IBM® Kenexa® BrassRing® on Cloud

BrassRing Object API
GetCandidate

Release Date: February, 2016
Edition Notice

Note: Before using this information and the product it supports, read the information in Notices.

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

Licensed Materials - Property of IBM

© Copyright IBM® Corporation, 2016.

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

Edition Notice .................................................................................................................. ii
Notices ................................................................................................................................. iii

Contents .................................................................................................................................. v

Introduction .............................................................................................................................. 6
Purpose ........................................................................................................................................ 6
Audience ...................................................................................................................................... 6
Privileges .................................................................................................................................... 6
Terms Used in this Guide .......................................................................................................... 6

What is the BrassRing Object API? .......................................................................................... 6

How Do Object APIs in the BrassRing Work? ........................................................................... 7
Restriction of Objects ............................................................................................................. 7
BrassRing Object Oriented APIs .............................................................................................. 7

Using XML .................................................................................................................................. 8
XML Syntax Format Guidelines .............................................................................................. 8
Special Conditions .................................................................................................................. 8
Encoding Standard .................................................................................................................. 8
CDATA Input ............................................................................................................................ 8
Security ....................................................................................................................................... 8
Certificate based encryption ..................................................................................................... 9

BrassRing API Library – Supported API Calls ......................................................................... 9

Order of Object during Requests and Responses ...................................................................... 9

Executing the GetCandidate API Call ..................................................................................... 9
Configuring Workbench Subscriptions .................................................................................... 10

Use Case ..................................................................................................................................... 12

GetCandidate Use Case HR Status Workflow Configuration ..................................................... 13
Use Case – Step One - Configuring HR Status Candidate Export .............................................. 13
Use Case – Step Two ................................................................................................................ 14
Use Case – Step Three ............................................................................................................. 15
Use Case – Step Four ............................................................................................................... 15

Configuring Workbench Subscriptions ................................................................................... 16

GetCandidate Request (Candidate Object) ................................................................................ 17
GetCandidate Response (Candidate Object) .............................................................................. 18

XML Element Table for BrassRing API Library ........................................................................ 22

Error Codes ............................................................................................................................... 24

Language/Site/LocaleIDs .......................................................................................................... 26
Introduction

This document provides information about IBM® Kenexa® BrassRing® on Cloud web service Application Programming Interfaces (APIs) for the BrassRing Object Oriented API workflow developed to provide a standardized workflow for the exchange of information between the IBM Kenexa BrassRing® (BR) application and client applications.

Purpose

The purpose of this document is to present an overview of the BrassRing GetCandidate Object Oriented API. These APIs can be used to integrate with third-party vendors. This API integration allows BrassRing clients to export their configuration and candidate data into Human Resource Information Systems (HRIS) or other 3rd party systems.

This document provides:
- An overview of BrassRing’s GetCandidate API.
- An overview of the supported workflow process, including information exchange with any external, third-party system that can send and receive XML requests and responses.
- A procedure for consuming the GetCandidate API and xml schemas

Audience

The intended audience for this document is:
- Engineering Services Team, Support Team, System Integrators, Technical Services Group, and Vendors.

Privileges

Power Users, Partner Users, Tech Services, and Super Users can access Workbench Subscription configuration to restrict the objects returned in the API call. Your Support Team can assist you with assigning your privileged users.

Terms Used in this Guide

The following terms are used in this guide. Some terms have been abbreviated for comprehension. Abbreviated terms are identified once.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>XML</td>
<td>extensible Markup Language</td>
</tr>
<tr>
<td>Clients</td>
<td>third-party vendors or client HRIS</td>
</tr>
</tbody>
</table>

What is the BrassRing Object API?

The BrassRing Object-Oriented (OO) API exchanges real-time data between the BrassRing application and clients, vendors, and client HRIS. For the purposes of this document, the term client is intended to be used interchangeably with vendors.

The BrassRing API supports an OO workflow allowing clients to retrieve data objects associated with their BrassRing accounts and configured in BrassRing Workbench. The OO API workflow allows for data categorization of objects and their associated attributes. For example, when the object oriented API requests the Candidates data object and no other data subsets are noted, the API response includes all of the Candidate data attributes that include email address, candidate full name, status (Active or Inactive) and date of last edit. See GetCandidate Response. (Candidate Objects).

The OO workflow can also be restricted and thereby limited to a subset of the retrieved object data. For example, if a client wants to request a candidate's HR Status alone, the client can restrict the OO response to return only those...
objects and attributes required. Workbench configuration permits clients to limit (restrict) the scope of the data (objects) returned in the OO response. For more information, please see Configuring Workbench Subscriptions and Restrictions of Objects.

How Do Object APIs in the BrassRing Work?

BrassRing APIs use eXtensible Markup Language (XML) messages to communicate. XML messages contain a defined set of elements for each specific API function. At a high level, these XML elements are the content containers and can contain configuration instructions that indicate, among other things, whether the XML message is a request or a response. At a more detailed level, the elements in the XML instruct the application on exactly what values to retrieve and display, and can contain authentication codes, encryption values and other necessary information. Refer to Using XML for guidelines.

Because the BrassRing API uses object oriented APIs, an API call can retrieve an "object" and all of its associated object attributes from the BrassRing application. If filtering restrictions are configured in Workbench, the API call returns only a subset of the object's attributes.

For example, when the API call GetCandidate initiates a request for an object named “Req Templates,” the API responds and returns the object, Req Templates and all of that object's attributes. These object attributes are common BrassRing Workbench attributes. For more information on object filters and order rules, refer to Order of Objects during Requests and Responses.

Restriction of Objects

When the API calls executes, the GetCandidate request can return up to 10,000 objects and their attributes. Clients have the option to restrict the object types and filter the object's attributes returned in the API response. Clients can use the Workbench Subscriptions to restrict data objects and the filters to restrict object attributes.

For example, if a client wanted to export only candidates of one requisition or one HR status, they can select these filters in Workbench. If a HR status or requisition id is not sent in the request by client/vendor 3rd party system, the export will fail.

Clients can also determine what fields they want to export by selecting them in the output fields. For example, if they want to export only the candidates’ contact details, they can select those fields in the output fields.

For more information, please see Configuring Workbench Subscriptions.

BrassRing Object Oriented APIs

Each of the BrassRing API calls can request and retrieve up to 10,000 objects including object attributes in one API call. If an API response needs to return more than 10,000 objects (in one API call) to fulfill the API request, the API creates a logical break point between objects at an object attribute level and returns a Globally Unique Identifier (GUID) with the first set of objects. For example, the following is a list of Candidate attributes:

Candidate Attributes
- Resume Key
- Email Address
- Candidate Full Name
- Status (Active or Inactive)
- Date of Last Edit

If a user requests the Candidate object and the 10,000 limit occurs within the Candidate "Type" attribute, the API response creates a logical break point at the Candidate Status attribute level and returns a Globally Unique Identifier (GUID) and all objects and their attributes up to the end of the Global/Member attribute. The subsequent API request containing the GUID identifies where the API response should begin so that the API response begins its response at the "Type" attribute and then returns the remaining Status and Date of Last Edit attributes up to 10,000 items.
Using XML

XML document are structured documents containing, among other things, elements. Each element contains an instruction that supports the processing of API requests or responses. XML documents and elements must be written in a defined (valid) format. Therefore, when using BrassRing APIs you must adhere to all syntax guidelines regarding the XML document format. If your XML is not valid, the following XML Syntax Format Guidelines must be followed when using BrassRing APIs.

- All XML must have closing tags.
- XML markup tags are case sensitive.
- XML elements must be properly nested.
- XML attributes must be quoted.
- XML must use encoding if XML contains special characters ($quot; represents a quotation mark) if you are not using CDATA.
- XML comments must be enclosed with <! Comment -->.
- XML element naming conventions:
  o Names cannot start with a number or punctuation character.
  o Names cannot start with the letters xml (or XML, or Xml, etc.).
  o Names cannot contain spaces.

* Start and End XML tags must match in case sensitivity. You cannot have a Start tag that begins with an upper case letter and an Ending tag that begins with a lower case letter. However, some applications you may use with XML that do have more restrictive case sensitivity. For example, XHTML requires lower case markup tags. Recommended practice is to check naming conventions of other applications you intend to use with XML. Using all lowercase for XML tags is the most universally accepted format.

Special Conditions

If transmitting an element field or form field that is a multi-select or a checkbox or contains items in a series, multiple values can be sent in a single node with ~|~ delimiter.

For example, <FormInput fieldid="12584"><![CDATA[ chk1~|~chk2]]></FormInput> or <FormInput fieldid="12584">chk1~|~chk2</FormInput>.

Encoding Standard

XML version 1.0, encoding UTF-8 is the encoding specification used in BrassRing XML documents.

CDATA Input

The following CDATA syntax guidelines must be followed when using BrassRing APIs.

- All CDATA input must start with "<![CDATA[" and ends with "]>")
- CDATA input items cannot be nested.
- Separate list items within CDATA using commas or ~|~.

Security

The BrassRing API Library uses a token based authentication system. Users logging into the BrassRing application, must supply their User ID, Password, and Manifest name in the API call XML request. When the BrassRing application receives this information it generates an authentication token that encrypts the received information, the GUID, and the Current UTC Datetime and returns the encrypted information in an XML element.
named Security. This XML Security element is then used to authenticate users during the API call process. If users do not submit the Security element, the system reverts to standard Username/Password mode for authentication.

Certificate based encryption

The BrassRing Object Oriented API provides certificate based encryption. When vendors submit API calls to the ATS must contain the User ID, Password, and Manifest name. BrassRing then:

1. Uses the Manifest name & User ID, the Xpath’s of the Nodes to be encrypted & signed and reads it from the XML in the Keymaster table’s SecurityXpath column.
2. Once read, the client certificate public key encrypts all the output data in the nodes.
3. The Vendor receives the certificate private key (IBM Kenexa’s Key) and uses it to digitally sign the nodes. If those nodes do not exist, they are created.

The client can then use the digital signature to decrypt the output data.

BrassRing API Library – Supported API Calls

The GetCandidate API uses the HTTP Post method. During Workbench API configuration, clients can designate which fields the API calls requests and returns.

Technical Details on this approach can be found on the following site: http://msdn.microsoft.com/en-us/library/debx8sh9%28v=vs.110%29.aspx

Each GetCandidate API web service call uses XML (parameters) to initiate requests and receive responses. XML parameters specify what is being requested and what response is expected.

The supported API call is:

- GetCandidate

During Workbench API configuration, clients can designate which fields the API calls requests and returns.

Order of Object during Requests and Responses

The business rule for the order of selection of objects is as follows:

- If identifier filters exist, they take the preference as they directly relate to the requested object. The system retrieves these objects first. If there is main object filter as well which was supposed to be applied on the REQUESTED object (and not the child) the result will be filtered further here to get the REQUESTED OBJECTS which falls under both (MAIN AND IDENTIFIER) filters.
- If there is requested child for the object then the system retrieves the children for the objects received in the previous step.
- If there is only identifier filter there for the Requested Object and Main filter needs to be applied to the child object, then the first step is to identify the Requested Object applied the Identifier filter. Then, the system retrieves the children of the Requested Object from the previous step and applies the MAIN Filter on them to get the final result.
- If there is no identifier filter on the Requested Object and only Main Filter applied to the Child object, then the system retrieves the children which fall under the Main Filter criteria and then get the related parent.

Executing the GetCandidate API Call

The general procedure to execute the OO API call requires clients to:

1. Configure Workbench subscriptions to choose the outputable fields and mark required filters.
2. Prepare XML documents.
3. Access BrassRing portal URL.
4. Execute API call (GetCandidate).
5. Examine results of API Call Response.
Configuring Workbench Subscriptions
Clients can configure subscriptions by selecting output fields and required filters in Workbench.

When a client selects an outputable field for the Candidate Export and the selected field is configured as encrypted at the field level and the Allow Encrypted Fields is unchecked, it means that this field value is transmitted as a NON_ENCRYPTED value.

For additional information about subscriptions and restrictions, see Restriction of Objects.

In Workbench:
While creating/editing a subscription, workbench users with appropriate permissions can see the below settings when the object type “Candidates” is selected.

These are applicable for only Get Candidate API.

1. **Tools > Integrations > Administration.**
2. Select the Subscription Admin Icon for **Object based Import/Export**.
   The integration type administration screen displays.
3. Select Candidate and then Select the **Edit Settings pencil** icon.
   The Edit Subscription Candidates dialog launches.

4. Scroll down and select the **Export** check box.
   There is no dependency between Form Types and Output Fields. For example, selecting a specific Form Type does not impact selectable output fields.
   Select Form Types. You can select the check box for All Form Types or select forms from the list.
5. Select **Output Fields** in Select Output Fields category.
   You can select the check box for All Fields or uncheck the All Fields checkbox and manually select each output field using the scroll bars to scroll within the Output Fields selection box.
Output fields are encrypted by default if the field is configured as encrypted at the field level. To maintain encryption, user must select "Allow encrypted fields".

6. Select **Required Filters** in the Required Filters category. You can select the check box for No Filters Required or uncheck the No Filters Required checkbox and manually select each output field using the scroll bars to scroll within the Output Fields selection box.

7. Select **Save**.
Use Case

This use case workflow diagram demonstrates a typical use case for extracting candidate details.

Client Export API

<table>
<thead>
<tr>
<th>Client Application</th>
<th>Request XML</th>
<th>Response XML</th>
<th>BrassRing</th>
</tr>
</thead>
<tbody>
<tr>
<td>HR User opens Client Application</td>
<td>Request xml includes Authentication Details, Object Filters, Data Subset</td>
<td>Response xml includes the requested data</td>
<td>BrassRing authenticates the requestor</td>
</tr>
<tr>
<td>Enters filters to pull Data from BrassRing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sample Filters</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example: User types: Self Service User, Quick Start User</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clicks Client Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data is converted into xml</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Recruiter receives an error message</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resulting User Types are displayed in a table</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
GetCandidate Use Case HR Status Workflow Configuration

This Use Case demonstrates how a client might configure a workflow to use the GetCandidate API to retrieve candidates with a specific HR status.

The general procedure to execute the OO API call requires clients to:

1. Configure Workbench subscriptions to chose the outputable fields and mark required filters.
2. Prepare XML documents.
3. Access BrassRing portal URL.
4. Execute API call (GetCandidate).
5. Examine results of API Call Response.

Use Case – Step One - Configuring HR Status Candidate Export

In this scenario, clients configure the Workbench Subscription and configure the output fields and filters to retrieve candidates that match that criteria. Client logs into Workbench and navigates to:

1. Tools > Integrations > Administration.
2. Selects the Subscription Admin Icon for Object based Import/Export.

   The integration type administration screen displays.

3. Selects Candidate and then Selects the Subscription Administration icon.

   The Edit Subscription Candidates dialog launches.

   Client scrolls to the Candidate section of the dialog.

4. Scroll downs to the Candidate section and selects the Export check box.
There is no dependency between Form Types and Output Fields. For example, selecting a specific Form Type does not impact selectable output fields.

Form Types. You can select the check box for All Form Types or select forms from the list.

5. Selects **Output Fields** in Select Output Fields category.

6. Unchecks the **All Fields** check box, leaves **Allow Encrypted Field** check box checked, and selects **HR Status** in the list box.

   Output fields are encrypted by default if the field is configured as encrypted at the field level. To maintain encryption, user must select "Allow encrypted fields".

7. Unchecks the **All Filters** check box and selects **Requisition ID** and **HR Status** in the **Required Filters** list box.

8. Selects **Save**.

**Use Case – Step Two**

The client must now prepare the XML documents. In the XML document as shown, the client specifies the HR status (2) to be retrieved and the requisition ID (789342). The XML for these queries looks like this.

```xml
<FILTER TYPE="CANDIDATE_HR_STATUS">
  <VALUE>2</VALUE>
</FILTER>

<FILTER TYPE="REQUISITION_ID">
  <VALUE>789342</VALUE>
</FILTER>
```
This means the API call will request and return the GetCandidate Object and all candidates that match the filters and both the specified Req ID (789342) AND the HR status (2).

```
<?xml version="1.0" encoding="utf-8"?>
<GET_DATA>
  <AUTHENTICATION>
    <CLIENT_ID>Client ID</CLIENT_ID>
    <INTEGRATION_USER>Integration User ID</INTEGRATION_USER>
    <REQUESTOR>The Requestor</REQUESTOR>
    <USERNAME>Username</USERNAME>
    <PASSWORD>Password</PASSWORD>
    <MANIFEST_NAME>"objectapi"</MANIFEST_NAME>
    <SECURITY_TOKEN />  
  </AUTHENTICATION>
  <REQUEST>
    <OBJECT NAME="CANDIDATES">
      <CLIENT_OBJECT_IDENTIFIER>The value of the integration ID</CLIENT_OBJECT_IDENTIFIER>
      <ADDITIONAL_RECORDS_GUID></ADDITIONAL_RECORDS_GUID>
    </OBJECT>
    <MAIN_OBJECT_FILTERS>
      <FILTER TYPE="CANDIDATE_HR_STATUS">
        <VALUE>2</VALUE>
      </FILTER>
      <FILTER TYPE="REQUISITION_ID">
        <VALUE>789342</VALUE>
      </FILTER>
    </IDENTIFIER_OBJECT_FILTERS>
    <OUTPUT>
      <DATA_SUBSET INCLUDE="CANDIDATE_HR_STATUS">TRUE</DATA_SUBSET>
      <DATA_SUBSET INCLUDE="CANDIDATE_REQUISITION_ID">TRUE</DATA_SUBSET>
    </OUTPUT>
  </REQUEST>
</GET_DATA>
```

Use Case – Step Three

IBM will configure and provide an endpoint URL for the Client. This URL will be used to execute the GetCandidate API call. This URL is a 2-way SSL integration that will require a certificate handshake between IBM and the Client.

Use Case – Step Four

The GetCandidate API returns all Candidates that meet the GetCandidate filters AND the criteria specified in the XML. In this case, the GetCandidate API returns all candidates associated with the Req ID of (789342) AND the HR status (2).
Configuring Workbench Subscriptions

Clients can configure subscriptions by selecting output fields and required filters in Workbench.

When a client selects an outputable field for the Candidate Export and the selected field is configured as encrypted at the field level and the Allow Encrypted Fields is unchecked, it means that this field value is transmitted as a NON_ENCRYPTED value.

For additional information about subscriptions and restrictions, see Restriction of Objects.

In Workbench:

While creating/editing a subscription, workbench users with appropriate permissions can see the below settings when the object type “Candidates” is selected.

These are applicable for only Get Candidate API.

1. Tools > Integrations > Administration.
2. Select the Subscription Admin Icon for Object based Import/Export.

   The integration type administration screen displays.

3. Select Candidate and then Select the Edit Settings pencil icon. The Edit Subscription Candidates dialog launches.

4. Scroll down and select the Export check box.

   There is no dependency between Form Types and Output Fields. For example, selecting a specific Form Type does not impact selectable output fields.

   Select Form Types. You can select the check box for All Form Types or select forms from the list.

5. Select Output Fields in Select Output Fields category.

   You can select the check box for All Fields or uncheck the All Fields checkbox and manually select each output field using the scroll bars to scroll within the Output Fields selection box.
Output fields are encrypted by default if the field is configured as encrypted at the field level. To maintain encryption, users must not select "Allow encrypted fields".

6. Select **Required Filters** in the Required Filters category.
   You can select the check box for No Filters Required or uncheck the No Filters Required checkbox and manually select each output field using the scroll bars to scroll within the Output Fields selection box.

7. Select **Save**.

**GetCandidate Request (Candidate Object)**

In the GetCandidate Request, the request can be crafted to return a list of candidates, or a single candidate with varying degrees of data for each candidate depending on the filter criteria used.

**Request XML**

API call: GetCandidate

```xml
<?xml version="1.0" encoding="utf-8" ?>
<GET_DATA>
  <AUTHENTICATION>
    <CLIENT_ID>Client ID</CLIENT_ID>
    <INTEGRATION_USER>Integration User ID</INTEGRATION_USER>
    <REQUESTOR>The Requestor</REQUESTOR>
    <USERNAME>Username</USERNAME>
    <PASSWORD>Password</PASSWORD>
    <MANIFEST_NAME>"objectapi"</MANIFEST_NAME>
    <SECURITY_TOKEN/>
  </AUTHENTICATION>
  <REQUEST>
    <OBJECT_NAME="CANDIDATES">
      <CLIENT_OBJECT_IDENTIFIER>The value of the integration ID</CLIENT_OBJECT_IDENTIFIER>
      <ADDITIONAL_RECORDS_GUID></ADDITIONAL_RECORDS_GUID>
    </OBJECT>
    <MAIN_OBJECT_FILTERS>
      <FILTER TYPE="DATETIME_FROM" FORMAT="YYYY-MM-DDThh:mm:ssZ">Date following the format</FILTER>
      <FILTER TYPE="DATETIME_TO" FORMAT="YYYY-MM-DDThh:mm:ssZ">Date following the format</FILTER>
      <FILTER TYPE="INCLUDE_ACTIVE">To include the element must be present with the value of "TRUE"</FILTER>
      <FILTER TYPE="INCLUDE_INACTIVE">To include the element must be present with the value of "TRUE"</FILTER>
    </MAIN_OBJECT_FILTERS>
    <IDENTIFIER_OBJECT_FILTERS>
      <FILTER TYPE="SYSTEM_ID">
        <VALUE>Resume Key of a candidate</VALUE>
      </FILTER>
      <FILTER TYPE="CONFIGURED_ID">
        <VALUE>Email Address of a candidate</VALUE>
      </FILTER>
    </IDENTIFIER_OBJECT_FILTERS>
  </REQUEST>
</GET_DATA>
```
GetCandidate Response (Candidate Object)

The response data set varies based on the requested object filters. Returned fields also vary based on the output data subsets in the request.

- If the no additional data subsets are selected for each candidate returned will be:
  - Resume Key
  - Email Address
  - Candidate Full Name
  - Status (Active or Inactive)
  - Date of Last Edit

- If the ADDITIONALDETAILS data set is selected the following additional fields will show up for each candidate:
  - Address
  - City
  - Zipcode
  - State
  - Country
• Home Phone
• Cell Phone
• Work Phone
• Fax
• Homepage
• BRUID
• Candidate Type

- If the CANDIDATE_HR_STATUS data set is selected the following additional fields will show up for each candidate and all of their associated job applications.
  • Job Requisition Number
  • HR Status
- If the CANDIDATE_RESUME_AND_COVERLETTER data set is selected the following additional fields will show up for each candidate and all of their associated job applications.
  • Resume Text
  • Cover Letter Text
- If the CANDIDATE_FORMS_DATA data set is selected the following additional fields will show up for each candidate form.
  • Candidate Form ID
  • Candidate Form Name
  • Talent Gateway ID
  • Status (Active or Inactive)
  • Form Type
  • Requisition ID
  • Questions
  • Answers
- If the CANDIDATE_EDUCATION_EXPERIENCE data set is selected the following additional fields will show up for each candidate.
  • For each Education item
    - School Name
    - Degree
    - GPA
    - Major
    - Most Recent Indicator
    - End Date
  • For each Work Experience Item
    - Employer Name
    - Job Description
    - Start Date
    - End Date
    - Most recent Indicator

Response XML

```xml
<?xml version="1.0" encoding="utf-8" ?>
```
<RESULTS>
  <CLIENT_ID>11721</CLIENT_ID>
  <REQUESTOR>Test</REQUESTOR>
  <OBJECT NAME="CANDIDATES">
    <CLIENT_OBJECT_IDENTIFIER>The value of the integration ID</CLIENT_OBJECT_IDENTIFIER>
    <ADDITIONAL_RECORDS_GUID></ADDITIONAL_RECORDS_GUID>
    <INSTANCE>
      <SYSTEM_ID>Candidate Resume Key</SYSTEM_ID>
      <CONFIGURED_ID>Candidate Email Address</CONFIGURED_ID>
      <DESCRIPTION>
        <VALUE LANGUAGE="EN">Candidate Full Name (last, first, middle initial)</VALUE>
      </DESCRIPTION>
      <STATUS>Active or Inactive</STATUS>
      <MODIFIED_DATE>Date of Last Edit</MODIFIED_DATE>
      <STANDARD_DETAILS TYPE="ADDRESS">Candidate Street Address</STANDARD_DETAILS>
      <STANDARD_DETAILS TYPE="CITY">Candidate City</STANDARD_DETAILS>
      <STANDARD_DETAILS TYPE="ZIPCODE">Candidate Zip Code</STANDARD_DETAILS>
      <STANDARD_DETAILS TYPE="STATE">Candidate State</STANDARD_DETAILS>
      <STANDARD_DETAILS TYPE="COUNTRY">Candidate Country</STANDARD_DETAILS>
      <STANDARD_DETAILS TYPE="HOMEPHONE">Candidate Home Phone #</STANDARD_DETAILS>
      <STANDARD_DETAILS TYPE="CELLPHONE">Candidate Cell Phone #</STANDARD_DETAILS>
      <STANDARD_DETAILS TYPE="WORKPHONE">Candidate Work Phone #</STANDARD_DETAILS>
      <STANDARD_DETAILS TYPE="FAX">Candidate Fax #</STANDARD_DETAILS>
      <STANDARD_DETAILS TYPE="HOMEPAGE">Candidate Homepage</STANDARD_DETAILS>
      <STANDARD_DETAILS TYPE="BRUID">Candidate BRUID</STANDARD_DETAILS>
      <STANDARD_DETAILS TYPE="CANDIDATE_TYPE">Candidate Type (internal or external)</STANDARD_DETAILS>
      <STANDARD_DETAILS TYPE="REQUISITION_STATUS">
        <VALUE TYPE="HR_STATUS">Candidate HR Status related to this job</VALUE>
      </STANDARD_DETAILS>
      <STANDARD_DETAILS TYPE="REQUISITION_ID">Job Requisition ID</STANDARD_DETAILS>
      <EDUCATION>
        <VALUE TYPE="SCHOOL_NAME">Name of School</VALUE>
        <VALUE TYPE="DEGREE">Type of Degree</VALUE>
        <VALUE TYPE="MAJOR">Major</VALUE>
        <GPA>GPA</GPA>
        <VALUE TYPE="ENDING_YEAR">Year of Graduation</VALUE>
        <VALUE TYPE="MOST_RECENT">Is this the most recent</VALUE>
      </EDUCATION>
      <EXPERIENCE>
        <VALUE TYPE="EMPLOYER_NAME">employer Name</VALUE>
        <VALUE TYPE="POSITION_DESCRIPTION">Job Description</VALUE>
        <VALUE TYPE="STARTING_YEAR">Year Started</VALUE>
        <VALUE TYPE="ENDING_YEAR">Year Ended</VALUE>
      </EXPERIENCE>
    </INSTANCE>
  </OBJECT>
<VALUE TYPE="MOST_RECENT">Is this the most recent</VALUE>
</STANDARD_DETAILS>
<STANDARD_DETAILS TYPE="RESUME_TEXT">Full Resume Text</STANDARD_DETAILS>
<STANDARD_DETAILS TYPE="COVERLETTER_TEXT"> Full Cover Letter Text</STANDARD_DETAILS>

<OBJECT NAME="CANDIDATE_FORMS_DATA">
  <INSTANCE>
    <SYSTEM_ID>Candidate Form ID</SYSTEM_ID>
    <CONFIGURED_ID>Candidate Form Name</CONFIGURED_ID>
    <DESCRIPTION>
      <VALUE LANGUAGE="EN">Talent Gateway ID</VALUE>
    </DESCRIPTION>
    <STATUS>Active or Inactive</STATUS>
    <MODIFIED_DATE>Last Date of Edit</MODIFIED_DATE>
    <STANDARD_DETAILS TYPE="FORM_TYPE">Form Type</STANDARD_DETAILS>
    <STANDARD_DETAILS TYPE="REQUITION_ID">Requisition ID</STANDARD_DETAILS>
    <STANDARD_DETAILS TYPE="CANDIDATE_RESPONSE">
      <VALUE TYPE="QUESTION IDENTIFIER=""QUESTION_DBFIELDNAME" PROPERTY=""This will be field type. Values are Radio, CheckBox, SingleSelect, MultiSelect, Text, Grid, Auto-Fill, and Query Select">Question Name</VALUE>
      <VALUE TYPE="RESPONSE IDENTIFIER=""No">Answer</VALUE>
    </STANDARD_DETAILS>
  </INSTANCE>
</OBJECT>

<OBJECT NAME="CANDIDATE_ATTACHMENTS">
  <INSTANCE>
    <SYSTEM_ID>File ID</SYSTEM_ID>
    <CONFIGURED_ID>File Name</CONFIGURED_ID>
    <DESCRIPTION>
      <VALUE LANGUAGE="EN">locale specific file name</VALUE>
    </DESCRIPTION>
    <STATUS>Active or Inactive</STATUS>
    <MODIFIED_DATE>Last date of edit</MODIFIED_DATE>
    <STANDARD_DETAILS TYPE="FILE_CONTENT">File Content</STANDARD_DETAILS>
    <STANDARD_DETAILS TYPE="ATTACHMENT_CATEGORY">Attachment Category</STANDARD_DETAILS>
  </INSTANCE>
</OBJECT>
</RESULTS>
XML Element Table for BrassRing API Library

<table>
<thead>
<tr>
<th>Request/Response</th>
<th>XML Element Tag</th>
<th>Required?</th>
<th>Functionality Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>?xml version=&quot;1.0&quot;/encoding=&quot;utf-8&quot;?</td>
<td></td>
<td>Indicates version of xml and encoding type For more information see: <a href="http://www.w3.org/standards/xml/">http://www.w3.org/standards/xml/</a></td>
</tr>
<tr>
<td>Request</td>
<td>GET_DATA</td>
<td></td>
<td>The root node for retrieving the data</td>
</tr>
<tr>
<td>Request</td>
<td>CLIENT_ID</td>
<td>Yes</td>
<td>Client_ID identifies the client.</td>
</tr>
<tr>
<td>Request</td>
<td>INTEGRATION_USER</td>
<td>Yes</td>
<td>Any active BrassRing user of the client. If the API is being used by a Partner, the partner must get this from the mutual client. No password is needed for this.</td>
</tr>
<tr>
<td>Request</td>
<td>REQUESTOR</td>
<td>Yes</td>
<td>Any text that can be used to identify the response for a request. Example Request1, Request2 etc for each request.</td>
</tr>
<tr>
<td>Request</td>
<td>USERNAME</td>
<td></td>
<td>Client(consumer of the API) Application specific user name. BrassRing will randomly create one if this is not provided.</td>
</tr>
<tr>
<td>Request</td>
<td>PASSWORD</td>
<td></td>
<td>Client(consumer of the API) Application specific password. BrassRing will randomly create one if this is not provided.</td>
</tr>
<tr>
<td>Request</td>
<td>MANIFEST_NAME</td>
<td></td>
<td>The value must always be “Objectapi”</td>
</tr>
<tr>
<td>Request</td>
<td>SECURITY_TOKEN</td>
<td></td>
<td>Clients use their Security Token to validate themselves when using the APIs. It is generally returned to Client after a successful login, and is used when Clients make additional API requests.</td>
</tr>
<tr>
<td>Request</td>
<td>OBJECT</td>
<td></td>
<td>Object type from Workbench configuration</td>
</tr>
<tr>
<td>Request</td>
<td>CLIENT_OBJECT_IDENTIFIER</td>
<td></td>
<td>Object specific unique identifier</td>
</tr>
<tr>
<td>Request</td>
<td>ADDITIONAL_RECORDS_GUID</td>
<td></td>
<td>Batch id for more than 10,000 records</td>
</tr>
<tr>
<td>Request</td>
<td>MAIN_OBJECT_FILTERS</td>
<td></td>
<td>Object specific filters. Ex: For candidate object, the filters would be Candidate Name, HR Status, Applied On, etc.</td>
</tr>
<tr>
<td>Request</td>
<td>IDENTIFIER_OBJECT_FILTERS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Request</td>
<td>OUTPUT</td>
<td></td>
<td>All the fields that we want to in the result.</td>
</tr>
<tr>
<td>Request</td>
<td>DATA_SUBSET</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Response</td>
<td>RESULT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLIENT_ID</td>
<td>Client_ID identifies the client.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REQUESTOR</td>
<td>Any text that can be used to identify the response for a request. Example Request1, Request2 etc for each request.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>OBJECT</td>
<td>Object type from Workbench configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLIENT_OBJECT_IDENTIFIER</td>
<td>Object specific unique identifier</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADDITIONAL_RECORDS_GUID</td>
<td>Batch id for more than 10,000 records</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INSTANCE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSTEM_ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONFIGURED_ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STATUS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MODIFIED_DATE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STANDARDDETAILS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Error Codes

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Error Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>AuthenticationToken node cannot be empty.</td>
</tr>
<tr>
<td>19</td>
<td>AuthenticationToken node is missing from input.</td>
</tr>
<tr>
<td>46</td>
<td>JobIDs node is missing from input.</td>
</tr>
<tr>
<td>47</td>
<td>JobIDs node is missing from input.</td>
</tr>
<tr>
<td>50</td>
<td>ResumeType node cannot be empty.</td>
</tr>
<tr>
<td>51</td>
<td>Invalid ResumeType specified in input.</td>
</tr>
<tr>
<td>52</td>
<td>ResumeType node is missing from input.</td>
</tr>
<tr>
<td>53</td>
<td>Resume node cannot be empty.</td>
</tr>
<tr>
<td>54</td>
<td>Resume node is missing in input.</td>
</tr>
<tr>
<td>55</td>
<td>ResumeName node cannot be empty.</td>
</tr>
<tr>
<td>56</td>
<td>ResumeName is missing from input.</td>
</tr>
<tr>
<td>57</td>
<td>FileExtension node cannot be empty when resumeType input is &quot;Upload&quot;</td>
</tr>
<tr>
<td>58</td>
<td>FileExtension node cannot be missing when resumeType input is &quot;Upload&quot;</td>
</tr>
<tr>
<td>61</td>
<td>ResumeID node cannot be empty.</td>
</tr>
<tr>
<td>62</td>
<td>ResumeID node is missing from input.</td>
</tr>
<tr>
<td>70</td>
<td>CoverLetterType node cannot be empty.</td>
</tr>
<tr>
<td>71</td>
<td>Invalid CoverLetterType specified in input.</td>
</tr>
<tr>
<td>72</td>
<td>CoverLetterType node is missing from input.</td>
</tr>
<tr>
<td>73</td>
<td>CoverLetter node cannot be empty.</td>
</tr>
<tr>
<td>74</td>
<td>CoverLetter node is missing from input.</td>
</tr>
<tr>
<td>75</td>
<td>CoverLetterName node cannot be empty.</td>
</tr>
<tr>
<td>76</td>
<td>CoverLetterName node is missing from input.</td>
</tr>
<tr>
<td>77</td>
<td>FileExtension node cannot be empty when CoverLetterType input is &quot;Upload&quot;</td>
</tr>
<tr>
<td>78</td>
<td>FileExtension node cannot be missing when CoverLetterType input is &quot;Upload&quot;</td>
</tr>
<tr>
<td>79</td>
<td>CoverLetterId node cannot be empty.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------------</td>
</tr>
<tr>
<td>80</td>
<td>CoverLetterId node is missing from input.</td>
</tr>
<tr>
<td>230</td>
<td>The authentication status is not valid for this request.</td>
</tr>
<tr>
<td>430</td>
<td>Resume name is not unique.</td>
</tr>
<tr>
<td>431</td>
<td>Maximum number of resumes exceeded.</td>
</tr>
<tr>
<td>432</td>
<td>The Resume Id does not exist.</td>
</tr>
<tr>
<td>433</td>
<td>File Type is not supported.</td>
</tr>
<tr>
<td>434</td>
<td>File size exceeds 3 MB.</td>
</tr>
<tr>
<td>435</td>
<td>Cover Letter Name is not unique.</td>
</tr>
<tr>
<td>436</td>
<td>Maximum number of Cover Letters allowed exceeded.</td>
</tr>
<tr>
<td>437</td>
<td>The passed Resume is already configured as default resume.</td>
</tr>
<tr>
<td>438</td>
<td>An internal error has occurred.</td>
</tr>
<tr>
<td>439</td>
<td>FileExtension node cannot be empty when resumeType input is &quot;Upload&quot;.</td>
</tr>
<tr>
<td>440</td>
<td>The Cover Letter Id does not exist.</td>
</tr>
</tbody>
</table>
Language/Site/LocaleIDs

This is a current table of Language, Site and Locale IDs.

<table>
<thead>
<tr>
<th>Language</th>
<th>Language ID</th>
<th>Site Locale ID</th>
<th>ISO State Code (Not US State Codes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>6</td>
<td>1025</td>
<td>AR</td>
</tr>
<tr>
<td>Azerbaijani</td>
<td>9</td>
<td>1068</td>
<td>AZ</td>
</tr>
<tr>
<td>Chinese</td>
<td>138</td>
<td>2052</td>
<td>ZH</td>
</tr>
<tr>
<td>Chinese (Traditional)</td>
<td>126</td>
<td>1028</td>
<td>TW</td>
</tr>
<tr>
<td>Croatian</td>
<td>44</td>
<td>1050</td>
<td>HR</td>
</tr>
<tr>
<td>Czech</td>
<td>20</td>
<td>1029</td>
<td>CS</td>
</tr>
<tr>
<td>Danish</td>
<td>22</td>
<td>1030</td>
<td>DA</td>
</tr>
<tr>
<td>Dutch</td>
<td>82</td>
<td>1043</td>
<td>NL</td>
</tr>
<tr>
<td>Dutch-informal</td>
<td>82</td>
<td>11043</td>
<td>NL-IN</td>
</tr>
<tr>
<td>English</td>
<td>1</td>
<td>1033</td>
<td>EN</td>
</tr>
<tr>
<td>English (Intl)</td>
<td>140</td>
<td>2057</td>
<td>GB</td>
</tr>
<tr>
<td>English RTL</td>
<td>144</td>
<td>21033</td>
<td>ER</td>
</tr>
<tr>
<td>English UK Government</td>
<td>145</td>
<td>12057</td>
<td>GG</td>
</tr>
<tr>
<td>English US Government</td>
<td>143</td>
<td>11033</td>
<td>GV</td>
</tr>
<tr>
<td>Finnish</td>
<td>31</td>
<td>1035</td>
<td>FI</td>
</tr>
<tr>
<td>French (Canada)</td>
<td>141</td>
<td>3084</td>
<td>FC</td>
</tr>
<tr>
<td>French (France)</td>
<td>34</td>
<td>1036</td>
<td>FR</td>
</tr>
<tr>
<td>German</td>
<td>23</td>
<td>1031</td>
<td>DE</td>
</tr>
<tr>
<td>German-informal</td>
<td>23</td>
<td>11031</td>
<td>DE-IN</td>
</tr>
<tr>
<td>Greek</td>
<td>25</td>
<td>1032</td>
<td>EL</td>
</tr>
<tr>
<td>Hebrew</td>
<td>42</td>
<td>1037</td>
<td>HE</td>
</tr>
<tr>
<td>Hungarian</td>
<td>45</td>
<td>1038</td>
<td>HU</td>
</tr>
<tr>
<td>Language</td>
<td>Language ID</td>
<td>Site Locale ID</td>
<td>ISO State Code (Not US State Codes)</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>Hungarian-informal</td>
<td>45</td>
<td>11038</td>
<td>HU-IN</td>
</tr>
<tr>
<td>Italian</td>
<td>52</td>
<td>1040</td>
<td>IT</td>
</tr>
<tr>
<td>Japanese</td>
<td>54</td>
<td>1041</td>
<td>JA</td>
</tr>
<tr>
<td>Korean</td>
<td>61</td>
<td>1042</td>
<td>KO</td>
</tr>
<tr>
<td>Norwegian</td>
<td>83</td>
<td>1044</td>
<td>NO</td>
</tr>
<tr>
<td>Polish</td>
<td>88</td>
<td>1045</td>
<td>PL</td>
</tr>
<tr>
<td>Portuguese (Brazil)</td>
<td>90</td>
<td>1046</td>
<td>PT</td>
</tr>
<tr>
<td>Portuguese (Portugal)</td>
<td>142</td>
<td>2070</td>
<td>PP</td>
</tr>
<tr>
<td>Romanian</td>
<td>94</td>
<td>1048</td>
<td>RO</td>
</tr>
<tr>
<td>Russian</td>
<td>95</td>
<td>1049</td>
<td>RU</td>
</tr>
<tr>
<td>Serbian</td>
<td>108</td>
<td>2074</td>
<td>SR</td>
</tr>
<tr>
<td>Slovak</td>
<td>102</td>
<td>1051</td>
<td>SK</td>
</tr>
<tr>
<td>Slovenian</td>
<td>103</td>
<td>1060</td>
<td>SL</td>
</tr>
<tr>
<td>Spanish</td>
<td>27</td>
<td>3082</td>
<td>ES</td>
</tr>
<tr>
<td>Spanish-informal</td>
<td>27</td>
<td>13082</td>
<td>ES-IN</td>
</tr>
<tr>
<td>Swedish</td>
<td>112</td>
<td>1053</td>
<td>SV</td>
</tr>
<tr>
<td>Turkish</td>
<td>123</td>
<td>1055</td>
<td>TR</td>
</tr>
<tr>
<td>Welsh</td>
<td>21</td>
<td>1106</td>
<td>CY</td>
</tr>
</tbody>
</table>