## Connectivity Verification via Other Platforms

As it is understood that web service calls actually are made from outside of the SOAP UI tool, once the connectivity testing with the SOAP UI tool is completed, one can go ahead with verifying the North Boarding API using another internal platform other than the SOAP UI one.

First Data must be informed of the result of connectivity tests with North Boarding API using the SOAP UI tool. This will help First Data stay up-to-date with the latest progress of partners and will also expedite the troubleshooting/support process in case there is any technical difficulty or issue involved.

This section provides additional information about how SOAP requests to North Boarding API should be constructed with an aim to assist with the generation of valid SOAP requests from First Data partners.

North Boarding SOAP API was designed to understand many different formats of SOAP requests. However to be considered valid by the web service, SOAP requests must meet defined minimum structural requirements to guarantee a successful integration with North Boarding FACS SOAP API. More details are available in:

* [SOAP Request Header](#_SOAP_Request_Header)
* [SOAP Request Body](#_SOAP_Request_Body)

The link to North Boarding API in UAT is:

<https://lsuat-api.fdportfoliomanager.com/boarding/north/NorthAsBoardingAPIService.svc>

### SOAP Request Header

Note: A SOAP header can leave out the attribute **wsu: Id**, e.g. wsu: Id="UsernameToken-699F92A9FFDD6BF4E7145557039343214" and still be fully understood by North FACS API.

The process to generate a valid SOAP can be in a general sense described as having the steps below:

1. Generate a random nonce value
2. Get current local date time in UTC format
3. Get username token (plain-text) from First Data
4. Get password token (plain-text) from First Data
5. Combine nonce value (plain-text) + created date (UTC) + password token (plain-text) to form a string. Apply SHA-1 encryption on the string. The encryption result is then converted to base-64 string format to obtain the password digest.
6. Send to North FACS API

* username token (plain-text)
* password digest (encrypted)
* nonce value (in base-64 string format)
* created date time (UTC)

Header of SOAP requests to North Boarding FACS API is always required contains a <wsse: UsernameToken> element of which XML Structure can be illustrated as:

|  |
| --- |
| <wsse:UsernameToken>  <wsse:Username>...</wsse:Username>  <wsse:Password Type="..."> ... </wsse:Password>  <wsse:Nonce EncodingType=" ... "> ... </wsse:Nonce>  <wsu:Created> ... </wsu:Created>  </wsse:UsernameToken> |

The four mandatory XML elements (with highlighted values) that must be specified within the <wsse: UsernameToken> element in the SOAP header request are:

**<wsse:Nonce> Encrypted value of a random nonce**

This element has an optional attribute EncodingType which can be used for specifying the encoding type of the nonce. If the attribute isn’t specified then the default Base64Binary encoding is used.

A new unique nonce value should be generated every single time a request is sent to North FACS SOAP API for optimal security. Please be advised that this is only optional and North FACS SOAP API does not require that or check whether the received nonce value is unique or not.

Example:

Clear-text nonce = 1458232109983 → Encoded nonce = MTQ1ODIzMjEwOTk4Mw==

**<wsu:Created> Creation time of the SOAP request**

This element specifies the timestamp when the SOAP request is generated.

The value can be obtained by converting the current local date time to Coordinated Universal Time (UTC).

Example: 3/17/2016 11:28 AM (EST) → 2016-03-17T16:28:29.983Z (UTC)

**<wsse:Username> Plain text userid (API user)**

Userid is provided by First Data.

**<wsse:Password> Hashed value of password of wsse:Username**

This element has attribute Type which is used for specifying the password type being provided. It should always be #PasswordDigest.

How the value for this element is generated can be described by the following pseudo-code:

Password\_Digest = Base64 (SHA-1 (nonce + created + password))

The generated nonce (clear-text value), creation timestamp, and the password (clear-text password) are concatenated into a combination. The combination is digested using the SHA-1 hash algorithm. The digest result is then encoded in Based64 encoding format and the final password digest value is produced.

The steps below demonstrate how the elements within SOAP request header is generated:

1. Generate random nonce: 1458232109983
2. Get Base64 of nonce: 1458232109983 → MTQ1ODIzMjEwOTk4Mw==
3. Get current date time and convert to UTC: 3/17/2016 11:28 AM (EST) → 2016-03-17T16:28:29.983Z (UTC)
4. Receive password: 123456
5. Create inner password: 14582321099832016-03-17T16:28:29.983Z123456
6. Apply SHA1 to inner password: 14582321099832016-03-17T16:28:29.983Z123456 → 3b781af3b3cdffc4286348490b4f42c50f6df221
7. Get Base64 of the SHA1 digest password: 3b781af3b3cdffc4286348490b4f42c50f6df221 → O3ga87PN/8QoY0hJC09CxQ9t8iE=

A valid SOAP request header with the above values would appear as follows;

|  |
| --- |
| <soapenv:Header>  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">  <wsse:UsernameToken>  <wsse:Username>apitest</wsse:Username>  <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordDigest"> O3ga87PN/8QoY0hJC09CxQ9t8iE=</wsse:Password>  <wsse:Nonce EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary">MTQ1ODIzMjEwOTk4Mw==</wsse:Nonce>  <wsu:Created>2016-03-17T16:28:29.983Z</wsu:Created>  </wsse:UsernameToken>  </wsse:Security>  </soapenv:Header> |

Or as follows, without the attribute EncodingType of element wsse:Nonce;

|  |
| --- |
| <soapenv:Header>  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">  <wsse:UsernameToken>  <wsse:Username>apitest</wsse:Username>  <wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordDigest"> O3ga87PN/8QoY0hJC09CxQ9t8iE=</wsse:Password>  <wsse:Nonce>MTQ1ODIzMjEwOTk4Mw==</wsse:Nonce>  <wsu:Created>2016-03-17T16:28:29.983Z</wsu:Created>  </wsse:UsernameToken>  </wsse:Security>  </soapenv:Header> |

### SOAP Request Body

How the SOAP request body is constructed is based on which among the two currently available web methods is called

#### SubmitMPA

The customizable contents are highlighted.

|  |
| --- |
| <soapenv:Body>  <tem:SubmitMPA>  <!--Optional:-->  <tem:xmlContent>  e  <!--You may enter ANY elements at this point-->  gero  </tem:xmlContent>  <!--Optional:-->  <tem:extRefID>?</tem:extRefID>  <!--Optional:-->  <tem:northChannel>?</tem:northChannel>  </tem:SubmitMPA>  </soapenv:Body> |

#### GetStatusByTimeSpan

|  |
| --- |
| <soapenv:Body>  <tem:GetStatusByTimeSpan>  <tem:fromTimeStamp> 2016-02-24T11:00:00Z</tem:fromTimeStamp>  <tem:toTimeStamp> 2016-02-24T19:00:00Z</tem:toTimeStamp>  </tem:GetStatusByTimeSpan>  </soapenv:Body> |

## Common Connectivity Issues

### Common issue #1: Generic Error

#### Indicator

A response as seen below is returned by North FACS API after the submission of a request;

|  |
| --- |
| <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">  <s:Body>  <s:Fault>  <faultcode>s:Client</faultcode>  <faultstring xml:lang="en-US">Generic error occurred.</faultstring>  </s:Fault>  </s:Body>  </s:Envelope> |

#### Cause

There are many possible causes to this particular type of error, among which the most common ones are:

* If it is a request to the web method GetStatusByTimeSpan, the provided dates are not in correct format.
* Either username token or password token in the request header is not correct.
* The element wsu:Created has expired. It must be re-generated at the time before the request is sent. A large delay would cause North FACS API to consider the provided wsu:Created expired.

#### Actions

1. Double-check username and password
2. Log a request for issue investigation to First Data. The request must detail:

* When the SOAP request was submitted
* The submitting username

### Common issue #2: Submission failure

#### Indicator

A response comparable to the one in the screenshot below is returned by North FACS API. More details are available in North FACS Boarding API - Basic Verification Guide.

|  |
| --- |
| <SubmitMPAResult xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  <ExternalRefId xmlns="http://tempuri.org/">0000001</ExternalRefId>  <Timestamp xmlns="http://tempuri.org/">02/16/2016 05:16:05 PM</Timestamp>  <Status xmlns="http://tempuri.org/">FAILURE</Status>  <Errors xmlns="http://tempuri.org/">  <MerchantError>  <ErrorId>1</ErrorId>  <ErrorDescription>MPA Info - Locations field does not reflect the number of locations added.</ErrorDescription>  </MerchantError>  </Errors>  </SubmitMPAResult> |

#### Cause

North Boarding API was implemented with numerous validation rules to prevent false MPA data from being entered into the First Data system.

#### Actions

If you are not aware of the existence of the validation rule or the error message(s) returned are not informative enough, please contact First Data for further support.

The support request must provide the following information:

* The username the MPA was submitted with
* The XML content of the MPA

### Common issue #3: Submission failure without specific errors

#### Indicator

A response comparable to the one in the screenshot below is returned by North FACS API. More details are available in [North FACS Boarding API - Basic Verification Guide.](#_Basic_Verification_Guides)

|  |
| --- |
| <SubmitMPAResult xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  <ExternalRefId xmlns="http://tempuri.org/">0000001</ExternalRefId>  <Timestamp xmlns="http://tempuri.org/">02/16/2016 05:19:12 PM</Timestamp>  <Status xmlns="http://tempuri.org/">FAILURE</Status>  <Errors xmlns="http://tempuri.org/" />  </SubmitMPAResult> |

#### Cause

As there should have been a clear error message indicating which validation rule was violated that led to the submission failure, this is likely an unknown defect of North Boarding FACS API.

#### Actions

Send a support request to First Data with information about:

* The username the MPA was submitted with