IBM® Kenexa® BrassRing on Cloud

**Candidate Export Technical Specifications API Guide**

*Version 1.2*

*Release Date: November 11, 2015*

Edition Notice

**Note:** Before using this information and the product it supports, read the information in the *Notices* section at the end of this document.

This edition applies to IBM® Kenexa® BrassRing on Cloud and to all subsequent releases and modifications until otherwise indication in new editions.

Licensed Materials - Property of IBM

**© Copyright IBM® Corporation, 2014.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

|  |
| --- |
| **Table of Contents** |

[Edition Notice I](#_Toc435003035)

[Chapter 1: Introduction 4](#_Toc435003036)

[1.1 What are BrassRing’s XML Integration APIs 4](#_Toc435003037)

[1.2 Purpose 4](#_Toc435003038)

[1.3 Audience 4](#_Toc435003039)

[Chapter 2: BrassRing’s Candidate Export Integration APIs 5](#_Toc435003040)

[2.1 Overview 5](#_Toc435003041)

[2.2 Communication Methodology 5](#_Toc435003042)

[2.3 Rules for the Candidate Export Integration 6](#_Toc435003043)

[2.4 Candidate Export Snapshot 6](#_Toc435003044)

[2.4.1 Figure 1: Trigger Page 6](#_Toc435003045)

[2.4.2 Figure 2: Success Page 7](#_Toc435003046)

[2.4.3 Figure 3: Form Import 7](#_Toc435003047)

[2.5 Candidate XML Definitions (Data-mapping) 8](#_Toc435003048)

[2.5.1 Mandatory Fields 8](#_Toc435003049)

[2.5.2 Custom Fields defined by Customer (examples below) 8](#_Toc435003050)

[2.6 Candidate Tag Definitions – Detailed Descriptions 9](#_Toc435003051)

[2.6.1 Mandatory Fields 9](#_Toc435003052)

[Chapter 3: XML Schemas 10](#_Toc435003053)

[3.1 Sample XML Export Request (Candidate data only) 10](#_Toc435003054)

[3.2 Sample XML export Request (Candidate data and Requisition data) 11](#_Toc435003055)

[3.3 Sample Synch XML Response (Success) 12](#_Toc435003056)

[3.3.1 Example 1: Standard Response 12](#_Toc435003057)

[3.3.2 Example 2: Redirect Response 13](#_Toc435003058)

[3.4 Sample XML Response (Failure) 14](#_Toc435003059)

[Chapter 4: Envelope Processing 15](#_Toc435003060)

[4.1 Sender 15](#_Toc435003061)

[4.2 Recipient 15](#_Toc435003062)

[4.3 TransactInfo 15](#_Toc435003063)

[4.4 Packet 15](#_Toc435003064)

[4.5 Payload 16](#_Toc435003065)

[Chapter 5: Security 17](#_Toc435003066)

[5.1 Standard Implementation Security 17](#_Toc435003067)

[5.1.1 Transport Layer Security (TLS) 17](#_Toc435003068)

[5.1.2 IP restrictions 17](#_Toc435003069)

[5.2 Optional Security Features 17](#_Toc435003070)

[5.2.1 Message Layer Encryption/Security (MLS) 17](#_Toc435003071)

[5.2.2 2-Way SSL / Certificate Based Authentication 18](#_Toc435003072)

[5.2.3 Basic Authentication 18](#_Toc435003073)

[Chapter 6: API Error Messages 19](#_Toc435003074)

[Notices 20](#_Toc435003075)

[Trademarks 21](#_Toc435003076)

1. Introduction
   1. What are BrassRing’s XML Integration APIs

BrassRing XML integrations feature flexible, secure, real-time integrations by sending XML over the Internet using web services or HTTP Posts. XML formats used in BrassRing XML integrations are based on HR-XML Consortium guidelines.

The BrassRing XML integrations offering allows BrassRing to work with the customers to configure multi-level, robust integration relationships between BrassRing and any systems that can send and receive XML. Examples of these systems include HRIS systems such as SAP/Oracle/PeopleSoft, etc., and systems that perform HR related services like background checks, assessments or compensation analysis.

The customer can choose to receive data through either a web service or HTTP post. BrassRing handles the exchange of XML requests and responses using its own messaging software. The customer builds their own messaging system to exchange XML web service messages or HTTP post with BrassRing.

*Note: Web Service exports require special PSE configuration. Self-Service exports configured in the Mapping Tool only support HTTP Post.*

* 1. Purpose

The purpose of this document is as follows:

To present a high-level architecture and design of BrassRing APIs and how BrassRing implements Candidate export integrations to meet its Customer’s needs to candidate data exchange.

To describe suggested high-level workflow processes that can be supported between BrassRing and any external, third-party system that can send and receive XML requests and responses.

To describe the steps required to implement Candidate/New Hire Export integration.

* 1. Audience

Client Decision Makers, HRIS Implementation Teams, Internal IT Teams, Systems Integrators and Support Teams.

BrassRing Engineering Services Team, Support Team, and Technical Services Group.

1. BrassRing’s Candidate Export Integration APIs
   1. Overview

Candidate export Integrations from BrassRing to an external system (Customer or a third-party vendor) is referred in many ways namely:

New Hire export

Candidate export for Assessments

Candidate export for Background check

Candidate data is any data that applies to a candidate such as name, address, etc. and any information that resides on a candidate or Req form in the BrassRing system.

These integrations are triggered real-time in BrassRing meaning Recruiters initiate the export by moving a candidate to a configured triggering HR Status which initiates the transaction of the data across. For example, a customer may wish to export candidate data each time a Recruiter moves a candidate to an “Offer Accepted” or “Initiate Background Check” HR Status.

This integration can be implemented as a one-way integration or a two-way integration.

**Unidirectional or One-way integration** – Candidate data flows one way from BrassRing to external system

**Bidirectional or Two-way integration** – Candidate data flows one way from BrassRing to external system and information such as assessment results, background check results or any candidate form data are sent from external system to BrassRing which then get attached to the candidate in BrassRing on a candidate form. Please refer to the BrassRing Form Import Technical Specifications API.docx documentation on details of bringing back results into BrassRing.

The data required to be transacted is defined in the data mapping document process.

* 1. Communication Methodology

Connection to the BrassRing system is accomplished via 2 mechanisms using XML as a data transfer vehicle.

HTTP Post

Web services

Both the above mechanisms are synchronous meaning all transactions are real-time with a request-response messaging and no message queuing.

The XML format is depicted in Section 3.

*Note: Web Service exports require special PSE configuration. Self-Service exports configured in the Mapping Tool only support HTTP Post.*

* 1. Rules for the Candidate Export Integration

Below are a few rules to consider when looking into the use of the Integration:

The XML schema will be provided by BrassRing and cannot be customized (see sample XML below).

Transactions are processed real-time via HTTPS Post or Web Services.

Field requirements (outlined in the Data Mapping sections).

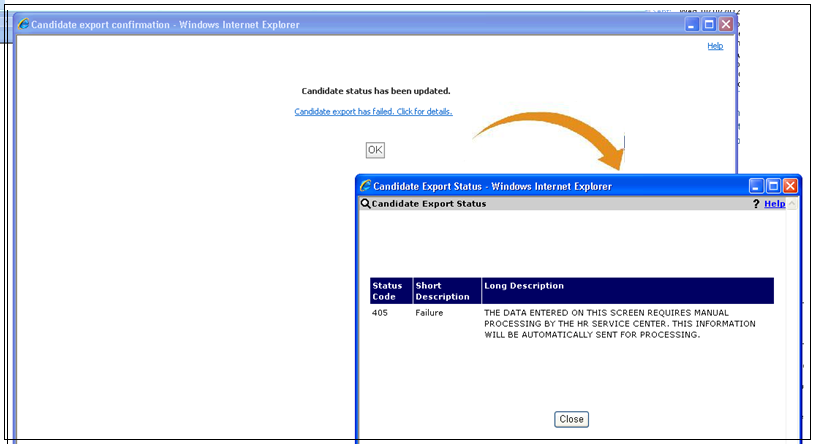
*Note: Web Service exports require special PSE configuration. Self-Service exports configured in the Mapping Tool only support HTTP Post.*

* 1. Candidate Export Snapshot

Recruiters in the system trigger the export. Below is a screen print of the HR Status screen for your reference.

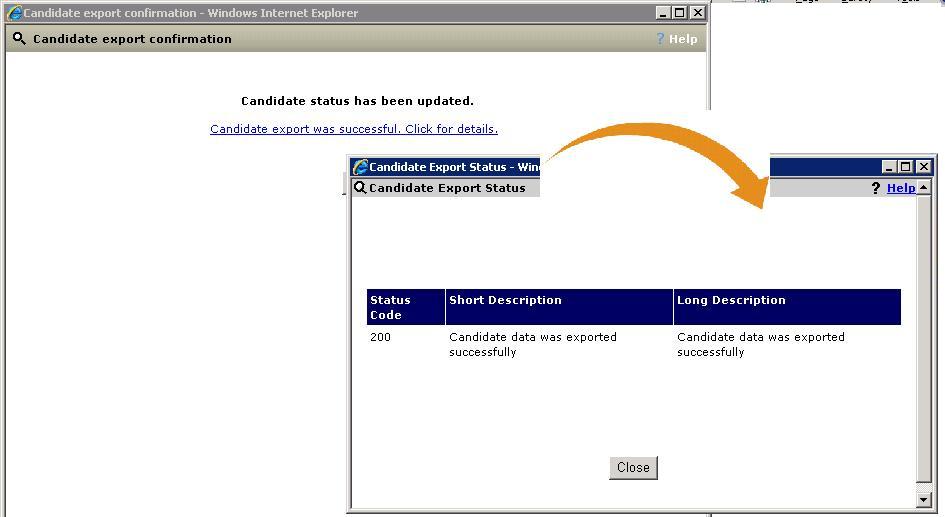
* + 1. Figure 1: Trigger Page

Triggering page and page displayed to the Recruiter when the export is unsuccessful.



* + 1. Figure 2: Success Page

Page displayed to the Recruiter when the export is successful.



* + 1. Figure 3: Form Import

Candidate results screen (in case of bi-directional integrations) . This is the screenshot of the Form when Form results are sent back as part of the Form Import (ILOAD) integrations.



* 1. Candidate XML Definitions (Data-mapping)
     1. Mandatory Fields

| Field Name | Format | Field Length | Form Name | XML Tag |
| --- | --- | --- | --- | --- |
| CANDIDATEID | Integer | 15 | This is the unique identifier of the Candidate in the BrassRing system | CANDIDATEID |
| REQUISITIONNUMBER | Varchar | 30 | This is the Client Req number also referred as the Optional Req Number. The value will be blank if client does not use it | REQUISITIONNUMBER |
| BRREQNUMBER | Varchar | 30 | This is the unique identifier of the Job in BrassRing. It is always a value that ends with “BR”, for example 100 BR, 3423BR, etc. | BRREQNUMBER |
| JOBCODE | Varchar | 35 | This is the standard job code associated with the Job | JOBCODE |
| STATUS | Varchar | 50 | This is the HR Status value that triggered the candidate/new hire integration | STATUS |

* + 1. Custom Fields defined by Customer (examples below)

| Field Name | Format | Field Length | Form Name | XML Tag |
| --- | --- | --- | --- | --- |
| Start Date | Varchar | 255 | Offer Form | XML tags for Custom fields are provided after data mapping requirements are provided for the integration |
| Employee ID | Varchar | 50 | Offer Form |
| Race | Varchar | 255 | EEO Form |
| Gender | Varchar | 255 | EEO Form |
| Recruiter | Varchar | 255 | Requisition Form |
| Job title | Varchar | 255 | Requisition Form |
| Department | Varchar | 100 | Requisition Form |

* 1. Candidate Tag Definitions – Detailed Descriptions
     1. Mandatory Fields

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| * + - 1. CANDIDATEID   The value in this tag contains the candidate’s unique number in BrassRing   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **XML Tag** | **Tag Present?** | **Value Present?** | **Format** | **Length** | | CANDIDATEID | Yes | Yes | Integer | 15 | |
| * + - 1. REQUISITIONNUMBER   The value in this tag contains the customer’s Req Number also referred as the Optional Req number in BrassRing   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **XML Tag** | **Tag Present?** | **Value Present?** | **Format** | **Length** | | REQUISITIONNUMER | Yes | Optional | Varchar | 30 |   \_ |
| * + - 1. BRREQNUMBER   The value in this tag contains the BrassRing generated Req number. This value is unique in BrassRing and always in “BR”. (i.e. 100BR).   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **XML Tag** | **Tag Present?** | **Value Present?** | **Format** | **Length** | | BRREQNUMBER | Yes | Yes | Varchar | 30 |   \_ |
| * + - 1. JOBCODE   The value in this tag contains the job code value association with the Requisition.   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **XML Tag** | **Tag Present?** | **Value Present?** | **Format** | **Length** | | JOBOCDE | Yes | Optional | Varchar | 35 |   \_ |
| * + - 1. STATUS   The value in this tag contains the HR Status name that triggered the export.   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **XML Tag** | **Tag Present?** | **Value Present?** | **Format** | **Length** | | STATUS | Yes | Yes | Varchar | 50 |   \_ |

1. XML Schemas
   1. Sample XML Export Request (Candidate data only)

Below is a sample Candidate XML file. The items highlighted in yellow are mandatory fields that are exported. Other fields are listed as an example and driven by the data mapping document requirements on the fields to be exported. Candidate data only represents all information that resides on Candidate forms in BrassRing.

<?xml version="1.0" encoding="UTF-8"?>

<Envelope version="01.00">

<Sender>

<!—The Id tag will always be included. It will hold the employee ID of the BrassRing user who is initiating the request. If the user has no employee ID, the Id tag will be exported with no value. -->

<Id/>

<Credential>BrassRing will provide</Credential>

</Sender>

<Recipient>

<Id/>

</Recipient>

<TransactInfo transactType="data">

<TransactId>HSCAND636096</TransactId>

<TimeStamp>2/28/2011 5:34:34 PM</TimeStamp>

</TransactInfo>

<Packet>

<PacketInfo packetType="data">

<PacketId>1</PacketId>

<Action>SET</Action>

<Manifest>MANIFEST\_NAME</Manifest>

</PacketInfo>

<Payload><![CDATA[<?xml version="1.0"?>

<CANDIDATE>

<CANDIDATEID>984577</CANDIDATEID>

<REQUISITIONNUMBER/>

<BRREQNUMBER>100BR</BRREQNUMBER>

<JOBCODE>09845145447</JOBCODE>

<STATUS>03-Offer Accepted</STATUS>

<CANDIDATEFIRSTNAME></CANDIDATEFIRSTNAME>

<CANDIDATELASTNAME></CANDIDATELASTNAME>

<DATEOFBIRTH>1977-02-15</DATEOFBIRTH>

<BIRTHPLACE>Boston</BIRTHPLACE>

<COUNTRYOFBIRTH>USA</COUNTRYOFBIRTH>

<GENDER>M</GENDER>

<MARITALSTATUS>Single</MARITALSTATUS>

<NATIONALITY>American</NATIONALITY>

<LANGUAGENAME>EN</LANGUAGENAME>

<ACTION\_REASON>Addition</ACTION\_REASON>

<ADDITIONAL\_COMPENSATION>0</ADDITIONAL\_COMPENSATION>

<ADDRESS1>4719 Fetteressa Drive</ADDRESS1>

<ADDRESS2>none</ADDRESS2>

<BASE\_SALARY\_FREQUENCY>H</BASE\_SALARY\_FREQUENCY>

<SALARY\_GRADE>OL6</SALARY\_GRADE>

<SIGN\_ON\_BONUS\_AMOUNT>0</SIGN\_ON\_BONUS\_AMOUNT>

<ETHNIC\_GROUP>White</ETHNIC\_GROUP>

<HIRE\_DT>2012-04-23</HIRE\_DT>

</CANDIDATE>

]]></Payload>

</Packet>

</Envelope>

* 1. Sample XML export Request (Candidate data and Requisition data)

Below is a sample Candidate XML file. The items highlighted in yellow are mandatory fields that are exported. Other fields are listed as an example and driven by the data mapping document requirements on the fields to be exported. Candidate data only represents all information that resides on Candidate forms in BrassRing. Requisition data represents information that reside on the Requisition forms in BrassRing and appears in a second Packet node in the XML.

<?xml version="1.0" encoding="UTF-8"?>

<Envelope version="01.00">

<Sender>

<Id>BrassRing will provide</Id>

<Credential>BrassRing will provide</Credential>

</Sender>

<Recipient>

<Id/>

</Recipient>

<TransactInfo transactType="data">

<TransactId>HSCAND636096</TransactId>

<TimeStamp>2/28/2011 5:34:34 PM</TimeStamp>

</TransactInfo>

<Packet>

<PacketInfo packetType="data">

<PacketId>1</PacketId>

<Action>SET</Action>

<Manifest>MANIFEST\_NAME</Manifest>

</PacketInfo>

<Payload><![CDATA[<?xml version="1.0"?>

<CANDIDATE>

<CANDIDATEID>984577</CANDIDATEID>

<REQUISITIONNUMBER/>

<BRREQNUMBER>100BR</BRREQNUMBER>

<JOBCODE>09845145447</JOBCODE>

<STATUS>03-Offer Accepted</STATUS>

<CANDIDATEFIRSTNAME></CANDIDATEFIRSTNAME>

<CANDIDATELASTNAME></CANDIDATELASTNAME>

<DATEOFBIRTH>1977-02-15</DATEOFBIRTH>

<BIRTHPLACE>Boston</BIRTHPLACE>

<COUNTRYOFBIRTH>USA</COUNTRYOFBIRTH>

<GENDER>M</GENDER>

<MARITALSTATUS>Single</MARITALSTATUS>

<NATIONALITY>American</NATIONALITY>

<LANGUAGENAME>EN</LANGUAGENAME>

</CANDIDATE>

]]></Payload>

</Packet>

<Packet>

<PacketInfo packetType="data">

<PacketId>2</PacketId>

<Action>SET</Action>

<Manifest>MANIFEST\_NAME</Manifest>

</PacketInfo>

<Payload><![CDATA[<REQUISITION>

<JOBCODE>09845145447</JOBCODE>

<SAPPOSITION>Manager of Operations</SAPPOSITION>

<COMPANYCODE>IBM</COMPANYCODE>

<JOBKEY>RD-54123</JOBKEY>

<ORGUNIT>Hosting</ORGUNIT>

<PERSONNELAREA>PA\_784</PERSONNELAREA>

<EMPLOYEEGROUP>Permanent</EMPLOYEEGROUP>

<EMPLOYEESUBGROUP>Labor</EMPLOYEESUBGROUP>

<BUSINESSAREA>Talent Management</BUSINESSAREA>

</REQUISITION> ]]></Payload>

</Packet>

</Envelope>

* 1. Sample Synch XML Response (Success)

Below is a sample synchronous success Response XML that must be sent back by the client’s URL on the Request. Code value of 200 indicates that the transaction was successfully received by the Client’s URL.

The synchronous response can be sent in 1 of 2 ways.

* + 1. Example 1: Standard Response

Below is how 90% of our client have implemented. It the standard response that acknowledges receipt of the XML and a successful message is displayed on the screen to the user who triggered the export.

<?xml version="1.0" encoding="UTF-8"?>

<Envelope version="01.00">

<Sender>

<Id>BrassRing will provide</Id>

<Credential>BrassRing will provide</Credential>

</Sender>

<Packet>

<PacketInfo packetType="response">

<PacketId>1</PacketId>

<Action>SET</Action>

<Manifest>MANIFEST\_NAME</Manifest>

<Status>

<Code>200</Code>

<ShortDescription>Candidate data was exported successfully</ShortDescription>

<LongDescription>Candidate data was exported successfully</LongDescription>

</Status>

</PacketInfo>

<Payload/>

</Packet>

<Recipient>

<Id/>

</Recipient>

<TransactInfo transactType="response">

<TransactId>11082520</TransactId>

<TimeStamp>11/23/2009 08:54:28 AM PST</TimeStamp>

<Status>

<Code>200</Code>

<ShortDescription>Candidate data was exported successfully</ShortDescription>

<LongDescription>Candidate data was exported successfully</LongDescription>

</Status>

</TransactInfo>

</Envelope>

* + 1. Example 2: Redirect Response

The below example adds an additional tag called RedirectURL. Including this tag with a valid URL in the Response will automatically redirect the user to the designated URL provided upon a successful transaction. You may choose such an option when you require the user to be automatically redirected into some other system upon a successful transaction of the candidate data from BrassRing. An example of such a scenario would be to redirect the user to an external on-boarding system after data transfer for the candidate data was successful.

<?xml version="1.0" encoding="UTF-8"?>

<Envelope version="01.00">

<Sender>

<Id>BrassRing will provide</Id>

<Credential>BrassRing will provide</Credential>

</Sender>

<Recipient>

<Id>3054485</Id>

</Recipient>

<TransactInfo transactType="response">

<TransactId>2193791</TransactId>

<TimeStamp>11/10/2011 12:25:22 PM</TimeStamp>

<Status>

<Code>200</Code>

<ShortDescription>Success</ShortDescription>

<LongDescription>Candidate data transfer was successful</LongDescription>

<RedirectURL>https://mydomain.com/showmessage.aspx?token=634e9e5ed8749c19</RedirectURL>

</Status>

</TransactInfo>

<Packet>

<PacketInfo packetType="response">

<PacketId>1</PacketId>

<Action>SET</Action>

<Manifest>DGTALX\_CANDIDATE\_EXPORT</Manifest>

<Status>

<Code>200</Code>

<ShortDescription>Success</ShortDescription>

<LongDescription>Candidate data transfer was successful</LongDescription>

<RedirectURL>https://mydomain.com/showmessage.aspx?token=634e9e5ed8749c19</RedirectURL>

</Status>

</PacketInfo>

<Payload/>

</Packet>

</Envelope>

* 1. Sample XML Response (Failure)

Below is a sample synchronous failure Response XML that must be sent back on the Request in the event an error is encountered. Code value of 405 indicates that the transaction was unsuccessfully processed by the Client’s URL.

<?xml version="1.0" encoding="UTF-8"?>

<Envelope version="01.00">

<Sender>

<Id>BrassRing will provide</Id>

<Credential>BrassRing will provide</Credential>

</Sender>

<Packet>

<PacketInfo packetType="response">

<PacketId>1</PacketId>

<Action>SET</Action>

<Manifest>MANIFEST\_NAME</Manifest>

<Status>

<Code>405</Code>

<ShortDescription>Background Request Submission Not Successful</ShortDescription>

<LongDescription>Missing Required: Applicant Region., Missing Required: Applicant SSN, Missing Required: Applicant Date of Birth., Missing Required: Conviction Question. </LongDescription>

</Status>

</PacketInfo>

<Payload/>

</Packet>

<Recipient>

<Id/>

</Recipient>

<TransactInfo transactType="response">

<TransactId/>

<TimeStamp>11/23/2009 03:54:19 PM PST</TimeStamp>

<Status>

<Code>405</Code>

<ShortDescription>Background Request Submission Not Successful</ShortDescription>

<LongDescription>Missing Required: Applicant Region., Missing Required: Applicant SSN, Missing Required: Applicant Date of Birth., Missing Required: Conviction Question. </LongDescription>

</Status>

</TransactInfo>

</Envelope>

1. Envelope Processing

The top level Envelope contains two attributes, version and encoding.

**Version** attribute identifies the version of the envelope. The only valid value at this time is “01.00”.

**Encoding** attribute identifies XML encoding and is optional. It can be set to “UTF-8” or “UTF-16”. The recommended setting is “UTF-16”.

The Envelope also has four top-level **mandatory** elements (Sender, Recipient, TransactInfo and Packet). These elements are described in detail below.

* 1. Sender

The values below will be provided at the time of implementation.

**Employee Id:** The **Id** tag will always be included. It will hold the employee ID of the BrassRing user who is initiating the request. If the user has no employee ID, the Id tag will be exported with no value

**Client Id :** Sent in the **Credential** tag.

* 1. Recipient

The URL of the recipient is passed in the Id tag. This helps to identify whether the data has been sent to the correct recipient destination.

* 1. TransactInfo

An identifier for the transaction should be included in the **TransactId**. This identifies the envelope transaction and not any of the constituent data, since there could be multiple documents within one envelope. The combination of the client Id, TransactId and PacketId creates a globally unique identifier for each packet.

A timestamp can be added in the optional **TimeStamp**. This date format should conform to ISO standard 8601 (i.e. “2000-10-09T14:14:11Z”)

**transactType** should be set to “data”. This attribute helps provide the envelope parser with a heads up on what to expect in the payloads.

* 1. Packet

The **Packet** contains the payload and metadata about the **payload**. The Payload element contains the actual XML and the **PacketInfo** element contains the metadata about the packet.

**PacketInfo** contains the following elements:

**packetType** This attribute describes the content of the packet. A packetType of data, the default.

**PacketId**: This element uniquely identifies the contents of the payload. PacketId is 1 unless suggested otherwise

**Action**: This optional element contains the transaction code – SET

**Manifest**: This is a mandatory element that identifies the contents of the payload. This field must be set to the value provided during implementation and should not be changed.

* 1. Payload

The **Payload** element contains one HR-XML document inside a CDATA section. This approach allows the payload to be parsed separately from the envelope and avoids any interdependency between envelope and contents when validating envelope documents.

1. Security

Several security protocols and measures have been implemented to ensure security on the data transmission, message integrity, and message confidentiality. These mechanisms can be used to accommodate a wide variety of security models and encryption technologies.

* 1. Standard Implementation Security
     1. Transport Layer Security (TLS)

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that provide communication security over the Internet. TLS and SSL encrypt the segments of network connections above the Transport Layer, using asymmetric cryptography for key exchange, symmetric encryption for privacy, and message authentication codes for message integrity.

BrassRing integration API URLs (web service and HTTP Post URLs) are HTTPS-enabled and utilize 128 bit encryption over SSL when exchanging and transmitting information over the Internet. This is a mandatory encryption of the data that is enforced.

* + 1. IP restrictions

To add another layer to security and protection, IP restrictions can also be enforced. At the time of implementation, IP addresses from Staging and Production environments can be provided that can be white-listed on an external firewall on the customer’s network. These IP addresses are provided on request.

* 1. Optional Security Features
     1. Message Layer Encryption/Security (MLS)

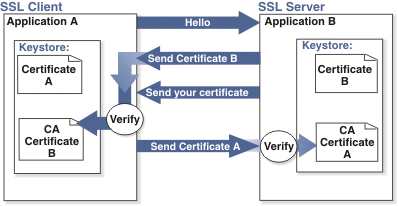
In additional to the above security protocols, message layer encryption is also available on the payload data.

Message layer encryption is facilitated using RSA encryption on the XML. Both Java and Microsoft .NET development environments support APIs for RSA encryption. BrassRing uses 1024-bit RSA encryption key pairs along with a public key generated by the client to implement functions to encrypt, sign, decrypt and authenticate data.

Adding MLS adds to the cost and timeline of the project as this is not part of standard implementations.

* + 1. 2-Way SSL / Certificate Based Authentication

Additionally, client certificate authentications or 2-way SSL or mutual SSL authentication is also available to enforce additional security around the integrations. The method of data exchange is accomplished with the use of certificates that confirm the source of the data before processing the information. SSL Session negotiated by Handshake Protocol via a X.509 public key Certificate of Peer.



Adding 2-way SSL adds to the cost and timeline to the project as this is not part of Standard implementations.

**Note**: 2-way SSL is currently not available to clients on the China data-center.

* + 1. Basic Authentication

In the context of an HTTP transaction, basic access authentication is a method for a HTTP Post URL to provide a user name and password when making a request. When the user agent wants to send the server authentication credentials it will make use of the Authorization.

This username and password will be used for authentication while sending data over to the client.  Upon successful validation of the credentials, we will transfer the candidate export data to the client.

Username and password be encoded in base 64 for transmission.

Note: Basic Authentication is not the same as Web service authentication (WSA). Basic authentication applies on HTTP Post URLs whereas WSA applies on web services. BrassRing infrastructure on candidate export integrations does not currently support WSA.

1. API Error Messages

Below is a list of error messages that you may encounter when using the Candidate export integration. Please refer to the explanation below in order to assist you with troubleshooting.

|  |  |  |
| --- | --- | --- |
| **Code** | **Error Description** | **Reasoning** |
| **401** | ValidateAccess: - Access not setup for Sender – DSDSDS | Value in the Sender tag is incorrect |
| **401** | Credential value is invalid | Value in the Credential tag in incorrect |
| **405** | getProfileInfo: - Inactive subscription | Value in the Manifest tag is incorrect |
| **405** | Error While trying to post :The remote server returned an error: (405) Method Not Allowed | The customer’s HTTP URL or the web service URL is not available or not responding |
| **405** | \*\*\*Custom error messages can be sent in the client’s Response that can be displayed to the end user | \*\*\* |

*NOTE: This list is not all inclusive but contains the errors that are most encountered.*

Notices

This information was developed for products and services offered in the U.S.A and other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan Ltd.  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
5 Technology Park Drive  
Westford Technology Park  
Westford, MA 01886

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only. All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

These terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

* IBM
* AIX
* Sametime
* WebSphere

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.