API Specifications

API Name: Camouflage **Interface Version:** v1.5

Initial Release Date: Oct 2016 **Last Revised Date: Oct 18, 2016**

Table of Contents

1. Client Application Registration

For general client registration, the process will include filling out and submitting the form via API Consumer Portal. Details for client application registration is available at https://dlife.discoverfinancial.com/docs/DOC-154149. This will create a new key/secret via Layer 7 and will make it available for use for JWT authentication between provider and consumers using JWTv2.

2. Client Application Authentication

In order to access the CLOAK services and Data Services the approved client application must be authenticated in real-time.

Authentication by Camouflage application follows the approved solution pattern DFS REST Client Authentication Security Pattern. https://dlife.discoverfinancial.com/docs/DOC-115443

3. Accessing CLOAK Service APIs and Data Services

To use the CLOAK APIs the client application must be authenticated first with the client application id, client application secret, and consumer application certificate. To authenticate the application, the Camouflage application makes HTTP GET request to end point https://intapistest.discoverfinancial.com/dfs/auth/jwt/v2/token.

When the DFS Open API gateway receives the request, it authenticates the client application by using the client application id, secret, and scope contained in the request.

Similar process needs to be followed after Data Services enables JWT Authentication.

4. Tokenize/Detokenize and Metadata APIs

4.1. Bulk Tokenize and Metadata

```
public TokenizeAndMetadataResponseVO[] tokenize(String[] input, String
dataElementName, Set<MetaDataAttributeName> metaDataAtrributeNames) throws
TokenizationServiceException, MetaDataServiceException;
```

Request Fields Description

Field Name	Required	Data Type	Description
Input	Y	Array	This is the value that needs to be tokenized.
Date Element Name	Y	String	Data Element Name. Examples: DE_PAN23, DE_SSN23.
Set<MetaDataAttributeName>	N	Set	Set of Metadata attributes to be returned by this API. MetaDataAttributeName.ALL returns all available Metadata. If null it does not return any metadata.

Please refer to Appendix A for structure of TokenizeAndMetadataResponseVO and list of available metadata attributes (MetaDataAttributeName)

Response Fields Description

Field Name	Required	Data Type	Description
TokenizeAndMetadataResponseVO	Y	Array	The tokenized value and metadata for input values.

Error Scenario

TokenizationServiceException will be throw if there is a connectivity problem or all inputs cannot be tokenized. Error for individual element will be represented by ErrorVO inside TokenizeAndMetadataResponseVO for each element. If IIN Range not found then EDIT, LENGTH, MOD10 and PAN_VERIFICATION will be returned as FAILED.

Input array max limit per call

Current input array limit is 5000 elements defines by Cloak Service.

4.2. Bulk Detokenize

```
public DetoknizeResponseVO[] detokenize(String[] token, String dataElementName)
throws TokenizationServiceException;
```

Request Fields Description

Field Name	Required	Data Type	Description
Token	Y	Number	This is the token that needs to be detokenized.
Date Element Name	Y	String	Data Element Name. Examples: DE_PAN23, DE_SSN23

Response Fields Description

Field Name	Required	Data Type	Description
DetoknizeResponseVO	Y	Object	The detokenized value of the token.

Error Scenario

TokenizationServiceException will be throw if there is a connectivity problem.
Error for individual element will be represented by ErrorVO inside `DetoknizeResponseVO` for each element.

Token array max limit per call

Current token array (input) limit is 5000 elements defines by Cloak Service.

4.3. Tokenize and Metadata (One Element)

```
public TokenizeAndMetadataResponseVO tokenize(String input, String dataElementName,
Set<MetaDataAttributeName> metaDataAtrributeNames) throws
TokenizationServiceException, MetaDataServiceException;
```

Request Fields Description

Field Name	Required	Data Type	Description
Input	Y	String	This is the value that needs to be tokenized.
Date Element Name	Y	String	Data Element Name. Examples: PAN, SSN
Set<MetaDataAttributeName>	N	Set	Set of Metadata attributes to be returned by this API. MetaDataAttributeName.ALL returns all available Metadata. If null it does not return any metadata.

Response Fields Description

Field Name	Required	Data Type	Description
TokenizeAndMetadataResponseVO	Y	Object	The tokenized value of the input String and metadata.

Please refer to Appendix A for structure of TokenizeAndMetadataResponseVO and list of available metadata attributes (MetaDataAttributeName)

Error Scenario

TokenizationServiceException will be thrown if input cannot be tokenized or unable to connect to Cloak service.

4.4. Detokenize

```
public DetoknizeResponseVO detokenize(String token, String dataElementName) throws
TokenizationServiceException;
```

Request Fields Description

Field Name	Required	Data Type	Description
Token	Y	String	This is the token that needs to be detokenized.
Date Element Name	Y	String	Data Element Name. Examples: PAN, SSN

Response Fields Description

Field Name	Required	Data Type	Description
DetoknizeResponseVO	Y	String	The detokenized value of the token.

Error Scenario

TokenizationServiceException will be thrown if input cannot be detokenized or unable to connect to Cloak service.

5. Camouflage integration

Following steps needs to be followed for integrating Camouflage into your code base.

5.1. Configure JWT Client Parameters for Cloak Service and Data Service

e.g.

```
jwt_client_cacheOffsetOfExpTime=86400
jwt_client_environment=DEV
jwt_client_requireGatewayAuthentication=true
jwt_client_tokenEndpointURL=https://apidev.discoverfinancial.com/dfs/auth/jwt
/v2/token
jwt_client_tokenTimeToLive=129600
jwt_client_issuer=<Application Name>
jwt_client_apiKey=xxx

#Configure either apiSecret or keyStore, keyStorePassword, keyStoreType
(Option# 2 is recommended approach by EA)
#Option 1
jwt_client_apiSecret=xxx (Note: Encrypted value using EncryptionEngine java
#class
#e.g. Base64.encode(EncryptionEngine.encrypt("propertyvalue".getBytes())))

#Option 2 (Configure keyStore, keyStorePassword, keyStoreType)
jwt_client_appSecretKeyStore=<Path to jceks file provided by EA>
jwt_client_appSecretKeyStorePassword=<KeyStore password provide by EA>
jwt_client_appSecretKeyStoreType=jceks
```

5.2. Add Camouflage Dependency

```
<dependency>
  <groupId>com.discover.cpp.security</groupId>
  <artifactId>edge-service-camouflage</artifactId>
  <version>0.30.0</version>
</dependency>
```

5.3. Configure Data Service URL, Cloak Service URL, Meta Data (IIN Ranges) Refresh rate, Data Service Request Headers

(Note: These parameters can be configured in property files or environment variable. Environment variable takes precedence over parameter configured in property file.)

Required Properties

```
#Cloak Service URL without JWT(Pre Prod)
#cloakServiceURL=
https://vcld000927.rw.discoverfinancial.com:18300/CloakService/enterprise/sec
urity/cloak/v1/data #non JWT
```

JWT Enabled Cloak Service URL (Pre Prod)

```
cloakServiceURL=
https://vcld000927.rw.discoverfinancial.com:18400/CloakService/enterprise/sec
urity/cloak/v1/data #JWT
```

iincache.properties

```
iincache.app-name=iin-cache
iincache.iin-data-service-api-uri=https://iin-data-services-devint-bdc-
cldfundry-0.cfd2.discoverfinancial.com/cpcd/reference/iin/pulse
iincache.search-strategy-lookup-range=2
iincache.bucket-granularity=120
 iincache.max-number-buckets=10
```

Optional Properties

```
metadata.refresh.cron.expression=03 24 15 * * * (optional. If not configured
it defaults midnight every day)
cloakService.ConnectionTimeout =5000 (optional. ConnectionTimeout in
milliseconds. If not configured it defaults to 10000 ms.)
```

5.4. Autowire TokenizeAndMetaDataService

```
@Resource(name="tokenizeAndMetaDataService")
private ITokenizeAndMetaDataService tokenizeAndMetaDataService;
```

6. Sample Code

```
public class TokenizeAndMetaDataSampleCode {
   @Resource(name="tokenizeAndMetaDataService")
   private ITokenizeAndMetaDataService tokenizeAndMetaDataService;
          public void test() {
          String[] input = new String[]{"6011123456781234", "6022223456781234",
   "6033333456781234"} ;

    Set<MetaDataAttributeName> metaDataAttributes = new HashSet<>();
    metaDataAttributes.add(MetaDataAttributeName.ALL);

           TokenizeAndMetadataResponseVO[]  tokenizeAndMetadataResponseVO =
   tokenizeAndMetaDataService.tokenize(input, "DE_PAN23" , metaDataAttributes);

          }
   }
```

Appendix A - Request/Response Structure

1. TokenizeAndMetadataResponseVO

Fields Description

Field Name	Required	Data Type	Description
Input	Y	String	This is the value that needs to be tokenized.
Token	Y	String	This is the token that needs to be detokenized.
Map<MetaDataAttributeName, String>	N	Map	This is the Map of Metadata
ErrorVO	N	Object	This object contains error for individual input during bulk tokenization.

2. DetoknizeResponseVO

Fields Description

Field Name	Required	Data Type	Description
Token	Y	String	This is the token that needs to be detokenized.
DetokenizedValue	Y	String	This is the token that needs to be detokenized.
ErrorVO	N	Object	This object contains error for individual input during bulk tokenization.

3. MetaDataAttributeName enum

Fields Description

Field Name	Description
EDIT	EDIT Check Result
MOD10	EDIT Check Result
LENGTH	LENGTH Check Result
PAN_VERIFICATION	PASSED if EDIT, MOD10 and LENGTH checks PASS and FLASE if EDIT, MOD10 or LENGTH check fails
IIN_RESOURCE_ID	IIN Resource ID
IIN_HREF	IIN Href
ALL	Used as input to tokenizeAndMetdata API to retrieve all Metadata fields

4. Error Codes

ERROR Code	Description
001	Unable to tokenize
002	Matching IIN Range not found
003	Unable to tokenize and Matching IIN Range not found
004	Unable to detokenize

5. Data Element Names

Element Name	Description
DE_PAN23	This value is used to tokenize/detokenize PAN numbers. DE_PAN23 is defined to tokenize numeric chars in input. Non-numeric chars are not tokenized. e.g. @Invalid -> @Invalid, 1234ABC567 -> 5678AB234
DE_SSN23	This value is used to tokenize/detokenize SSN
DE_BAN23	Bank Account Number

Appendix B – Configuration Properties

Required Configuration Properties

Property Name	**Description**
cloakServiceURL	Cloak Service URL
dataServiceURL	Data Service URL
dataservice.requestor_id	Requestor Id is allocated by the CPP platform. This will be the requestor’s well-known name.
dataservice.target_id	The well-known name of this service allocated by the CPP platform

Optional Configuration Properties

Property Name	Description	Default
metadata.refresh.cron.expression	Frequency of metadata refresh from Data Service represented as Cron expression	00 00 00 * * * (Refreshed every midnight)
cloakService.ConnectionTimeout	Cloak Service connection timeout in milliseconds	10000 ms

CLOAK Service and Data Service DFS Open API Gateway Configuration Properties

Following properties needs to be set as environment variable to access CLOAK API and Data Service (IIN Metadata).

Environmental Variable Name	Data Type	Required	Description
jwt_client_environment	String	Yes	Define the runtime environment in which the generator runs, e.g. “DEV”, “TSYS”, “PROD”, or “Sandbox”. The client application has to be run in the same environment as its calling services’ environment.
jwt_client_requireDynamicallyUpdate	String	No	Define whether or not an application wants to update the configuration dynamically in runtime. If this property is not provided, by default, it is set to “false”. In general, the configuration information required by the JWT generator are not changed often. It is not necessary to dynamically update the configuration information at runtime. If really needed, set to field to "true". If it is set to “true”, the configuration update task timer configuration information must be provided. This property itself can’t be dynamically changed. It is only used when the WSJWTGenerator instance is initialized. In most of operating systems, when a program or process is started, the environmental variables are loaded into the program/process memory and without

			restarting the program/process, the in-memory environmental variables can't be updated. So by using environmental variables to define WSJWTGenerator configuration information, this property would be set to "false" or not be set. If the environmental variables are able to be updated during program running, the environmental variable jwt_client_config_change_timestamp must be updated to reflect when the configuration changes happened. The timed-up configuration update task checks if the variable provided timestamp is larger than the timestamp of the last time of loading WSJWTGenerator's configuration. If it is, the task reloads the configuration; otherwise, not.
jwt_client_config_change_timestamp	String	No	The string representation of the unix timestamp. The environmental variable specifies when the changes are made to the environmental variables defined for WSJWTGenerator configuration information.
jwt_client_timerInterval	String	No	If property requireDynamicallyUpdate is set to "true", a timer will be created which invokes the configuration update task periodically and this property must be provided. The firing interval of the timer is given by this property. It has to be a string representing a numeric number and is larger than 0 in order to create a timer. This property can't be dynamically changed. It is only used when the WSJWTGenerator instance is initialized. Its value must be larger than 0.
jwt_client_timerIntervalUnit	String	No	If property requireDynamicallyUpdate is set to "true", the property must be provided and its value must be one of the following values: "MINUTES","MILLISECONDS","DAYS","HOURS","MICROSECONDS","SECONDS", and "NANOSECONDS". For example, if

			timerInterval is 10, if timerIntervalUnit is “HOURS”, then the timer firing interval is 10 hours; if timerIntervalUnit is “DAYS”, then the timer firing interval is 10 days. This property can’t be dynamically changed. It is only used when the WSJWTGenerator instance is initialized.
jwt_client_timerShutdownWaitTime	String	No	If property requireDynamicallyUpdate is set to “true”, the property must be provided and its value is measured as milliseconds. Assign the property to a proper value to allow the timer is shutdown gracefully such that the on-going configuration update can be completed without exceptions. It has to be a string representation of a numeric number and is larger than 0. By default, it is set to 1000 (1 second).
jwt_client_outboundProxyHost	String	No	If the generator is required to authenticate the application against an authentication authority (i.e. DFS API gateway) outside its running zone, the outbound proxy host name needs to be provided, e.g. "proxy.discoverfinancial.com". If this property is not provided, https request to DFS API gateway JWT token authentication end point will not use a proxy.
jwt_client_outboundProxyPort	String	No	If the generator is required to authenticate the application against an authentication authority (i.e. DFS API gateway) outside its running zone, the outbound proxy port number needs to be provided, e.g. “8080”. If this property is not provided, https request to DFS API gateway JWT token authentication end point will not use a proxy. It has to be a string representation of a numeric number and is larger than 0.
jwt_client_cacheOffsetOfExpTime	String	Yes	The generator can cache a generated token for reuse. The token cache time must be less than the token life time. The field defines the difference between the

			token life time and the token cache time which must be positive. It has to be a string representation of a numeric number and is larger than 0. For example, if the token life time is “3600” and cacheOffsetOfExpTime is “300”, then the token cache time is 3600-300=3300. Its value is measured as seconds. Its recommended value is 24 hours, i.e. “86400” seconds.
jwt_client_tokenTimeToLive	String	Yes	Defines the generated token life time. It has to be a string representation of a numeric number and is larger than 0. For example, if it is set to “3600”, the generated JWT token will be expired after 1 hour. Its value is measured as seconds. Its recommended value is 36 hours, i.e. “129600” seconds.
jwt_client_continueWithTimeoutCachedToken	String	No	When a cached JWT token reaches its cache interval, a new JWT token needs to be created and replaces the cached token. It may happen that the DFS API gateway is down when renews a JWT token. If this property is set to “true”, when the DFS API gateway is down and the cached token interval is reached but the token has not been expired, the generator will return the out-of-date cached token and the application can continue to use it to invoke service calls until the token is expired; if the token TTL is reached too, then the generator throws exception WSJWTException. If this property is set to “false”, when cache interval of a token is reached and the token can’t be renewed from the DFS API gateway, the generator will throws exception WSJWTException. By default, it is set to “true”.
jwt_client_issuer	String	Yes	Defines who issues the token in the JWT claim. For a client application, it is the name of the application issuing the token, e.g. "Fraud Alerts Service v1 Sandbox Test"

jwt_client_apiKey	String	Yes	Defines the identifier of the application in the JWT claim, i.e. also called subject. The application identifier is assigned when the application is registered to the authentication authority (i.e. DFA API Portal), e.g. "l7xx338e006c524d4285a955f9b8ab7e4f51"
jwt_client_requireGatewayAuthentication	String	No	DFS API gateway is the authentication authority. If the application is required to authenticate to the gateway, the field must be set as "true", otherwise as "false". If the field is set as "false", the generator will only generate a HS256 JWT token by using given issuer, apiKey, api secret, and environment.
jwt_client_gatewayConnectionTimeout	string	No	If property requireGatewayAuthentication is set to "true", the property must be provided and its value must be larger than 0. It has to be a string representation of a numeric number and is larger than 0. By default, it is set to "2000" (2 seconds). It is measured as milliseconds.
jwt_client_gatewayResponseTimeout	String	No	If property requireGatewayAuthentication is set to "true", the property must be provided and its value must be larger than 0. It has to be a string representation of a numeric number and is larger than 0. By default, it is set to "30000" (30 seconds). It is measured as milliseconds.
jwt_client_tokenEndpointURL	String	No	Define JWT token authentication end point. If property requireGatewayAuthentication is set to true, the property must be provided and its value can't be empty string. The end point is provided by DFS API gateway. For example, the end point for development gateway is https://apidev.discoverfinancial.com/dfs/auth/jwt/v1/token. In some case, the URL may contains some characters (e.g. '=') which are interpreted for different special purpose in different OS. To be consistent,

Field Code Changed

			it is required that the environment variable is assigned to the base64 encoded value of the URL.
jwt_client_secret	String	Yes	Define the application secret used to authenticate the client application to DFS API gateway JWT authentication endpoint and to generate a JWT token which is used to invoke APIs. It is provided when the application is registered on DFS API portal. It is required that the environment variable is assigned to the encrypted value of the original application secret encrypted by using ECC encryption utility packaged in CommonConfigReader-R8_CCR_1.3.7.jar file.
jwt_client_consumerAppCertificate	String	Yes	It is another secret the client application must provide in order to be authenticated by DFS API gateway to obtain the JWT token used by making service calls. It is provided when the client application is registered on DFS API Portal. It is required that the environment variable is assigned to the encrypted value of the original application secret encrypted by using ECC encryption utility packaged in CommonConfigReader-R8_CCR_1.3.7.jar file.