content_hash	String	SHA-256 digest of request body. Body value can be encrypted (optional)

3. The JWT signature is the cryptographic value which is added to the JWT header. It's comprised of the base64URL safe-encoded variant and concatenated together with a

period (`.`) in the JWT header and claims. It uses the JWT signature algorithm RS256 with the partner JWT signing private key.

Below is the final token structure:

```
<base64URLSafeEncoded(UTF-8(header))>
.
<base64URLSafeEncoded(payload)>
.
<base64URLSafeEncoded(signature)>
```

Using the JWS Token

Add the token to the HTTP request header:

```
X-DFS-C-APP-JWT: XXXXX.YYYYYY.ZZZZZZ
```

A successful response returns the 200 HTTP status code. The JWT token is signed by a Discover gateway signing private key and its corresponding public key and certificate. The `DISCOVER_SIGNATURE_CERT` is provided at this JWKS endpoint:

```
https://apis.discover.com/dfs/jwk/v1/public-keys
```

Payload Signature & Encryption

The body of the HTTP request and response may contain payload data which may require encryption. When that is the case, an API will require the implementation of either a JWE or nested JWT to encrypt the payload data.

> **Note:** each API may have different variations of payload encryption.

JSON Web Encryption (JWE)

Field	Description
Header	How the token is signed
Encrypted key	The key used to encrypt the payload containing the recipient's public key
Initialization Vector (IV)	A random value used to confirm the encryption is unique

Encrypted payload	The encrypted payload
Authentication tag	Ensures the integrity and authenticity of the encrypted message

Here is the header example:

```
{
  "alg": "RSA-OAEP-256",
  "enc": "A256GCM",
  "kid": "xxx",
  "typ": "JWE"
}
```

Field	Type	Description
alg	String	An algorithm for asymmetric (public key) encryption
enc	String	The encryption algorithm used to secure the payload of a JWT with a 256 bit key
kid	String	The key ID used to encrypt **DISCOVER_ENCRYPTION_CERT** and the **PARTNER_ENCRYPTION_CERT**
typ	String	The type of JWE (optional)

Below is the structure of JWE encryption following the header. Each of these fields are base64URL-encoded and concatenated together with a period. The JWE is then added as the field of the body.

```
<base64URLSafeEncoded(UTF-8(header))>
.
<base64URLSafeEncoded(encrypted_key)>
.
<base64URLSafeEncoded(iv)>
.
<base64URLSafeEncoded(ciphertext)>
.
<base64URLSafeEncoded(tag)>
Encrypted Key ('encrypted_key'):
```

Field	Type	Description
encrypted_key	Byte sequence	Generates a random key according to the value of enc in the JWE header
iv	Byte sequence	The initialization vector generates a random 16-byte initialization vector
ciphertext	Byte sequence	Performs authenticated encryption on the plaintext using the algorithm represented by enc from the JWE header. The Content Encryption Key (CEK) is used as the encryption key. The Content Encryption IV (initialization vector) and the Additional Authenticated Data (AAD)value requests a 128-bit authentication tag output
tag		The value used to confirm the integrity of the encrypted payload

Nested JWT (JWS in JWE)

The nested JWT is composed of a JWS token of the payload that is then nested inside a JWE token. You must create a JWS with a header and payload (JSON formatted). Then sign it with a private key to ensure the integrity and authenticity of the payload.
The JWS header (protected) is in the form of a JSON object.

Header example:

```
{
  "typ": "JWT",
  "alg": "RS256",
  "kid": "xxx"
}
```

Field	Type	Description
typ	String	An optional type defining the JWT. Default is JWT

alg	String	The algorithm used to sign the JWT
kid	String	The key ID used sign **PARTNER_SIGNATURE_CERT** (partner to DFS Services LLC) and the **DISCOVER_SIGNATURE_CERT** (DFS Services LLC to partner)

JWS Final Structure

```
<base64URLSafeEncoded(UTF-8(header))>
.
<base64URLSafeEncoded(payload)>
.
<base64URLSafeEncoded(signature)>
```

DFS Services LLC generates a unique key pair for JWS.

The Nested JWT

The field, or payload that needs to be sent must be encrypted using the JWE method. You must sign the JSON payload. Use the signed payload and apply the JWE encryption method to it.

Account Notification API

The Account Notification API connects DFS Services LLC with the token requestor by transmitting issuer-initiated updates applied to Cardholder accounts.

> **Note:** Token requestors must integrate with this endpoint to receive issuer notifications via webhooks.

Account Profile

POST

Updates information about the card issuer.

Endpoint

```
/globalpymt/ddx/stored-payment-token/aggregator/v1/issuer
```

Example Request


```
{
  "taskId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "requestId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "addressContext": {
    "zip": "60115",
    "city": "Riverwoods",
    "line1": "Discover Financial Services",
    "line2": "Attn: Marketing Department",
    "line3": "2500 Lake Cook Road",
    "state": "Illinois",
    "countryCode": "US",
    "setAsShippingAddress": "true"
  },
  "profileUpdates": [
    {
      "reason": "Suspended Token",
      "issuerContext": {
        "email": "jondaun@discover.com",
        "website": "https://www.discover.com",
        "issuerName": "Discover Network",
        "contactNumber": "800-867-5309",
        "addressContext": {
          "zip": "60115",
          "city": "Riverwoods",
          "line1": "Discover Financial Services",
          "line2": "Attn: Marketing Department",
          "line3": "2500 Lake Cook Road",
          "state": "Illinois",
          "countryCode": "US",
          "setAsShippingAddress": "false"
        },
        "privacyPolicyURL": "https://www.discover.com/privacypolicy.html",
        "termsConditionsURL": "https://www.discover.com/privacypolicy.html
      },
      "tokenReferenceId": "df106a60805440dcdf1f864c24060",
      "accountMetadataContext": {
        "cardType": "Credit",
        "panSuffix": "0289",
        "labelColor": "#FFFFFF",
        "cardImageId": "df106a670805440d8cdf1f8647c24060",
        "networkLogoId": "logo123",
        "backgroundColor": "#FFFFFF",
        "foregroundColor": "#FFFFFF",
        "productDescription": "Discover It"
      },
```

```
      "encryptedAccountContext": "==VGhpcyBpcyBhbiBleGFtcGxlHZhbHVIIHdoaW
    }
  ],
  "tokenRequestorId": "62345678910",
  "tokenRequestorPartyId": "66221"
}
```


Example Response


```
{
  "responseId": "26e9d8e2-c483-4e96-8e4a-0123456789ab",
  "taskId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "requestId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "status": "COMPLETED",
  "statusDateTime": "2025-04-25T15:00:00Z",
  "profileUpdateResults": [
    {
      "tokenReferenceId": "df106a60805440dcdf1f864c24060",
      "status": "SUCCESS",
      "message": "Token profile successfully updated."
    }
  ],
  "tokenRequestorId": "62345678910",
  "tokenRequestorPartyId": "66221"
}
```


Parameter Definitions

Parameter	Description
accountMetadataContext	The account metadata such as card type, PAN, and suffix.
addressContext	The issuer address and shipping address settings.
encryptedAccountContext	A base64 or binary-encoded string containing encrypted account-level metadata such as Cardholder account information and verification details.
panSuffix	The last four digits of the PAN.

profileUpdates	An array of the updates to the token profiles including issuer and account metadata.
profileUpdateResults	An array of results of each token profile update attempt.
requestId	A client-generated UUID to trace and correlate the request and response.
responseId	The identifier of the response.
status	The status result of the response.
statusDateTime	The timestamp when the status was created.
taskId	The identifier representing the batch or the group of the request.
tokenRequestorId	Numeric ID assigned to the token requestor by the network.
tokenRequestorPartyId	An ID of the business entity or legal party of the token requestor.

Tokenized Transaction

POST

Updates transaction activities that use a token.

Endpoint

```
/globalpymt/ddx/stored-payment-token/aggregator/v1/transaction
```

Example Request

```
{
  "requestId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "encryptedContext": "base64.encoded.jwe.encrypted.content",
```

```
  "tokenRequestorId": "60000014610",
  "tokenRequestorPartyId": "123456"
}
```

Example Response

```
{
  "responseId": "cb0f2785-9b2b-4e9d-8e61-1234567890ab",
  "requestId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "status": "COMPLETED",
  "statusDateTime": "2025-04-25T15:30:00Z",
  "decryptedContext": {
    "tokenReferenceId": "TOKEN-1234567890ABCDE",
    "tokenStatus": "ACTIVE",
    "tokenType": "CL",
    "tokenRequestorId": "60000014610",
    "tokenRequestorPartyId": "123456",
    "accountLastFour": "1111",
    "expirationMonth": "12",
    "expirationYear": "2027"
  }
}
```

Parameter Definitions

Parameter	Description
accountMetadataContext	The account metadata such as card type, PAN, and suffix.
accountLastFour	The last four digits of the PAN.
decryptedContext	The decrypted payload containing the token and account details.
encryptedContext	The Base64-encoded JWE string containing the tokenized data.
expirationMonth	The expiration month of the account.
expirationYear	The expiration year of the account.

requestId	The expiration year of the account.
sdgsdg	Client-generated UUID to trace and correlate the request and response.
responseId	The identifier of the response.
status	The result of the response.
statusDateTime	The timestamp when the task reached the reported status.
tokenReferenceId	The identifier assigned to the stored payment token.
tokenRequestorId	The ID assigned to the token requestor by the network.
tokenRequestorPartyId	An ID of the business entity or legal party of the token requestor.
tokenStatus	The resulting token status after operation.
tokenType	The type of token issued.

Lifecycle

POST

Updates the token lifecycle status.

Endpoint

```
/globalpymt/ddx/stored-payment-token/aggregator/v1/tokens/state
```

Example Request

```
{
  "reason": "User requested unlinked due to lost device",
  "taskId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "tokens": [
    {
      "tokenReferenceId": "tokenCH8u9dUvWSMqQs0XWzOEo4kQ9I78CWtx"
    }
  ],
  "requestId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "operationType": "Unlink",
  "tokenRequestorId": "62345678910",
  "tokenRequestorPartyId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
}
```


Example Response


```
{
  "responseId": "4bcd9386-2d6a-45e2-9f33-1234567890ab",
  "taskId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "requestId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "status": "COMPLETED",
  "statusDateTime": "2025-04-25T16:00:00Z",
  "unlinkResults": [
    {
      "tokenReferenceId": "tokenCH8u9dUvWSMqQs0XWzOEo4kQ9I78CWtx",
      "status": "SUCCESS",
      "message": "Token successfully unlinked from device."
    }
  ],
  "operationType": "Unlink",
  "tokenRequestorId": "62345678910",
  "tokenRequestorPartyId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
}
```


Parameter Definitions

Parameter	Description
message	The result message of the unlinking operation.
operationType	The type of operation being performed.

reason	Explanation why the unlinking operation is requested.
responseId	The identifier of the response.
status	The status of the unlinking operation.
statusDateTime	The timestamp when the unlinking operation was updated.
taskId	The identifier to track the unlinking operation.
tokens	A list containing token object to be unlinked.
tokenReferenceId	The identifier of the token being unlinked.
requestId	The client-generated UUID to trace and correlate the request and response.
tokenRequestorId	The numeric ID assigned to the token requestor by the network.
tokenRequestorPartyId	The ID of the business entity or legal party of the token requestor.
unlinkResults	Array containing the result details.

Boarding API

The Boarding service API automates token requestor onboarding and profile management for the token requestor and the Token Requestor Aggregator (TRA) accounts associated with the DDX platform. These APIs process payment tokenization for an efficient token management experience.

Token Requestor Boarding

Update Token Requestor

PUT

Updates the token requestor information for the given token requestors.

Endpoint

Example Request

```
{
  "requestId": "RequestId1234",
  "featureFlags": {
    "transactionNotificationsEnabled": true
  },
  "tokenRequestorId": 62345678910,
  "tokenRequestorKYCInfo": {
    "networkMid": "1234",
    "merchantUrl": "www.xyzlimited.com",
    "merchantName": "Merchant Name",
    "merchantAddress": {
      "zip": "60015",
      "city": "Riverwoods",
      "line1": "2500 Lake Cook Road",
      "line2": "Business Technology Department",
      "line3": "ATTN: CIO",
      "state": "Illinois",
      "countryCode": "US"
    },
    "merchantPhoneNumber": {
      "number": "8003472683",
      "phoneCountryCode": "1"
    },
    "merchantCategoryCode": [742]
  },
  "tokenRequestorPartyId": "123456",
  "profileEffectiveEndDate": "2025-04-21T15:07:00.786Z",
  "profileEffectiveStartDate": "2025-04-21T15:07:00.786Z"
}
```

Example Response

```
{
  "tokenType": "01",
  "responseId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "tokenRequestorId": 62345678910,
  "tokenRequestorName": "XYZ Limited"
}
```


Parameter Definitions

Parameter	Description
featureFlags	The Object containing feature toggles or flags related to the requestor's preference.
transactionNotificationsEnabled	A boolean flag indicating the status of the transaction notification for the token requestor.
merchantAddress	The object containing the token requestor Merchant business address.
merchantCategoryCode	hghfh
sgh	The code of the Merchant business type.
merchantName	The name of the Merchant.
merchantPhoneNumber	The Merchant phone number.
merchantUrl	The URL of the Merchant.
NetworkMid	The network Merchant identifier assigned by the payment network.
profileEffectiveEndDate	The date/time when the token requestor profile expires.

requestId	A client-generated UUID to trace and correlate the request and response.
responseId	The identifier of the response.
tokenRequestorId	A numeric ID assigned to the token requestor by the network.
tokenRequestorKYCInfo	An object containing the Know Your Customer (KYC) details.
tokenRequestorName	The registered name of the token requestor.
tokenRequestorPartyId	An ID of the business entity or legal party of the token requestor.
tokenType	The type of token issued.
sgh	hghfh

Board Token Requestors

POST

Boards one-to-many new token requestors under the given token aggregator.

Endpoint

Example Request

```
{
    "requestId": "wLLc39bD3acR",
    "tokenRequestorInfo": [
        {
```

```
            "tokenType": "CL",
            "featureFlags": {
                "transactionNotificationsEnabled": true
            },
            "recordLevelId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
            "tokenLocation": "01",
            "drcCategoryCode": "COF_RUL1",
            "tokenRequestorName": "Target",
            "tokenRequestorType": "ACQUIRER",
            "tokenRequestorKYCInfo": {
                "networkMid": "1234",
                "merchantUrl": "www.discover.com",
                "merchantAddress": {
                    "zip": "60015",
                    "city": "Riverwoods",
                    "line1": "2500 Lake Cook Road",
                    "line2": "Business Technology Department",
                    "line3": "ATTN: CIO",
                    "state": "Illinois",
                    "countryCode": "US"
                },
                "merchantPhoneNumber": {
                    "number": "8003472683",
                    "phoneCountryCode": "1"
                },
                "merchantCategoryCode": [742]
            },
            "parentTokenRequestorId": 99999999999,
            "profileEffectiveEndDate": "1861-98-11T34:75:14?575Z",
            "profileEffectiveStartDate": "8501-07-53T68:79:54%445Z"
        }
    ],
    "tokenRequestorPartyId": "123456"
}
```


Response


```
{
    "responseId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "tokenRequestorDataList": [
        {
            "tokenType": "string",
            "recordLevelId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
            "tokenRequestorId": 62345678910,
            "tokenRequestorName": "XYZ Limited",
            "assignedDRCCategoryCode": "string"
        }
    ],
    "tokenRequestorErrorList": [
```

```
        {
            "errorCode": "30001",
            "recordLevelId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
            "errorDescription": "Error Occurred During Processing - Contac
            "tokenRequestorName": "Target"
        }
    ]
}
```


Parameter Definitions

Parameter	Description
assignedDRCCategoryCode	Domain Restricted Code (DRC) category code assigned to the token requestor during processing.
drcCategoryCode	The DRC classification of the token requestor.
errorCode	The response error code.
errorDescription	The human-readable cause of the error.
featureFlags	The object holding optional toggles to control requestor features.
merchantAddress	The object containing the token requestor Merchant business address.
merchantPhoneNumber	The Merchant phone number.
merchantUrl	The URL of the Merchant.
parentTokenRequestorId	Identifier for the parent token requestor entity.
profileEffectiveEndDate	The date/time when the token requestor profile expires.

recordLevelId	The identifier of the record used for tracking individual entities.
requestId	A client-generated UUID to trace and correlate the request and response.
responseId	The identifier of the response.
tokenLocation	The environment of the token (i.e. 01 = e-Commerce).
tokenRequestorDataList	A list of token requestor data entries in the response.
tokenRequestorErrorList	A list of error objects of the failed token requestor operations.
tokenRequestorId	A numeric ID assigned to the token requestor by the network.
tokenRequestorInfo	The array of the token requestor configurations submitted in the request.
tokenRequestorKYCInfo	An object containing the Know Your Customer (KYC) details.
tokenRequestorName	The registered name of the token requestor.
tokenRequestorPartyId	An ID of the business entity or legal party of the token requestor.
tokenRequestorType	The classification of the token requestor (i.e. issuer, acquirer).
tokenType	The type of token issued.

Token Requestor Lookup

GET

Provides functions to lookup token requestor details and account hierarchy information. Used by the token requestors, aggregators, and payment facilitators to retrieve the hierarchy information of the tokenRequestorId.

Endpoint

Example Request

Example Response


```
{
  "pagination": {
    "limit": 2147483647,
    "maxOffset": 2147483647,
    "nextOffset": 2147483647,
    "totalRecords": 9223372036854776000,
    "currentOffset": 2147483647,
    "previousOffset": 2147483647
  },
  "responseId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "tokenRequestorId": 69999999999,
  "childTokenRequestors": [
    {
      "profileActive": true,
      "tokenRequestorId": 69999999999,
      "tokenRequestorName": "Al's Toybarn",
      "parentTokenRequestorId": 69999999999
    }
  ],
  "tokenRequestorPartyId": "123456"
}
```


Parameter Definitions

Parameter	Description
childTokenRequestors	A list of child token requestors.

`client_id`	Your API key issued by DFS Services LLC.
`client_secret`	Your client secret issued by DFS Services LLC.
`networkMid`	The Merchant ID assigned by DFS Services LLC.
`pagination`	Meta data for paging through results.
`requestId`	A client-generated UUID to trace and correlate the request and response.
`responseId`	The identifier of the response.
`taskId`	Optional tracking ID for batch/task operations.
`tokenRequestorId`	A numeric ID assigned to the token requestor by the network.
`tokenRequestorPartyId`	An ID of the business entity or legal party of the token requestor.

Retrieve Token Requestor Details

GET

Retrieves detailed information of onboarded token requestors.

Endpoint

Example Request

Example Response

```
{
  "tokenType": "CL",
  "responseId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "featureFlags": {
    "futureFeature": false,
    "transactionNotificationsEnabled": true
  },
  "profileActive": true,
  "tokenLocation": "07",
  "tokenRequestorId": 69999999999,
  "tokenRequestorName": "Al's Toybarn",
  "tokenRequestorType": "ACQUIRER",
  "tokenRequestorKYCInfo": {
    "networkMid": "1234",
    "merchantUrl": "www.discover.com",
    "merchantAddress": {
      "zip": "60015",
      "city": "Riverwoods",
      "line1": "2500 Lake Cook Road",
      "line2": "Business Technology Department",
      "line3": "ATTN: CIO",
      "state": "Illinois",
      "countryCode": "US"
    },
    "merchantPhoneNumber": {
      "number": "8003472683",
      "phoneCountryCode": "1"
    },
    "merchantCategoryCode": [742]
  },
  "tokenRequestorPartyId": "123456",
  "parentTokenRequestorId": 69999999999,
  "profileEffectiveEndDate": "2025-04-22T13:12:33.822Z",
  "profileEffectiveStartDate": "2025-04-22T13:12:33.822Z"
}
```


Parameter Definitions

Parameter	Description
childTokenRequestors	A list of child token requestors.
client_id	Your API key issued by DFS Services LLC.

featureFlags	The Object containing feature toggles or flags related to the requestor's preference.
NetworkMid	The network Merchant identifier assigned by the payment network.
parentTokenRequestorId	Identifier for the parent token requestor entity.
profileActive	Indicates whether the token requestor profile is currently active.
ProfileEffectiveStartDate	Date when the token requestor profile begins.

Health Check

POST

Tests the API to evaluate the overall health of the boarding endpoint.

Endpoint

```
/globalpymt/ddx/stored-payment-token/boarding/v2/healthcheck
```

Example Request

```
{
  "requestId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
}
```

Example Response

```
{
  "healthy": true,
  "message": "DDX Stored Payment Boarding Service is healthy!",
  "version": "1.02.1-Release",
  "platform": "epp-ddx-trd-boarding-service",
```

```
  "timestamp": "2021-10-22T21:37:58.555Z"
}
```

Parameter Definitions

Parameter	Description
healthy	The API health status.
message	A status message of the API.
platform	Identifies the environment platform.
requestId	UUID connecting the request and response.
timestamp	The server's response time in ISO 8601 format.
version	The API version.

Cryptogram Generation API

The Cryptogram Generation API is used to create dynamic cryptograms for token-based transactions. The cryptograms support payment transactions initiated by consumers on a Merchant's website. The result is an extra layer of protection against fraud during payment transaction processing.

On Demand Credentials

POST

Generates one or many dynamic cryptograms for a payment token.

Endpoint

```
/globalpymt/ddx/stored-payment-token/cryptogram/v1/cryptogram
```

Example Request


```
{
  "requestId": "VHtafwC7xSA",
  "tokenReferenceId": "tokenCH8u9dUvWSMqQs0XWzOEo4kQ9I78CWtx",
  "tokenRequestorId": 62345678910,
  "transactionTimestamp": "2017-02-16T21:00:00+01:00",
  "tokenRequestorPartyId": "123456",
  "totalTransactionAmount": 100.28,
  "cryptogramRequestInfoList": [
    {
      "merchantURI": "www.xyzlimited.com",
      "merchantCity": "Chicago",
      "merchantName": "XYZ Limited",
      "recordLevelId": "3fa85f64-5717-4562-b3fc-2c963f66afc2",
      "cryptogramType": "DCID",
      "merchantCountry": "US",
      "transactionAmount": 100.28,
      "merchantPostalCode": "60606",
      "transactionCurrencyCode": "840"
    }
  ]
}
```


Example Response


```
{
  "responseId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "cryptogramResponseInfoList": [
    {
      "merchantName": "XYZ Limited",
      "recordLevelId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
      "cryptogramType": "DCID",
      "remoteCryptogram": "-A-Za-z0-9999999999999999999999999999999999999
      "transactionAmount": 100.28
    }
  ]
}
```


Parameter Definitions

Parameter	Description

cryptogramRequestInfoList	A list of one or more Merchant-level cryptogram requests.
cryptogramResponseInfoLis	A list of generated cryptograms.
cryptogramType	The type of cryptogram (DCID or TVI).
merchantCity	City where the Merchant resides.
merchantCountry	Two letter ISO country code where the Merchant resides.
merchantName	The name of the Merchant.
merchantPostalCode	Postal code of the Merchant.
merchantURI	The Merchant website address.
recordLevelId	The UUID used to map each cryptogram to a Merchant.
remoteCryptogram	JWE-encrypted value.
responseId	The identifier of the response.
requestId	Client-generated UUID to trace and correlate the request and response.
tokenReferenceId	The payment token ID with 32-64 alphanumeric characters.
tokenRequestorId	An ID assigned by DFS Services LL to the token requestor.
tokenRequestorPartyId	The Identifier of the party associated with the token requestor.

totalTransactionAmount	Total amount of the transaction.
transactionCurrencyCode	The ISO 4217 currency code.
transactionAmount	Amount of the transaction.
transactionTimestamp	Transaction timestamp in ISO 8601 format.

Health Check Controller

POST

Tests the health of the API.

Endpoint

```
/globalpymt/ddx/stored-payment-token/cryptogram/v1/healthcheck
```

Example Request

```
{
  "requestId": "fO-Uv4uZ3AtzUjaQdqTqu-lyR4fcsqUoPhiBPs2H8WZK78t4LSNZGvLvPf
}
```

Example Response

```
{
  "healthy": true,
  "message": "DDX Stored Payment Crypto Service is healthy!",
  "version": "1.02.1-Release",
  "platform": "epp-ddx-trd-crypto-service",
  "timestamp": "2021-10-22T21:37:58.555Z"
}
```

Parameter Definitions

Parameter	Description
healthy	The API health status.
message	A status message of the API.
platform	Identifies the environment platform.
requestId	UUID connecting the request and response.
timestamp	The server's response time in ISO 8601 format.
version	The API version.

Token Operations API

The Token Operations API offers token provisioning and lifecycle management. This API enables our partners to securely generate and manage payment tokens.

Token Provisioning

POST

Provisions a single token.

Endpoint

```
/globalpymt/ddx/stored-payment-token/operations/v1/token
```

Example Request

```
{
  "taskId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "requestId": "phQiXzcoUecabMdqI74cUqEiEhcwDJMOg1Eb1OlKevMCKz2nw",
  "consumerData": {
```

```
    "email": "johnsmith@discover.com",
    "consumerName": "ABC",
    "consumerIdentifier": "2fa55f64-5717-2634-b3fc-2c963f66aa22"
  },
  "billingAddress": {
    "zip": "60012",
    "city": "Riverwoods",
    "line1": "Discover",
    "line2": "C/O Marketing Department",
    "line3": "2500 Lake Cook Rd",
    "state": "Illinois",
    "countryCode": "US"
  },
  "cardHolderData": {
    "email": "johnsmith@discover.com",
    "phoneNumber": "Cardholder Phone Number",
    "cardholderName": "John Smith",
    "emailAddressAge": 1,
    "consumerIdentifier": "2fa55f64-5717-2634-b3fc-2c963f66aa22"
  },
  "tokenRequestorId": 62345678910,
  "countryOfProvision": "US",
  "cardHolderDeviceInfo": {
    "latitude": "65",
    "ipAddress": "123.255.0.1",
    "longitude": "43",
    "riskScore": 2,
    "deviceType": "MOBILE"
  },
  "tokenAssuranceContext": {
    "age": 1,
    "data": "testData",
    "dataType": "CRYPTOGRAM",
    "tokenAssuranceMethod": "01"
  },
  "tokenRequestorPartyId": "66221",
  "sourceOfTokenRequestor": "ADD_DEVICE",
  "encryptedProvisionAccountContext": "-A-Za-z0-9999999999999999999999999
}
```


Example Response


```
{
  "taskId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "responseId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "issuerContext": {
    "email": "info@discover.com",
    "website": "discover.com",
```

```
    "issuerName": "Discover",
    "contactNumber": "18003472687",
    "privacyPolicyURL": "discover.com/privacy",
    "termsConditionsURL": "discover.com/terms"
  },
  "tokenReferenceId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "tokenRequestorId": 62345678910,
  "termsAndConditionsId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "tokenAssuranceMethod": "01",
  "encryptedTokenContext": "-A-Za-z0-9999999999999999999999999999999999999
  "tokenRequestorPartyId": "66221",
  "paymentAccountReference": "6001D13PL7D1AJYG4BWONDXDLKPP1",
  "tokenProvisioningDecision": "APPROVED",
  "tokenizationDecisionReason": "Good",
  "provisionAccountMetadataContext": {
    "cardType": "CREDIT",
    "panSuffix": "6011",
    "cardImageId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "productDescription": "Discover Card Original",
    "currentExpirationYear": "2034",
    "currentExpirationMonth": "02"
  }
}
```


Parameter Definitions

Parameter	Description
`BillingAddress`	The billing address associated with the account.
`cardHolderData`	Details of the Cardholder name and PAN.
`consumerData`	Alternative parameter to cardHolderData that is used for additional user metadata.
`countryOfProvision`	The two digit country code where the token is provisioned.
`encryptedProvisionAccountContext`	JWE-encrypted data used for account provisioning.

EncryptedTokenContext	Contains the provisioned token information in JWE format.
issuerContext	An array of the metadata returned by the issuer.
issuerName	The name of the issuer.
panSuffix	Last four digits of the PAN.
paymentAccountReference	A 29-character reference of the provisioned account.
provisionAccountMetadataContext	A structured object containing metadata about the consumer provisioned account, device, channel, or the platform initiating the request.
requestId	Client-generated UUID to trace and correlate the request and response.
responseId	The identifier of the response.
riskScore	A numeric value indicating the risk of the transaction or token request.
sourceOfTokenRequestor	The metadata that identifies the origin of the token requestor.
taskId	UUID of the provisioned task.
termsAndConditionsId	UUID of the terms and conditions agreed upon by the user.
tokenAssuranceContext	Data assurance level of Cardholder verification.

tokenAssuranceMethod	The Europay Mastercard Visa Consortium (EMVCo) token assurance method code. 01=non-card issuer.
tokenProvisioningDecision	The description of the privisioning authorization.
tokenReferenceId	The payment token ID with 32-64 alphanumeric characters.
tokenRequestorId	An ID assigned to the token requestor by DFS Services LLC.
tokenRequestorPartyId	The Identifier of the party associated with the token requestor.
tokenizationDecisionReason	Resulting reason for the decline or an alternate tokenization result.

Provision One-to-Many Tokens

POST

Initiates the request to provision one-to-many tokens (async) and returns information required for polling.

Endpoint

```
/globalpymt/ddx/stored-payment-token/operations/v1/tokens
```

Example Request

```
{
  "taskId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "requestId": "pGRJBAz8U7-CW188-Dwiu9RpavPphLHWJhFCY6RE3x9QM",
  "tokenRequestList": [
    {
      "tokenType": "CLOUD",
      "consumerData": {
```

```
        "email": "johnsmith@discover.com",
        "consumerName": "ABC",
        "consumerIdentifier": "2fa55f64-5717-2634-b3fc-2c963f66aa22"
      },
      "recordLevelId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
      "billingAddress": {
        "zip": "60012",
        "city": "Riverwoods",
        "line1": "Discover",
        "line2": "C/O Marketing Department",
        "line3": "2500 Lake Cook Rd",
        "state": "Illinois",
        "countryCode": "US"
      },
      "cardHolderData": {
        "email": "johnsmith@discover.com",
        "phoneNumber": "Cardholder Phone Number",
        "cardholderName": "John Smith",
        "emailAddressAge": 1,
        "consumerIdentifier": "2fa55f64-5717-2634-b3fc-2c963f66aa22"
      },
      "countryOfProvision": "US",
      "cardHolderDeviceInfo": {
        "latitude": "65",
        "ipAddress": "123.255.0.1",
        "longitude": "43",
        "riskScore": 2,
        "deviceType": "MOBILE"
      },
      "tokenAssuranceContext": {
        "age": 1,
        "data": "testData",
        "dataType": "CRYPTOGRAM",
        "tokenAssuranceMethod": "01"
      },
      "sourceOfTokenRequestor": "ADD_DEVICE",
      "encryptedProvisionAccountContext": "-A-Za-z0-9999999999999999999999
    }
  ],
  "tokenRequestorId": 62345678910,
  "tokenRequestorPartyId": "66221"
}
```


Example Response

```
curl -X 'POST' \
  'https://p3-portal-aws-useast1-apps-stage-1.apps.aws-useast1-apps-stage-
  -H 'accept: application/json' \
```

```
-H 'Content-Type: application/json' \
-d '{
"taskId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
"requestId": "oXEDkBj-N1RlG-wjHa9-LzD5rMHqc14e",
"tokenRequestList": [
  {
    "tokenType": "CLOUD",
    "consumerData": {
      "email": "johnsmith@discover.com",
      "consumerName": "ABC",
      "consumerIdentifier": "2fa55f64-5717-2634-b3fc-2c963f66aa22"
    },
    "recordLevelId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "billingAddress": {
      "zip": "60012",
      "city": "Riverwoods",
      "line1": "Discover",
      "line2": "C/O Marketing Department",
      "line3": "2500 Lake Cook Rd",
      "state": "Illinois",
      "countryCode": "US"
    },
    "cardHolderData": {
      "email": "johnsmith@discover.com",
      "phoneNumber": "Cardholder Phone Number",
      "cardholderName": "John Smith",
      "emailAddressAge": 1,
      "consumerIdentifier": "2fa55f64-5717-2634-b3fc-2c963f66aa22"
    },
    "countryOfProvision": "US",
    "cardHolderDeviceInfo": {
      "latitude": "65",
      "ipAddress": "123.255.0.1",
      "longitude": "43",
      "riskScore": 2,
      "deviceType": "MOBILE"
    },
    "tokenAssuranceContext": {
      "age": 1,
      "data": "testData",
      "dataType": "CRYPTOGRAM",
      "tokenAssuranceMethod": "01"
    },
    "sourceOfTokenRequestor": "ADD_DEVICE",
    "encryptedProvisionAccountContext": "-A-Za-z0-99999999999999999999
  }
],
"tokenRequestorId": 62345678910,
```

```
  "tokenRequestorPartyId": "66221"
}'
```

Parameter Definitions

Parameter	Description
BillingAddress	The billing address associated with the account.
cardHolderData	Details of the Cardholder name and PAN.
cardHolderDeviceInfo	Device metadata of the Cardholder.
consumerData	Alternative parameter to cardHolderData that is used for additional user metadata.
countryOfProvision	The two digit country code where the token is provisioned.
encryptedProvisionAccountContext	JWE-encrypted data used for account provisioning.
EncryptedTokenContext	Contains the provisioned token information in JWE format.
issuerContext	An array of the metadata returned by the issuer.
issuerName	The name of the issuer.
panSuffix	Last four digits of the PAN.
paymentAccountReference	A 29-character reference of the provisioned account.
provisionAccountMetadataContext	A structured object containing metadata about the consumer provisioned account, device, channel, or

	the platform initiating the request.
requestId	Client-generated UUID to trace and correlate the request and response.
recordLevelId	ID for tracking the request at the record level.
responseId	The identifier of the response.
riskScore	A numeric value indicating the risk of the transaction or token request.
sourceOfTokenRequestor	The metadata that identifies the origin of the token requestor.
taskId	UUID of the provisioned task.
termsAndConditionsId	UUID of the terms and conditions agreed upon by the user.
tokenAssuranceContext	Data assurance level of Cardholder verification.
tokenAssuranceMethod	The Europay Mastercard Visa Consortium (EMVCo) token assurance method code. 01=non-card issuer.
tokenType	The type of token being requested (i.e. CLOUD).
tokenRequestorId	An ID assigned to the token requestor by DFS Services LLC.
tokenRequestList	An array containing one or more token request objects.

tokenRequestorPartyId	The Identifier of the party associated with the token requestor.
tokenizationDecisionReason	Resulting reason for the decline or an alternate tokenization result.

Retrieve Bulk Provision Results

GET

Retrieves the results of the given asynchronous bulk provision request.

Endpoint

```
/globalpymt/ddx/stored-payment-token/operations/v1/request/{requestId}
```

Example Request

```
GET https://partner.discoverglobalnetwork.com/globalpymt/ddx/stored-paymen
```

Example Response


```
{
  "responseId": "b4de98ab-2ccf-4de4-b7a5-9b6e2a3d8e42",
  "taskId": "3d9a2a4e-b9c9-4a33-aeec-22f28a5d926e",
  "requestId": "f2a1b7cc-b8e3-4c76-b1b1-dfdc8887ef66",
  "tokens": [
    {
      "tokenReferenceId": "TKN-1234567890",
      "tokenRequestorId": "TRX12345",
      "token": "411111******1111",
      "tokenType": "CL",
      "tokenStatus": "ACTIVE",
      "expirationDate": "2027-12",
      "provisionedDate": "2025-04-01T12:00:00Z"
    },
    {
      "tokenReferenceId": "TKN-0987654321",
      "tokenRequestorId": "TRX67890",
      "token": "550000******0004",
      "tokenType": "01",
```

```
      "tokenStatus": "ACTIVE",
      "expirationDate": "2026-11",
      "provisionedDate": "2025-03-28T09:15:00Z"
    }
  ]
}
```

Parameter Definitions

Parameter	Description
client_id	Your API key issued by DFS Services LLC.
client_secret	Your client secret issued by DFS Services LLC.
expirationDate	Expiry of the provisioned token (month/year).
provisionedDate	Timestamp when the token was provisioned.
requestId	Client-generated UUID to trace and correlate the request and response.
responseId	The identifier of the response.
taskId	An optional tracking ID for batch/task operations.
token	The masked token value.
tokenReferenceId	The reference identifier associated with the token.
tokenStatus	The resulting token status after operation.
tokenType	Token usage type: CL=card on file; 01=single use.

GET

Checks the current status of an asynchronous bulk provision request.

Endpoint

```
/globalpymt/ddx/stored-payment-token/operations/v1/request/{requestId}
```

Example Request

```
GET https://partner.discoverglobalnetwork.com/globalpymt/ddx/stored-paymen
```

Example Response

```
{
  "requestId": "abc123-request-id",
  "taskId": "def456-task-id",
  "status": "COMPLETED",
  "timestamp": "2025-05-15T14:23:00Z",
  "description": "Token processing completed successfully.",
  "tokensGenerated": 3
}
```

Parameter Definitions

Parameter	Description
description	Status check return.
requestId	The UUID that connects the request and response.
status	Task processing and token operation status results.
taskId	An optional tracking ID for batch/task operations.
timestamp	When the status check response is generated.

tokensGenerated	The quantity of tokens created.

Token Metadata

POST

Retrieves metadata for the given token.

Endpoint

```
/globalpymt/ddx/stored-payment-token/operations/v1/token/metadata
```

Example Request


```
{
  "taskId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "requestId": "Rii2uzlDTm0lnxtf40C3ox7NQz99csD2pez8nQbw0fU5",
  "encryptedToken": "-A-Za-z0-9999999999999999999999999999999999999999999
  "tokenReferenceId": "tokenCH8u9dUvWSMqQs0XWzOEo4kQ9I78CWtx",
  "tokenRequestorId": 62345678910,
  "tokenRequestorPartyId": "66221",
  "expandedAccountMetadataRequested": [
    "string"
  ]
}
```


Example Response


```
{
  "token": {
    "tokenStatus": "ACTIVE",
    "tokenReferenceId": "tokenCH8u9dUvWSMqQs0XWzOEo4kQ9I78CWtx",
    "tokenIssuerContext": {
      "email": "johnsmith@discover.com",
      "website": "discover.com",
      "issuerName": "Discover",
      "contactNumber": "18003472683",
      "privacyPolicyURL": "discover.com/privacy"
    },
    "tokenMetadataContext": {
      "tokenExpYear": "2024",
      "tokenExpMonth": "12",
```

```
      "tokenLastFour": 9012
    },
    "tokenAccountMetadataContext": {
      "cardType": "CREDIT",
      "productDescription": "Discover Card Original"
    },
    "encryptedTokenAccountContext": "string"
  },
  "taskId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "parData": {
    "paymentAccountReference": "12345678987654321123"
  },
  "responseId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "cardImageData": {
    "labelColor": "#hf1234",
    "cardImageId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "backgroundColor": "#ff1130",
    "foregroundColor": "#ff0000"
  },
  "tokenRequestorId": 62345678910,
  "tokenRequestorPartyId": "12345",
  "termsAndConditionsData": {
    "termsAndConditionsId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "termsAndConditionsURL": "ZFhp0huPsyGUuUO7fH8y+w9vfpUyjzppFDgkKrSMZAAR
  }
}
```


Parameter Definitions

Parameter	Description
cardImageData	Contains visual design data for the tokenized physical card (i.e. color, branding).
encryptedToken	The encrypted form of the payment token to be decrypted or queried.
encryptedTokenAccountContext	Encrypted metadata associated with the token account.
ExpandedAccountMetadataRequested	Optional account fields requested in the response.

parData	Contains the paymentAccountReference value of the underlying payment account number. Data is returned in the response if the expandedAccountMetadataRequested in the original request contains the parData value.
requestId	Client-generated UUID to trace and correlate the request and response.
responseId	The identifier of the response.
taskId	UUID of the provisioned task.
termsAndConditionsData	An object containing the applicable terms and conditions.
token	Contains the token detail.
tokenAccountMetadataContext	Provides the token attributes.
tokenIssuerContext	The issuer contact information and branding metadata (i.e. issuer name, website, phone number).
tokenRequestorId	An ID assigned to the token requestor by DFS Services LLC.
tokenRequestorPartyId	The Identifier of the party associated with the token requestor.
tokenStatus	The resulting token status after operation is processed.

Token Lifecycle Operations

The Token Lifecycle Operations API is used by token requestors to manage the state of the token.

PUT

Endpoint

```
/globalpymt/ddx/stored-payment-token/operations/v1/token/state
```

Example Request

```
{
  "taskId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "requestId": "Pt0ILfvObKObNvAcHoy28OBQ4Ase5y7SCOIDc",
  "tokenReferenceId": "66221ASDFSADFASDLFKJASDFLKJASDFKLJASDF",
  "tokenRequestorId": 62345678910,
  "tokenRequestedState": "ACTIVE",
  "tokenRequestorPartyId": "66221",
  "tokenStateChangeReason": "FRAUD RISK"
}
```

Example Response

```
{
  "taskId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "responseId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "tokenStatus": "ACTIVE",
  "tokenReferenceId": "66221",
  "tokenRequestorId": 62345678910,
  "tokenRequestorPartyId": "66221"
}
```

Parameter Definitions

Parameter	Description
requestId	Client-generated UUID to trace and correlate the request and response.
responseId	The identifier of the response.

taskId	UUID of the provisioned task.
tokenReferenceId	A unique reference identifier for the token being modified or queried.
tokenRequestedState	The returned new state of the token (i.e. active, suspended).
tokenRequestorId	An ID assigned to the token requestor by DFS Services LLC.
tokenRequestorPartyId	The Identifier of the party associated with the token requestor.
tokenStateChangeReason	The reason for the change of the token state provided by the requestor.
tokenStatus	The resulting token status after operation is processed.
termsAndConditionsData	An object containing the applicable terms and conditions.
token	Contains the token detail.
tokenAccountMetadataContext	Provides the token attributes.
tokenIssuerContext	The issuer contact information and branding metadata (i.e. issuer name, website, phone number).
tokenRequestorId	An identifier assigned to the token requestor by DFS Services LLC.
tokenRequestorPartyId	Identifier for the party/entity associated with the token requestor.
tokenStatus	The resulting token status after operation.

Update Lifecycle Status

PUT

Initiates the request to update the lifecycle status of one-to-many tokens.

Endpoint

```
/globalpymt/ddx/stored-payment-token/operations/v1/tokens/state
```

Example Request

```
{
  "method": "POST",
  "url": "https://apis.discover.com/globalpymt/ddx/stored-payment-token/op
  "headers": {
    "Accept": "application/json",
    "Content-Type": "application/json",
    "Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.MockToke
    "client_id": "your-client-id",
    "client_secret": "your-client-secret"
  },
  "body": {
    "taskId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
    "requestId": "tPcZ6lwlj9nixfJj34-VqlFZikAmFbwyI6c7xzen-A",
    "tokenOperations": [
      {
        "tokenReferenceId": "66221ASDFSADFASDLFKJASDFLKJASDFKLJASDF",
        "tokenRequestedState": "ACTIVE",
        "tokenStateChangeReason": "Suspended Token"
      }
    ],
    "tokenRequestorId": 62345678910,
    "tokenRequestorPartyId": "66221"
  }
}
```

Example Response

```
{
  "responseId": "a97cd602-b2e2-456b-a68a-9f145e9e26f4",
  "taskId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "requestId": "tPcZ6lwlj9nixfJj34-VqlFZikAmFbwyI6c7xzen-A",
```

```
  "status": "COMPLETED",
  "statusDateTime": "2025-04-23T14:36:00Z",
  "tokenOperations": [
    {
      "tokenReferenceId": "66221ASDFSADFASDLFKJASDFLKJASDFKLJASDF",
      "tokenStatus": "ACTIVE",
      "status": "SUCCESS",
      "message": "Token successfully moved to ACTIVE state."
    }
  ]
}
```


Parameter Definitions

Parameter	Description
client_id	Your DFS Services LLC-issued API key.
client_secret	Your client secret issued by DFS Services LLC.
message	the token operation result.
requestId	Client-generated UUID to trace and correlate the request and response.
responseId	An identifier of the response.
status	The token operation status results.
statusDateTime	Timestamp when the task reached the reported status.
taskId	Identifier of the task associated with the token state change operation.
tokenOperations	An array of token operations included in the request and echoed in the response.

tokenReferenceId	A unique reference identifier of the token being modified or queried.
tokenRequestedState	The new state of the token (i.e. active, suspended).
tokenRequestorId	An identifier assigned to the token requestor by DFS Services LLC.
tokenRequestorPartyId	Identifier of the party/entity associated with the token requestor.
tokenStateChangeReason	The reason provided by the requestor of the token state change.
tokenStatus	The resulting token status after operation.

Retrieve Bulk Token Results

GET

Receives the results of the given asynchronous bulk token status request.

Endpoint

```
/globalpymt/ddx/stored-payment-token/operations/v1/request/{requestId}
```

Example Request

```
GET https://partner.discoverglobalnetwork.com/globalpymt/ddx/stored-paymen
```

Example Response

```
{
  "taskId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "responseId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "tokenStatusResults": [
    {
      "tokenStatus": "ACTIVE",
      "tokenReferenceId": "66221ASDFSADFASDLFKJASDFLKJASDFKLJASDF",
```

```
      "tokenRequestorId": "62345678910"
    }
  ]
}
```

Parameter Definitions

Parameter	Description
responseId	The identifier of the response.
taskId	An optional tracking ID for batch/task operations.
tokenRequestorId	A numeric ID assigned to the token requestor by the network.
tokenReferenceId	A unique reference identifier for the token being modified or queried.
tokenRequestorId	The ID assigned to the token requestor by DFS Services LLC.
tokenStatusResults	An array of objects representing the result of a token state change such as active, suspension, or termination.
tokenStatus	The resulting token status after operation.

Check Bulk Token Status

GET

Checks the current status of the given asynchronous bulk token status request.

Endpoint

```
/globalpymt/ddx/stored-payment-token/operations/v1/request/{requestId}
```

Example Request

```
GET https://partner.discoverglobalnetwork.com/globalpymt/ddx/stored-paymen
```

Example Response


```
{
  "responseId": "0ba5c230-abe1-4b39-8f01-f17ef72a9d45",
  "requestId": "f2a1b7cc-b8e3-4c76-b1b1-dfdc8887ef66",
  "taskId": "3d9a2a4e-b9c9-4a33-aeec-22f28a5d926e",
  "status": "COMPLETED",
  "statusDateTime": "2025-04-23T15:20:00Z",
  "tokenStateResults": [
    {
      "tokenReferenceId": "66221ASDFSADFASDLFKJASDFLKJASDFKLJASDF",
      "tokenRequestorId": "62345678910",
      "status": "SUCCESS",
      "message": "Token successfully moved to ACTIVE state.",
      "tokenStatus": "ACTIVE"
    }
  ]
}
```


Parameter Definitions

Parameter	Description
message	A message about the token operation result.
requestId	UUID which connects the request and response.
responseId	The response identifier.
taskId	An optional tracking ID for batch/task operations.
tokenReferenceId	A unique reference identifier of the token being modified or queried.

tokenRequestorId	The numeric ID assigned to the token requestor by the network.
tokenStatusResults	An array of objects representing the result of a token state change such as active, suspension, or termination.
tokenStatus	The resulting token status after operation.
status	The token operation status results.
statusDateTime	A timestamp when the task reached the reported status.

Token Account Lookup

This API is used to look up token information using a PAN or token ID.

POST

Endpoint

```
/globalpymt/ddx/stored-payment-token/operations/v1/token-account-detai
```

Example Request

```
{
  "taskId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "requestId": "-NsflY2fX570J",
  "tokenReferenceId": "STIDDN58x67b4412ef4aabff5612ac8affa9b2",
  "tokenRequestorId": 62345678910,
  "tokenRequestorPartyId": "66221",
  "encryptedAccountContext": "==VGhpcyBpcyBhbiBleGFtcGxlIHZhbHVlIHdoaWNoIE
}
```

Example Response

```
{
  "taskId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "responseId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "tokenRequestorId": 62345678910,
```

```
  "tokenRequestorPartyId": "66221",
  "encryptedTokenLookupResponse": "==VGhpcyBpcyBhbiBleGFtcGxlIHZhbHVlIHdoa
}
```

Parameter Definitions

Parameter	Description
encryptedAccountContext	A base64 or binary-encoded string containing encrypted account-level metadata such as Cardholder account information and verification details.
encryptedTokenLookupResponse	A base64-encoded and cryptographically protected string that contains the full decrypted details of a token lookup result.
requestId	A client-generated UUID used to trace and correlate the request and response.
responseId	The identifier of the response.
taskId	Identifier for the task associated with the token state change operation.
tokenReferenceId	The unique reference identifier for the token being modified or queried.
tokenRequestorId	ID assigned to the token requestor by DFS Services LLC.
tokenRequestorPartyId	An ID of the business entity or legal party of the token requestor.

Health Check Controller

POST

This method checks the health status of the endpoint.

Endpoint

```
/globalpymt/ddx/stored-payment-token/operations/v1/healthcheck
```

Example Request


```
{
  "method": "POST",
  "url": "https://apis.discover.com/globalpymt/ddx/stored-payment-token/op
  "headers": {
    "Accept": "application/json",
    "Content-Type": "application/json",
    "Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.MockAcce
    "client_id": "your-client-id",
    "client_secret": "your-client-secret"
  },
  "body": {
    "requestId": "8b93a1f7-dc6a-432f-bc03-cdf99b3e1297",
    "taskId": "21f4c111-1820-470b-89c0-0370ac4a6a33",
    "platform": "UAT",
    "version": "v1.0"
  }
}
```


Example Response


```
{
  "healthy": true,
  "message": "DDX Stored Payment Token Operations Service is healthy!",
  "version": "1.02.1-Release",
  "platform": "epp-ddx-trd-token-operations-service",
  "timestamp": "2021-10-22T21:37:58.555Z"
}
```


Parameter Definitions

Parameter	Description

client_id	Your API secret you use to get access tokens.
encryptedAccountContext	A base64 or binary-encoded string containing encrypted account-level metadata such as Cardholder account information and verification details.
encryptedTokenLookupResponse	A base64-encoded and cryptographically protected string that contains the full decrypted details of a token lookup result.
healthy	The health status of the API .
message	API status description.
platform	Identifies the environment platform.
requestId	A client-generated UUID used to trace and correlate the request and response.
responseId	The identifier of the response.
taskId	Identifier for the task associated with the token state change operation.
tokenReferenceId	The unique reference identifier for the token being modified or queried.
tokenRequestorId	ID assigned to the token requestor by DFS Services LLC.
tokenRequestorPartyId	An ID of the business entity or legal party of the token requestor.
requestId	The UUID which connects the request and response.

taskId	The identifier for the task associated with the token state change operation.
timestamp	When the health check response is generated.
version	API version.

Card Asset API

Asset Service

The Asset Service API is used by the token requestors, issuers and digital wallet partners to retrieve personalized card details and additional asset resources using the `assetId`.

Servers

```
https://trda-asset-service-epp-ddx-token-requestors-dev.apps.aws-useas
```

Health Check

POST

Checks the overall health of the system.

Endpoint

```
/globalpymt/ddx/stored-payment-token/assets/v1/healthcheck
```

Example Request

```
{
  "requestId": "0-g-rPp1cgHk2i4RMKQo72EWdElqHqt9Vg6VZ8wAueW6MCYPFMN843"
}
```

Example Response

```
{
  "healthy": true,
  "message": "DDX Stored Payment Asset Service is healthy!",
  "version": "1.02.1-Release",
  "platform": "epp-ddx-trda-asset-service",
  "timestamp": "2021-10-22T21:37:58.555Z"
}
```


Parameter Definitions

Parameter	Description
healthy	The API health status.
message	A status message of the API.
platform	Identifies the environment platform.
requestId	UUID connecting the request and response.
timestamp	The server's response time in ISO 8601 format.
version	The API version.

Card Asset Retrieval

GET

Retrieves card asset information.

Endpoint

```
/globalpymt/ddx/stored-payment-token/assets/v1/request/{requestId}/ass
```

Example Request

```
GET https://trda-asset-service-epp-ddx-token-requestors-dev.apps.aws-u
```

Example Response

```
{
  "mediaType": "text/html",
  "textAsset": "This is plain text",
  "responseId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "assetDescription": "TermsAndConditions"
}
```

Parameter Definitions

Parameter	Description
assetDescription	A description of the returned asset.
mediaType	The type of media returned.
responseId	An identifier of the response.
textAsset	The format of the returned asset.

API Scopes

DFS Services LLC uses API scopes to manage your permissions using access tokens for reading data, writing data, and deleting data. Scopes validate your API requests for authorization. Your provisioned scopes are in your account. Login here.

API Plan

Your API plan controls routing to the endpoints to manage rate limits and quotas based on your organization's service agreement.

Plan	Endpoint
Sandbox	sandbox.apis.discover.com

Certification	sandbox.apis.discover.com
Production	apis.discover.com

Your API plan must be specified in the HTTP header of all API requests including OAuth 2.0 access token requests.

Response Error Codes

Error Code	Description	HTTP Status Code
6	The system has difficulty processing the request.	500
00000	The response was successful.	200
00001	Unknown Wallet Program.	400
00002	Unsupported Wallet Program.	400
00003	Unsupported API.	400
00004	Unknown Issuer. The issuerId is unknown in the system.	400
1001	User authentication failed	401
1003	Not authorized to access the resource	400
1006	The system is in maintenance mode.	503
10001	Invalid PAN.	400
10002	PAN Ineligible.	400

10003	PAN provisioning count exceeded.	400
10004	Cryptographic error. This is returned when you try to decrypt or encrypt sensitive information.	400
10005	PAR doesn't exist.	204
10006	Card identifier validation failed.	400
10007	Account/user can't be added at this time.	400
10008	The token requestor can't be updated at this time.	400
10009	Account is already provisioned in the wallet.	400
10010	Name verification failed.	400
10011	Account/user can't be updated at this time.	400
10012	The card expired.	400
10013	User is locked out from provisioning.	400
10014	Account not found.	400
10015	The OTP verification failed.	400
10016	Invalid terms and conditions Id.	400
10017	The tokenRequestorPartyId isn't eligible for token aggregator boarding.	400

10018	Device has reached provisioning limit.	400
10019	Invalid device type.	400
10020	Invalid asset. The UUID provided is invalid.	400
10021	Invalid provision correlation Id	400
10022	Expired provision correlation Id	400
10023	Duplicate request. The request was identified as duplicate for the given transaction.	400
10024	Unknown programId. This is returned when the programId is invalid.	400
10025	Invalid cryptoType. This is returned when the provided cryptoType is invalid.	400
10026	The token is not valid,	400
10027	Invalid SessionId.	400
10028	Invalid softwareVersion. The softwareVersion passed in is not valid.	400
10029	Invalid tokenRequestorPartyId. The named field is invalid in the request.	400
10030	Invalid existing account details.	400
10031	Token re-personalization in progress.	400

10034	Invalid taskId.	400
10035	Expired profilecorrelationId.	400
10036	Token re-personalization retry count exceeded.	400
10037	Invalid token requestor and payment token identifiers. The mapped values are not valid.	400
10038	Invalid wbcVersion. The wbcVersion passed in is not valid.	400
10039	Device is already registered.	400
10040	Device is already initialized and the pocket keys have been generated for the wallet and device combination.	400
10041	Device is already de-registered.	400
10042	Device not found in the database.	400
10043	Device registration in progress.	400
10044	Device is not registered in the database.	400
10045	Device is de-registered in the database.	400
10046	Device initialization failed.	400
10047	Bad external status due to the account being closed.	400
10048	Bad external status (i.e. lost or stolen device).	400

10102	Passed tokenId is invalid.	400
10103	The status associated with the passed-in tokenId is invalid.	400
10104	Invalid channel identifier in the request.	400
10105	Error occurred while retrieving status from one or more systems.	400
10106	This is returned when the operation type isn't supported by the wallet.	400
10107	Invalid PAN identifier. The PAN identifier wasn't found in the NWP system.	400
10108	There's no tokens found for the panId.	400
10109	This is returned when the index sent in the request isn't unique.	400
10110	Invalid one-time password (OTP) channel identifier. The selectedChannelId passed in isn't valid.	400
10111	The one-time password (OTP) request limit was reached. Often a rate limit issue.	400
10112	The user is locked out from provisioning due to too many unsuccessful one-time password (OTP) submissions.	400
10113	Invalid one-time password (OTP).	400
10114	The time-to-live of the one-time password (OTP) value is expired.	400

10115	Invalid card art elements.	400
10116	The one-time password (OTP) isn't present in the request for the issuer who is configured to receive it.	400
10117	The AccountContext/TokenContext not present in request when Issuer is configured to receive it	400
10118	Invalid consumer and Cardholder object combination. make sure that either the cardHolderData or the consumerData object is included in the request.	400
10119	Invalid token and tokenReferenceId field combination. Double-check that either the token or the tokenReferenceId is included in your request.	400
10120	Invalid tokenRequestorId. The named field is invalid in your request.	400
10121	There are no results found for the given taskId.	412
10122	The request failed because the data sent in the request is too large.	413
10123	Duplicate recordLevelId in the request. Make sure the unique record level identifiers are in the request.	400
10124	The tokenReferenceId and the PAN may not be present in the request.	400
10125	The token not activ.e	400
10131	No contact channels were retrieved for the Cardholder.	400

10132	The tokenId wasn't found.	400
10133	Unsupported resource type.	400
30001	The request was valid and while processing but it returned an unexpected error.	500
30002	Downstream system unavailable.	500
30003	Invalid one-time password (OTP) request limit was reached.	400
30004	The one-time password (OTP) value is invalid.	400
30005	Expired one-time password (OTP) value.	400
30006	The one-time password (OTP) selectedChannelIdentifier is invalid.	400
30007	The user is locked out from provisioning due to too many unsuccessful one-time password (OTP) submissions.	400
70001	The issuer details were not found for the tokenRequestorPartyId and the bankIdentification.	200
70002	Issuer details for the IIN ResourceId are missing.	200
70003	PAN whitelisting check failed.	400
70004	Luhn validation check failed for the PAN.	400
70005	Bin range refresh not enabled for the issuer network partyId.	200

70006	Duplicate issuer bin range refresh.	200
70007	Bin range refresh processing error.	400
70008	No eligible bin range to add or update.	200
70010	The token requestor party details not found for the tokenrequestorpartyId and the tokenrequestorId.	xxx
90000	Payload couldn't be parsed.	400
90001	Mandatory named field isn't present in the request.	400
90002	Invalid field length. The named field is not present in the request.	400
90003	The field type is invalid	400
90004	Invalid field value. The named field is invalid in the request.	400
90005	Invalid HTTP header.	400
90006	No content-type in HTTP header. The named field is not present in the request.	400
90007	Too many items in the request.	400
90008	The panId wasn't found or resulted in an error.	400
90009	Could not generate the one-time password (OTP) .	500

10126	Token is locked. To complete the checkout please call XXX-XXX-XXXX.	400
10127	Token not found.	400
10128	Unable to send the one-time password (OTP) requested for this transaction.	400
10129	Unable to perform the one-time password (OTP) send operation that was requested for this transaction.	400
10130	Invalid token type.	400

Versioning

This API uses versioned endpoints (i.e. /v1/ and /v2/) to indicate the release level of individual resources.
Questions? Contact your DFS Services LLC integration representative.

Glossary

Term	Description
Acquirer	A business entity that facilitates card processing between a Merchant and a payment network.
Bank Identification Number (BIN)	A unique identifier attached to a token.
Certificate Authority (CA)	A trusted entity that issues digital certificates to verify the identity of websites, organizations, and individuals.
Card-Not-Present (CNP)	A transaction where the card isn't present at the time of purchase.

Card Present Indicator (CID)	An Identifier that indicates whether a card was swiped (in-person purchase) or a card-not-present transaction (online purchase).
Certified Token Requestor (CTR)	An entity authorized by a token service provider such as Visa and Mastercard that requests and manages payment tokens on behalf of cardholders and merchants.
Claims (payload)	This is the content of the JWS which contains the claims. Claims are a hash of the body content (which could be encrypted) and the metadata.
Cloud Wallet	A type of digital wallet product interacting between the Cardholder and the Cloud Wallet.
Cloud Wallet Providers	A company or service providing digital wallet cloud solutions. They manage payment credentials of credit cards and tokens.
Consumer Application Certificate	A digital certificate to secure communication between the client and the API.