Card Security

This section provides an overview of security implementation for all the APIs provided by Network International.

1. SSL Implementation

API communication happens over a secure TLS1.2 channel for enforcing encryption of API payload.

- Client Hello and Server Hello: Server provides the information which includes SSL version number, cipher settings, session-specific data that is needed to communicate with the client
- Authentication and Pre-Master Secret: Client authenticates the server certificate. (e.g. Common Name / Date / Issuer) Client (depending on the cipher) creates the pre-master secret for the session, encrypts with the server's public key and sends the encrypted pre-master secret to the server.
- Decryption and Master Secret: Server uses its private key to decrypt the pre-master secret. Both Server and Client perform steps to generate the master secret with the agreed cipher.
- Encryption with Session Key: Both client and server exchange messages to inform that future messages will be encrypted.

2. OAUTH Flow Description

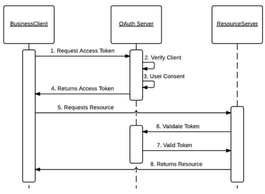
Client will send authorization request to NI API Gateway. Once API Gateway receives the authorization request, it will respond with access token to client. Client will trigger the API request with access token received from API Gateway. API Gateway will verify the access token received from client for each API request processing.

The access token will be placed in the standard authorization header of the client requests,' where it can then be validated by the API Gateway.

Example: Authorization: Bearer XXXXXX_ACCESS_TOKEN_XXXXX

Client application has to retrieve a new access token before the expiry of existing token.

3. Flow Diagram



4. Creation of Access Token

API Gateway will share full URI for requesting access token e.g.https://api.network.ae/oauth2/token

Client application also needs to send a 'client ID' and a 'Client secret' pair along with each API call.

4.1 Token Request

HTTP Method

POST

Headers

Field	Value
Accept	application/json
Content-Type	application/x-www-form-urlencoded

Body Form Data

Field	Value
client_id	*** Issuer generated value ***
client_secret	*** Issuer generated value ***
grant_type	client_credentials

Example Body Form Data

client_id=9d70c6bbad8ad20262828222fc0f3fdd&client_secret=a3d5566e8ca0d6da823eb7815c1c2b&grant_type=client_credentials

4.2 Token Response

HTTP Status Codes Supported

Client supports some basic HTTP Status codes today. Most import to send a 200 on successful creation.

Code	Description
200	The tokens were successfully created and are returned in the response body.
400	Client error; the request is malformed in some way and must be corrected.
401	Unauthorized request; the authorization code has expired.
403	You're not authorized to access this endpoint.
404	Endpoint not found
429	The request has been rejected because of rate limiting – you've sent too many
500	The API encountered an error while attempting to communicate with the back end.
502	Internal connection failure.
503	Backend at capacity error.
504	Gateway Timeout.

4.3 Response Body JSON Properties

Client supports the following on the create token response. Most importantly **access_token** and **expires**_in properties.

Field	Description
Token_type	Always set to "Bearer"
Expires_in	The Time length/validity of Token in seconds.
Access_token	The token value created by the Issuer system. This token will be presented in the header of subsequent requests.

Example Response Body

{"access_token":"cxac2rxtgw45xcst2495s2pa","token_type":"bearer","expires_in":300}