IBM® Kenexa® BrassRing® on Cloud

Application Programming Interface (API) Reference

Release Date: March, 2014

IBM®

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, 2014.

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

API Reference Guide

Revision Table

Author	Date updated	Summary
Bobbi Hennessey	16 July 2012 (started)	Incorporating LDP 118 into work flow. Added draft of updated error codes. Still in draft stage.
Bobbi Hennessey	Sept. 6, 2012	Restructure of Document Incorporating new ideas/solutions
Bobbi Hennessey	Oct. 6, 2012	Release 13.2 API features added to guide. New workflow examples, updated structure. New XML element table.
Bobbi Hennessey	Oct .23, 2012	Incorporate RDP 658 XML from SE into document. Updated new user and form import. Noted most element values in XML element table.
Bobbi Hennessey	Dec. 17, 2012	Incorporating changes from PM team review meeting, including but not limited to the following: • API error codes tables • API XML element tables • Release 13.2 APIs • Language, Site and Locale IDs table added
Bobbi Hennessey	Jan 2, 2013	Incorporated new error codes for Simultaneously Login API.
Bobbi Hennessey	May 30, 2013	• Updated Forgot Password included operation to reset password without security question if authentication token still valid • Updated SubmitFormResponses, UpdateProfile • Added NewUser Creation Fields

Bobbi Hennessey	October, 2013	• Adding Release 13.3 API projects including:
		  • Assessment API
		  • Additions and enhancements to Job Cart API
		  • Updated TG Job Search API with XML for Job Search Results element
		  • Updated TG Search API with additional information regarding XML element use

Bobbi Hennessey	March, 2014	• Republished IBM Branded API Reference Guide and including Release 13.4 API projects:
		  • Updated TG Search API functionality to support new API node, ReturnUnpostedJobs, that allows clients to retrieve and display unposted jobs
		  • TG Candidate API enhancement allows candidates to change security question(s) and/ or answer(s) from a client's BrassRing talent gateway website
		  • Updated WB path to Client and Site ID

Introduction

This document provides information about IBM Kenexa BrassRing on Cloud web service Application Programming Interfaces (APIs).

Purpose

The purpose of this document is to:

- Present a high-level overview of the BrassRing API architecture.
- Provide high level overview of supported workflow processes, including information exchange with any external, third-party system that can send and receive XML requests and responses.
- Provide procedures for initiating BrassRing APIs.

Audience

The intended audience for this document is:

- Client Decision Makers, Internal IT Teams, Systems Integrators, and Support Teams.
- Engineering Services Team, Support Team, and Technical Services Group.

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.

API	Application Programming Interface
XML	extensible Markup Language

Typographical Conventions

This document uses the following typographical conventions:

> Use Ctrl + Click to follow hyperlinks. Use ALT + LEFT ARROW to return to point of origin.

What are BrassRing APIs?

An API is actually an interface, a common boundary, between two separate systems, the BrassRing application and a client's website. BrassRing APIs allow clients to configure APIs that automatically exchange real-time data between their job websites and their BrassRing application. Once configured, APIs can automatically populate a client's job website and respond to candidate's input as well as job search requests. Once an API is configured, clients no longer need to manually populate job site websites or respond to each and every candidate inquiry – the API does it for them. Clients can maintain one up-to-date job repository in BrassRing connected to their job websites.

How Do APIs Work?

How do the two separate systems, the BrassRing application and a client's website, communicate using an API? What is the communication channel? These two systems communicate using a transport mechanism (envelope) and transport language (content).

BrassRing APIs can use either HTTP Post or Web Service Descriptive Language (WSDL) as the transport mechanism. BrassRing APIs use eXtensible Markup Language (XML) messages (requests and response) as content containers. The real-time API data exchanges (XML messages) are known as requests and responses. Each XML message contains a defined set of elements for each specific API function. At a high level, these XML elements are 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.

XML messages are non-editable.

For more detailed technical information on configuring APIs, see Configuring APIs.

> Use Ctrl + Click to follow hyperlinks. Use ALT + LEFT ARROW to return to point of origin.

API Types

BrassRing offers two types of APIs designed to enhance a client's talent management website, Talent Gateway APIs and Candidate APIs. Talent Gateway APIs focus on job seeker functions such as job search, TG account creation, and how the API delivers content to a talent gateway or client's website. Candidate APIs enhance a candidate's experience after they have applied to a position. Examples include accessing their job carts, viewing their communication history, and resetting their passwords. Each API's functionality is briefly outlined in the following sections.

Job Search, a TG API, can be used in conjunction with Candidate APIs. Consider this scenario where a candidate re-visits a client's website (Candidate API – Authenticate User) and searches for a job (TG Job Search API) and views job results. Candidate clicks the hyperlinked job to view the job details (TG Job Search API) and decides to apply. Candidate clicks their profile and then views their resume (Candidate API – Get Resume) and applies to the position. The How Can I Use an API? This section walks through a typical API workflow.

API Functionality

Each of the API types, Talent Gateway and Candidate, support a defined functionality. BrassRing offers the following Talent Gateway and Candidate APIs.

Web Service API (Talent Gateway)

Talent Gateway APIs focus on talent management functions such as job search, account creation, and content delivery methods used to populate a talent gateway or client's website.

- **Job Search –** Provides job search field options clients can include for use when performing job searches and returning job results. This is the most commonly used AP because it enables customers to allow job seekers to perform job searches from many different locations such as branded career sites, microsites, Facebook or other social media channels.

The Job Search function on a talent gateway can return responses in one of two formats:

- Proprietary XML
- Rich Site Summary (RSS) WML web feed

Proprietary XML is used in situations where clients are using a proprietary XML system. This feature allows the BrassRing application to interface directly with a client's proprietary system. RSS XML web feeds are used in situations where clients wish to continuously push new and updated content to candidates on their web sites. Once configured, clients are assured of real-time automatic delivery of new content.

Candidate APIs

Candidate specific APIs are APIs that directly communicate with candidates. Each of these APIs is outlined below.

- **Access Page -** Allows clients to navigate directly to one of the Talent Gateway pages from one of their Web pages. For example: users could be directed to the Search openings page, Edit profile page, or Job cart page.
- **Assessments - Allows** authenticated candidates to view and access their pending assessments.
- **Authenticate User–** Provides single sign-on (SSO) support. To Support SSO Talent Gateways, the user has to send the Client Id / Site Id / SSO ID. The system then generates the new user and sends the Authentication Token back to client. Client can use that token to access other APIs. This API can also be used in a non-SSO situation when another application needs to send the userid and password to the Talent Gateway.
- **Candidate Forms** – Allows candidates to view forms on their job portals that have been eLinked to them.
- **Change Security Questions** – Allows candidates to change security question(s) and/ or answer(s) from a client's BrassRing talent gateway website.
- **Communications History**– Enables candidates to view a log of all communications received by clients and communications with clients. This is helpful if a candidate misplaced an email and needs to see the contents or fill out a form that was associated with the email.
- **Event Manager Search - Candidate Specific –** Retrieves all the events scheduled in Event Manager that the candidate can self-schedule. The request can be time bound and can be restricted to only public events.
- **Document/Document Packet** – Enables re-authenticated candidates to view and access their document packets on their job portal
- **Forgot Password –** Provides candidate reauthentication so candidates can change forgotten passwords.
- **Get Resume–** Enables users to see direct links to their resumes that they have uploaded in the system.
- **Get Jobs in Job Cart–** Allows users to see the jobs to which they have previously applied to on a client's website and to add or remove jobs from their job carts.
- **Get Job Status** – Retrieves the current HR status of all jobs to which the user has applied on a client's website. This is useful for clients who want to communicate the candidate's application status.
- **New User Creation–** Enables systems to create a totally new Talent Gateway user. This is useful where a candidate was already created in another application and that candidate needs to log in directly to the Talent Gateway later on. This is common if the application began in a different system, but the candidate will log in to the Talent Gateway during the post-application process.
- **Reset Password–** Provides a way for users to reset their passwords. This API is rarely used but can be useful when password expiration is in use.
- **Simultaneous Login to Multiple Gateways** – Provides a way to authenticate candidates into multiple Talent Gateways simultaneously via the API authentication process in order to ensure all links returned via Search API, Saved Jobs, and Job Cart are clickable and actionable.
- **Talent Gateway Form Import -** Allows candidates to submit and update form data for Talent Gateway forms.
- **Update Existing Users** – Allows clients to update existing TG user profiles, including the updating of Talent Records.

API Reference Guide

> **Note**: To view more detailed information about configuring Candidate APIs, see
> Configuring Candidate APIs.

How Can I Use an API?

BrassRing provides a number of APIs that automate talent management functions between BrassRing and a client's system or website. Talent Gateways APIs support talent management functions related to the talent gateway and content delivery. Candidate APIs directly support a candidate's interactions with a client's BrassRing application or website. Although these two API types focus on a different functionality, BrassRing APIs work seamlessly together.

Talent Gateway APIs support job search functions and determine how a client's content is delivered to their talent gateways or websites. Candidate APIs allow candidates to perform tasks related to their job application. For example, viewing their job carts, communication history, and uploaded resumes. Clients can also use candidate APIs to authenticate users and to allow simultaneous login to multiple gateways.

> Use Ctrl + Click to follow hyperlinks. Use ALT + LEFT ARROW to return to point of origin.

BrassRing Job Search – Talent Gateway API

The BrassRing job search API is an example of a talent gateway API. When configured, candidates can search (request) jobs on a client's website. The API receives the candidate's request and responds (response) with job search results. Talent gateways using this functionality can be resident on a client's website site or on third party vendor websites. Once configured, clients can update content within their BrassRing application once and this content becomes automatically available on their talent management websites.

This example demonstrates the BrassRing Job Search API functionality on a client's talent management website.

1. Candidate views client's job website and begins a job search.

2. Candidate enters search criteria and clicks **Search**.
   The search results display.

3. Candidate clicks Job Description hyperlink to view the job.
   The jobs details page displays.

4. Candidate can then use the Apply button to apply to the position.

When the candidate completes the job application, clients can configure the Candidate APIs to provide a robust candidate experience. See BrassRing – Candidate APIs.

> Use Ctrl + Click to follow hyperlinks. Use ALT + LEFT ARROW to return to point of origin.

BrassRing APIs – Candidate APIs

Candidate APIs can automatically personalize a candidate's experience after application. For example, by using the Authenticate User API, you can personalize the greeting a candidate views when they return to your job website. The

following workflow is an example of a subset of BrassRing Candidate APIs available. For a full list of all BrassRing Candidate APIs, see Candidate APIs.
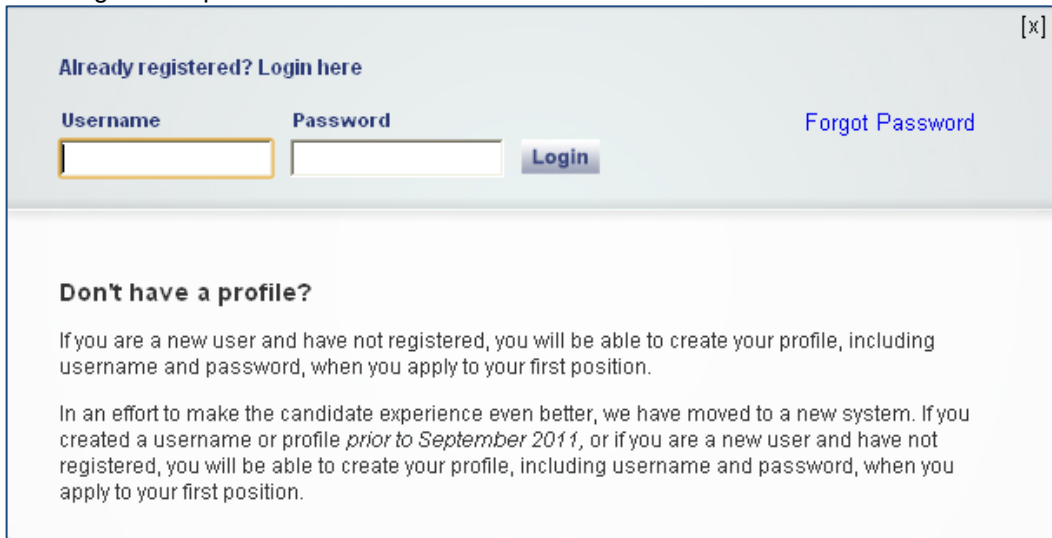
This example demonstrates the functionality of a subset of the BrassRing candidate APIs:

1. Candidate logs into the client's job website. If the candidate's first visit to the login screen, the New User API executes. If a returning user either forgets their password or has timed out (password expiration) for the last used password, he/she can use one of the two following methods to initiate re-authentication:

Forgot Password – enables candidates to reset forgotten passwords for re-authentication

Reset Password – provides a way for users to retrieve their passwords, useful when password expiration is in use

2. If the candidate is returning to the login screen, the Authenticate User API executes. This example shows a returning user experience.
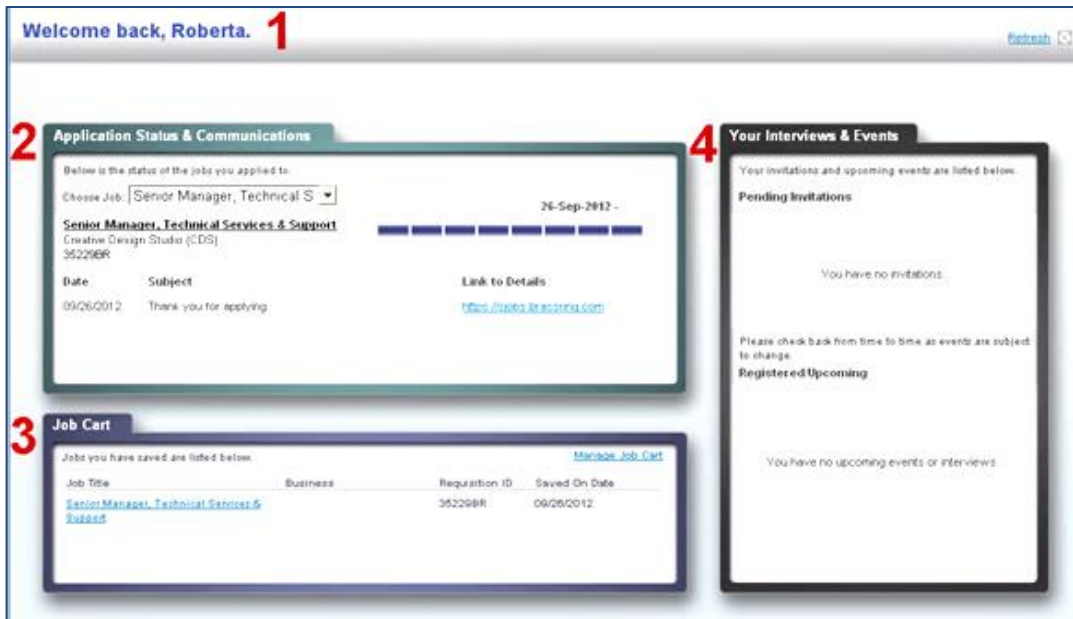
After logging into their account, the Welcome Screen and the configured APIs display.

The Authenticate User API authenticates and recognizes the candidate, displays a personalized message (**1**) for the candidate, and also executes the following Candidate APIs:

Get Job Status (**2**) – enables view of details regarding their application

API Reference Guide

Get Job Cart (**3**) – links to jobs in candidate's job cart

Event Manager Search (**4**) – allows candidates to self-register for events

Additional Candidate APIs can be configured that display the following links and information:
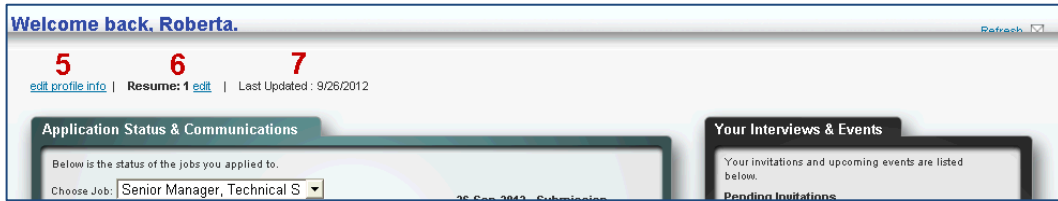
Access Page (**5**) – allows candidates to navigate to edit profile page (or other specified location)

Get Resume (**6**) - enables candidates to access and edit their resumes

Communication History (**7**) – displays last date candidate logged into portal

Candidate Forms – allows candidates to view forms received via eLink (not shown)

Document/Document Packet - enables re-authenticated candidates to access their document packets on the job portal (not shown)

When the candidate logs out, this visit is added to this candidate's Communication History on this website by the Communication History API.

Configuring Talent Gateway APIs

Talent Gateway APIs focus on talent management functions such as job search, TG account creation, and how the API delivers content to a talent gateway or client's website. When clients configure job search APIs, they can define exactly what the candidate can view and access when performing job searches on their websites.

Configuration of the API includes defining search values (using valid XML and XML elements) to return when a candidate's performs a job search. For example, if a client wants job search results to include job title and job location, the client must define the search values (XML elements) when configuring their job search API. When the BrassRing application receives the API XML request for the defined values contained in the XML elements, it immediately displays the requested information on the client's website.

Clients can also choose a specific API delivery method: static, dynamic, or proprietary. Content delivered statically does not change, but dynamically delivered content is constantly updated using Real Simple Syndication (RSS). Proprietary delivery is chosen when content is delivered to the client side of a site. RSS allows a client's website to continuously update the originally posted information, ensuring that the most current version always displays. For detailed information on configuring RSS feeds to your client's website, see Configuring RSS Feeds for Talent Gateway APIs.

As with all BrassRing APIs, each API requires Workbench configuration. The following procedure outlines how to configure searches in Workbench and how to execute the APIs.

BrassRing Job Search – Talent Gateway API

The BrassRing job search API is a talent gateway API. When configured, candidates can search (request) jobs (request) on a client's website. The API receives the candidate's request and responds (response) with job search results. Talent gateways using this functionality can be resident on a client's website site or on third party vendor websites. Once configured, clients can update content within their BrassRing application once and this content becomes automatically available on their talent management websites.

This example demonstrates the BrassRing Job Search API functionality on a client's talent management website.

1. Candidate views client's job website and begins a job search.

2. Candidate enters search criteria and clicks **Search**.
   The search results display.

3. Candidate clicks Job Description hyperlink to view the job.
   The jobs details page displays.

4. Candidate clicks **Apply** to apply to the position.

When the candidate completes the job application, clients can configure the Candidate APIs in conjunction with TG APIs to provide a robust candidate experience. See Candidate APIs for a workflow example.

Talent Gateway API Response Types

The Talent Gateway API sends API response in one of three ways:

- Static link to TG search results

  o If you want to take candidates into the search results page, or create static links to "canned" search results to use on your site, you want a link to the search results returned from your query. Static links do not update automatically.

- Proprietary XML (for example, Microsoft's version of XML)

- o If you want candidates to view job detail fields on the client side of the site, you need to define your API request to return proprietary XML which will also provide direct links into the Talent Gateway job details page.

- RSS XML web feed

  - o If you want to set up an RSS feed to which a candidates can subscribe, you want your query to return an RSS web feed so that you can update your XML store to which the candidates are subscribing. (TGs do not support RSS feeds, but clients can set them up on their sites). For information on configuring RSS feeds, see Configuring RSS.

Example 1. - Static Job Search Results

This example shows static job search results. In this example, the job search results sent from BrassRing are non-changed, static, and do not update on a regular basis. The other content on this page also remains unchanged.
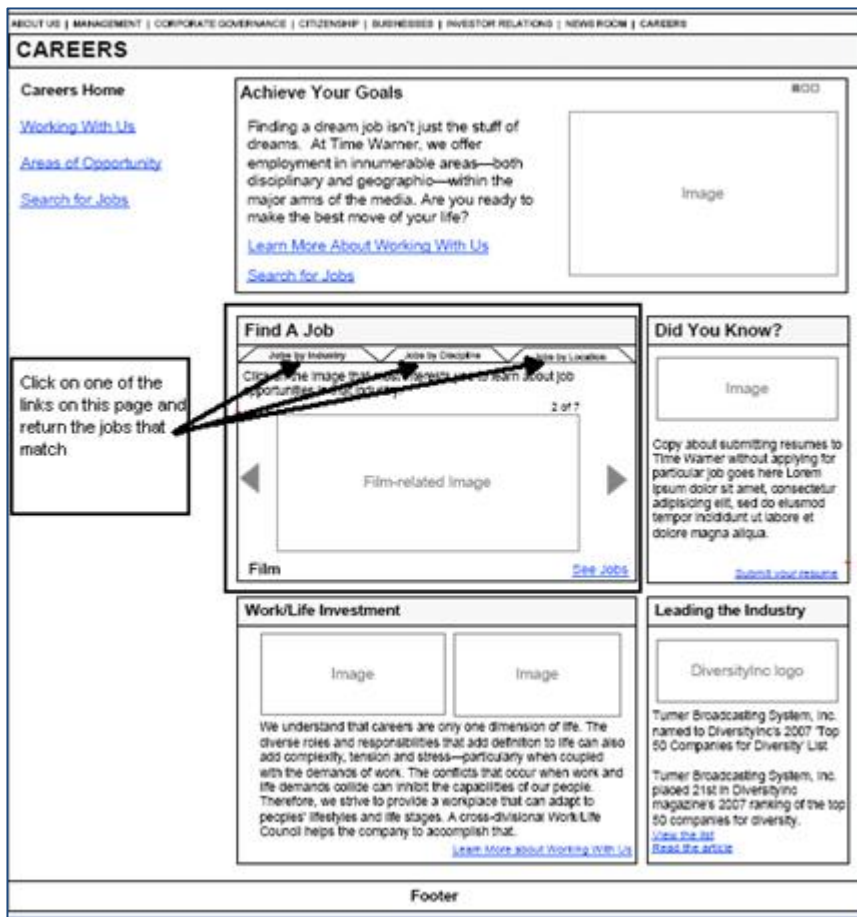
Example 2. – RSS Job Search Results

This example shows job search results returned as RSS feeds. If the job position or description changes in BrassRing while the candidate is viewing the job search results, this job search results page will change accordingly. In addition, the jobs listed in the Most Viewed Jobs fields will also update continuously.
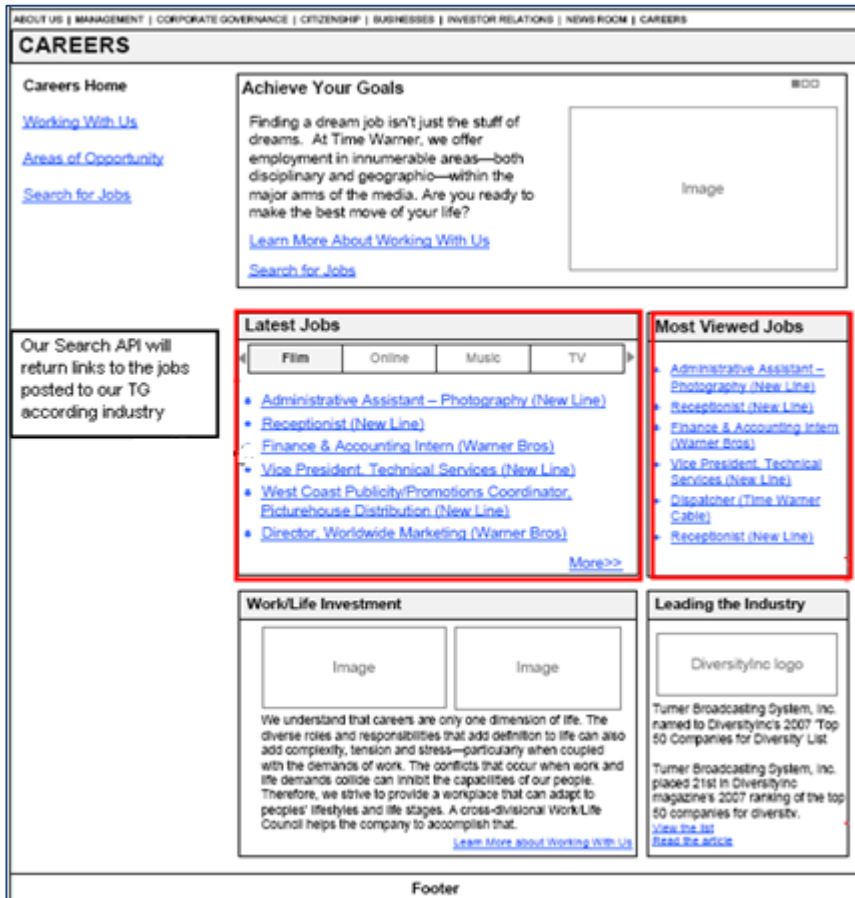
A client's requirements dictate which option works best for their particular situation.

Before You Begin

Preparing for initiating BrassRing APIs ensures successful deployment. Before you begin the process for initiating APIs for your clients, obtain the following information from your clients:

- Client ID
- UserID
- KeyName
- Encryption Key
- Specify encryption method: Advanced Encryption Standard (AES) or Rivest, Shamir and Adleman (RSA)

Client IDs and Site IDs are used to identify clients when clients are submitting XML documents. You can find ClientID and Site ID in Workbench: Tools > Talent Gateway > Edit Icon > Talent Gateway Details. Refer to Configuring the Job Search API.

If clients want to use encryption, contact your Support Team. You can also refer to Configuring Encryption for additional information.

> Use Ctrl + Click to follow hyperlinks. Use ALT + LEFT ARROW to return to point of origin.

XML Element Tag	Functionality Details
Client ID/Site ID	API web service calls use XML elements to identify the client and the client's site when sending job search requests. XML documents and elements define job search criteria. Refer to Enabling the TG to allow Web Service Access.
KeyName/ Encryption Key/Encryption Method	Use these XML elements when clients want to use encryption for their implementation. Refer to Configuring Encryption for additional information.

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

The following XML syntax 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 Start and End tags must match in case sensitivity. You cannot have a Start tag begin with an upper case letter and an Ending tag begin with a lower case letter. However, some applications you may use with XML 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 ~|~

Configuring the Talent Gateway Job Search API

All BrassRing APIs use XML documents (content containers) and elements to define what is contained in an XML request and an XML response. Whereas elements within Candidate API XML documents are generally fixed, elements within Job Search XML documents can vary. Clients can pick and choose elements for their Job Search APIs. For example, a client may want to return only the job location and not the job details in job results where another client may want to return job results using a proximity search. Each XML document can be tailored to meet each client's needs by including or excluding specific XML elements.

Job search API configuration begins in Workbench where clients enable the TG to allow the web service access, define the elements of their job search, and preview the XML output. Clients next determine the API Response Type - how the job search results display. The next step is to access the web service using one of BrassRing router addresses to establish the transport medium and invoke the web service.

If you wish to use the Job Search feature for more than one TG, you must enable and configure each TG separately.

> **Note:** Each client or third party vendor will have a preferred method (standard network protocol) to access and invoke the web service. Documenting this part of the process is out of the scope of this document.

Step 1: Enabling the TG to allow Web Service Access

a. In WB: **Tools >Talent Gateway.**

The Talent Administration page displays.

b. Select your TG and click the edit pencil icon.

c. Note the Client and Site ID.

d. Check the checkbox for:

- **Allow Web service to query this gateway**

- Require **encryption**

This setting applies to full TGs only. On a Global TG, this applies to member-level gateways only. If ***Allow Web service to query this gateway*** is later unchecked, Workbench automatically removes the check mark and disables the encryption setting.

e. Scroll to the ***Job Search engine*** and ***Job search results*** fields. Select the search fields from the **Job search engine** selection list and the job search results fields from the **Job search results** selection list.
Specifying the Search and Search Results fields.

> Job Search fields (requests) are sent to the BrassRing API as search requests when candidate defines their searches.

> Job Search Result fields (responses) are what are returned to the candidate during the job search process.

f. Click Save.

Step 2: Enabling the Preview XML Feature

a. In WB: Admin > Manage Clients > Edit Client Settings or click the pencil icon to edit the client settings.

b. Select the **Yes** radio button for Talent Gateway Search API – Enable Preview XML.

c. Click **Save**.

Step 3: Using the Preview XML Feature to Build Your Search Criteria

a. In WB: Tools > Forms > Reqs > Req Forms.

b. Select the radio button for your TG.

c. Click **Preview XML Talent Gateway Integration**.

d. Click the checkbox for each field you wish to include in your search criteria. You can also click **Select all** field to choose all of the fields. The list shows available Search Engine fields—both custom and standard. Fields displayed here are the ***Job search engine*** fields previously selected.
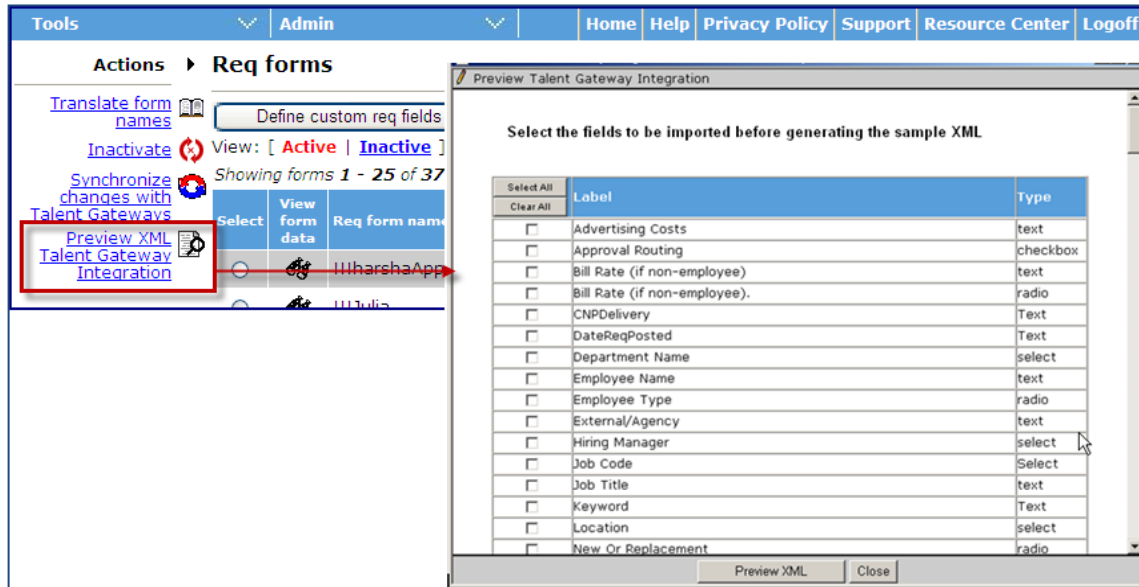
The list includes:

- Alphabetized Standard Req fields and Custom Req fields of the standard requisition template "req field index."

- Active fields only that are searchable and mapped to the Search Engine.

- The list *does not include* the translated fields (includes only the base language).

e. Click **Preview XML**.
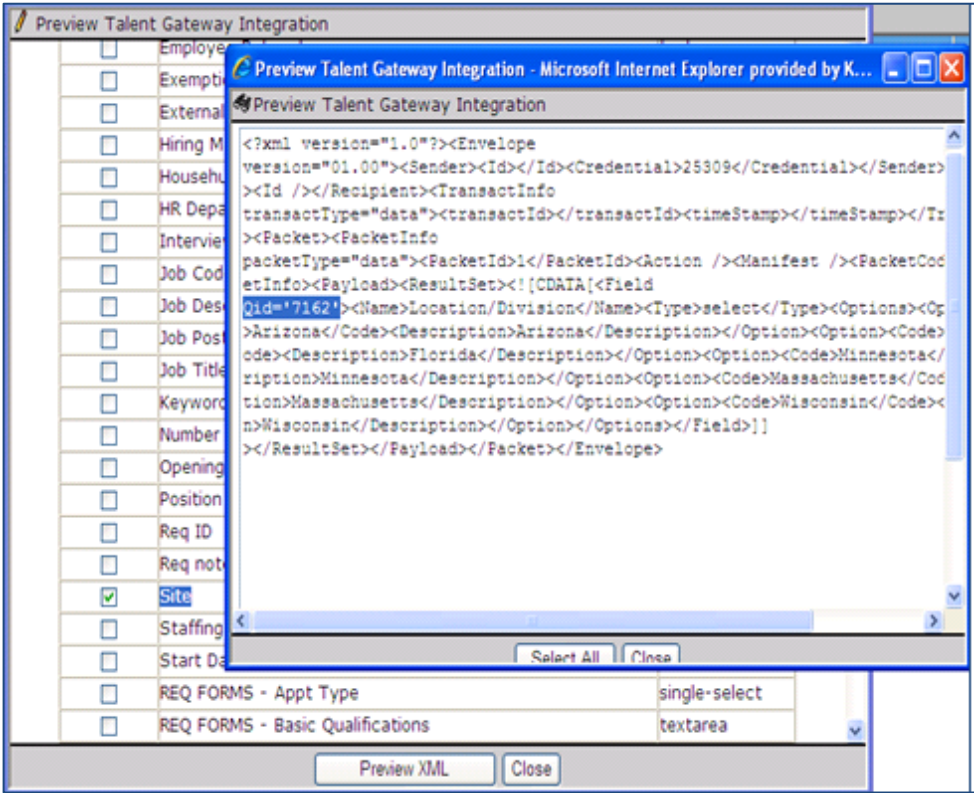
f. Click the **Select All** button, then copy (Ctrl + C) and paste (Ctrl + V) this XML into your XML Editor. This is an example of the request that is sent to BrassRing's Message Router (Web Service) to obtain results. In addition to this XML, it is necessary to insert the Input String from Step 5 (Specifying the Input String) after the '<Payload>' element identifier.

Step 4: Viewing the XML Generated for the Search

The following is an example of a web service XML template. This XML template request example shows the information request sent to the BrassRing message router. When the BrassRing message router receives this request, it responds and displays the results of the request (response).

Web Service XML Template

```
<?xml version="1.0"?>
<Envelope version="01.00">
<Sender>
        <Id></Id>
        <Credential>CLIENTID</Credential>
</Sender>
<Recipient>
        <Id/>
</Recipient>
<TransactInfo transactType="data">
        <transactId></transactId>
        <timeStamp></timeStamp>
</TransactInfo>
<Packet>
        <PacketInfo packetType="data">
        <PacketId>1</PacketId>
        <Action/>
        <Manifest/>
        <PacketCode/>
        </PacketInfo>
        <Payload>
```

BR ← Client

```
        <ResultSet>
        <![CDATA[
        <Field qid="123">
        <Name></Name>
        <Type></Type>
        <Options>
        <Option>
        <Code></Code>
<Description>
</Description>
        </Option>
        </Options>
        </Field>
        ]]>
        </ResultSet>
        </Payload>
</Packet>
</Envelope>
```

BR ← Client

Step 5: Specifying the Input String

The following XML example shows the input string's elements and the required format. The input string defines each API's specific request variables. [Input String table].

```
<InputString>
<ClientId/>
<SiteId/>
        <NumberOfJobsPerPage/>
<PageNumber/>
<OutputXMLFormat/>
<AuthenticationToken/>
<HotJobs/>
<JobDescription/>
    <outputFields/>
<ReturnJobsCount/>
    <ReturnJobDetailQues>
<QuestionsForORCondition>
        <Groups>
        <Group><![CDATA[comma separated list of questions which support OR]]>
          </Group>
        <Group><![CDATA[comma separated list of questions which support OR]]>
          </Group>
   <Groups>
</QuestionsForORCondition>
        <ProximitySearch>
        <Distance/>
        <Measurement/>
        <Country/>
        <State/>
        <City/>
        <zipCode/>
        </ProximitySearch>
        <JobMatchCriteriaText/>
        <DatePosted>
              <Date/>
        </DatePosted>
        <SelectedSearchLocaleId/>
        <Questions>
        <Question Sort="Yes/No" Sortorder="ASC/DESC">
        <id/>
        <Value/>
        </Question>
        </Questions>
<DatePosted>ALL</DatePosted>
```

BR ← Client

```
<ReturnUnpostedJobs>Yes</ReturnUnpostedJobs>
</InputString>
```

BR ← **Client**

Input String Element (field) Table

Input String Fields	Description
ClientId	The client's Identification number, also known as partner ID
SiteId	The client's site ID.

PageNumber	Options for this element field are:
	
	    1 – 50 job search results are returned per page
	
	    2 – if more than 50 jobs search results are required, use page numbers to request additional job search results using incremental page numbers
	
	For other options see ReturnJobsCount.

OutputXMLFormat	Desired XML format, either:
	
	    0—XML
	
	    1—RSS
	
	    2—Link

AuthenticationToken	Sent by the Authentication API after a successful login, otherwise remove this tag
HotJobs	This is the Hot Jobs flag that determines if the featured job's value is Yes or No

ProximitySearch	If the Proximity Search setting is on for this client, this applies to a proximity search and includes the following information:
	
	    Distance
	
	    Measurement—either **MI** (Mile) or **KM** (kilometer)
	
	    Country—The selected country ID
	
	    State—The selected state name
	
	    City—The selected city name
	
	    zipCode
	
	If the user wants to search using the proximity Search feature, then use this tag. Otherwise remove this tag.

ReturnJobsCount	Returns the number of jobs clients want in one request. This element should NOT be used when requesting job description or job details fields. If you want all jobs to be returned in one request set the value as TG_SEARCH_ALL. If you want some specific job count, enter that count in this node. If you want default 50 jobs per page then do not specify this node in the input xml. Refer to PageNumber.

Input String Fields	Description
ReturnJobDetailQues	This node contains the list of comma separated questions which client want in search output and those are configured as Job Details Question in talent Gateway.
JobDescription	This is the Job Description flag that determines if you need job description to be sent in output. YES – Job description field is returned in the output. NO – Job description field is not returned. If this node value is YES then it will ignore the value entered in the ReturnJobsCount node and will just return 50 jobs at a time.
outputFields	This node contains comma separated list of question Ids (fields - common Questions across all sites) which client has configured in workbench. This input parameter is optional. These can be configured at: WB -> Tools -> Settings -> Social media configuration ->Administer job widget -> Select output fields
JobMatchCriteriaText	This job match criteria text displays if the **EnableJobMatch** setting is on for this client. If the user wants to search using the Job Match Criteria feature then use this tag; otherwise remove this tag

DatePosted	This indicates the requisition posting date(s) to be included in the search, and is one of the following:
	
	    All posting dates—Use the value **ALL**
	
	    Includes All jobs updated after a specific date in **yyyy-mm-dd** date format

ReturnUnpostedJobs	The ReturnUnpostedJobs node allows clients to retrieve and display unposted jobs as of a specified date.
	
	Values for this node:
	
	    Yes
	
	    To request all posting dates - —Use the value **ALL**
	
	    Specific date – Enter date in yyyy-mm-dd format
	
	ReturnUnpostedJobs response includes Job IDs and date and time of unposting. Unposted jobs will be in addition to posted jobs (25,000 maximum).

SelectedSearchLocaleId	This is the Locale ID and is present for Global TGs when the user wants to search on different locales. to view LocaleIDs, click here.
OutputFields	Question Ids (QID) of fields that the client wants as an output fields. This ID should be mapped to BrassRing's Search Engine.

Input String Fields	Description
Questions	The question node is based on the values entered/selected by the user on the Search screen. Options include:
	Question Sort—either **Yes** or **No**
	Sort order –either **ASC** (ascending) or **DESC** (descending)
	The Sort option is available on only one field in the search criteria. If you specify more than one sort item, sorting is performed on the last available field in the search criteria. If you do not specify a sort order, the default sort order is used, which is the Last Updated time of the req.
	If the user selects the **All** option, use TG_SEARCH_ALL, otherwise use the actual selected option's Answer Value.
	If the user selects more than one option code, then those codes should be added into the value tag and should be separated by commas.
	To obtain the ***question ID*** for building your search criteria, refer to the Workbench Preview XML section of this document.
QuestionsForORCondition	• For example, if 1304 is the Keyword, 1323 is the primary Category, 1325 is the secondary Category, 1336 is primary location and 1339 is the secondary location. The values are provided in the \<Value\> node as usual.
	• In addition to that the ids 1323 and 1325 are provided as a group in the \<QuestionsForORCondition\> node. Similarly, 1336 and 1339 are grouped together.
	• So, this search API will result in all Jobs that have Keyword as "JAVA" AND (Primary Category OR Secondary Category as "Human Resources") AND (Primary location OR Secondary location as "Boston").
	This OR based logic should be applicable only to Search Opening questions and exclude following:
	• Proximity Search fields
	• Keyword
	• JobMatchCriteriaText
	• Date Posted

Sample Input String from the Client to the Message Router

The following XML demonstrates a web service call requesting one page of job search results. For information on element fields, refer to the Input String Element Table.

> If an element does not contain any data, the opening and closing tags can be combined as shown for \<HotJobs/\>, \<JobDescription/\>, \<ReturnJobCount/\>, etc.

The following XML code describes the request input sent by the Client to the Message Router.

```
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
<Sender>
        <!—Valid Employee ID. This is mandatory -->
        <Id>10540</Id>
        <!—Client Id.  This is mandatory -->
        <Credential>11721</Credential>
</Sender>
<TransactInfo transactType="data" >
        <TransactId>10/24/2008</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
```

BR ← Client

```
</TransactInfo>
<Unit UnitProcessor="SearchAPI">
        <Packet>
        <PacketInfo packetType="data">
        <packetId>1</packetId>
        </PacketInfo>
        <Payload>
        <InputString>
        <ClientId>11721</ClientId>
        <SiteId>6129</SiteId>
        <NumberOfJobsPerPage>30<</NumberOfJobsPerPage>
        <PageNumber>1</PageNumber>
        <OutputXMLFormat>0</OutputXMLFormat>
        <!—If you have valid token from login API then use otherwise remove this tag-->
        <AuthenticationToken/>
        <HotJobs/>
         <JobDescription/>
         <!—For all jobs mention ALL otherwise any number of jobs if do not want to use
         then so not include this node in input xml →
         <outputFields/>
         <!—Comma separated list of search field questions selected in workbench→
         <ReturnJobsCount/>
        <!—Comma separated list of questions which are configured as Job
        Details fields in Gateway and client want those in search output→
         < ReturnJobDetailQues/>
        <!—If not using the proximity search then remove this ProximitySearch  tag-->
        <ProximitySearch>
        <Distance/>
        <Measurement/>
        <Country/>
        <State/>
        <City/>
        <zipCode/>
        </ProximitySearch>
        <!—If not using the Job Match Criteria then remove this tag-->
        <JobMatchCriteriaText/>
        <SelectedSearchLocaleId/>
        <Questions>
        <Question Sort="No" Sortorder="ASC">
         <Id>1323</Id>
         <Value><![CDATA[TG_SEARCH_ALL]]></Value>
        </Question>
        <Question Sort="No" Sortorder="ASC">
         <Id>1304</Id>
        <!—This is the option code of the question you want to search on-->
         <Value><![CDATA[TG_SEARCH_ALL]]></Value>
        </Question>
        <Question Sort="No" Sortorder="ASC">
         <Id>1292</Id>
         <Value><![CDATA[TG_SEARCH_ALL]]></Value>
        </Question>
        </Questions>
        </InputString>
        </Payload>
        </Packet>
</Unit>
</Envelope>
```

BR ← Client

Configuring Encryption

Use the instructions below to begin configuration:

1. Client requests a share access key by completing a Hive Ticket.

- The most commonly used name for this key is Integration.

- Specify encryption method: AES or RSA

Your Support Team representative completes the Hive ticket and sends you the required values for your API configuration:

Client Name, UserId, KeyName, and Encyrption Key. These values are used to set up your API.

Sample Input String in Advanced Encryption Standard (AES) Encrypted Format

The BrassRing web service message router supports AES encryption. This follow example shows a sample input string sent from the Client to the Message Router in AES encrypted format. To use this functionality, define the AES keys and use those keys to encrypt the input string to BrassRing. When BrassRing receives the input string, it decrypts and reads the data. When the message router responds with the output string, it encrypts the data using the same key.

> To use AES encryption, the **Require encryption** setting in Workbench must be enabled.

```
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
<Sender>
        <!—Valid Employee ID. This is mandatory -->
        <Id>10540</Id>
<!—Client Id.  This is mandatory -->
        <Credential>GodgiyZlqRvShLWJjD8zZA==</Credential>
</Sender>
<Recipient/>
<TransactInfo transactType="data">
        <TransactId>10/24/2008</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="Decryptor">
        <EncryptedData xmlns="http://www.w3.org/2001/04/xmlenc#"
Type="http://www.w3.org/2001/04/xmlenc#Element">
        <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:KeyName>Integration</ds:KeyName>
        </ds:KeyInfo>
        <CipherData>
        <CipherValue>Uy93UnZMVFIrZHIvMDBFQmcvbFhCZz09a0xQak50WTlEZzE0Qn
ZpcTJlS0QxRUxoanEzT3JyKzVSVi80cUhoQkcwcXFIK1NTN2o0ZUc3TXBjNndPSUhDaV
o0QnFZMlNQWkk5NkUwM2N4eHRSMzF5R1QwV0MrRWYxY0pQVVRXSm1DZIIrbmdIb2
ZpdW1WY1FweS9KMWUwU2tsTkF6UHdPd0RUNDlkcWQ0SW5sYWjRwanQwOHlMMU8v
L3VhbE9lMTIMlCamF5ZTV1ekpzak02OG9iV1BYSngzT3Bva0czS2xYK0dlYjdUc25hRUFTLz
B0a3pFWE9iYnRqWEljaalYvZXh5VzJNWTVKNXFpZzZQ2TFRnaTdooaUhicSsvb0NJRGU5e
C9Rd05PM1NkL1k3VVlxdlF3MlgwMG1rVFc0aG1TdVpoUWRNRNQ0JhY0t3ZmsrNllzWittMEI
vVkRJTUhOL2ZLeitrQ1lmUUFUFUZStITzlhUXRBbHUwMFlETE5WUEE0amtYUTVVpd1UyeX
RXcFlTalJhczNxVmcyK2RWdlUWFhKa0l5MW9LWWRRabmsyQ09DSEFYMWozbkV1eX
RJL25TdTRFNUhhYak5LS3kzb0FHMXMvZXE4S0tlbGddhNTdFTGRIVCt1Y1YdPV3d2OStjd0
12TWJFMHNnVmd1akNRcmdTL3g5OTMvQ0RPRW5FFQlh6bVB3cW1FMWpReXE0b1B2
VGpzdTZ2VEpSZ2U4TGJTc2hsQnUxZnoyaXpLU3VJUmtQNGdRZnAyOWVTdjJwwRzJqZz
RFRmxiZ2h2MTVnem1uaXlQbmNNNWS90ajdPcEZhQm5zSExCbjRHcjFxTHJMaGGFqQkwx
RC9Ha3BrV2wzdnU4UWoxbEJWZGxNUmlEZnRrYVpNUDM2aFhmMTB5V3gzZytuMndB
cERJZ0hNL3ZyY1hjZNU0zaFJQcGh3Mzh2RVVvvM0hpU0k4a2NNLa3JjSmxrdWXaaTUpjQ21J
RGZiaW4yRkNNubjNoTUM4NE1MMHH5bUFINWhmOE8zVW5vM2IFL3ptVWVVbkk1LzVu
Y09XcTlIQK3dRVWVKKRUx2dHRLZWc0bXZyYU40c1JppQ1ZjSThxeVUzeHhhPVzI2M1oxL3c
0MFk3OWpqMkkl0T3prOWlxYVRmdkkhxd0ptUUNVcWdhUlplNVRaL2poR1lnWjlQMFZxRjln
dUhoRDQySTJPam1RU1RRMTZmdm9sWGVYNEVXVzlGOEZ4c0hwcSSS9QL1JYWGxZZn
I2VlQrMnNobWV6ZnU3VjBnVVQyOFFFQVEVPbjEzSzFKVlFjcVpQQR2lQbU9Od0dyMloydW</CipherValue>
```

R2ek5rTEJ1OUppSmdIdUhPV2pNRjlFWkVwNDJ1b3Jpd3lCS2RWSG1Ra09zOFovWW53
b0RqbFZ6SVNXS2NsMDV5a004M0w4MFpuTWx0eG5HTXNZYmZEeDhGTCtOZXVIZkth
QjNlYUNvM1lLRG5PU0FUdkh2OEJ4TG04eWJ3bGZicGFOOVVEK2NVQUxNeWhMKys0
Nnh6SGZOUHFIRUhpYkZTRVZudWFycFkxZE9Tb0lENDMwUVpkNUd0c0o1V3E4eGNIV
mNEK1FBNEQ0ZnhXTlg0V0hiakEzVVVGR2dva3gyUDY3bWtUGxSRUU4aVhpZEV6czU
vbFRvWnRyS0x1VEJqWUQ0VVFOOHVwOFJmcUJYQnpjNzhXeDR5SVp6MjhiVFJSQVJj
a2FIMnRBVGRHRUG1HSlo4YmJvVCtISzZwWExWd0dIemgzdS9lbkNMdDN3SCtkWHBvSn
FVbzdnSEgzcEJNa3FrV1Qrcy9pZU9VY3FMbnQzS0t6K2ptaDhrbS9DSU9heVduUk8rNU9
uMURRmRGh5WHRLSnNZaE1aV3RvQTUrNC9uRjh0NmtSR3JMeG9pZFhhCTVhuRDY1TG
RHclJ5dEdDdVUxK3BZdU1ia2hlNHpPaUDJPczA3UEhSeHdUWStvbWZ2aWZFVnZtMnpC
UHJDY3FQM0s3NnBzcjg5UE9LaHJLR25VdEpXSm1WZVlocjVPUGRFVFV5SkdRRUZjTy
9qV2RBV096OENHRGdCKzUxbk1qWllERmMyNlhGZG02djA0eW52b2Uxc3B4Nm5sVWJ
yMnpoVDlHRGxEQkhMTXVacWhRZGhJUjlYUjFGRFdxY2V1MXlSeSt6SFhaZkxGemMzZ
283Y2hOemJYdXQrVzNObm5tanYvS3J0NE1lbHdBQTZJVU81amhvMnNBb29wbzNxNTU
2anZYVmxMMEVxbkZ1Q3lpVjVhaXp5SFZ1MFZrNmFRSW9xSFc1UUtuclJZUnk4MndZY1
E2dEtBdWNtZzNabkZVc2kzOW9DM2hScURwUXlWN3F6dGd2aDZibS81NDJvMXdyb0Fw
ajMwcEdMdHhqYjZscm9mMVFNc3J6amJqR2dPU3ppb0Q2Z3ZKQjFaVFVQVERRbWtwQ
k5NZ1k5MENNncmd4eTBPU2NmYkJkZ1I1cG1jNDVnaDc4U1pBS3NNeW54dkZnenVlTml
MWGx3Ykd3TWNJSkEzcVhCSHZTc00xbUxiMFFwcDhZSVFqM0ZTWUd1ZDZ3aDdrSFN
JWWdEL25HY3NhM3JIUE5TQmRobkhBcXRZN09HUXQ1UEE0RjIxSXdhZFN6bStEUTk4
bW9lL0NyMDRnTGdTRXJKcUFzb0ZwT0MxaGJCSENNwYWxyWFAzMEoxNHhUR0pUMX
FHaUdyTlFENEZrTWN6Q09TNDZVUXpERU9OS2ZhbjdYZUpjaXBwdlF2aDdzREcrVzB1b
IA2cExSYUhDMm1yRDY=</CipherValue>
        </CipherData>
        </EncryptedData>
</Unit>
</Envelope>

BR ← Client

Step 6: Accessing the Services

The BrassRing web service uses a message router address as the entry point for the APIs. Each client will use their own method for accessing the message router to submit their search request.

When a client accesses any one of these message router addresses, the router internally invokes the web service. The invoked web services then performs these functions in chorological order:

- Validates the client credentials (sending into the <Sender> tag)
- Validates the input xml (if any node is incorrect or there is an invalid xml string)
- Performs the encryption/decryption if client has enabled encryption.
- Sends input string to web service
- Receives search results from web service
- Returns search results to client

Your Team Support member can provide clients with a shared key.

Message Router Addresses

The BrassRing message router addresses are:

US – Staging Environment:

http://stagingimport.brassring.com/WebRouter/WebRouter.asmx

US – Production Environment:

http://Import.brassring.com/WebRouter/WebRouter.asmx

EU – Staging Environment:

http://stagingkrb-jobs.brassring.com/WebRouter/WebRouter.asmx

EU – Production Environment:

http://krb-jobs.brassring.com/WebRouter/WebRouter.asmx

Web Service Description Language Paths

The Web Service Description Language (WSDL) paths for message router web services are as follows:

Staging Environment:

http://stagingimport.brassring.com/WebRouter/WebRouter.asmx?wsdl

Production Environment:

http://Import.brassring.com/WebRouter/WebRouter.asmx?wsdl

Step 7: Reviewing the API Service Sample Output Strings

The web service returns search results in an output string to the client in one of four formats. When the client sends a search request to the message router, the input string specifies the format of the output string, or how results are returned to the client. For further information regarding input string descriptions, click here.

Web Service Output String Formats

- XML format

- RSS feed format

- Proprietary link format

- AES Encrypted for proprietary XML format

Web Service Output String Format Examples

The following section contains XML output string format examples returned by the BrassRing web service.

XML Format

The following example shows an output string in XML format.

```xml
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
 <Sender>
        <Id>123</Id>
        <Credential>25082</Credential>
</Sender>
<TransactInfo transactType="response" transactId="1fba18f3-60b0-46e6-b1f0-66510baff4e1">
        <TransactId>10/24/2008</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="SearchAPI">
        <Packet>
        <PacketInfo packetType="data">
        <packetId>1</packetId>
        </PacketInfo>
        <Payload>
        <InputString>
        <ClientId>25082</ClientId>
        <SiteId>5038</SiteId>
        <NumberOfJobsPerPage>30<</NumberOfJobsPerPage>
        <PageNumber>1</PageNumber>
        <OutputXMLFormat>0</OutputXMLFormat>
        <AuthenticationToken/>
        <HotJobs/>
           <JobDescription/>
           <outputFields/>
           <ReturnJobsCount/>
   < ReturnJobDetailQues/>
        <ProximitySearch>
        <Distance/>
        <Measurement/>
```

BR ⟶ Client

```
              <Country/>
              <State/>
              <City/>
              <zipCode/>
              </ProximitySearch>
              <JobMatchCriteriaText/>
              <SelectedSearchLocaleId/>
              <Questions>
              <Question Sort="No" Sortorder="ASC">
              <Id>6555</Id>
              <Value>TG_SEARCH_ALL</Value>
              </Question>
              <Question Sort="No" Sortorder="ASC">
              <Id>6754</Id>
              <Value>TG_SEARCH_ALL</Value>
              </Question>
              <Question Sort="No" Sortorder="ASC">
              <Id>6557</Id>
              <Value>TG_SEARCH_ALL</Value>
              </Question>
              <Question Sort="No" Sortorder="ASC">
              <Id>6558</Id>
              <Value>TG_SEARCH_ALL</Value>
              </Question>
              <Question Sort="No" Sortorder="ASC">
              <Id>7386</Id>
              <Value>TG_SEARCH_ALL</Value>
              </Question>
              <Question Sort="No" Sortorder="ASC">
              <Id>6932</Id>
              <Value>TG_SEARCH_ALL</Value>
              </Question>
              <Question Sort="No" Sortorder="ASC">
              <Id>6455</Id>
              <Value>TG_SEARCH_ALL</Value>
              </Question>
              </Questions>
              </InputString>
              <ResultSet>
              <Resultset>
              <Jobs>
              <Job>
              <Question Id="6928">211BR</Question>
              <Question Id="6936">412 ACCOUNT CLERK II, LACERA </Question>
              <Question Id="6952">ACCOUNT CLERK II, LACERA</Question>
              <Question Id="6932">Human Resources </Question>
              <Question Id="6405">Open Continuous </Question>
              <Question Id="6455">&amp;nbsp;</Question>
              <Question Id="6410">&amp;nbsp;</Question>
              <Question Id="6411">&amp;nbsp;</Question>
              <HotJob>Yes </HotJob>
              <LastUpdated>Date and Time </LastUpdated>
<JobDetailLink>http://dev3-tgweb-
02.brassring.com/1033/ASP/TG/cim_jobdetail.asp?AuthToken=dfdsfsdfdsff&amp;partnerid=250
82&amp;siteid=5038&amp;jobid=12607</JobDetailLink>
              </Job>
              </Jobs>
              <OtherInformation>
              <TotalRecordsFound>19</TotalRecordsFound>
              <MaxPages>1</MaxPages>
              <StartDoc>1</StartDoc>
              <PageNumber>1</PageNumber>
              </OtherInformation>
              </Resultset>
```

BR ⟶ Client

```
        </ResultSet>
        </Payload>
        <Status>
        <Code>200</Code>
        <ShortDescription>Successfull Search API Call</ShortDescription>
<LongDescription>The Search API call has been successfully completed and returned the
results.</LongDescription>
        </Status>
        </Packet>
        <Status>
        <Code>200</Code>
        <ShortDescription>Successfull Search API Call</ShortDescription>
<LongDescription>The Search API call has been successfully completed and returned the
results.</LongDescription>
        </Status>
</Unit>
<Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>All units are processed with success codes</LongDescription>
</Status>
</Envelope>
```

BR ⟶ Client

RSS Format Sample Output

The following example shows an output string in RSS format.

```
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
<Sender>
        <Id>10540</Id>
        <Credential>11721</Credential>
</Sender>
<TransactInfo transactType="response" transactId="827db146-c576-4b2b-b997-
9eb59a7ea7ec">
        <TransactId>10/24/2008</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="SearchAPI">
        <Packet>
        <PacketInfo packetType="data">
        <packetId>1</packetId>
        </PacketInfo>
        <Payload>
        <InputString>
        <ClientId>11721</ClientId>
        <SiteId>6129</SiteId>
        <NumberOfJobsPerPage>30<</NumberOfJobsPerPage>
        <PageNumber>1</PageNumber>
        <OutputXMLFormat>1</OutputXMLFormat>
        <AuthenticationToken/>
        <HotJobs/>
          <JobDescription/>
          <outputFields/>
          <ReturnJobsCount/>
   < ReturnJobDetailQues/>
        <ProximitySearch>
        <Distance/>
        <Measurement/>
        <Country></Country>
        <State/>
        <City/>
        <zipCode/>
        </ProximitySearch>
```

BR ⟶ Client

```
              <JobMatchCriteriaText/>
              <SelectedSearchLocaleId/>
              <Questions>
              <Question Sort="No" Sortorder="ASC">
              <Id>1323</Id>
              <Value>TG_SEARCH_ALL</Value>
              </Question>
              <Question Sort="No" Sortorder="ASC">
              <Id>1304</Id>
              <Value>TG_SEARCH_ALL</Value>
              </Question>
              <Question Sort="No" Sortorder="ASC">
              <Id>1292</Id>
              <Value>TG_SEARCH_ALL</Value>
              </Question>
              </Questions>
              </InputString>
              <ResultSet>
              <rss version="2.0">
              <channel>
              <title>Stryker Job Postings</title>
<link>https://sdev3-tgweb-
02.brassring.com/1033/ASP/TG/cim_home.asp?partnerid=11721&amp;siteid=6129</link>
              <description>Lists the requisitions actively posted on a job board</description>
              <language>en-US</language>
              <item>
              <title>test</title>
<description>Division:  Marketing&lt;P&gt;Location (Country - State):  Boston&lt;P&gt;City or
Sales Region:  MA&lt;P&gt;Business Function:  Sales &lt;P&gt;Requisition-Number:
3908BR&lt;P&gt; &lt;P&gt;HotJob:  Yes&lt; &lt;P&gt;LastUpdated:  2009-01-
20&lt;</description>
<guid>https://sdev3-tgweb-
02.brassring.com/1033/ASP/TG/cim_jobdetail.asp?AuthToken=sadsadsadsd&amp;partnerid=1
1721&amp;siteid=6129&amp;jobid=146489</guid>
<link>https://sdev3-tgweb-
02.brassring.com/1033/ASP/TG/cim_jobdetail.asp?AuthToken=sadsadsadsd&amp;partnerid=1
1721&amp;siteid=6129&amp;jobid=146489</link>


               <pubDate>Fri, 07 Mar 2008 05:00:00 GMT</pubDate>
              </item>
              </channel>
              </rss>
              <OtherInformation>
              <TotalRecordsFound>1</TotalRecordsFound>
              <MaxPages>1</MaxPages>
              <StartDoc>1</StartDoc>
              <PageNumber>1</PageNumber>
              </OtherInformation>
              </ResultSet>
              </Payload>
              <Status>
              <Code>200</Code>
              <ShortDescription>Successfull Search API Call</ShortDescription>
<LongDescription>The Search API call has been successfully completed and returned the
results.</LongDescription>
              </Status>
              </Packet>
              <Status>
              <Code>200</Code>
              <ShortDescription>Successfull Search API Call</ShortDescription>
<LongDescription>The Search API call has been successfully completed and returned the
results.</LongDescription>
              </Status>
</Unit>
```

BR ⟶ Client

```
<Status>
      <Code>200</Code>
      <ShortDescription>Success</ShortDescription>
      <LongDescription>All units are processed with success codes</LongDescription>
</Status>
```

BR ────▶ Client

Proprietary Link Format Sample Output

The following example shows an output string in Proprietary Link format.

```
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
<Sender>
      <Id>10540</Id>
      <Credential>11721</Credential>
</Sender>
<TransactInfo transactType="response" transactId="64ec6add-81c2-40ea-a879-
34bfa929a648">
      <TransactId>10/24/2008</TransactId>
      <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="SearchAPI">
      <Packet>
      <PacketInfo packetType="data">
      <packetId>1</packetId>
      </PacketInfo>
      <Payload>
      <InputString>
      <ClientId>11721</ClientId>
      <SiteId>6129</SiteId>
      <NumberOfJobsPerPage>30<</NumberOfJobsPerPage>
      <PageNumber>1</PageNumber>
      <OutputXMLFormat>2</OutputXMLFormat>
      <AuthenticationToken/>
      <HotJobs/>
          <JobDescription/>
          <outputFields/>
          <ReturnJobsCount/>
   < ReturnJobDetailQues/>
      <ProximitySearch>
      <Distance/>
      <Measurement/>
      <Country></Country>
      <State/>
      <City/>
      <zipCode/>
      </ProximitySearch>
      <JobMatchCriteriaText/>
      <SelectedSearchLocaleId/>
      <Questions>
      <Question Sort="No" Sortorder="ASC">
      <Id>1323</Id>
      <Value>TG_SEARCH_ALL</Value>
      </Question>
      <Question Sort="No" Sortorder="ASC">
      <Id>1304</Id>
      <Value>TG_SEARCH_ALL</Value>
      </Question>
      <Question Sort="No" Sortorder="ASC">
      <Id>1292</Id>
      <Value>TG_SEARCH_ALL</Value>
      </Question>
      </Questions>
      </InputString>
```

BR ────▶ Client

```
        <ResultSet>
        <!—Replace &amp; with the & sign -- >
<Link>https://sdev3-tgweb-
02.brassring.com/1033/ASP/TG/cim_searchresults.asp?AuthToken=sadsadsadsd&amp;Functi
on=LinkQuery&amp;LinkId=18</Link>
        </ResultSet>
        </Payload>
        <Status>
        <Code>200</Code>
        <ShortDescription>Successfull Search API Call</ShortDescription>
<LongDescription>The Search API call has been successfully completed and returned the
results.</LongDescription>
        </Status>
        </Packet>
        <Status>
        <Code>200</Code>
        <ShortDescription>Successfull Search API Call</ShortDescription>
<LongDescription>The Search API call has been successfully completed and returned the
results.</LongDescription>
        </Status>
</Unit>
<Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>All units are processed with success codes</LongDescription>
</Status>
</Envelope>
```

AES Encrypted for Proprietary XML Format – Sample Output

The following example shows an output string in AES Encrypted for Proprietary Link format.

```
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
<Sender>
        <Id>123</Id>
        <Credential>Kq1LfgDx8RzLeHkYXVHGIA==</Credential>
</Sender>
<Recipient/>
<TransactInfo transactType="response" transactId="3bfe2ca7-59ac-421c-a992-
8e281235c0ee">
        <TransactId>10/24/2008</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="Decryptor">
<EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element"
xmlns="http://www.w3.org/2001/04/xmlenc#">
        <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:KeyName>Integration</ds:KeyName>
        </ds:KeyInfo>
        <CipherData>

        <CipherValue>aUl1ZlY0RDdWRllZV05HODdOaXg1dz09QllQS0FVcVpNYllSem5vR1Rr
aDJQL2RKMi96aDJBVkcydHYyYXEyTVhrY3FVN3hEVGVUZGhSZmx6MW4wdFNGYWtPdVJ
BOG9uUDJndGRucHphcjVYS0ZoWUFjTmhGSll2dFZZMbnVjdnlEYktYMkRITkpzSnREbDlvMW
N4cEJENUJXM3dmTTRZWUZpOEFYcW03YnphaGNlUk5xGJqSUhJWXlRSDljVjNlcHVtQlFQ
Yk9oME45NWhlWGFTNElUQnBMQmU5cUp3L2d5aFBEdU1oVkk1aTVVOTzJSOTR2RDFDddm
FHWWRlZmtvZXMrZ3Y2NU40V1hkkM3NXMGdaTTFyV0w0WUpGNUllKRWdFSjBqBqV0did3JhQ
3B1V2NUYXpZWVMzZUZOVzB0NXBldEJ4VTcvRWZsTzh2cjFYcnVSOUltRTVyZXhhGenpDYk
JLL3lBZzdVTTlHK1JssbE04a1VMUUIlWjVlRTZxZkQ2aTgxc1BvUmRUcXowwWlllY05KdnF5NThk
TEYrN0pROWhmYytNMDcyU1lGazBKSCtYcEIrRGhjUUI1OGdYUHlvdWhQNHR5L1JEUXk5Q
203VUwwRkpsbmt4d0xpNGw3SzBJTytVWlBSU25wRHHrwa3hTWmM0Tzd4dW40RUM1bWhP
dGxXa0lYYWWNZOTRPRzhWYVhOVStWMG1UaHpodGGZRVEzsYVd4TDc1Z0JCeHZHZXb00xU3
```

QxWUxRb0lsbWxSdE13TGtDZnRtMFNwQjg5ZXEyOEFOQlBmVkNrdmlpbWRVZFVIVndaL1d
FVjVnRE1ubHNWMkdVRldwRXhmcFh3MzhvelJjYmhQYU5mcjFpUkdiNWZBcDl2Sm9CTm50e
kZlWUFWUmV6S3laaNUZ1UVpaaGRDMnMrQndPa3RTQmhsMkZpZ1N6aE1WRk90Njc4eDNB
Q1ZmZlZZT3dNcUhJN0JHQmJkeDR4SzlaVnhmU210Q0FjVVBBbno3SFlxL2NuRGJGTE81Uz
ZZdlFJMnU5VzFzTy9JYnd6NFVJUnFXSU1qK01KMGx6T3N3NUlTUEpLejVuRnpjbVJGJR3djU
HVWSWhUcXlXdHM4dUFHaNjNmYXltNjdVeFpicnJ4NmpiczlycY2dVp1dTB1L2IyVmZ5dzhQT
DB2RTZtOHZSMmRzODI0V0xRSDdGc0hKMWE0blV6U0xiUXcxamRHNE50Yndzb1c0ekVEZ
zVoSloxeFlxelZML0F3Sy8ya2lQYzZyK0t5OW1xcmkwanl0aHhzOGlMNmlUWHVWTEVtN3BH
R3ZiVTRnakh1OUlEeUNCQSt1OENJZytoZ0hGVTlGRkpzV0I2RHpjeGVRL3VVSWVJS0p4MH
hLcm5qL2VxaTY1Z1VqdjlzVWorNHFJaU11aHUzcnBsQnBGbDhDNWltRWwyTHljS3ZnYUhpR
UNoTUpObVlMZkZHU2RvTkxZbnlwa2lrR1RjNlhiSVhvQ0pNSElCaFc1a21lSnlTUWQ4SU5SU
GN4aTliK0xOdDFVNVBFd0llOXczcXNyQjNveHlNbTc1KzRJWU5wRllSeC9iWVVoVDloZyt4Y3
R0dS8xVUZhQitSZU1KQURjaHZiN2N3T05qamF3RXJNRmVya2tOUUhyMmhGWVhBRjZVND
ArdHllaFM2cHhsVTJVQTFHUzEyeXY4c2MvaHNyb080OFAxajUyVE5pdDFvcFpkRC93RFFKa
WlJd0gwajAwZUVJK1d1SGVmSG1aQ1U2OWNyLytLRDRNem85Nk5PYXJFNFI2TWhreVZsV
WEyRTh5L1pic3VSOERQL2lLR29IL1lBZHhIbGs1RC9pd1VxTHV0Z01rQ2xlVXRRZFFMMy92
dG4wd1dKWTJIb216WXdmTzE2NndzaWVFY2Q3N1pmYzEvUVVHOFd4dlZKcUNNQeXorZjRwL
2VPaEZkRzkzN2FyUXBoSnZoZS3ZQM24wUS80K2xkb1JnQXYzeHhmYUZad2tPTzdkR21qV3p
XTFM3djRXWHIwOFNGSW1lNnhzSb3VYa05MaVR0czlXYzlaRzZGbzBKd1pKUVdWYmgwSDd
rWHRCK0Zqb2xoVk5MTWNkRm51eVJBdkJQTEJQL2d0S3JON3dJWUw1VDVIaHZ4NTI3TlB
PRFFqd21ISG4yWVdyV1Y4SXJnREF5QU9BVW91Z2RoMTBudS9ISXBDZ1JUbzIzSW5VMEE
xU0x6dUg1elN3dy94a1Fmei9BY3lobXppUXRNbXdkJZ0xqL2VGTnJhMmE0M1pTNVhiV3hpam
N2M3FrTitlNVZMUjJ2V2p3aWdUUGRNbmRwRUUxbDhlKzZxQ1hscTE2SkxFeTE4Q1Bpd1k5V
UJ3UFVXdFBMaXp6dHB2K3JlMzkI1MlV14NGNGN2UyZlpzbWxRYVVnbEFReHZvcTQ1YjB0a
FhOME0rMGtvM1NiL2dBMnJSMmVEYzRaZnlLbkFkaXo3Ui9QWXliRnR3c1RtMXNDZDdTSXI
wbW12aE4xaHJmQk9rZzUwa254TSszL3B1SzhLLS2x3UT09</CipherValue>
         </CipherData>
         </EncryptedData>
         <Status>
         <Code>200</Code>
         <ShortDescription>Message decrypted and processed without
errors.</ShortDescription>
         <LongDescription/>
         </Status>
</Unit>
<Status>
         <Code>200</Code>
         <ShortDescription>Success</ShortDescription>
         <LongDescription>All units are processed with success codes</LongDescription>
</Status>
</Envelope>

BR ——→ Client

RSS for Talent Gateway Web API

You can also use the API to obtain search results as an RSS feed. RSS web feeds (also known as XML store) allow clients to update their content on their websites continuously. Candidates can choose to subscribe to these web feeds for updated information.

> **Note:** TGs do not support RSS feeds, but clients can set them up on their sites.

The following is an example of an RSS web feeds on a client website.

To configure RSS feeds on a client's website, clients needs to:

- Enable the following two TG client settings:
    - Talent Gateway Search API – Enable Preview XML
    - Allow Web service to query this gateway
- Save the candidate's search criteria
- Configure API calls to BrassRing's web service
- Save the results into an XML file
- Provide that link to the candidates for their RSS reader
- Update the XML file at appropriate intervals.

To configure RSS feeds:

- BrassRing sends the RSS.xml to the Support Team or Client.

- Client creates an HTML page using this RSS.xml and sends BrassRing the link for that page with an image.
- BrassRing updates the link/image on Talent Gateways.

RSS Sample Input XML sent to the Web service

This following RSS XML example requests all the featured jobs that appear on the TG:

```xml
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
<Sender>
        <Id>10540</Id>
        <Credential>11721</Credential>
</Sender>
<TransactInfo transactType="response" transactId="827db146-c576-4b2b-b997-
9eb59a7ea7ec">
        <TransactId>10/24/2008</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="SearchAPI">
        <Packet>
        <PacketInfo packetType="data">
        <packetId>1</packetId>
        </PacketInfo>
        <Payload>
        <InputString>
        <ClientId>11721</ClientId>
        <SiteId>6129</SiteId>
        <PageNumber>1</PageNumber>
        <OutputXMLFormat>1</OutputXMLFormat>
        <AuthenticationToken/>
        <HotJobs/>
        <ProximitySearch>
        <Distance/>
        <Measurement/>
        <Country></Country>
        <State/>
        <City/>
        <zipCode/>
        </ProximitySearch>
        <JobMatchCriteriaText/>
        <SelectedSearchLocaleId/>
        <Questions>
        <Question Sort="No" Sortorder="ASC">
        <Id>1323</Id>
        <Value>TG_SEARCH_ALL</Value>
        </Question>
        <Question Sort="No" Sortorder="ASC">
        <Id>1304</Id>
        <Value>TG_SEARCH_ALL</Value>
        </Question>
        <Question Sort="No" Sortorder="ASC">
        <Id>1292</Id>
        <Value>TG_SEARCH_ALL</Value>
        </Question>
<![CDATA[TG_SEARCH_ALL]]></Value> </Question>   </Questions>
 </Questions>
 </InputString>
 </Payload>
 </Packet>
 </Unit>
 </Envelope>
```

BR ➝ Client

RSS XML Feed Tags

This table describes the XML tags used in the RSS XML document.

RSS XML Tags	Description
ID	Employee ID of an active BrassRing user. The client's Identification number, also known as partner ID. This is available in WB > Tools > Users > Administer BrassRing users > Edit user > Employee ID.
Credential/ClientID	These two tags are the partner id of the TG that BrassRing needs to perform the web search.
SiteID	TG site ID.
OutputXMLFormat	This tag defines the output format that needs to be generated after performing the web search.Use the value 1 for RSS output format (0 to get XML, 2 to get links)Desired XML format, either:

Error Codes for Web Service API

Code	Long Description
100	An unknown error has occurred.
200	The Search API call has been successfully completed and returned the results.
201	The site provided does not support logins.
202	Unable to find that email/password in the system.
203	The account has been locked out due to excessive invalid login attempts.
204	The login was successful; however, a password reset is required to request any pages.
205	The site requested is inactive or does not exist.
206	The session for this authentication token has been deleted.
207	The user has logged out of the Talent Gateway. Please request a new Authentication Token.
208	The session for this authentication token has expired.
209	Password must be less than 25 characters.
210	Password must contain one of the following special characters. -~`!@#$%^&*()_+={}[|<>,.:;'?\"\\]
211	Password cannot be one of the recent passwords used.
212	Password cannot be the same as the username.
213	Password cannot contain spaces.
214	The old password entry is incorrect. Please try again.
215	The information requested is not supported by the authenticated site.
216	No StatusCheck data found for this candidate.
217	No JobCode data found for this candidate.
218	Please enter the measurement for proximity search.
219	Please enter the distance for proximity search.
220	Client Input string should be in encrypted format.
221	Client Input string should be in clear text format.
222	Please enter the state for proximity search.

Code	Long Description
223	Please enter the city for proximity search.
224	Mismatch data found for Country/State/City. Please enter the valid data for proximity search.
225	The token provided does not exist.
226	The token provided has expired.
227	The token provided must reset the password first.
228	The page requested is not supported by the authenticated site.
229	You have requested a page ID that does not exist.
230	The authentication status is not valid for this request.
231	Password must be at least six characters.
232	Number of jobs per page should not be greater than 50
237	ReturnUnpostedJobs node only accepts a Yes value. Please correct or remove.
251	Password must be at least eight characters.
252	Your password must contain at least one of the following special characters. -~`!@#$%^&*()_+={}[|<>,.:;'?\"\\]
253	Your password may not be the same as any of the five preceding passwords.
311	Your login credentials cannot be validated. You may have created an account in the past, if so please click the "forgot your password" link on the Welcome page to reset your password.
312	Your login credentials cannot be validated. You may have created an account in the past, if so please click the "forgot your password" link on the Welcome page to reset your password.
313	You cannot use your email address as your password.
314	You cannot use your username as your password.
315	Your password may not be the same as your login email address.
316	Your password may not be the same as your username.
317	Your password must be a minimum of 6 and a maximum of 25 characters.
318	Your password may not contain spaces.
319	Your password must contain at least one of the following special characters. -~`!@#$%^&*()_+={}[|<>,.:;'?\"\\]
320	Your password must be a minimum of 8 a maximum of 25 characters.

Code	Long Description
321	Suspected replay attack - 2 – attempting re-login with the same link.
322	Invalid Security Questions.
323	Gateway is not SSO enabled.
324	Please enter a valid email address. Delete all the invalid characters, including spaces, from your email address.
325	The username must be less than 100 characters.
326	The username must not include any spaces.
327	The username must not include the following characters
351	The sites requested must all be active gateways.
352	The sites requested must be full gateways.
353	SSO ID is required for authentication into SSO gateways.
354	The sites requested must all be configured for either SSO or non-SSO.
355	The sites requested must all be configured with the same language.

Web Service API with Multiple Languages

Job data is presented for the appropriate language even when reqs are posted in multiple languages. When clients perform a job search with the search API, the web service request is made to one job site.

This is because when you query our search API, you are querying against one site. For a non-Global Talent Gateway (GTG), the web service response displays jobs posted to that site only, and therefore, jobs from one language only. For a GTG, clients still specify the site, but in a separate node they can specify all of the locales making up that GTG. To perform this type of web service request, add the following node to your request:

- <SelectedSearchLocaleId>*1033,2057,3084*</SelectedSearchLocaleId>

Search results do not return multiple translations for a single req, however. It mimics a candidate selecting different languages on the TG search openings page. For example, assume a GTG has the following languages:

- English
- Spanish
- French
- Italian
- German

If a candidate searches for English and Spanish jobs on a member site of the GTG, the search results page shows all of the reqs available in English on that site, and all of the reqs that are translated in Spanish but not in English. The search API works the same way.

Member sites of a GTG all have their own unique SiteId. You can map the preferred language for each candidate to the corresponding member site on the GTG, and then pass in only the locale for that site in the SelectedSearchLocaleId node. That returns only jobs that have been translated in the language of the member site you're querying against.

Configuring Candidate APIs

Each candidate API requires some level of Workbench or BrassRing configuration.

The configuration process for each candidate APIs is a three-step process.

- For each candidate API you want to configure, you must ensure the corresponding Workbench configuration is in place. Each API has its own separate XML document. Elements within the XML document request and return specific information from BrassRing to the client. If the required configurations do not exist, the XML response will return empty strings instead of the requested values. See Configuring Workbench.
- The next step involves preparing the two-way communication channel (transport mechanism) between BrassRing and the client. See Establishing the Transport Mechanism.
- The third step of the procedure involves using predefined XML documents to send requests and receive responses from BrassRing. Elements with the XML document define who is requesting the information and what information is being requested. Elements are the configuration instructions within the XML document. Candidate APIs have some common elements and some candidate-specific elements. For example, common elements in XML requests indicate the sender, important for authentication. Candidate specific elements indicate the values (functionality) each candidate API is requesting.

Let's see how one candidate-specific element in the Access Page XML document works. This API allows clients to redirect candidates to a specific landing page on their Talent Gateway website using the <DestinationPage> element within the XML document. When the client's server sends the API request to BrassRing, pre-defined element values in the XML indicate the landing page using a numerical value. For example, if you want your candidates to land on the Search Openings page, your request XML element (<DestinationPage>) contains the numerical value 1; if you want your candidates to land on the Edit Profile page, your XML element (<DestinationPage>) contains the numerical value 2, and so on.

> Use Ctrl + Click to follow hyperlinks. Use ALT + LEFT ARROW to return to point of origin.

Each candidate API uses its own specific XML document. See Defining the XML Document.

Executing the Candidate APIs

When the candidate API configuration is in place, the next step is to execute the candidate APIs. Executing the candidate APIs means invoking the BrassRing message router (server). This means you are requesting, receiving, and displaying the information for each candidate API on your Talent Gateway website.

Because Candidate APIs display candidate-specific information, each candidate must be authenticated when accessing their profiles on a client's talent gateway. Clients perform this authentication using an Authentication Token. Refer to Authenticate User for more information on authentication.

When you access the message router (server), the router internally invokes the web service. Invoking the message router (server) performs these functions in chorological order:

- Validates the client credentials (sending into the <Sender> tag)
- Validates the input XML (if any node is incorrect or there is an invalid xml string)
- Performs the encryption/decryption if client has enabled encryption.
- Sends the XML document to the web service
- Receives the results from web service
- Displays results on the client's Talent Gateway website.

For more information, see Invoking the Server.

To see an example of the Candidate API workflow, see BrassRing APIs – Candidate APIs.

> Use Ctrl + Click to follow hyperlinks. Use ALT + LEFT ARROW to return to point of origin.

Before You Begin

Preparing for initiating BrassRing APIs ensures successful deployment.

Using XML

XML documents are structured documents that contain, among other things, XML 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.

XML Syntax Format Guidelines

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

- All XML must have closing tags.
- XML is case sensitive.
- XML elements must be properly nested
- XML attributes must be quoted
- XML comments must be enclosed with <! Comment -->
- XML element naming conventions:

Names cannot start with a number or punctuation character

Names cannot start with the letters xml (or XML, or Xml, etc.)

Names cannot contain spaces

Encoding Standard

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

CDATA Input

CDATA is the content clients can input when using BrassRing's APIs. CDATA can be text or numerical and must follow syntax guidelines.

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

Candidate API Configuration

Configuring BrassRing Workbench

Candidate APIs rely on BrassRing Workbench configurations to support the information request and response workflow. For each candidate API you want to configure, you must ensure the corresponding BrassRing Workbench configuration is in place.

For more information concerning BrassRing Workbench configuration, please contact your Support Team member.

Establishing the Transport Mechanism

Configuring any of the candidate APIs requires the initial set up of the communication between a client's BrassRing system and a client's job website.

> **Note:** As with all BrassRing APIs, each API requires BrassRing Workbench configuration.

Configuring Encryption

Use the instructions below to begin configuration:

2. Client requests a share access key by completing a Hive Ticket.

   - The most commonly used name for this key is Integration.
   - Specify encryption method: AES or RSA

Your Support Team representative completes the Hive ticket and sends you the required values for your API configuration:

Client Name, UserId, KeyName, and Encyrption Key. These values are used to set up your API.

3. Determine your web service transport mechanism. BrassRing offers either the HTTP Post or Web Service Descriptive Language (WSDL). Both transport mechanisms are synchronous, meaning all transactions are real-time with a request-response messaging and no message queuing.

Defining the XML Document

1. Select your Candidate API. Access the corresponding XML document here:
2. Copy or enter the text of the XML document into your XML editor. Adhere to all XML document syntax formatting guidelines.
3. Validate your XML document in your XML editor. If necessary, correct XML and re-validate.

Invoking the Message Router (Server)

The BrassRing web service uses a message router address as the entry point for the APIs. Each client will use their own method for accessing the message router to submit their search request.

When a client accesses any one of these message router addresses, the router internally invokes the web service. The invoked web services then performs these functions in chronological order:

- Validates the client credentials (sending into the <Sender> tag)
- Validates the input xml (if any node is incorrect or there is an invalid xml string)
- Performs the encryption/decryption if client has enabled encryption.
- Sends the XML document to the web service
- Receives the results from web service
- Displays results on the client's Talent Gateway website.

> If you are using encryption, contact your Support Team member to obtain a shared encryption key.

The general procedure to invoke the message router is:

1. Open your programming software application used to send commands.
2. Access the message router address for the web service transport mechanism.
3. Transmit your XML document.

Message Router Addresses

The BrassRing message router addresses are:

US – Staging Environment:

http://stagingimport.brassring.com/WebRouter/WebRouter.asmx

US – Production Environment:

http://Import.brassring.com/WebRouter/WebRouter.asmx

EU – Staging Environment:

http://stagingkrb-jobs.brassring.com/WebRouter/WebRouter.asmx

EU – Production Environment:

http://krb-jobs.brassring.com/WebRouter/WebRouter.asmx

Web Service Description Language Paths

The Web Service Description Language (WSDL) paths for message router web services are as follows:

Staging Environment:

http://stagingimport.brassring.com/WebRouter/WebRouter.asmx?wsdl

Production Environment:

http://Import.brassring.com/WebRouter/WebRouter.asmx?wsdl

> Use Ctrl + Click to follow hyperlinks. Use ALT + LEFT ARROW to return to point of origin.

Candidate APIs Preparing the XML Documents

XML documents serve as the configuration instructions for API requests and responses. Each Candidate API supports a specified functionality and has its own XML document. Elements within this XML document contain the instruction for a request or a response.

For example, the Job Status API, contains an element called <ResultSet>. In the case of a request, when the BrassRing XML processor receives an XML document with the <ResultSet> element, it knows the element is requesting a job status. The BrassRing XML processor responds to this <ResultSet> request by replying with a job status and displaying it on the client's website.

> All Candidate APIs require candidate authentication before web service call can be executed.

Candidate APIs

The following is a list of BrassRing's candidate APIs links. For more information concerning BrassRing Workbench configuration, please contact your Support Team member.

To access the required XML document for each candidate API, click the candidate API hyperlink.

- Access Page
- Authenticate User
- Candidate Forms
- Change Secret Question
- Communications History
- Event Manager Search – Candidate Search
- Document/Document Packet
- Forgot Password
- Get Resume
- Get Jobs in Job Cart
- Get Job Status
- New User Creation
- Reset Password
- Simultaneous Login
- Talent Gateway Form Import
- Update Existing User

Candidate API XML Documents

The following are the Candidate Specific XML documents. A table listing all elements used in the Candidate APIs follows the XML documents. You can find it here.

Access Page

Allows clients to navigate directly to one of the Talent Gateway pages from one of their web pages. For example: users could be directed to the Search openings page, Edit profile page, or Job cart page.

Clients must have the corresponding Talent Gateway pages configured in BrassRing Workbench in order for the API request to be fulfilled. For example, if you want candidates to land on the Job Cart page, you must have the Job Cart configured in BrassRing Workbench.

For more information concerning BrassRing Workbench configuration, please contact your Support Team member.

Input String

Each API has a required XML input string that clients must use when requesting responses from the BrassRing API.

```
<InputString>
        <AuthenticationToken>fa7309b1-46f0-4176-891d-
4d916f287d98</AuthenticationToken>
        <DestinationPage>0</DestinationPage>
        <!-- HomePage = 0,SearchOpenings = 1,EditProfile = 2,StatusCheck =
3,ResumeManager = 4,AgentManager = 5,JobCart = 6,SavedDrafts = 7,Logout = 8,SubmitNow
= 9 -->
</InputString>
```

2xB ← **Client**

Access Page Response XML (formerly Results XML)

```
<?xml version="1.0" encoding="utf-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by  () -->
<Envelope version="01.00">
 <Sender>
        <Id>QE</Id>
        <Credential>AXc9N72IF+3UhS8HUZq+FA==</Credential>
```

BR → **Client**

```
</Sender>
<Recipient/>
<TransactInfo transactType="response" transactId="bdd18f76-abb1-4097-ae60-
9660e61ae85c">
        <TransactId>4/26/2011</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="Decryptor">
        <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element" encoding="utf-8"
xmlns="http://www.w3.org/2001/04/xmlenc#">
        <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:KeyName>Integration</ds:KeyName>
        </ds:KeyInfo>
        <CipherData>
        <CipherValue>
        <Unit UnitProcessor="AccessTGPage">
        <Packet>
        <PacketInfo packetType="data" encrypted="yes">
        <packetId>1</packetId>
        </PacketInfo>
        <Payload>
        <InputString>
        <AuthenticationToken>fa7309b1-46f0-4176-891d-
4d916f287d98</AuthenticationToken>
        <DestinationPage>0</DestinationPage>
        <!-- HomePage = 0,SearchOpenings = 1,EditProfile = 2,StatusCheck =
3,ResumeManager = 4,AgentManager = 5,JobCart = 6,SavedDrafts = 7,Logout = 8,SubmitNow
= 9 -->
        </InputString>
```

BR ⟶ Client

Access Page Response XML (formerly Results XML)

```
        <ResultSet>
        <RedirectURL>https://sqa-tgweb-
01.brassring.com/1033/asp/tg/redirectcandidate.asp?q=%5eudzQFPVk94hM37MiZWaGjTn1S
BeuHiNco%2fAyx63VjjIb6TNzQe%2fAd517PnJSqyWxMyVy%2befTmM2x%0d%0aZuiU4cWz
mgxB%2bUHoJh9GWOWbO1rD0RaKcxxGTj9BBktYK2u2rUCxXCvvOTdi9oG4j%2bkCbUhBZp
3F%0d%0aJT54nO9%2bpxN2tUk%2fHc4zRMGhv18upixQGQ%2fW9bK2RnvpywSuHGs2FXq
xhz9tqshrKyEiBvU%2bHJYx%0d%0axaDw5kPwosXVt3akng%3d%3d</RedirectURL>
        </ResultSet>
        </Payload>
        <Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>Call was successfully processed without error.</LongDescription>
        </Status>
        </Packet>
        <Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>Call was successfully processed without error.</LongDescription>
        </Status>
        </Unit>
        </CipherValue>
        </CipherData>
        </EncryptedData>
        <Status>
        <Code>200</Code>
        <ShortDescription>Message decrypted and processed without
errors.</ShortDescription>
        <LongDescription/>
        </Status>
</Unit>
```

BR ⟶ Client

```
<Status>
      <Code>200</Code>
      <ShortDescription>Success</ShortDescription>
      <LongDescription>All units are processed with success codes</LongDescription>
</Status>
</Envelope>
```

BR ⟶ Client

To view the XML Element Table for this API, click here.

To view error codes for this API, click here.

Assessments (Pending)

Web service call: GetCandidateAssessmentInfo

Enables clients to pull candidate pending assessment data into their customized gateway. Candidate must be authenticated into the client's Talent Gateway before initiating this web service call request. For each authenticated candidate, the GetCandidateAssessmentInfo call returns and displays the requested information in the form of a Pending Assessment page. The Pending Assessment page displays the following data:

- AutoReqID (or CustomReqID based on client configuration)
- Job Title
- Assessment Link (if link is not yet clickable (Assessment Batch) no value returns.
- Assessment Name
- Status

If there are no pending assessments, the client can determine what message displays to the candidate. For more information concerning BrassRing Workbench configuration, please contact your Support Team member.

Input String

Each API has a required XML input string that clients must use when requesting responses from the BrassRing API. This input string is a candidate authentication request that must be processed before requesting a candidate's pending assessment data.

Assessment Page Candidate Authentication Request XML

```
         <?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
<Sender>
        <Id>10540</Id>
        <Credential>11721</Credential>
</Sender>
<TransactInfo transactType="data" transactId="075cbfe6-b9dc-49f9-8f12-8eae62b62055">
        <TransactId>10/24/2008</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
</TransactInfo>
<Unit UnitProcessor="AuthenticateTGUser">
        <Packet>
        <PacketInfo packetType="data">
        <packetId>1</packetId>
        </PacketInfo>
        <Payload>
        <InputString>
        <ClientId></ClientId>
        <SiteId></SiteId>
        <Username></Username>
```

BR ⟶ Client

```
     <Password></Password>
     </InputString>
     </Payload>
     </Packet>
 </Unit>
</Envelope>
```

BR ⟶ Client

Assessment Page Candidate Authentication Response XML

```
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
 <Sender>
        <Id>1364</Id>
        <Credential>1sU9JELZ8z9tPsHw8FDjyw==</Credential>
</Sender>
<Recipient/>
<TransactInfo transactType="response" transactId="2347cf88-3c8d-4f6c-ab51-bfa0679f2bbe">
        <TransactId>7/1/2011</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="Decryptor">
        <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element" encoding="utf-8"
xmlns="http://www.w3.org/2001/04/xmlenc#">
        <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:KeyName>Integration</ds:KeyName>
        </ds:KeyInfo>
        <CipherData>
        <CipherValue>
        <Unit UnitProcessor="AuthenticateTGUser">
        <Packet>
        <PacketInfo packetType="data" encrypted="yes">
        <packetId>1</packetId>
        </PacketInfo>
        <Payload>
        <InputString>
        <ClientId></ClientId>
        <SiteId></SiteId>
        </InputString>
        <ResultSet>
        <AuthenticationStatus></AuthenticationStatus>
        <AuthenticationToken></AuthenticationToken>
        </ResultSet>
        </Payload>
        <Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>Call was successfully processed without error.</LongDescription>
        </Status>
        </Packet>
<Code>200</Code>
        <Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>Call was successfully processed without error.</LongDescription>
        </Status>
        </Unit>
        </CipherValue>
        </CipherData>
        </EncryptedData>
        <Status>
        <Code>200</Code>
```

BR ⟶ Client

```
            <ShortDescription>Message decrypted and processed without
errors.</ShortDescription>
        <LongDescription/>
<Code>200</Code>
        <Status>
         <Code>200</Code>
         <ShortDescription>Success</ShortDescription>
         <LongDescription>Call was successfully processed without error.</LongDescription>
         </Status>
         </Unit>
         </CipherValue>
         </CipherData>
         </EncryptedData>
         <Status>
         <Code>200</Code>
         <ShortDescription>Message decrypted and processed without
errors.</ShortDescription>
         <LongDescription/>
         </Status>
</Unit>
<Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>All units are processed with success codes</LongDescription>
</Status>
</Envelope>
```

BR ➝ Client

Clients use the web service call: GetCandidateAssessmentInfo to display the pending assessments. Candidates authenticated by the client's Talent Gateway can then request and access their pending assessments.

Assessment Page Candidate Pending Assessment Request

```
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
    <Sender>
        <Id>01845</Id>
        <Credential>20355</Credential>
    </Sender>
    <TransactInfo transactType="data" transactId="075cbfe6-b9dc-49f9-8f12-8eae62b62055">
        <TransactId>10/24/2008</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
    </TransactInfo>
    <Unit UnitProcessor="GetCandidateAssessmentInfo">
        <Packet>
            <PacketInfo packetType="data">
                <packetId>1</packetId>
            </PacketInfo>
        <Payload>
          <InputString>
        <ClientId>13746</ClientId>
        <SiteId>6623</SiteId>
        <AuthenticationToken>
<![CDATA[a828257b-b92e-4736-88e2-0761dc4c48b]]>
</AuthenticationToken>
<RedirectURL>http://clientsite.com</RedirectURL>
                </InputString>
            </Payload>
        </Packet>
    </Unit>
</Envelope>
```

BR ⬅ Client

Assessment Page Response XML

```
<?xml version="1.0" encoding="utf-8" ?>
- <Envelope version="01.00">
- <Sender>
  <Id>01845</Id>
  <Credential>/dwZJPx+zFd1D4MOvWHrJg==</Credential>
  </Sender>
  <Recipient />
- <TransactInfo transactType="response" transactId="fad44a1d-e4d8-4702-838e-9714665feb53">
  <TransactId>1/2/2013</TransactId>
  <TimeStamp>12:00:00 AM</TimeStamp>
  </TransactInfo>
- <Unit UnitProcessor="Decryptor">
- <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element" encoding="utf-8" xmlns="http://www.w3.org/2001/04/xmlenc#">
  <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />
- <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:KeyName>Integration</ds:KeyName>
  </ds:KeyInfo>
- <CipherData>
- <CipherValue>
- <Unit UnitProcessor="GetCandidateAssessmentInfo">
- <Packet>
- <PacketInfo packetType="data" encrypted="yes">
  <packetId>1</packetId>
  </PacketInfo>
- <Payload>
- <InputString>
  <ClientId>20355</ClientId>
  <SiteId />
  <AuthenticationToken>0cff5819-47ee-4333-b9ad-e459bfac0dec</AuthenticationToken>
  <AssessRedirectURL>http://msn.com</AssessRedirectURL>
  </InputString>
- <ResultSet>
- <Reqs>
- <Req Id="4BR">
  <JobTitle>Req with Batch - No Hurdle</JobTitle>
- <Assessments>
- <Assessment Id="C577480F64EC07B4E040A8C0F2C87996">
  <AssessmentName>Numerical Reasoning (EN-US)</AssessmentName>
  <AssessmentURL>https://qa-import.brassring.com/AssessmentOrder/AssessmentOrderRequest.aspx?ck=2xademo1&sid=C577480F64EC07B4E040A8C0F2C87996&bid=2420&aid=20355&gid=4BR&ad1=Jeeva&ad2=Jeevan&userid=38776&RFIResponse=False&cpath=https%3a%2f%2fsqa-tgweb-01.brassring.com%2fcss%2fPartner_20355_6532.css&return=http%3a%2f%2fmsn.com&reqid=93946&siteid=6532&languageid=1&etid=&posturl=http://qa-tgweb-01.brassring.com/xmlgateway/xmltransformer.aspx?t__EQUALS__2XA&clientid=20355&IsAnony=0&new=y</AssessmentURL>
  <AssessmentStatus>In Progress</AssessmentStatus>
  </Assessment>
- <Assessment Id="C56186056C80A116E040A8C0F2C874A0">
  <AssessmentName>Logical Reasoning</AssessmentName>
  <AssessmentURL />
  <AssessmentStatus>Not Started</AssessmentStatus>
  </Assessment>
  </Assessments>
  </Req>
  </Reqs>
  </ResultSet>
  </Payload>
- <Status>
  <Code>200</Code>
```

```
      <ShortDescription>Success</ShortDescription>
      <LongDescription>Call was successfully processed without error.</LongDescription>
      </Status>
      </Packet>
  - <Status>
      <Code>200</Code>
      <ShortDescription>Success</ShortDescription>
      <LongDescription>Call was successfully processed without error.</LongDescription>
      </Status>
      </Unit>
      </CipherValue>
      </CipherData>
      </EncryptedData>
  - <Status>
      <Code>200</Code>
      <ShortDescription>Message decrypted and processed without errors.</ShortDescription>
      <LongDescription />
      </Status>
      </Unit>
  - <Status>
      <Code>200</Code>
      <ShortDescription>Success</ShortDescription>
      <LongDescription>All units are processed with success codes</LongDescription>
      </Status>
      </Envelope>
```

BR ➝ Client

To view the XML Element Table for this API, click here.

To view error codes for this API, click here.

Authenticate User

Provides single sign-on (SSO) support. To Support SSO Talent Gateways, the user has to send the Client Id / Site Id / SSO ID. BrassRing then generates the new user and sends the Authentication Token back to client. Client can use that token to access other APIs. This API can also be used in a non-SSO situation when another application needs to send the userid and password to the Talent Gateway.

For more information concerning SSO BrassRing Workbench configuration, please contact your Support Team member.

Authenticate User Request XML (formerly Input Spring to Candidate Web Service)

Web Service Call: AuthenticateTGUser

```
      <?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
<Sender>
      <Id>10540</Id>
      <Credential>11721</Credential>
</Sender>
<TransactInfo transactType="data" transactId="075cbfe6-b9dc-49f9-8f12-8eae62b62055">
      <TransactId>10/24/2008</TransactId>
      <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
</TransactInfo>
<Unit UnitProcessor="AuthenticateTGUser">
      <Packet>
      <PacketInfo packetType="data">
       <packetId>1</packetId>
      </PacketInfo>
```

BR ⟵ Client

```
        <Payload>
        <InputString>
        <ClientId></ClientId>
        <SiteId></SiteId>
        <Username></Username>
        <Password></Password>
        </InputString>
        </Payload>
        </Packet>
 </Unit>
</Envelope>
```

BR ← Client

SSO Support

To Support SSO TGs, we are using existing API (Authenticate User) where user has to send the Client Id / Site Id / SSO ID and we will generate the new user at our end and then send the Authentication Token back to client. Client can use that token to access other APIs.

Input String

Each API has a required XML input string that clients must use when requesting responses from the BrassRing API.

```
Input String
        <InputString>
        <ClientId></ClientId>
        <SiteId></SiteId>
        <SSOID></SSOID>
        </InputString>
```

BR ← Client

Business Rules:

For SSO support, we are using existing Authenticate User API with addition of new node called SSOSessionID, which will have valid SSO Session ID. If we don't find entries for that SSOSessionID in our system we will treat that as new user and create a new user and send Authentication Token which is used to access other API calls.

Authenticate User Request XML

Web Service Call: AuthenticateTGUser (SSO)

```
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
 <Sender>
        <Id>10540</Id>
        <Credential>11721</Credential>
 </Sender>
<TransactInfo transactType="data" transactId="075cbfe6-b9dc-49f9-8f12-8eae62b62055">
        <TransactId>10/24/2008</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="AuthenticateTGUser">
        <Packet>
        <PacketInfo packetType="data">
        <packetId>1</packetId>
        </PacketInfo>
        <Payload>
        <InputString>
        <ClientId></ClientId>
        <SiteId></SiteId>
        <SSOID></SSOID>
        </InputString>
        </Payload>
        </Packet>
 </Unit>
</Envelope>
```

BR ← Client

Authenticate User Response XML (Final Output from Candidate Web Service)

```
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
<Sender>
        <Id>1364</Id>
        <Credential>1sU9JELZ8z9tPsHw8FDjyw==</Credential>
</Sender>
<Recipient/>
<TransactInfo transactType="response" transactId="2347cf88-3c8d-4f6c-ab51-
bfa0679f2bbe">
        <TransactId>7/1/2011</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="Decryptor">
        <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element" encoding="utf-8"
xmlns="http://www.w3.org/2001/04/xmlenc#">
        <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:KeyName>Integration</ds:KeyName>
        </ds:KeyInfo>
        <CipherData>
        <CipherValue>
        <Unit UnitProcessor="AuthenticateTGUser">
        <Packet>
        <PacketInfo packetType="data" encrypted="yes">
        <packetId>1</packetId>
        </PacketInfo>
        <Payload>
        <InputString>
        <ClientId></ClientId>
        <SiteId></SiteId>
        <SSOID></SSOID>
        </InputString>
        <ResultSet>
        <AuthenticationStatus></AuthenticationStatus>
        <AuthenticationToken></AuthenticationToken>
        </ResultSet>
        </Payload>
        <Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>Call was successfully processed without error.</LongDescription>
        </Status>
        </Packet>
<Code>200</Code>
        <Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>Call was successfully processed without error.</LongDescription>
        </Status>
        </Unit>
        </CipherValue>
        </CipherData>
        </EncryptedData>
        <Status>
        <Code>200</Code>
        <ShortDescription>Message decrypted and processed without
errors.</ShortDescription>
        <LongDescription/>
```

BR ⟶ Client

Authenticate User Response XML (Final Output from Candidate Web Service) (continued)

```
<Code>200</Code>
        <Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>Call was successfully processed without error.</LongDescription>
        </Status>
        </Unit>
        </CipherValue>
        </CipherData>
        </EncryptedData>
        <Status>
        <Code>200</Code>
        <ShortDescription>Message decrypted and processed without errors.</ShortDescription>
        <LongDescription/>
        </Status>
</Unit>
<Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>All units are processed with success codes</LongDescription>
</Status>
</Envelope>
```

BR ⟶ Client

To view the XML Element Table for this API, click here.

To view the error codes for Assessments this Authenticate User API, click here.

To view the error codes for this Authenticate User API SSO, click here.

> Use Ctrl + Click to follow hyperlinks. Use ALT + LEFT ARROW to return to point of origin.

Candidate Forms

Web service call: GetCandidateForms

Enables clients to display eLinked forms on a candidate's portal page. Candidates can access, edit, or complete eLinked candidate forms on their candidate portal page. Clients can also restrict display of forms by specifying a particular date range or form template for eLinked forms. Result set includes the form name, the candidate name, candidate reference number, Req ID and Status, Job Title, dates forms were sent, added and/or edited and dates drafts may have been saved or edited. Forms display in descending order, with the most recent forms being listed first.

Candidate forms must be configured before using this API. For more information concerning configuration, please contact your Support Team member.

Input String

```
<InputString>
        <ClientId>516</ClientId>
        <SiteId>8723</SiteId>
        <AuthenticationToken>bfd0bfed-c763-4a88-9833-947cf3dcd50d</AuthenticationToken>
        <Interval>no of days</Interval>
        <FormTypeIds><![CDATA[Comma separated list of FormTypeIds]]></FormTypeIds>
</InputString>
```

BR ⟵ Client

Candidate Forms Response XML (Formerly Candidate Forms XML)

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="UTF-8"?>
<forms>
<row>
        <mode>view</mode>
```

BR ⟶ Client

```
                <resumekey>160918</resumekey>
                <formname><![CDATA[Test115EST]]></formname>
                <displayformname><![CDATA[Test]]></displayformname>
                <formurl><![CDATA[https:///JetStream/500/Presentation/Template/ASP/Candidate/Forms/
        ViewForm.asp?api&isgq=@7jmvKabnpHc=&apprvl=&localeid=@IsSbRwlc1jxKpUy0Sbstn
        A==&mode=@1yjS1Hi6/q/uNq32m6X08Q==&where=@TPEBTmFaU8Z5hrgmRqSygA==
        &encryptedvalues=@5BQpjzUXs5GMq/AmdCT0bA==*@jJ/IiMFNw5fRNohUQz$A6eREX
        StAhCRu*@6m1QHPwOkrEsurR1k5TDKQ==*@G6OALy2D/Sw=*@EPGSpPrMoa4ffltB1
        4q/Gw==*@OpuOLH2aDJI=*@7jmvKabnpHc=*@Em2CCdA4S6sHdyynXiGBww==*@G6
        OALy2D/Sw=*@amtcAhHDhl8=**@s/8UkTuFFVU=*@s/8UkTuFFVU=]]></formurl>
                <candidatename><![CDATA[Christopher Cowles]]></candidatename>
                <jobrequisitionid><![CDATA[1312BR]]></jobrequisitionid>
                <optionaljobrequisitionid><![CDATA[Test123]]></optionaljobrequisitionid>
                <jobtitle><![CDATA[Attach Assessment - XML - 2]]></jobtitle>
                <reqstatus>0</reqstatus>
                <addedon>25 May 2012</addedon>
                <editedon>21 Jun 2012</editedon>
                <datesent>20 May 2012</datesent>
                <draftcreatedon>23 May 2012</draftcreatedon>
                <drafteditedon>24 May 2012</drafteditedon>
        </row>
</forms>
```


BR ⟶ Client

To view the XML Element Table for this API, click here.

To view the error codes for this API, click here.

Business Rules

The !CDATA[ ] included in the sample tags is necessary to accommodate for single quotes, double quotes and other data separating characters that may be legitimately used in the value.

If <Interval> is blank, then ALL forms sent/eLinked to candidate will be returned (regardless of sent date). The client will be able to filter the list and present the desired forms as necessary.

If <FormtypeIDs> is blank, ALL forms sent/eLinked to the candidate will be returned (regardless of form type). The client will be able to filter the list and present the desired forms as necessary.

The result set will include only forms that have been sent to the candidate.

Only forms where the To: (non-system user) = the current email address in the Talent Record is included

This DOES include when a form is attached via an email communication (sent manually or automated).

NOTE: If the candidate email address has changed, then previous forms may not be included

Response XML will list results by last edited date in descending order, beginning with most recent.

Send existing form in edit mode form setting:

If a single per candidate or single candidate / req form configured with the form setting "**Send existing form in edit mode**" is initially 'eLinked' in Add mode (eLink Blank form), the eLink supplied in the API will be an 'edit mode' eLink after the form is saved to the Talent Record.

If a single per candidate or single candidate / req form configured with the form setting "**Send existing form in edit mode**" = **No** is initially 'eLinked' in Add mode (eLink Blank form), the form will be included in the API listing of forms in VIEW mode.

DRAFT forms

If a form is sent in Add mode and the candidate saves it as a draft. The link will remain unchanged (it will still be 'Add mode') but if there are any DRAFTS saved for that form for that candidate (and perhaps for that req), then the DRAFT will be presented to the candidate upon re-click of the link.

The <mode> will be returned as "draft", if the draft exists.

Once the draft form is saved as a form, the 'Draft created date' and 'Draft edited date' will be set to 'blank' because the draft no longer exists as it was 'converted' into a Form.

If the Draft is edited (but not yet saved as a form) the 'Draft edited date' will be updated.

Use Scenarios/Workflows: How a form will included in the API table

1. Blank form is eLinked (Add mode) from the Add form action

   - This can be either from the Candidate results panel (list of candidates) or from within the Talent Record

   - Select candidate in check box

   - Action | Add new form

   - Select form from list

   - Click 'eLink blank form' button at top of form

   - Type candidate's email address (that is currently in the Talent Record) into the To: (non-system users)

2. Blank form is eLinked (Add mode) from the Talent Record (Forms list)

   - Click on Add/View link next to Forms in the Talent Record ribbon

   - Select form from single select drop down list

   - Click 'eLink blank form' button to the right of the selected form in the single select drop down list

   - Type candidate's email address (that is currently in the Talent Record) into the To: (non-system users)

3. Blank form is eLinked (Add mode) via Send Communications

   - This can be done either from (1) the candidate's listing, (2) from within the Talent Record or (3) via Automation Manager triggers. Blank form will be sent if the form does not yet exist for the candidate or for the candidate for this requisition (depending on form type)

   - Action | Send communications

   - Select a communication that has a form attached in the template

   - NOTE: Unable to tell from drop down listing of Communication template names which email template contains an attached form.

   - Fill in rest of email information

   - Click Send

4. Blank form is eLinked (Add mode) via eLink (talent record)

   - Unlikely scenario as the whole Talent Record is RARELY (if ever) eLinked to the candidate. However, this can be done either from the candidate's listing or from within the Talent Record.

   - Action | Send elink

   - Select form from 'Forms to complete' list

   - Fill in rest of eLink information

   - Type candidate's email address (that is currently in the Talent Record) into the To: (non-system users)

5. Form is eLinked (Edit or View mode) from the Talent Record (Forms list)

   - Click on Add/View link next to Forms in the Talent Record ribbon

   - Select the **eLink form** icon in the row of the desired form to send

   - Choose 'Edit' for the **eLink form to:**

   - Type candidate's email address (that is currently in the Talent Record) into the To: (non-system users)

6. Form is eLinked (Edit or View mode) from the Talent Record (Action log)

   - Navigate to desired form in the Action log

- RIGHT CLICK in the row of the desired form to send and select **eLink form**

- Choose 'Edit' for the **eLink form to:**

- Type candidate's email address (that is currently in the Talent Record) into the To: (non-system users)

7. Form is eLinked (Edit mode) via Send Communications

- This can be done either from (1) the candidate's listing, (2) from within the Talent Record or (3) via Automation Manager triggers. Edited form will be sent if the form *does* exist for the candidate or for the candidate for this requisition (depending on form type) AND the setting the form template setting 'Send existing form in edit mode' = YES.

- Action | Send communications

- Select a communication that has a form attached in the template

- NOTE: Unable to tell from drop down listing of Communication template names which email template contains an attached form.

- Fill in rest of email information

- Click Send

8. Form is eLinked (Edit mode) via eLink (talent record)

- Unlikely scenario as the whole Talent Record is RARELY (if ever) eLinked to the candidate. However, this can be done either from the candidate's listing or from within the Talent Record.

- Action | Send elink

- Select form from 'Forms to complete' list

- Fill in rest of eLink information

- Type candidate's email address (that is currently in the Talent Record) into the To: (non-system users)

9. Form is eLinked (View mode) via eLink (talent record)

- Unlikely scenario as the whole Talent Record is RARELY (if ever) eLinked to the candidate. However, this can be done either from the candidate's listing or from within the Talent Record.

- Action | Send elink

- Select form from 'Forms to view' list

- Fill in rest of eLink information

- Type candidate's email address (that is currently in the Talent Record) into the To: (non-system users)

> There may be other scenarios that include a form in the eLink form API table that are not accounted for in this list.

Change Secret Question/Answer

Web service call: ChangeCandidateSecurityInfo

This web service call enables clients to display an option for candidates to change their security question(s) and/or answer(s) when authenticated candidates access a client's BrassRing talent gateway website. This web service call displays one or three security questions depending upon Talent Gateway configuration:

- If Enhanced Self-Service Login Management is disabled on talent gateways, only one SecurityQID & Answer is required.

- If Enhanced Self-Service Login Management is enabled on talent gateways, all three SecurityQIDs & Answers are required in order to change any secret questions and answers.

- The existing security questions & the corresponding answers are replaced with the new values sent in the xml.

For more information concerning BrassRing Workbench configuration, please contact your Support Team member.

Input String (Modified)

This XML request is an example of the web service API call when enhanced login message is not enabled.

```
<Payload>
<InputString>
        <AuthenticationToken/>
        <SecurityInfo>
        <SecurityQuestionOne>
           <SecurityQID></SecurityQID>
           <Answer></Answer>
        </SecurityQuestionOne>
    </SecurityInfo>
        </InputString>
</Payload>
```

BR ← Client

This XML request is an example of the web service API call when enhanced login message is enabled.

```
<Payload>
<InputString>
        <AuthenticationToken/>
        <SecurityInfo>
        <SecurityQuestionOne>
           <SecurityQID></SecurityQID>
           <Answer></Answer>
        </SecurityQuestionOne>

        <SecurityQuestionTwo>
           <SecurityQID></SecurityQID>
           <Answer></Answer>
        </SecurityQuestionTwo>

        <SecurityQuestionThree>
           <SecurityQID></SecurityQID>
           <Answer></Answer>
        </SecurityQuestionThree>

    </SecurityInfo>
        </InputString>
</Payload>
```

BR ← Client

Change Secret Question/Answer Response XML

This XML response is an example of the web service API call when the client setting 'TG login details management' is configured as Enhanced self-service login management and Account Retrieval Tool.

```
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
  <Sender>
   <Id>01845</Id>
   <Credential>22022</Credential>
  </Sender>
  <TransactInfo transactType="data"
         transactId="075cbfe6-b9dc-49f9-8f12-8eae62b62055">
   <TransactId>11/20/2013</TransactId>
   <TimeStamp>12:00:00 AM</TimeStamp>
  </TransactInfo>
  <Unit UnitProcessor="ChangeCandidateSecurityInfo">
   <Packet>
    <PacketInfo packetType="data">
      <packetId>1</packetId>
    </PacketInfo>
    <Payload>
```

BR → Client

```
      <InputString>
        <AuthenticationToken>bd0afb7a-20b0-4b59-b857-357c97cc6db9</AuthenticationToken>
        <SecurityInfo>

          <SecurityQuestionOne>
            <SecurityQID>3</SecurityQID>
            <Answer>SAMPLE</Answer>
          </SecurityQuestionOne>

          <SecurityQuestionTwo>
            <SecurityQID>7</SecurityQID>
            <Answer>TEST</Answer>
          </SecurityQuestionTwo>

          <SecurityQuestionThree>
            <SecurityQID>2</SecurityQID>
            <Answer>TRIAL</Answer>
          </SecurityQuestionThree>

        </SecurityInfo>
      </InputString>
  <ResultSet>Success</ResultSet>
 </Payload>
  <Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>Call was successfully processed without error.</LongDescription>
      </Status>
    </Packet>
    <Status>
      <Code>200</Code>
      <ShortDescription>Success</ShortDescription>
      <LongDescription>Call was successfully processed without error.</LongDescription>
    </Status>
  </Packet>
<Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>Call was successfully processed without error.</LongDescription>
      </Status>
    </Unit>
   </CipherValue>
  </CipherData>
 </EncryptedData>
 <Status>
  <Code>200</Code>
  <ShortDescription>Message decrypted and processed without errors.</ShortDescription>
  <LongDescription />
 </Status>
</Envelope>
```

BR ➝ Client

To view the XML Element Table for this API, click here.

To view error codes for this API, click here.

Communications History

Web service call: GetCommunicationHistory

Enables clients to display a communications history link on the candidate portal page. When configured, candidates can view a log of some or all communications with clients. If a candidate misplaced an email and needs to see the email contents or needs to fill out a form associated with the email, they can access it when they log into the candidate portal page. Communications history only includes communications currently tracked within BrassRing.

For more information concerning configuration, please contact your Support Team member.

Input String (Modified)

This XML requests all message sent to Candidate.

```
<Payload>
 <InputString>
        <AuthenticationToken/>
        <PageNumber></PageNumber>
      <ReturnMessagesCount></ReturnMessagesCount>
        </InputString>
</Payload>
```

BR ← Client

Communications Response XML

This Results XML response displays all candidate communications on client's website.

```
<ResultSet>
<Messages>
<Message>
 <CorrespondenceID>12345</CorrespondenceID>
 <Date>MM/DD/YYYY</Date>
 <JobID></JobID>
  <JobTitle></JobTitle>
 <From></From>
 <TemplateName></TemplateName>
 <Type></Type>
 <Subject></Subject>
<MessageLink></MessageLink>
  </Message>
```

BR → Client

Communications Response XML- Full Decrypted Results

```
<?xml version="1.0" encoding="utf-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by  () -->
<Envelope version="01.00">
 <Sender>
        <Id>QE</Id>
        <Credential>AXc9N72lF+3UhS8HUZq+FA==</Credential>
</Sender>
<Recipient/>
<TransactInfo transactType="response" transactId="0b02e774-d6a2-4bab-a9a8-466e9b8c2fdc">
        <TransactId>4/26/2011</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="Decryptor">
        <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element" encoding="utf-8"
xmlns="http://www.w3.org/2001/04/xmlenc#">
        <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:KeyName>Integration</ds:KeyName>
        </ds:KeyInfo>
        <CipherData>
        <CipherValue>
        <Unit UnitProcessor="GetCommunicationHistory">
        <Packet>
        <PacketInfo packetType="data" encrypted="yes">
        <packetId>1</packetId>
        </PacketInfo>
        <Payload>
        <InputString>
        <AuthenticationToken>3fe272e2-06e1-4266-b009-f75a2998f92f</AuthenticationToken>
        <PageNumber/>
        <ReturnMessagesCount/>
        </InputString>
        <ResultSet>
```

BR → Client

```xml
        <Messages>
        <Message>
        <CorrespondenceID>423</CorrespondenceID>
        <Date>02/11/2011</Date>
        <Subject>KM Email Template - Subject</Subject>
        <Type>1</Type>
        <TemplateName>KM Email Template</TemplateName>
        <From>McEachern, Kevin</From>
        <JobID>92476</JobID>
        <JobTitle>Test 4000 Characters in JD 2</JobTitle>
        <MessageLink>https://sqa-tgweb-
01.brassring.com/1033/asp/tg/redirectcandidate.asp?q=%5eudzQFPVk94hM37MiZWaGjTn1SBeuHi
NcQ6R4O1vBGsJdh4ZzRXSEsrJYm%2f2SqUWQ2WM7IqU3Mf8Y%0d%0aqrbKFgPwNpDKlvQnu9
2TtVGV7GCVjiSjGu%2fuZE8yrszMnoFtLM0T4oFMifa2k4nQ4WqNULe0t4V6%0d%0acbnvVrj8W5U
TcnkzsKpCqg6e9zILtTShfgedWj3NUhS4vBfAAqiijbTZXCtsT98TaOsPFaNGD4Nf%0d%0aJcI4O9rK8
ymOBwdj7%2fRXJ4F8KcF1lqPBFYy9lFIUlen2v1RXjw%3d%3d</MessageLink>
        </Message>
        <Message>
        <CorrespondenceID>422</CorrespondenceID>
        <Date>02/11/2011</Date>
        <Subject>Ad Hoc 170BR</Subject>
        <Type>4</Type>
        <TemplateName>Send candidate email</TemplateName>
        <From>McEachern, Kevin</From>
        <JobID>92476</JobID>
        <JobTitle>Test 4000 Characters in JD 2</JobTitle>
        <MessageLink>https://sqa-tgweb-
01.brassring.com/1033/asp/tg/redirectcandidate.asp?q=%5eudzQFPVk94hM37MiZWaGjTn1SBeuHi
NcQ6R4O1vBGsJdh4ZzRXSEsrJYm%2f2SqUWQ2WM7IqU3Mf8Y%0d%0aqrbKFgPwNpDKlvQnu9
2TtVGV7GCVjiSjGu%2fuZE8yrszMnoFtLM0T4oFMifa2k4nQ4WqNULe0t4V6%0d%0acbnvVrj8W5U
TcnkzsKpCqg6e9zILtTShfgedWj3NUhS4vBfAAqiijbTZXCtsT98TaOsPFaNGD4Nf%0d%0aJcI4O9rK8
ymOBwdj7%2fRXJ4F8KcF1lqPBFYy9lFL5aoBPbKMvnQ%3d%3d</MessageLink>
        </Message>
        <Message>
        <CorrespondenceID>421</CorrespondenceID>
        <Date>02/11/2011</Date>
        <Subject/>
        <Type>0</Type>
        <TemplateName>KM Letter Temp 2</TemplateName>
        <From>McEachern, Kevin</From>
        <JobID>92475</JobID>
        <JobTitle>Test 4000 Characters in JD</JobTitle>
        <MessageLink>https://sqa-tgweb-
01.brassring.com/1033/asp/tg/redirectcandidate.asp?q=%5eudzQFPVk94hM37MiZWaGjTn1SBeuHi
NcQ6R4O1vBGsJdh4ZzRXSEsrJYm%2f2SqUWQ2WM7IqU3Mf8Y%0d%0aqrbKFgPwNpDKlvQnu9
2TtVGV7GCVjiSjGu%2fuZE8yrszMnoFtLM0T4oFMifa2k4nQ4WqNULe0t4V6%0d%0acbnvVrj8W5U
TcnkzsKpCqg6e9zILtTShfgedWj3NUhS4vBfAAqiijbTZXCtsT98TaOsPFaNGD4Nf%0d%0aJcI4O9rK8
ymOBwdj7%2fRXJ4F8KcF1lqPBFYy9lFK%2fxS%2bX7zrThg%3d%3d</MessageLink>
        </Message>
        <Message>
        <CorrespondenceID>420</CorrespondenceID>
        <Date>02/11/2011</Date>
        <Subject>KM Email Template 2</Subject>
        <Type>1</Type>
        <TemplateName>KM Email Template 2</TemplateName>
        <From>McEachern, Kevin</From>
        <JobID>92475</JobID>
        <JobTitle>Test 4000 Characters in JD</JobTitle>
        <MessageLink>https://sqa-tgweb-
01.brassring.com/1033/asp/tg/redirectcandidate.asp?q=%5eudzQFPVk94hM37MiZWaGjTn1SBeuHi
NcQ6R4O1vBGsJdh4ZzRXSEsrJYm%2f2SqUWQ2WM7IqU3Mf8Y%0d%0aqrbKFgPwNpDKlvQnu9
2TtVGV7GCVjiSjGu%2fuZE8yrszMnoFtLM0T4oFMifa2k4nQ4WqNULe0t4V6%0d%0acbnvVrj8W5U
TcnkzsKpCqg6e9zILtTShfgedWj3NUhS4vBfAAqiijbTZXCtsT98TaOsPFaNGD4Nf%0d%0aJcI4O9rK8
ymOBwdj7%2fRXJ4F8KcF1lqPBFYy9lFIWiKteCTmQAQ%3d%3d</MessageLink>
```

BR ⟶ Client

```
            </Message>
            <Message>
            <CorrespondenceID>419</CorrespondenceID>
            <Date>02/11/2011</Date>
            <Subject>Ad Hoc 169BR 2</Subject>
            <Type>4</Type>
            <TemplateName>Send candidate email</TemplateName>
            <From>McEachern, Kevin</From>
            <JobID>92475</JobID>
            <JobTitle>Test 4000 Characters in JD</JobTitle>
            <MessageLink>https://sqa-tgweb-
01.brassring.com/1033/asp/tg/redirectcandidate.asp?q=%5eudzQFPVk94hM37MiZWaGjTn1SBeuHi
NcQ6R4O1vBGsJdh4ZzRXSEsrJYm%2f2SqUWQ2WM7IqU3Mf8Y%0d%0aqrbKFgPwNpDKlvQnu9
2TtVGV7GCVjiSjGu%2fuZE8yrszMnoFtLM0T4oFMifa2k4nQ4WqNULe0t4V6%0d%0acbnvVrj8W5U
TcnkzsKpCqg6e9zILtTShfgedWj3NUhS4vBfAAqiijbTZXCtsT98TaOsPFaNGD4Nf%0d%0aJcl4O9rK8
ymOBwdj7%2fRXJ4F8KcF1lqPBFYy9lFLEKfE2USWiXg%3d%3d</MessageLink>
            </Message>
            <Message>
            <CorrespondenceID>418</CorrespondenceID>
            <Date>02/11/2011</Date>
            <Subject/>
            <Type>0</Type>
            <TemplateName>KM Letter Template</TemplateName>
            <From>McEachern, Kevin</From>
            <JobID>92475</JobID>
            <JobTitle>Test 4000 Characters in JD</JobTitle>
            <MessageLink>https://sqa-tgweb-
01.brassring.com/1033/asp/tg/redirectcandidate.asp?q=%5eudzQFPVk94hM37MiZWaGjTn1SBeuHi
NcQ6R4O1vBGsJdh4ZzRXSEsrJYm%2f2SqUWQ2WM7IqU3Mf8Y%0d%0aqrbKFgPwNpDKlvQnu9
2TtVGV7GCVjiSjGu%2fuZE8yrszMnoFtLM0T4oFMifa2k4nQ4WqNULe0t4V6%0d%0acbnvVrj8W5U
TcnkzsKpCqg6e9zILtTShfgedWj3NUhS4vBfAAqiijbTZXCtsT98TaOsPFaNGD4Nf%0d%0aJcl4O9rK8
ymOBwdj7%2fRXJ4F8KcF1lqPBFYy9lFJ0uHFzYupuMA%3d%3d</MessageLink>
            </Message>
            </Messages>
            </ResultSet>
            </Payload>
            <Status>
            <Code>200</Code>
            <ShortDescription>Success</ShortDescription>
            <LongDescription>Call was successfully processed without error.</LongDescription>
            </Status>
            </Packet>
            <Status>
            <Code>200</Code>
            <ShortDescription>Success</ShortDescription>
            <LongDescription>Call was successfully processed without error.</LongDescription>
            </Status>
            </Unit>
            </CipherValue>
            </CipherData>
            </EncryptedData>
            <Status>
            <Code>200</Code>
            <ShortDescription>Message decrypted and processed without errors.</ShortDescription>
            <LongDescription/>
            </Status>
</Unit>
<Status>
            <Code>200</Code>
            <ShortDescription>Success</ShortDescription>
            <LongDescription>All units are processed with success codes</LongDescription>
</Status>
</Envelope>
```

BR ⟶ Client

To view the XML Element Table for this API, click <u>here.</u>

To view error codes for this API, click <u>here</u>.

Event Manager Event Search

The Event Manager XML search request can retrieve and display scheduled events that candidates have been invited or registered to attend via self-registration. In addition, this API can also retrieve all public events that meet defined search criteria.

> **Note:** The GetEvents API does NOT retrieve events completed through manually scheduled for the candidate by an Event Manager Administrator. Only events to which the candidate has been invited and self scheduled are returned via the API.

Clients can use this API to return and display the following types of information for one instance or a series of events:

- event types (public/candidate specific)
- events scheduled within a specific date range

Clients can use any of the variables for each event type when defining their event XML requests. For example, a client's request could include candidate specific and date range information as well as location.

<u>Event Types:</u> Clients can invite candidates to public events or candidate specific events. Public events are events candidates have not been invited to attend. Candidate specific events are those events candidates have registered for or been invited to attend.

- If request XML does not include the Client ID or authentication token, only public events display.

<u>Events Scheduled within a Specific date range</u>: Clients can request start and end date range for events. If no date range is included in the XML request, the system returns the next 90 days of public events. In general, the XML response returns and displays the following event information for candidate events:

- Event ID
- Event Type (Instance or Event Series)
- Event Name
- Event Date/Time*
- Event Location*
- Event Description**
- Candidate Status (Yes if candidate
- Event URL
- Req ID/Title
- Custom Field information (If Event Custom Field information is configured in Event Manager)
- Requisition ID/Job Title

* If a candidate is invited to an event series these fields are not populated in the response. In the event the candidate is invited to "any scheduled event" only the candidate status, event URL and Req data will be available.

Once configured, clients can choose to display the results on an event search results page or on the candidate portal page. Requests, please review business rules <u>here.</u> Configuration of Event Manager is required for this API. For more information concerning configuration, please contact your Support Team member.

Event Manager Request XML (Formerly Sample Request XML)

Web Service Call: GetEvents

```
<Payload>
 <InputString>
 <AuthenticationToken></AuthenticationToken>
 <ClientID></ClientID>
<Criteria>
        <EventType>
        <Value></Value>
        <Value></Value>
```

BR ← Client

```
            </EventType>
<Location>
        <Value></Value>
        <Value></Value>
</Location>
<City>
<Value></Value>
        <Value></Value>
</City>
<Region>
<Value></Value>
        <Value></Value>
</Region>
<PostalCode>
<Value></Value>
        <Value></Value>
</PostalCode>
<Country>
<Value></Value>
        <Value></Value>
</Country>
<Status>
        <Value></Value>
        <Value></Value>
        </Status>
<StartDate>yyyy/mm/dd hh:mm</StartDate>
        <EndDate>yyyy/mm/dd hh:mm</EndDate>
</Criteria>
 </InputString>
 </Payload>
```

BR ← Client

Business Rules:

- If the Request XML does not include an authentication token then only public events will be returned in the results set.
- If "Criteria" is not provided then the results set will not include any public events.
- Any" Values" provided must match Event Manager data exactly to be considered a match and be returned in the results set.
- Multiple "Values" may be provided for each criteria
- If start date and/or end date is missing then the results set will default to returning the next 90 days worth of public events which match the search criteria entered.
- Within an individual criteria the API will return results based on an 'OR' logic (Event Type = A OR B), across criteria the API will return results based on 'AND' logic (Event Type = X AND (Location = Y OR Z)

The following is an example of the XML response that displays events to candidates. There are also specific business rules related to Event Manager XML Responses, to review the business rules click here.

Event Manager Response XML

```
<ResultSet>
 <CandidateEvents>
<Event>
<Event ID></EventID>
<EventTemplate></EventTemplate>
<EventSeries></EventSeries>
<EventInstance></EventInstance>
<EventDateTime></EventDateTime>
<EventLocation></EventLocation>
<EventCity></EventCity>
<EventRegion></EventRegion>
<EventPostalCode></EventPostalCode>
<EventCountry></EventCountry>
<EventDescription></EventDescription>
```

BR → Client

```
<CandidateStatus></CandidateStatus>
<EventURL></EventURL>
        <Reqs>
<Req ID="" AutoReqID="">
        <JobTitle/>
</Req>
<CustomFields>
<Field>
<Name></Name>
<Value></Value>
</Field>
<Field>
<Name></Name>
<Value></Value>
</Field>
</CustomFields>
</Event>
 </CandidateEvents>
 <PublicEvents>
<Event>
<Event ID></EventID>
<EventTemplate></EventTemplate>
<EventSeries></EventSeries>
< EventInstance></EventInstance>
<Event Date></EventDate>
<EventTime></EventTime>
<EventTimeZone></EventTimeZone>
<EventLocation></EventLocation>
<EventCity></EventCity>
<EventRegion></EventRegion>
<EventPostalCode></EventPostalCode>
<EventCountry></EventCountry>
<EventDescription></EventDescription>
 <CustomFields>
<Field>
<Name></Name>
<Value></Value>
</Field>
<Field>
<Name></Name>
<Value></Value>
</Field>
</CustomFields>
        </Event>
 </PublicEvents>
</ResultSet>
```

BR ⟶ Client

Event Manager Response XML

```
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
 <Sender>
        <Id>1364</Id>
        <Credential>1sU9JELZ8z9tPsHw8FDjyw==</Credential>
</Sender>
<Recipient/>
<TransactInfo transactType="response" transactId="e570c9cd-f86c-4e89-951c-
e359dfaf23f3">
        <TransactId>6/1/2011</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="Decryptor">
        <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element" encoding="utf-8"
xmlns="http://www.w3.org/2001/04/xmlenc#">
```

BR ⟶ Client

```
                    <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
                    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                    <ds:KeyName>Integration</ds:KeyName>
                    </ds:KeyInfo>
                    <CipherData>
                    <CipherValue>
                    <Unit UnitProcessor="GetEvents">
                    <Packet>
                    <PacketInfo packetType="data" encrypted="yes">
                    <packetId>1</packetId>
                    </PacketInfo>
                    <Payload>
                    <InputString>
<AuthenticationToken>763b477b-d70d-406b-a456-5997958a7300</AuthenticationToken>
        <ClientId>11714</ClientId>
        <Criteria>
<EventType>
<Value/>
</EventType>
<Location>
<Value/>
</Location>
        <City>
<Value/>
        </City>
        <Region>
<Value/>
        </Region>
<PostalCode>
        <Value/>
</PostalCode>
        <Country>
<Value/>
        </Country>
        <Status>
<Value/>
        </Status>
<StartDate>2007/12/12 01:01</StartDate>
<EndDate>2011/12/12 01:01</EndDate>
        </Criteria>
        </InputString>
        <ResultSet>
        <CandidateEvents>
        <Events>
<Event>
<EventID/>
<EventTemplate>KM Template 2 - Private Event</EventTemplate><EventSeries>KM Event 7
- Private -Series</EventSeries>
<EventInstance/>
<EventDateTime/>
        <EventLocation/>
<EventCity/>
<EventRegion/>
<EventPostalCode/>
<EventCountry/>
<EventDescription/>
<CandidateStatus>Invited</CandidateStatus>
<EventURL>https://qalabs1.brassring.com/satori2/eventmanager/register.aspx?BRUrlParams
=bEJjZ01HU2VjK003WXdLMWV4eUM2anRnYmhiTDB2N1Y3UlZ2Z2tWTFgzalhIdDhySEdiZD
BXMlNVa1lsWVJIeEttSVBDa3htclvMUZOdXRGZU5CU0poNmNydXVNNHBXcEd6cXBraGGR
1aVdQeVY1elVaanJacjdYSG1ZV0N3WlNGbDIHdDBaWjAzZGJPYkVIdDVlmOHNYdzhLT3M3K
29hblUyRzJRTIczb0dxdDBxL3lXWGl6YXFpZjZBSjRRdEhtS0ZxK0hpdXJmQnZnoT0FreUhWWY
21WaXR6cllkNnhJWCtaSmJvNWFGdHgyMzYzZzlERnA2NzZJdUFodWEMWNrNHhHRWF
SMU9GU21PM3R3Z29iVWIrHlJZFpjWFdhdjlEaVBsZ3N3Z0FRRGoyddo3RlNudVY4V2lGU
```

```
FNjanBrdW00NWZoYUp1cCtCVFBjT2tBQVVPYmVPcmF6MFRNamRyc2lyOHV3RzRLKzFve
TVQM1psb2RqUFQ4SXNwb09KZ2tlT3h5WjhMUnN6U2RjODBIQXI3VFkxV0RWdkZTTGtiL2h
iV004Z0ZaeDFJditwUElEbC91WXJ3dmtjMFFFbmNWJFWmRSaUFsSUxHK1BZWk5FUGgvL2c
4dzQwd1p0QnVJM01q0</EventURL>
<CustomFields/>
</Event>
        </Events>
</CandidateEvents>
        <PublicEvents>
        <Events>
<Event>
<EventID>7467</EventID>
<EventTemplate>KM Event Template with all</EventTemplate>
<EventSeries/>
<EventInstance>KM Event 1</EventInstance>
<EventDate>29 Apr 2011</EventDate>
<EventTime>9:00AM</EventTime>
<EventTimeZone>Eastern Time (US &amp; Canada)</EventTimeZone>
<EventLocation>Vizag</EventLocation>
<EventCity/>
<EventRegion/>
<EventPostalCode/>
<EventCountry/>
<EventDescription>KM Event Template - Description field for KM Event 1</EventDescription>
<CustomFields>
<Field>
        <Name>KM Custom Field 1</Name>
        <Value>KM Custom Field 1 - Text</Value>
        </Field>
        <Field>
        <Name>KM Custom Field 3 Date</Name>
        <Value>4/22/2011 12:00:00 AM</Value>
        </Field>
        <Field>
        <Name>KM Custom Fiield 2</Name>
        <Value>KM Custom Field 2 - Text</Value>
</Field>
</CustomFields>
</Event>
<Event>
<EventID>7527</EventID>
<EventTemplate>KM Event Template with all</EventTemplate> <EventSeries>KM Event 10 -
Public Series</EventSeries> <EventInstance>KM Event 10 - Public Series</EventInstance>
<EventDate>01 May 2011</EventDate>
<EventTime>9:00AM</EventTime>
<EventTimeZone>Eastern Time (US &amp; Canada)</EventTimeZone>
        <EventLocation>Burlington</EventLocation>
<EventCity>Burlington</EventCity>
<EventRegion>MA</EventRegion>
<EventPostalCode>01802</EventPostalCode>
<EventCountry>US</EventCountry>
<EventDescription>KM Event Template - Description field</EventDescription>
        <CustomFields>
<Field>
<Name>KM Custom Field 1</Name>
<Value>Custom 1 slkjsd</Value>
        </Field>
<Field>
<Name>KM Custom Fiield 2</Name>
<Value>Custom 2 aslkdjf</Value>
</Field>
<Field>
        <Name>KM Custom Field 3 Date</Name>
        <Value>4/30/2011 12:00:00 AM</Value>
```

BR ⟶ Client

```
        </Field>
</CustomFields>
</Event>
<Event>
<EventID>7475</EventID>
<EventTemplate>KM Event Template with all</EventTemplate>
<EventSeries/>
<EventInstance>KM Event 2</EventInstance>
<EventDate>05 May 2011</EventDate>
<EventTime>9:00AM</EventTime>
        <EventTimeZone>Eastern Time (US &amp; Canada)</EventTimeZone>
        <EventLocation>Vizag</EventLocation>
<EventCity/>
<EventRegion/>
        <EventPostalCode/>
        <EventCountry/>
<EventDescription>KM Event Template - Description field for KM Event 1</EventDescription>
<CustomFields>
        <Field>
         <Name>KM Custom Field 1</Name>
         <Value>KM Custom Field 1 - Text</Value>
        </Field>
        <Field>
         <Name>KM Custom Field 3 Date</Name>
         <Value>4/22/2011 12:00:00 AM</Value>
        </Field>
        <Field>
         <Name>KM Custom Fiield 2</Name>
         <Value>KM Custom Field 2 - Text</Value>
        </Field>
</CustomFields>
</Event>
<Event>
<EventID>7630</EventID>
<EventTemplate>KM Public Event Template</EventTemplate>
<EventSeries>sss</EventSeries>
<EventInstance>sss222</EventInstance>
<EventDate>24 May 2011</EventDate>
<EventTime>9:00AM</EventTime>
<EventTimeZone>Pacific Time (US &amp; Canada)</EventTimeZone>
<EventLocation/>
<EventCity/>
<EventRegion/>
<EventPostalCode/>
<EventCountry/>
<EventDescription>KM Public Event Template -</EventDescription>
<CustomFields/>
        </Event>
<Event>
<EventID>7476</EventID>
<EventTemplate>KM Event Template with all</EventTemplate>
<EventSeries/>
<EventInstance>KM Event 3</EventInstance>
<EventDate>26 May 2011</EventDate>
<EventTime>9:00AM</EventTime>
<EventTimeZone>Eastern Time (US &amp; Canada)</EventTimeZone>
<EventLocation>Vizag</EventLocation>
<EventCity/>
<EventRegion/>
<EventPostalCode/>
<EventCountry/>
<EventDescription>KM Event Template - Description field for KM Event 1</EventDescription>
<CustomFields>
        <Field>
```

BR ⟶ Client

```
                <Name>KM Custom Field 1</Name>
                <Value>KM Custom Field 1 - Text</Value>
                </Field>
                <Field>
                <Name>KM Custom Field 3 Date</Name>
                <Value>4/22/2011 12:00:00 AM</Value>
                </Field>
                <Field>
                <Name>KM Custom Fiield 2</Name>
                <Value>KM Custom Field 2 - Text</Value>
                </Field>
</CustomFields>
</Event>
<Event>
        <EventID>7545</EventID>
<EventTemplate>KM Event Template with all</EventTemplate>
<EventSeries/>
<EventInstance>KM Event 11 - Public</EventInstance>
<EventDate>26 May 2011</EventDate>
<EventTime>9:00AM</EventTime>
<EventTimeZone>Eastern Time (US &amp; Canada)</EventTimeZone>
<EventLocation>Watertown</EventLocation>
<EventCity/>
<EventRegion/>
<EventPostalCode/>
<EventCountry/>
        <EventDescription>KM Event Template - Description field</EventDescription>
        <CustomFields>
        <Field>
        <Name>KM Custom Field 1</Name>
        <Value>askdfjskdl</Value>
        </Field>
        <Field>
        <Name>KM Custom Fiield 2</Name>
        <Value>asdflkajsd</Value>
        </Field>
        <Field>
        <Name>KM Custom Field 3 Date</Name>
        <Value>5/31/2011 12:00:00 AM</Value>
        </Field>
        </CustomFields>
        </Event>
        <Event>
        <EventID>7528</EventID>
        <EventTemplate>KM Event Template with all</EventTemplate>
        <EventSeries>KM Event 10 - Public Series</EventSeries>
        <EventInstance>KM Event 10 - Public Series</EventInstance>
        <EventDate>01 Jun 2011</EventDate>
        <EventTime>9:00AM</EventTime>
        <EventTimeZone>Eastern Time (US &amp; Canada)</EventTimeZone>
        <EventLocation>Burlington</EventLocation>
        <EventCity/>
        <EventRegion/>
        <EventPostalCode/>
        <EventCountry/>
        <EventDescription>KM Event Template - Description field</EventDescription>
        <CustomFields>
        <Field>
        <Name>KM Custom Field 1</Name>
        <Value>Custom 1 slkjsd</Value>
        </Field>
        <Field>
        <Name>KM Custom Fiield 2</Name>
        <Value>Custom 2 aslkdjf</Value>
```

BR ⟶ Client

```
        </Field>
        <Field>
        <Name>KM Custom Field 3 Date</Name>
        <Value>4/30/2011 12:00:00 AM</Value>
        </Field>
        </CustomFields>
        </Event>
        <Event>
        <EventID>7506</EventID>
        <EventTemplate>KM Event Template with all</EventTemplate>
        <EventSeries/>
        <EventInstance>KM Event 9 - Public</EventInstance>
        <EventDate>30 Jun 2011</EventDate>
        <EventTime>9:00AM</EventTime>
        <EventTimeZone>Greenwich Mean Time : Dublin, Edinburgh, Lisbon,
London</EventTimeZone>
        <EventLocation>Cambridge, MA</EventLocation>
        <EventCity/>
        <EventRegion/>
        <EventPostalCode/>
        <EventCountry/>
        <EventDescription>KM Event Template - Description field</EventDescription>
        <CustomFields>
        <Field>
        <Name>KM Custom Field 1</Name>
        <Value>Custom Text 1 aasdfsd</Value>
        </Field>
        <Field>
        <Name>KM Custom Field 3 Date</Name>
        <Value>6/29/2011 12:00:00 AM</Value>
        </Field>
        <Field>
        <Name>KM Custom Fiield 2</Name>
        <Value>Custom Text 2 asdpoifj</Value>
        </Field>
        </CustomFields>
        </Event>
        <Event>
        <EventID>7529</EventID>
        <EventTemplate>KM Event Template with all</EventTemplate>
        <EventSeries>KM Event 10 - Public Series</EventSeries>
        <EventInstance>KM Event 10 - Public Series</EventInstance>
        <EventDate>01 Jul 2011</EventDate>
        <EventTime>9:00AM</EventTime>
        <EventTimeZone>Eastern Time (US &amp; Canada)</EventTimeZone>
        <EventLocation>Burlington</EventLocation>
        <EventCity/>
        <EventRegion/>
        <EventPostalCode/>
        <EventCountry/>
        <EventDescription>KM Event Template - Description field</EventDescription>
        <CustomFields>
        <Field>
        <Name>KM Custom Field 1</Name>
        <Value>Custom 1 slkjsd</Value>
        </Field>
        <Field>
        <Name>KM Custom Fiield 2</Name>
        <Value>Custom 2 aslkdjf</Value>
        </Field>
        <Field>
        <Name>KM Custom Field 3 Date</Name>
        <Value>4/30/2011 12:00:00 AM</Value>
        </Field>
```

BR ⟶ Client

```
            </CustomFields>
            </Event>
            <Event>
            <EventID>7534</EventID>
            <EventTemplate>KM Event Template with all</EventTemplate>
            <EventSeries>KM Event 10 - Public Series</EventSeries>
            <EventInstance>KM Event 10 - Public Series</EventInstance>
            <EventDate>01 Dec 2011</EventDate>
            <EventTime>9:00AM</EventTime>
            <EventTimeZone>Eastern Time (US &amp; Canada)</EventTimeZone>
            <EventLocation>Burlington</EventLocation>
            <EventCity/>
            <EventRegion/>
            <EventPostalCode/>
            <EventCountry/>
            <EventDescription>KM Event Template - Description field</EventDescription>
            <CustomFields>
            <Field>
            <Name>KM Custom Field 1</Name>
            <Value>Custom 1 slkjsd</Value>
            </Field>
            <Field>
            <Name>KM Custom Fiield 2</Name>
            <Value>Custom 2 aslkdjf</Value>
            </Field>
            <Field>
            <Name>KM Custom Field 3 Date</Name>
            <Value>4/30/2011 12:00:00 AM</Value>
            </Field>
            </CustomFields>
            </Event>
            </Events>
            </PublicEvents>
            </ResultSet>
            </Payload>
            <Status>
            <Code>200</Code>
            <ShortDescription>Success</ShortDescription>
            <LongDescription>Call was successfully processed without error.</LongDescription>
            </Status>
            </Packet>
            <Status>
            <Code>200</Code>
            <ShortDescription>Success</ShortDescription>
            <LongDescription>Call was successfully processed without error.</LongDescription>
            </Status>
            </Unit>
            </CipherValue>
            </CipherData>
            </EncryptedData>
            <Status>
            <Code>200</Code>
            <ShortDescription>Message decrypted and processed without
errors.</ShortDescription>
            <LongDescription/>
            </Status>
</Unit>
<Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>All units are processed with success codes</LongDescription>
</Status>
</Envelope>
```

Business Rules:

- The Response XML will list events by date sent, in ascending order.
- Any custom field associated with the Event will be provided in the results set.
- The PublicEvents results set will include only Events that are marked as public, regardless of whether the candidate has been invited or registered, these will also be included.
- If an authentication token is provided in the request XML then both candidate specific events and public events will be provided in the results set. If the authentication token is missing then only the public events will be provided.
- If there is no search criteria provided then no public events will be returned.
- When a candidate has been invited/registered to an Event Instance, the following nodes are populated

    o EventID - Yes

    o EventTemplate - Yes

    o EventSeries - Yes

    o EventInstance - Yes

    o EventDate/Time – Yes

    o EventLocation – Yes

    o EventDescription – Yes

    o CandidateStatus – Yes

    o EventURL – Yes

    o Req – Yes (if available)

    o CustomFields - Yes

- When a candidate is invited to an Event Series, the following nodes are populated

    o EventID - Yes (available when candidate is registered, not available when invited)

    o EventTemplate - Yes

    o EventSeries - Yes

    o EventInstance - No

    o EventDate/Time – No

    o EventLocation – No

    o EventDescription – No

    o CandidateStatus – Yes

    o EventURL – Yes

    o Req – Yes (if available)

    o CutomFields – No

- When the candidate is invited to an EventTemplate, the following nodes are populated

    o EventID - Yes (available when candidate is registered, not when invited)

    o EventTemplate - Yes

    o EventSeries - No

    o EventInstance - No

    o EventDate/Time – No

    o EventLocation – No

    o EventDescription – No

- o CandidateStatus – Yes

- o EventURL – Yes

- o Req – Yes (if available)

- o CutomFields – No

- In the event the candidate is invited to "any scheduled event" only the candidate status, eventURL and Req data will be available.
- Client "Best Practice": to present the Event "name" clients should use only one of the 3 following nodes: EventInstance, EventSeries, or EventTemple, using whichever of the most granular is available. EventTemple is least granular, EventSeries has the next level of granularity, EventInstance is the most granular.

> Use Ctrl + Click to follow hyperlinks. Use ALT + LEFT ARROW to return to point of origin.

To view the XML Element Table for this API, click here.

To view error codes for this API, click here.

Document/Document Packet

This API allows clients to post links to document packets on a Candidate Portal page. When candidates click the document packet link, they are redirected to their document packet that contains active links to documents.

Configuration of Candidate Portal pages and posting documents and document packets must be completed in BrassRing before enabling this API. For more information concerning these configurations, please contact your Support Team member.

Each API has a required XML input string that clients must use when requesting responses from the BrassRing API.

```
<InputString>
<ClientId>516</ClientId>
<SiteId>8723</SiteId>
<AuthenticationToken>bfd0bfed-c763-4a88-9833-947cf3dcd50d</AuthenticationToken>
 </InputString>
```

BR ← Client

Document/Document Packet Request XML

Web Service Call: GetCandidatePortalInfo

```
  <?xml version="1.0" encoding="utf-8" ?>
- <Envelope version="01.00">
- <Sender>
  <Id>01801</Id>
  <Credential>516</Credential>
  </Sender>
- <TransactInfo transactType="data" transactId="075cbfe6-b9dc-49f9-8f12-8eae62b62055">
  <TransactId>10/24/2008</TransactId>
  <TimeStamp>12:00:00 AM</TimeStamp>
  </TransactInfo>
- <Unit UnitProcessor="GetCandidatePortalInfo">
- <Packet>
- <PacketInfo packetType="data">
  <packetId>1</packetId>
  </PacketInfo>
- <Payload>
- <InputString>
  <AuthenticationToken>a15cd9f4-84a8-4483-819d-f4d149b3a410</AuthenticationToken>
  </InputString>
  </Payload>
  </Packet>
  </Unit>
  </Envelope>
```

BR ← Client

XML Document/Document Packet Response XML

```
<?xml version="1.0" encoding="utf-8" ?>
- <Envelope version="01.00">
- <Sender>
  <Id>!@#$%^*()</Id>
  <Credential>pd3HhL1lfYDp/xqxPPjCCA==</Credential>
  </Sender>
  <Recipient />
- <TransactInfo transactType="response" transactId="8c2cb404-f935-4668-81f4-b12280a8c91b">
  <TransactId>10/16/2012</TransactId>
  <TimeStamp>12:00:00 AM</TimeStamp>
  </TransactInfo>
- <Unit UnitProcessor="Decryptor">
- <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element" encoding="utf-8"
  xmlns="http://www.w3.org/2001/04/xmlenc#">
  <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />
- <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:KeyName>Integration</ds:KeyName>
  </ds:KeyInfo>
- <CipherData>
- <CipherValue>
- <Unit UnitProcessor="GetCandidatePortalInfo">
- <Packet>
- <PacketInfo packetType="data" encrypted="yes">
  <packetId>1</packetId>
  </PacketInfo>
- <Payload>
- <InputString>
  <AuthenticationToken>28ca81d5-6420-4a31-bec9-830d3e1ba2ba</AuthenticationToken>
  </InputString>
- <ResultSet>
- <Documents>
- <Document>
  <DatePosted><![CDATA[10/12/2012]]></DatePosted>
  <Name><![CDATA[Doc with subform - oct12th.pdf]]></Name>
  <Type>portal</Type>
  <JobID>33BR</JobID>
  <JobTitle><![CDATA[Testing LDP096]]></JobTitle>
  <SentBy><![CDATA[Anandakumar, Usha]]></SentBy>
  <RemovalDate><![CDATA[01/01/1900]]></RemovalDate>
  <Url><![CDATA[https://sqa-tgweb-
01.brassring.com/tgwebhost/redirectcandidate.aspx?q=%5e8cjY8MLROUrd4DgSm8F9sgctAOl0hMd
zsek2p8gKlnZUupcfAc5St5nuCMe%2b18KkHclB7gUbAmdL%0d%0adhXF9S8p2OpFh6S91hTOG7t
oEdddYw8i1XUE3pQXlQ%2b%2bVmPUVvJy4y3YZPT86NKtvNZYtRrrwceu%0d%0arjxwXSW4n7N
m7OBegEbHL4j70npPszrdHnel2rOcpwKkoU7HWMvwVubHSfGTFlvwZwNondw8g7vV%0d%0alWIL
xEP1NNRJagHPLloPFH2PnPW4B2Wzf3XQZSl4gl4td3aONOqucCML0z9IyVco9JkE5fKNGB9D%0d
%0atNYy9tXPRKZLWQbCoj6wgmBLIpynLnCONhw1tbRUoKvQbBlzkXtVngt%2be2X%2f%2bRkAYZ
ZBfpwMy1IK%0d%0aagjNFAQJrs7zom8cqwDzcgZ4n%2bfbzKkj2yP6YBxpa0ni9ZeioSp%2f6Rb8d1U
efUHdB9vekr3islWlt%0d%0amoMyOms5dVfV%2bvZ5%2fAK%2fAJB8hIYf0TrCp%2bIsKF9h%2bOQ
CRBn7r1ktVKSW%2fxIWcoP7NbxtyYqUfSwd%0d%0a0Q%3d%3d]]></Url>
  </Document>
- <Document>
  <DatePosted><![CDATA[10/12/2012]]></DatePosted>
  <Name><![CDATA[Doc packet with form in template]]></Name>
  <Type>DocumentPacket</Type>
  <JobID>33BR</JobID>
  <JobTitle><![CDATA[Testing LDP096]]></JobTitle>
  <SentBy><![CDATA[Anandakumar, Usha]]></SentBy>
  <RemovalDate><![CDATA[12/11/2012]]></RemovalDate>
  <Url><![CDATA[https://sqa-tgweb-
01.brassring.com/tgwebhost/redirectcandidate.aspx?q=%5e8cjY8MLROUrd4DgSm8F9sgctAOl0hMd
zsek2p8gKlnZUupcfAc5St5nuCMe%2b18KkHclB7gUbAmdL%0d%0adhXF9S8p2OpFh6S91hTOG7t
oEdddYw8i1XUE3pQXlQ%2b%2bVmPUVvJy4y3YZPT86NKtvNZYtRrrwRs7%0d%0aUmJFbbtaGcJ
```

BR ➝ Client

4eK9cQAKzCSZxYgQp1pxuZigbeICbCiFS5nYxIasLGHKHJA%2bACtJVILTCDKAtg4T7%0d%0a6YI
zbzHNj4YyEz4kMFrPE6HZqsMXNkm%2b3j0GZxtwEiRmdlmLwfxmyUNJ2P7LOOHL3sKdXyl7Tv8V
%0d%0aYbRcVREe4LWiFZP5WwHwEAL8xyxEXOqIwbiQxWWpHVqiNCOJAPvaHmPG1qbM4GUd
8cdT5HDoVIU9%0d%0arz6XZ5j3SBW5UkdBgIgYn7qZsw%3d%3d]]></Url>
```
 </Document>
 </Documents>
 </ResultSet>
 </Payload>
- <Status>
 <Code>200</Code>
 <ShortDescription>Success</ShortDescription>
 <LongDescription>Call was successfully processed without error.</LongDescription>
 </Status>
 </Packet>
- <Status>
 <Code>200</Code>
 <ShortDescription>Success</ShortDescription>
 <LongDescription>Call was successfully processed without error.</LongDescription>
 </Status>
 </Unit>
 </CipherValue>
 </CipherData>
 </EncryptedData>
- <Status>
 <Code>200</Code>
 <ShortDescription>Message decrypted and processed without errors.</ShortDescription>
 <LongDescription />
 </Status>
 </Unit>
- <Status>
 <Code>200</Code>
 <ShortDescription>Success</ShortDescription>
 <LongDescription>All units are processed with success codes</LongDescription>
 </Status>
 </Envelope>
```


To view the XML Element table for this API, click here.

To view the error codes for this API, click here.

Forgot/Change Password

The Forgot Password API allows candidates to recover forgotten passwords or to change their passwords during logins to a client's website. During initial login, candidates set security questions that are used to validate identity and allow password reset.

When a candidate forgets their password, they must first answer one of their security questions (GetSecurity Questions API) and then the candidate is allowed to resest their password (Forgot Password API). A username or token is required to use this API..

For more information concerning configuration, please contact your Support Team member.

Get Security Question Request using Username

Web Service Call: GetSecurityQuestions

```
  <?xml version="1.0" encoding="utf-8" ?>
- <Envelope version="01.00">
- <Sender>
  <Id>UCA12</Id>
  <Credential>22022</Credential>
  </Sender>
- <TransactInfo transactType="data" transactId="075cbfe6-b9dc-49f9-8f12-8eae62b62055">
  <TransactId>10/24/2008</TransactId>
  <TimeStamp>12:00:00 AM</TimeStamp>
  </TransactInfo>
- <Unit UnitProcessor="GetSecurityQuestions">
- <Packet>
- <PacketInfo packetType="data">
  <packetId>1</packetId>
  </PacketInfo>
- <Payload>
- <InputString>
  <ClientId>22022</ClientId>
  <SiteId>5493</SiteId>
  <Username>october12</Username>
  </InputString>
  </Payload>
  </Packet>
  </Unit>
  </Envelope>
```

BR ← Client

Get Security Question with Authentication Token Request

Web Service Call: GetSecurityQuestions (authenticated)

```
<?xml version="1.0" encoding="utf-8" ?>
- <Envelope version="01.00">
- <Sender>
  <Id>UCA12</Id>
  <Credential>22022</Credential>
  </Sender>
- <TransactInfo transactType="data" transactId="075cbfe6-b9dc-49f9-8f12-8eae62b62055">
  <TransactId>10/24/2008</TransactId>
  <TimeStamp>12:00:00 AM</TimeStamp>
  </TransactInfo>
- <Unit UnitProcessor="GetSecurityQuestions">
- <Packet>
- <PacketInfo packetType="data">
  <packetId>1</packetId>
  </PacketInfo>
- <Payload>
- <InputString>
  <ClientId>22022</ClientId>
  <SiteId>5493</SiteId>
- <AuthenticationToken>
- <![CDATA[ 50c6f255-77c0-4d51-a670-b11deffbbce0
  ]]>
  </AuthenticationToken>
  </InputString>
  </Payload>
  </Packet>
  </Unit>
  </Envelope>
```

BR ← Client

To view the Error Codes for this API, click here.

Forgot/Change Password Request XML

Web Service Call: ForgotChangePassword

```
<?xml version="1.0" encoding="utf-8" ?>
- <Envelope version="01.00">
- <Sender>
  <Id>UCA12</Id>
  <Credential>22022</Credential>
  </Sender>
- <TransactInfo transactType="data" transactId="075cbfe6-b9dc-49f9-8f12-8eae62b62055">
  <TransactId>10/24/2008</TransactId>
  <TimeStamp>12:00:00 AM</TimeStamp>
  </TransactInfo>
- <Unit UnitProcessor="ForgotChangePassword">
- <Packet>
- <PacketInfo packetType="data">
  <packetId>1</packetId>
  </PacketInfo>
- <Payload>
- <InputString>
  <ClientId>22022</ClientId>
  <SiteId>5493</SiteId>
  <Username>october12</Username>
- <securityQuestions>
- <securityQuestion>
  <ID>1</ID>
- <Answer>
- <![CDATA[ stanes
  ]]>
  </Answer>
  </securityQuestion>
  </securityQuestions>
- <NewPassword>
- <![CDATA[ password@
  ]]>
  </NewPassword>
  </InputString>
  </Payload>
  </Packet>
  </Unit>
  </Envelope>
```

BR ← Client

Forgot/Change Password Request XML (no security question required)

The ForgotChangePassword API also allows authenticated candidates to reset their passwords without having to provide security question/response when authentication token used is still valid.

Web Service Call: ForgotChangePassword (no security question)

```
<?xml version="1.0" encoding="utf-8" ?>
- <Envelope version="01.00">
- <Sender>
  <Id>UCA12</Id>
  <Credential>22022</Credential>
  </Sender>
- <TransactInfo transactType="data" transactId="075cbfe6-b9dc-49f9-8f12-8eae62b62055">
  <TransactId>10/24/2008</TransactId>
  <TimeStamp>12:00:00 AM</TimeStamp>
  </TransactInfo>
- <Unit UnitProcessor="ForgotChangePassword">
- <Packet>
- <PacketInfo packetType="data">
  <packetId>1</packetId>
  </PacketInfo>
```

BR ← Client

```
- <Payload>
- <InputString>
  <ClientId>22022</ClientId>
  <SiteId>5390</SiteId>
- <AuthenticationToken>
- <![CDATA[
0ca815b6-4750-40ce-92bb-59e088226e56
  ]]>
  </AuthenticationToken>
- <NewPassword>
- <![CDATA[
password$$
  ]]>
  </NewPassword>
  </InputString>
  </Payload>
  </Packet>
  </Unit>
  </Envelope>
```

ForgotChangePassword Response

```
<?xml version="1.0" encoding="utf-8" ?>

  <string xmlns="http://integrationuri.org/"><?xml version="1.0" encoding="utf-

8"?>
<Envelope version="01.00">
<Sender>
<Id>UCA12</Id>
<Credential>22022</Credential>
</Sender>
<TransactInfo transactType="response" transactId="0914ba86-8784-4fc0-b3be-

afb9ac61a1a7">
<TransactId>10/24/2008</TransactId>
<TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="ForgotChangePassword">
<Packet>
<PacketInfo packetType="data" encrypted="no">
<packetId>1</packetId>
</PacketInfo>
<Payload>
<InputString>
<ClientId>22022</ClientId>
<SiteId>5390</SiteId>
<AuthenticationToken>54e30318-e441-41e1-992b-99e06bd52f28</AuthenticationToken>
<NewPassword>password$$</NewPassword>
</InputString>
<ResultSet>Success</ResultSet>
</Payload>
<Status><Code>200</Code>
<ShortDescription>Success</ShortDescription>
<LongDescription>Call was successfully processed without error.</LongDescription>
</Status></Packet><Status>
<Code>200</Code>
<ShortDescription>Success</ShortDescription>
<LongDescription>Call was successfully processed without error.
</LongDescription>
</Status>
</Unit>
<Status><Code>200</Code>
```

```
<ShortDescription>Success</ShortDescription>
<LongDescription>All units are processed with success codes</LongDescription>
</Status>
</Envelope>
</string>
```

BR ⟶ Client

To view the XML Element Table for this API, click here.

To view the Error Codes for this API, click here.

Get Resume

Enables clients to display direct links to a candidate's resume on a candidate portal page. Access to a candidate's resume can include editing capabilities and resume details such as cover letters, revision dates, and submission dates.

For more information concerning configuration, please contact your Support Team member.

Input String

Each API has a required XML input string that clients must use when requesting responses from the BrassRing API.

```
Users input string:
<InputString>
 <AuthenticationToken>fa7309b1-46f0-4176-891d-4d916f287d98</AuthenticationToken>
</InputString>
```

BR ⟵ Client

Get Resume Request XML

```
<ResultSet>
<Resumes>
        <Resume Language="1">
<ResumeId>1</ResumeId>
<ResumeDisplayName></ResumeDisplayName>
<ResumeLink></ResumeLink>
<ResumeRevised>MM/DD/YYYY</ResumeRevised>
<AppliedJobs>
        <Job>
        <JobID></JobID>
        <JobTitle></JobTitle>
        <JobLink></JobLink>
        <CoverLetterID>1</CoverLetterID>
        <CoverLetterName></CoverLetterName>
        <DateSubmitted></DateSubmitted>
        </Job>
</AppliedJobs>
</Resume>
</Resumes>
</ResultSet>
```

BR ⟵ Client

Get Resume Response XML

```
<?xml version="1.0" encoding="utf-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by  () -->
<Envelope version="01.00">
<Sender>
        <Id>QE</Id>
        <Credential>AXc9N72lF+3UhS8HUZq+FA==</Credential>
</Sender>
<Recipient/>
<TransactInfo transactType="response" transactId="230d6ba9-6e79-4ecb-88b8-69bf8c650e02">
        <TransactId>4/26/2011</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
        <Unit UnitProcessor="Decryptor">
```

BR ⟶ Client

```
        <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element" encoding="utf-8"
xmlns="http://www.w3.org/2001/04/xmlenc#">
<resumename>My resume/CV2</resumename>
        <resumeurl>https://sqa-tgweb-
01.brassring.com/1033/asp/tg/redirectcandidate.asp?q=%5eudzQFPVk94hM37MiZWaGjTn1SBe
uHiNco%2fAyx63VjjIb6TNzQe%2fAd517PnJSqyWxMyVy%2befTmM2x%0d%0aZuiU4cWzmgxB
%2bUHoJh9GWOWbO1rD0RaKcxxGTj9BBktYK2u2rUCxXCvvOTdi9oG4j%2bkCbUhBZnAw%0d
%0aThw2eyinuHxQmmBq4oiGqsWaOWko2C2JRFGYcP1In%2bpsL0le1D%2bAIHZs7dJTy2Daj
W4Xci93UvHZ%0d%0a8lErtzz7Nix1x9Fvnit7yf07eJianxIs%2bpYybAa%2foHYFHjF16zg8Qnlelmj
7</resumeurl>
        <resumerevised>12/6/2010</resumerevised>
        <defaultresume>false</defaultresume>
        <jobs/>
        </resume>
        <resume>
        <resumename>My resume/CV3</resumename>
        <resumeurl>https://sqa-tgweb-
01.brassring.com/1033/asp/tg/redirectcandidate.asp?q=%5eudzQFPVk94hM37MiZWaGjTn1SBe
uHiNco%2fAyx63VjjIb6TNzQe%2fAd517PnJSqyWxMyVy%2befTmM2x%0d%0aZuiU4cWzmgxB
%2bUHoJh9GWOWbO1rD0RaKcxxGTj9BBktYK2u2rUCxXCvvOTdi9oG4j%2bkCbUhBZnAw%0d
%0aThw2eyinuHxQmmBq4oiGqsWaOWko2C2JRFGYcP1In%2bpsL0le1D%2bAIHZs7dJTy2Daj
W4Xci93UvHZ%0d%0a8lErtzz7Nix1x9FvnkoieVLH0zvG43FcNe%2brTkJIkBAQ19XjFA%3d%3d<
/resumeurl>
        <resumerevised>12/6/2010</resumerevised>
        <defaultresume>false</defaultresume>
        <jobs/>
        </resume>
        </resumes>
        </ResultSet>
        </Payload>
        <Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>Call was successfully processed without error.</LongDescription>
        </Status>
        </Packet>
        <Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>Call was successfully processed without error.</LongDescription>
</Status>
        </Unit>
        </CipherValue>
        </CipherData>
        </EncryptedData>
        <Status>
        <Code>200</Code>
        <ShortDescription>Message decrypted and processed without errors.</ShortDescription>
        <LongDescription/>
        </Status>
</Unit>
<Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>All units are processed with success codes</LongDescription>
</Status>
</Envelope>
```

BR ⟶ Client

To view the XML Element Table for this API, click here.

To view the error codes for this API, click here.

Get Job Cart

Web Service Call: GetJobCart

The job cart API web service call GetJobCart enables clients to display a candidate's job cart information on a candidate portal page. Clients can also enable the Action element, Add/Remove, that allows candiddates to add or remove jobs from job carts.

Job carts can contain a maximum of 50 jobs. Both GetJobCart options require candidate authentication.

GetJobCart

The GetJobCart web service call enables clients to:

* display a candidate's job cart information on a candidate portal page, including requisition IDs, job titles, job status, date added to cart, job details, and job apply hyperlinks.

The GetJobCart web service call requires candidate authentication before a candidate can access their job cart.

GetJobCart Action Add/Remove

The GetJobCart web service call that includes one of the Actions, Add or Remove, enables candidates to:

* add or remove jobs from their job carts

The GetJobCart web service call action requires candidate authentication before a candidate can access their job cart.

> Site ID should be provided only when Job has to be Added/Removed against non-default member TG (GTG or additional TG)

For more information concerning configuration, please contact your BrassRing Support Team member.

Candidate Authentication

Both GetJobCart web service call options require candidate authentication. When a candidate is authenticated into the Talent Gateway, the API service returns an authentication token used to identify the candidate when sending API web service requests. The authentication token is required to authenticate the candidate's identity when the candidate accesses their job cart.

Input String

Candidate Authentication Request XML

```
        <?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
<Sender>
        <Id>10540</Id>
        <Credential>11721</Credential>
</Sender>
<TransactInfo transactType="data" transactId="075cbfe6-b9dc-49f9-8f12-8eae62b62055">
        <TransactId>10/24/2008</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
</TransactInfo>
<Unit UnitProcessor="AuthenticateTGUser">
        <Packet>
        <PacketInfo packetType="data">
        <packetId>1</packetId>
        </PacketInfo>
        <Payload>
        <InputString>
        <ClientId></ClientId>
        <SiteId></SiteId>
        <Username></Username>
```

BR ← Client

```
    <Password></Password>
    </InputString>
    </Payload>
    </Packet>
 </Unit>
</Envelope>
```

BR ← Client

Candidate Authentication Response XML

BR → Client

```
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
<Sender>
        <Id>1364</Id>
        <Credential>1sU9JELZ8z9tPsHw8FDjyw==</Credential>
</Sender>
<Recipient/>
<TransactInfo transactType="response" transactId="2347cf88-3c8d-4f6c-ab51-bfa0679f2bbe">
        <TransactId>7/1/2011</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="Decryptor">
        <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element" encoding="utf-8"
xmlns="http://www.w3.org/2001/04/xmlenc#">
        <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:KeyName>Integration</ds:KeyName>
        </ds:KeyInfo>
        <CipherData>
        <CipherValue>
        <Unit UnitProcessor="AuthenticateTGUser">
        <Packet>
        <PacketInfo packetType="data" encrypted="yes">
        <packetId>1</packetId>
        </PacketInfo>
        <Payload>
        <InputString>
        <ClientId></ClientId>
        <SiteId></SiteId>
        </InputString>
        <ResultSet>
        <AuthenticationStatus></AuthenticationStatus>
        <AuthenticationToken></AuthenticationToken>
        </ResultSet>
        </Payload>
        <Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>Call was successfully processed without error.</LongDescription>
        </Status>
        </Packet>
<Code>200</Code>
        <Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>Call was successfully processed without error.</LongDescription>
        </Status>
        </Unit>
        </CipherValue>
        </CipherData>
        </EncryptedData>
        <Status>
```

```
        <Code>200</Code>
        <ShortDescription>Message decrypted and processed without
errors.</ShortDescription>
        <LongDescription/>
<Code>200</Code>
         <Status>
         <Code>200</Code>
         <ShortDescription>Success</ShortDescription>
         <LongDescription>Call was successfully processed without error.</LongDescription>
         </Status>
         </Unit>
         </CipherValue>
         </CipherData>
         </EncryptedData>
         <Status>
         <Code>200</Code>
         <ShortDescription>Message decrypted and processed without
errors.</ShortDescription>
         <LongDescription/>
         </Status>
</Unit>
<Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>All units are processed with success codes</LongDescription>
</Status>
</Envelope>
```

BR ⟶ Client

Candidates authenticated by the client's Talent Gateway can access their job carts or add or remove jobs from their job carts.

GetJobCart Request XML

Web Service Call: GetJobCart

```xml
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
 <Sender>
  <Id>10540</Id>
  <Credential>11721</Credential>
 </Sender>
 <TransactInfo transactType="data" transactId="075cbfe6-b9dc-49f9-8f12-8eae62b62055">
  <TransactId>10/24/2008</TransactId>
  <TimeStamp>12:00:00 AM</TimeStamp>
 </TransactInfo>
 <Unit UnitProcessor="GetJobCart">
  <Packet>
   <PacketInfo packetType="data">
    <packetId>1</packetId>
   </PacketInfo>
   <Payload>
    <InputString>
     <AuthenticationToken>7eb3d08c-dbaa-4fbf-9def-35c383300428</AuthenticationToken>
    </InputString>
   </Payload>
  </Packet>
 </Unit>
</Envelope>
```

BR ← Client

GetJobCart Response XML

```xml
<?xml version='1.0' encoding='utf-8'?>
<Envelope version='01.00'>
 <Sender>
  <Id>10540</Id>
  <Credential>11721</Credential>
 </Sender>
 <Recipient/>
 <TransactInfo transactType='data'>
  <TransactId>6/22/2012</TransactId>
  <TimeStamp>12:00:00 AM</TimeStamp>
 </TransactInfo>
 <Packet>
  <PacketInfo packetType="data">
   <packetId>1</packetId>
  </PacketInfo>
  <Payload>
   <InputString>
    <AuthenticationToken>543e0179-a7de-4350-b58c-91ea00d8d1d0</AuthenticationToken>
   </InputString>
   <ResultSet>
    <jobs>
     <job>
      <jobrequisitionid>165BR</jobrequisitionid>
      <jobstatus>Open</jobstatus>
      <jobtitle>Editor</jobtitle>
<url>https://sjobs.brassring.com/tgwebhost/jobdetails.aspx?authtoken=543e0179-a7de-4350-
b58c-91ea00d8d1d0&amp;partnerid=11721&amp;siteid=523&amp;jobid=56543</url>
      <dateadded>11/18/2010</dateadded>
     </job>
    </jobs>
   </ResultSet>
  </Payload>
  <Status>
   <Code>200</Code>
   <ShortDescription>Success</ShortDescription>
```

BR → Client

```
    <LongDescription>Call was successfully processed without error.</LongDescription>
   </Status>
  </Packet>
  <Status>
   <Code>200</Code>
   <ShortDescription>Success</ShortDescription>
   <LongDescription>Call was successfully processed without error.</LongDescription>
  </Status>
</Envelope>
```

BR ➔ Client

GetJobCart Request Action Add/Remove XML

Web Service Call: GetJobCart Add/Remove Jobs

This example demonstates the web service call adding a job to a job cart. When adding or removing jobs from a job cart, each Action must be performed separately. For example, in one web service call, you can add jobs and then in a separate web service call you can remove jobs. The two actions cannot be performed simultaneously.

This web service call requires the following:

- Candidate Authentication Token
- Action (Add or Remove)
- Req ID (Auto Req ID or Optional Req ID)
- Site ID

Web Service Call: GetJobCart Add Jobs

```
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
  <Sender>
  <Id>10540</Id>
  <Credential>11721</Credential>
  </Sender>
  <TransactInfo transactType="data" transactId="075cbfe6-b9dc-49f9-8f12-8eae62b62055">
  <TransactId>10/24/2008</TransactId>
  <TimeStamp>12:00:00 AM</TimeStamp>
  </TransactInfo>
  <Unit UnitProcessor="JobCart">
   <Packet>
   <PacketInfo packetType="data">
   <packetId>1</packetId>
   </PacketInfo>
   <Payload>
   <InputString>
     <AuthenticationToken>7eb3d08c-dbaa-4fbf-9def-35c383300428</AuthenticationToken>
     <Action>Add</Action>
      <Reqs>
      <Req>
       <ReqId>38BR</ReqId>
      </Req>
       <Req>
       <ReqId>37BR</ReqId>
       <SiteId></SiteId>
       </Req>
       </Reqs>
    </InputString>
    </Payload>
    </Packet>
    </Unit>
</Envelope>
```

BR ⬅ Client

GetJobCart Request Action Add Response XML

```
<?xml version='1.0' encoding='utf-8'?>
<Envelope version='01.00'>
 <Sender>
 <Id>10540</Id>
 <Credential>11721</Credential>
 </Sender>
 <Recipient/>
 <TransactInfo transactType='data'>
 <TransactId>6/22/2012</TransactId>
 <TimeStamp>12:00:00 AM</TimeStamp>
 </TransactInfo>
 <Packet>
 <PacketInfo packetType="data">
 <packetId>1</packetId>
 </PacketInfo>
 <Payload>
   <InputString>
    <AuthenticationToken>543e0179-a7de-4350-b58c-91ea00d8d1d0</AuthenticationToken>
    <Action>Add</Action>
    <Reqs>
    <Req>
       <ReqId>37BR</ReqId>
    </Req>
    </Reqs>
   </InputString>
   <ResultSet>
    <Reqs>
    <Req>
       <ReqId>37BR</ReqId>
       <SiteID>5401</SiteID>
       <Status>Job has been saved to job cart.</Status>
    </Req>
    </Reqs>
   </ResultSet>
 </Payload>
 <Status>
   <Code>200</Code>
   <ShortDescription>Success</ShortDescription>
   <LongDescription>Call was successfully processed without error.</LongDescription>
 </Status>
 </Packet>
 <Status>
   <Code>200</Code>
   <ShortDescription>Success</ShortDescription>
   <LongDescription>Call was successfully processed without error.</LongDescription>
 </Status>
</Envelope>
```

BR ⟶ Client

To view the XML Element Table for this API, click here.

To view error codes for this API, click here.

Get Job Status

Enables clients to display a candidate's job status on a candidate portal page. Clients can configure job status information to include job titles, requisition IDs, HR statuses and hyperlinks to job details pages.

For more information concerning configuration, please contact your BrassRing Support Team member.

Input String

Each API has a required XML input string that clients must use when requesting responses from the BrassRing API.

```
Uses input string:
 <InputString>
 <AuthenticationToken>fa7309b1-46f0-4176-891d-4d916f287d98</AuthenticationToken>
</InputString>
```

BR ← Client

Jobs Status Request XML

```
<ResultSet>
<Jobs>
<Job>
<Jobsubmissiondate></Jobsubmissiondate>
 <JobrequisitionID></JobrequisitionID>
<Jobtitle> </Jobtitle>
 <jobstatus> </jobstatus>
 <Jobstatusdate></Jobstatusdate>
<url></url>
<HRstatus> </HRstatus>
<HRstatusdate></HRstatusdate>
</Job>
</Jobs>
</ResultSet>
```

BR ← Client

Jobs Status Response XML

```
<?xml version="1.0" encoding="utf-8"?>
<!-- edited with XML Spy v3.5 NT (http://www.xmlspy.com) by  () -->
<Envelope version="01.00">
 <Sender>
        <Id>QE</Id>
        <Credential>AXc9N72lF+3UhS8HUZq+FA==</Credential>
</Sender>
<Recipient/>
<TransactInfo transactType="response" transactId="3c54ee09-e0ee-4a6a-8227-1a3cc6522c4a">
        <TransactId>4/26/2011</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="Decryptor">
        <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element" encoding="utf-8"
xmlns="http://www.w3.org/2001/04/xmlenc#">
        <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:KeyName>Integration</ds:KeyName>
        </ds:KeyInfo>
        <CipherData>
        <CipherValue>
        <Unit UnitProcessor="GetStatusCheck">
        <Packet>
        <PacketInfo packetType="data" encrypted="yes">
        <packetId>1</packetId>
        </PacketInfo>
        <Payload>
        <InputString>
        <AuthenticationToken>fa7309b1-46f0-4176-891d-4d916f287d98</AuthenticationToken>
        </InputString>
        <ResultSet>
        <jobs>
        <job>
        <jobsubmissiondate>11/11/2010</jobsubmissiondate>
        <jobrequisitionid>164BR</jobrequisitionid>
        <jobtitle>System Developer</jobtitle>
```

BR → Client

```
            <jobstatus>Open</jobstatus>
            <url>https://sqa-tgweb-
01.brassring.com/1033/asp/tg/redirectcandidate.asp?q=%5eudzQFPVk94hM37MiZWaGjTn1SBeuH
iNco%2fAyx63VjjIb6TNzQe%2fAd517PnJSqyWxMyVy%2befTmM2x%0d%0aZuiU4cWzmgxB%2bU
HoJh9GWOWbO1rD0RaKcxxGTj9BBktYK2u2rUCxXCvvOTdi9oG4j%2bkCbUhBZjPA%0d%0aqbOf
K3X79KwylBbfhWNT0k4668eJ%2bmJWZR1TcT4TFrkoDLC2G4P6F8yKM3hdFqv9wZueKe0J0NS
m%0d%0aRwhv7uEFalYeaFTfIXzml%2bP5LCRvubUw18RymiA%3d</url>
            <jobstatusdate>03/08/2011</jobstatusdate>
            <hrstatus>Phone Screen</hrstatus>
            <hrstatusdate>11/11/2010</hrstatusdate>
            </job>
            </jobs>
            </ResultSet>
            </Payload>
            <Status>
            <Code>200</Code>
            <ShortDescription>Success</ShortDescription>
            <LongDescription>Call was successfully processed without error.</LongDescription>
            </Status>
            </Packet>
            <Status>
            <Code>200</Code>
            <ShortDescription>Success</ShortDescription>
            <LongDescription>Call was successfully processed without error.</LongDescription>
            </Status>
            </Unit>
            </CipherValue>
            </CipherData>
            </EncryptedData>
<Status>
            <Code>200</Code>
            <ShortDescription>Message decrypted and processed without errors.</ShortDescription>
            <LongDescription/>
            </Status>
</Unit>
<Status>
            <Code>200</Code>
            <ShortDescription>Success</ShortDescription>
            <LongDescription>All units are processed with success codes</LongDescription>
</Status>
</Envelope>
```

To view the XML Element Table for this API, click here.

To view error codes for this API, click here.

New User

This API feature enhancement allows clients to create new Talent Gateway (TG) users (candidates) with additional profile information. Previously, when candidates established new TG accounts, the information stored in their profiles was limited to username, password, security questions and answers.

Although not included in the XML examples, new profiles fields include the following:

- Language code.
- First, Middle, and Last Name fields (with pronunciation key)
- Full addresses (2 line addresses, city, state, zip, postal code, country)
- Home, Work, Other, and Fax phone fields
- Resume name and text
- Full educational information
- Work Experience
- Optional node to create candidate's Talent Record
- Optional node to allow reset of password for authenticated candidates

- Optional node to pass the resume key

Example of these additional profile fields is included in the Update Existing Users XML.

This API creates and authenticates new users and returns the authentication token to the client. Clients can later use this authentication token to authenticate this candidate and update that candidate's profile and Talent Record. See Update Existing User.

For more information concerning configuration, please contact your Support Team member.

Input String

Each API has a required XML input string that clients must use when requesting responses from the BrassRing API. This input string in this instance identifies the Client ID, Site ID, and username, password, and security question and answer for each candidate.

```
<InputString>
<ClientId></ClientId>
<SiteId></SiteId>
<Username></Username>
<Password></Password>
<SecurityQuestion></SecurityQuestion>
<SecurityQuestionAnswer></SecurityQuestionAnswer>
</InputString>
```

BR ← Client

Create User Request XML

Web Service Call: Create User

```
<?xml version="1.0" encoding="utf-8" ?>
- <Envelope version="01.00">
- <Sender>
  <Id>eurweiur</Id>
- <!--  you need to either use employeeid of the user who initiate this request. It is required, so if user
  doesn't have it - you may use some default emploeeid you need to configure per client(it has to be
  valid on our side)
  -->
  <Credential>6788</Credential>
- <!-- clientid required
  --> </Sender>
- <TransactInfo transactType="data" transactId="075cbfe6-b9dc-49f9-8f12-8eae62b62055">
  <TransactId>10/24/2008</TransactId>
  <TimeStamp>12:00:00 AM</TimeStamp>
  </TransactInfo>
- <Unit UnitProcessor="CreateUser">
- <Packet>
- <PacketInfo packetType="data">
  <packetId>1</packetId>
  </PacketInfo>
- <Payload>
- <InputString>
  <ClientId>6788</ClientId>
  <SiteId>541</SiteId>
  <Username>ushjune00</Username>
  <Password>password!</Password>
  <SecurityQuestion>1</SecurityQuestion>
  <SecurityQuestionAnswer>stanes</SecurityQuestionAnswer>
  <resetPassword>Yes</resetPassword>
- <!--  Request to reset password when the authentication method is invoked the next time
  -->
  <resumekey>17727</resumekey>
- <!-- candidate unique id
  -->
  </InputString>
  </Payload>
```

BR ← Client

```
</Packet>
</Unit>
</Envelope>
```

BR ← Client

Business Rules:

For new user creation we will follow the same rules and validation which we are following today when user creates new account in TG as we are using the same functions which TG uses today.

Create User Response to Client

```
<?xml version="1.0" encoding="utf-8" ?>
- <Envelope version="01.00">
- <Sender>
  <Id>!@#$%^*()</Id>
  <Credential>pd3HhL1lfYDp/xqxPPjCCA==</Credential>
  </Sender>
  <Recipient />
- <TransactInfo transactType="response" transactId="a591c2f3-3682-4498-a182-
90b0ea53d439">
  <TransactId>8/31/2012</TransactId>
  <TimeStamp>12:00:00 AM</TimeStamp>
  </TransactInfo>
- <Unit UnitProcessor="Decryptor">
- <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element" encoding="utf-8"
xmlns="http://www.w3.org/2001/04/xmlenc#">
  <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />
- <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:KeyName>Integration</ds:KeyName>
  </ds:KeyInfo>
- <CipherData>
- <CipherValue>
- <Unit UnitProcessor="CreateUser">
- <Packet>
- <PacketInfo packetType="data" encrypted="yes">
  <packetId>1</packetId>
  </PacketInfo>
- <Payload>
- <InputString>
  <ClientId>516</ClientId>
  <SiteId>8723</SiteId>
  <Username>aug31st</Username>
  <Password>password!</Password>
  <SecurityQuestion>1</SecurityQuestion>
  <SecurityQuestionAnswer>stanes</SecurityQuestionAnswer>
  <resetPassword>Yes</resetPassword>
  <resumekey>17727</resumekey>
  </InputString>
- <ResultSet>
  <AuthenticationStatus />
  <AuthenticationToken>49b5ce27-9ca5-4f96-a943-5c67a4bf9f91</AuthenticationToken>
  </ResultSet>
  </Payload>
- <Status>
  <Code>200</Code>
  <ShortDescription>Success</ShortDescription>
  <LongDescription>Call was successfully processed without error.</LongDescription>
  </Status>
  </Packet>
- <Status>
  <Code>200</Code>
  <ShortDescription>Success</ShortDescription>
  <LongDescription>Call was successfully processed without error.</LongDescription>
  </Status>
```

BR → Client

```
  </Unit>
  </CipherValue>
  </CipherData>
  </EncryptedData>
- <Status>
  <Code>200</Code>
  <ShortDescription>Message decrypted and processed without errors.</ShortDescription>
  <LongDescription />
  </Status>
  </Unit>
- <Status>
  <Code>200</Code>
  <ShortDescription>Success</ShortDescription>
  <LongDescription>All units are processed with success codes</LongDescription>
  </Status>
  </Envelope>
```

BR ➡️ Client

To view the XML Element Table for this API, click here.

To view error codes for this API, click here.

Reset Password

Provides a way for users to retrieve their password. **Required only if authenticate user call returns code/message: 204:** *The login was successful, however a password reset is required to request any pages.*

Each API has a required XML input string that clients must use when requesting responses from the BrassRing API.

Input String

```
<InputString>
        <AuthenticationToken>fa7309b1-46f0-4176-891d-4d916f287d98</AuthenticationToken>
        <OldPassword>password!</OldPassword>
        <NewPassword>password@</NewPassword>
</InputString>
```

BR ⬅️ Client

Reset Password Request XML

Web Service Call: ResetTGPassword

```
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
 <Sender>
        <Id>QE</Id>
        <Credential>AXc9N72lF+3UhS8HUZq+FA==</Credential>
</Sender>
<Recipient/>
<TransactInfo transactType="response" transactId="2fbfb764-8dd2-4767-8842-6c07fac315a0">
        <TransactId>4/26/2011</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="Decryptor">
        <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element" encoding="utf-8"
xmlns="http://www.w3.org/2001/04/xmlenc#">
        <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:KeyName>Integration</ds:KeyName>
        </ds:KeyInfo>
        <CipherData>
        <CipherValue>
        <Unit UnitProcessor="ResetTGPassword">
        <Packet>
        <PacketInfo packetType="data" encrypted="yes">
        <packetId>1</packetId>
        </PacketInfo>
```

BR ⬅️ Client

```
        <Payload>
        <InputString>
        <AuthenticationToken>fa7309b1-46f0-4176-891d-
4d916f287d98</AuthenticationToken>
        <OldPassword>password!</OldPassword>
        <NewPassword>password@</NewPassword>
        </InputString>
        <ResultSet/>
        </Payload>
        <Status>
        <Code>200</Code>
<ShortDescription>Success</ShortDescription>
        <LongDescription>Call was successfully processed without error.</LongDescription>
        </Status>
</Packet>
        <Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>Call was successfully processed without error.</LongDescription>
        </Status>
        </Unit>
        </CipherValue>
        </CipherData>
        </EncryptedData>
        <Status>
        <Code>200</Code>
        <ShortDescription>Message decrypted and processed without
errors.</ShortDescription>
        <LongDescription/>
        </Status>
</Unit>
<Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>All units are processed with success codes</LongDescription>
</Status>
</Envelope>
```

To view the XML Element Table for this API, click here.

To view the error codes for this API, click here.

> Use Ctrl + Click to follow hyperlinks. Use ALT + LEFT ARROW to return to point of origin.

Simultaneous Login to Multiple Gateway

Candidates using the Search API may view job results on third party websites that originate on different Talent Gateways. When candidates click to view job details, they may be directed to multiple gateways. Simultaneous login ensures that candidates can be authenticated using the API service in order to view Job Details from any job search results, view Saved Jobs, or view the Job Cart, even if those jobs are listed on different gateways.

For more information on business rules for this API, click here.

Input String
Existing AuthenticateTGUser XML Input (Non-SSO) (incudes new node, Additional Sites)


```
<?xml version="1.0" encoding="utf-8"?>
<Envelope version="01.00">
<Sender>
        <Id>10540</Id>
        <Credential>11721</Credential>
</Sender>
```

```
<TransactInfo transactType="data" transactId="075cbfe6-b9dc-49f9-8f12-8eae62b62055">
        <TransactId>10/24/2008</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="AuthenticateTGUser">
        <Packet>
        <PacketInfo packetType="data">
        <packetId>1</packetId>
        </PacketInfo>
        <Payload>
        <InputString>
        <ClientId></ClientId>
        <SiteId></SiteId>
        <AdditionalSites>
<SiteID/>
<SiteID>
</AdditionalSites>
<Username></Username>
        <Password></Password>
        </InputString>
        </Payload>
        </Packet>
  </Unit>
</Envelope>
```

BR ← Client

Create New User XML (incudes new node, Additional Sites)

```
<InputString>
 <ClientId></ClientId>
 <SiteId></SiteId>
 <AdditionalSites>
 <SiteID/>
 <SiteID>
 </AdditionalSites>
 <Username></Username>
 <Password></Password>
 <SecurityQuestion></SecurityQuestion>
 <SecurityQuestionAnswer></SecurityQuestionAnswer>
 </InputString>
```

BR ← Client

Authentication Token in Search API

```
</Sender>
<TransactInfo transactType="data" >
        <TransactId>10/24/2008</TransactId>
        <TimeStamp>12:00:00 AM</TimeStamp>
</TransactInfo>
<Unit UnitProcessor="SearchAPI">
        <Packet>
        <PacketInfo packetType="data">
        <packetId>1</packetId>
        </PacketInfo>
        <Payload>
        <InputString>
        <ClientId>11721</ClientId>
        <SiteId>6129</SiteId>
        <NumberOfJobsPerPage>30<</NumberOfJobsPerPage>
        <PageNumber>1</PageNumber>
        <OutputXMLFormat>0</OutputXMLFormat>
        <!—If you have valid token from login API then use otherwise remove this tag-->
        <AuthenticationToken/>
        <HotJobs/>
        <JobDescription/>
        <!—For all jobs mention ALL otherwise any number of jobs if do not want to use
```

BR ← Client

```
            then so not include this node in input xml □
            <outputFields/>
            <!—Comma separated list of search field questions selected in workbench□
            <ReturnJobsCount/>
        <!—Comma separated list of questions which are configured as Job
        Details fields in Gateway and client want those in search output□
            < ReturnJobDetailQues/>
        <!—If not using the proximity search then remove this ProximitySearch  tag-->
        <ProximitySearch>
        <Distance/>
        <Measurement/>
        <Country/>
        <State/>
        <City/>
        <zipCode/>
        </ProximitySearch>
        <!—If not using the Job Match Criteria then remove this tag-->
        <JobMatchCriteriaText/>
        <SelectedSearchLocaleId/>
        <Questions>
        <Question Sort="No" Sortorder="ASC">
        <Id>1323</Id>
        <Value><![CDATA[TG_SEARCH_ALL]]></Value>
        </Question>
        <Question Sort="No" Sortorder="ASC">
        <Id>1304</Id>
        <!—This is the option code of the question you want to search on-->
        <Value><![CDATA[TG_SEARCH_ALL]]></Value>
        </Question>
        <Question Sort="No" Sortorder="ASC">
        <Id>1292</Id>
        <Value><![CDATA[TG_SEARCH_ALL]]></Value>
        </Question>
        </Questions>
        </InputString>
        </Payload>
        </Packet>
    </Unit>
    </Envelope>
```

BR ← Client

Business Rules

- A new node w/ child nodes will be available in the authentication and new user XML's "<AdditionalSites>" by which clients may indicate additional Talent Gateways which the user should be authenticated into.
- The original "<SiteID>" node will remain unchanged and will determine which site the user should land on, in the event of an AccessPage call. This is not applicable for 'deep links' since these URLs are site specific so the TG is predetermined.
- The new node is not required, the authentication process will continue as it currently works if the new node is not present.
- Regardless of the number of sites included, a single authentication token will always be returned in the results call. BrassRing will track which sites it is applicable for in the session details.
- The multiple site authentications will be applicable for the following workflows: Job Apply via Search API results, Job Cart and Saved Draft.
- APIs for Search and Job Cart both return 'deep links' URLs by which the user may directly access a specific job. These URLs are site specific. In this workflow, the user may be presented with multiple URLs on a single 3rd-party hosted page. The user will have success when clicking on any link for which the site ID was included in the "AdditionalSites" node. There will be no changes made to the Job Cart experience when accessed via "AccessPage", only the GetJobCart API will be modified to support multiple site authentication)
- For Saved Draft (provided via the "AccessPage" functionality), the user is simply redirected to the primary TG page and will be presented with all of their Saved Drafts, regardless of which site the draft is associated with. When selecting Continue, the application will automatically open up to where they left off if it is associated with the current

TG. If it is associated with a different TG, the user will be presented with a popup message with OK and Cancel buttons (much like the language toggle on a GTG): [The current career site will change to match the career site of the selected draft.]. Clicking OK will redirect the user to the appropriate TG as long as it is a site which was included in the "AdditionalSites" node, Cancel will return them to the current page with no action taken. Once redirected to the appropriate gateway, any subsequent actions (such as Edit Profile) will apply to the gateway they have been redirected to, as opposed to returning them to the primary gateway.

Note about existing functionality:

If SSOID is sent then the Username and password sent in input xml are ignored. If using SSO, all gateways must be SSO gateways

- Only sites of a matching language may be sent in a single request, if multiple site IDs are sent which are configured with differing languages, an error would be returned. [The sites requested must all be configured with the same language.] For a multiple language experience, a GTG would need to be used.
- If all gateways requested are SSO gateway, and in input XML SSOID is empty, then an error will be returned. [SSOID is required for authentication into SSO gateways.]
- If any of the gateways in the list requested is a SSO gateway and other site Ids are non SSO, then an error will be returned: [The sites requested must all be configured for either SSO or non-SSO.]
- If any of the gateways in a list is a basic gateway, then an error will be returned: [The sites requested must all be full gateways.]
- If any of the sites in the requested list are inactive then an error will be returned: [The sites requested must all be active gateways.].

To view the XML Element Table for this API, click here.

To view the error codes for this Authenticate User API SSO, click here.

To view the error codes for Simultaneous Login, click here.

Talent Gateway Form Import

The Talent Gateway Form Import API allows candidates to submit and update form data for Talent Gateway forms. Clients can use the authentication token returned when a new user is created or when using the Authenticate User API.

This feature does not support per req forms. Form type IDs and field IDs are sent in the request to identify the candidate. If transmitting a form field that is a multi-select or checkbox, multiple values can be sent in a single node with ~|~ delimiter. For example, - <![CDATA[chk1~|~chk2]]>.

For more information concerning configuration, please contact your BrassRing Support Team member.

> Client can send either one form or multiple forms. Reqs are not supported. If client wants to send single form then only one node of Form will be present.

Talent Gateway Form Request Import XML

Web Service Call: SaveFormResponse

```
<?xml version="1.0" encoding="utf-8" ?>
- <Envelope version="01.00">
- <Sender>
  <Id>10540</Id>
  <Credential>11721</Credential>
  </Sender>
- <TransactInfo transactType="data" transactId="075cbfe6-b9dc-49f9-8f12-8eae62b62055">
  <TransactId>10/24/2008</TransactId>
  <TimeStamp>12:00:00 AM</TimeStamp>
  </TransactInfo>
- <Unit UnitProcessor="SaveFormResponses">
- <Packet>
- <PacketInfo packetType="data">
  <packetId>1</packetId>
  </PacketInfo>
```

BR ← Client

```
- <Payload>
- <InputString>
  <ClientId />
  <SiteId />
  <AuthenticationToken>valid token</AuthenticationToken>
- <Forms>
- <Form formtypeid="200" formname="Disposition Form">
- <FormInput fieldid="3471">
- <![CDATA[ Did not pass Reference Check
  ]]>
  </FormInput>
- <FormInput fieldid="3475">
- <![CDATA[ Offer Details
  ]]>
  </FormInput>
- <FormInput fieldid="3476">
- <![CDATA[ Comments1
  ]]>
  </FormInput>
- <FormInput fieldid="12584">
- <![CDATA[ chk1~|~chk2
  ]]>
  </FormInput>
- <FormInput fieldid="12585">
- <![CDATA[ Radio1
  ]]>
  </FormInput>
- <FormInput fieldid="12586">
- <![CDATA[ m
  ]]>
  </FormInput>
  </Form>
- <Form formtypeid="969" formname="Another-One">
- <FormInput fieldid="12319">
- <![CDATA[ 2006-2-1
  ]]>
  </FormInput>
- <FormInput fieldid="12321">
- <![CDATA[ Mahajan
  ]]>
  </FormInput>
- <FormInput fieldid="12322">
- <![CDATA[ Rupesh
  ]]>
  </FormInput>
- <FormInput fieldid="12326">
- <![CDATA[ Waltham
  ]]>
  </FormInput>
- <FormInput fieldid="12327">
- <![CDATA[ Yes
  ]]>
  </FormInput>
- <FormInput fieldid="12328">
- <![CDATA[ 12
  ]]>
  </FormInput>
- <FormInput fieldid="1056811">
- <![CDATA[ AF
  ]]>
  </FormInput>
- <FormInput fieldid="1056812">
- <![CDATA[ AnotherOne-TextArea1
  ]]>
```

```
  </FormInput>
- <FormInput fieldid="1056813">
- <![CDATA[ Chk1
  ]]>
  </FormInput>
- <FormInput fieldid="1056814">
- <![CDATA[ Radio2
  ]]>
  </FormInput>
- <FormInput fieldid="1056815">
- <![CDATA[ 111-11-1111
  ]]>
  </FormInput>
- <FormInput fieldid="1056816">
- <![CDATA[ rmahajan@kenexa.com
  ]]>
  </FormInput>
  </Form>
  </Forms>
  </InputString>
  </Payload>
  </Packet>
  </Unit>
  </Envelope>
```

BR ← **Client**

Talent Gateway Form Import Response XML

```
  <?xml version="1.0" encoding="utf-8" ?>
- <Envelope version="01.00">
- <Sender>
  <Id>!@#$%^*()</Id>
  <Credential>pd3HhL1lfYDp/xqxPPjCCA==</Credential>
  </Sender>
  <Recipient />
- <TransactInfo transactType="response" transactId="1e3d5353-e474-493b-b10d-c98b15e0f4e8">
  <TransactId>7/18/2012</TransactId>
  <TimeStamp>12:00:00 AM</TimeStamp>
  </TransactInfo>
- <Unit UnitProcessor="Decryptor">
- <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element" encoding="utf-8"
  xmlns="http://www.w3.org/2001/04/xmlenc#">
  <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />
- <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:KeyName>Integration</ds:KeyName>
  </ds:KeyInfo>
- <CipherData>
- <CipherValue>
- <Unit UnitProcessor="SaveFormResponses">
- <Packet>
- <PacketInfo packetType="data" encrypted="yes">
  <packetId>1</packetId>
  </PacketInfo>
- <Payload>
- <InputString>
  <ClientId>516</ClientId>
  <SiteId>8723</SiteId>
  <AuthenticationToken>ea57ea5d-24f1-47c5-881f-aa59098e09b2</AuthenticationToken>
  Forms>
  <Form formtypeid="200" formname="Disposition Form">
        <FormInput fieldid="3471"><![CDATA[Did not pass Reference Check]]></FormInput>
        <FormInput fieldid="3475"><![CDATA[Offer Details]]></FormInput>
        <FormInput fieldid="3476"><![CDATA[Comments1]]></FormInput>
        <FormInput fieldid="12584"><![CDATA[chk1~|~chk2]]></FormInput>
        <FormInput fieldid="12585"><![CDATA[Radio1]]></FormInput>
```

BR → **Client**

```
            <FormInput fieldid="12586"><![CDATA[m]]></FormInput>
</Form>
<Form formtypeid="969" formname="Another-One">
        <FormInput fieldid="12319"><![CDATA[2006-2-1]]></FormInput>
        <FormInput fieldid="12321"><![CDATA[Mahajan]]></FormInput>
        <FormInput fieldid="12322"><![CDATA[Rupesh]]></FormInput>
        <FormInput fieldid="12326"><![CDATA[Waltham]]></FormInput>
        <FormInput fieldid="12327"><![CDATA[Yes]]></FormInput>
        <FormInput fieldid="12328"><![CDATA[12]]></FormInput>
        <FormInput fieldid="1056811"><![CDATA[AF ]]></FormInput>
        <FormInput fieldid="1056812"><![CDATA[AnotherOne-TextArea1]]></FormInput>
        <FormInput fieldid="1056813"><![CDATA[Chk1]]></FormInput>
        <FormInput fieldid="1056814"><![CDATA[Radio2]]></FormInput>
        <FormInput fieldid="1056815"><![CDATA[111-11-1111]]></FormInput>
        <FormInput fieldid="1056816"><![CDATA[rmahajan@kenexa.com]]></FormInput>
</Form>
          </Forms>  </InputString>
        <ResultSet>
         <AuthenticationStatus>0</AuthenticationStatus>
         <AuthenticationToken>3bbaa10e-e596-4f46-b772-624ae49475d6</AuthenticationToken>
         </ResultSet>
         </Payload>
         <Status>
         <Code>200</Code>
         <ShortDescription>Success</ShortDescription>
         <LongDescription>Call was successfully processed without error.</LongDescription>
         </Status>
         </Packet>
         <Status>
         <Code>200</Code>
         <ShortDescription>Success</ShortDescription>
         <LongDescription>Call was successfully processed without error.</LongDescription>
         </Status>
         </Unit>
         </CipherValue>
         </CipherData>
         </EncryptedData>
         <Status>
         <Code>200</Code>
         <ShortDescription>Message decrypted and processed without errors.</ShortDescription>
         <LongDescription/>
         </Status>
</Unit>
<Status>
        <Code>200</Code>
        <ShortDescription>Success</ShortDescription>
        <LongDescription>All units are processed with success codes</LongDescription>
</Status>
</Envelope>
```

To view the XML Element Table for this API, click here.

To view error codes for this API, click here.

Update Existing User

This Update Existing User API allows clients to add or update existing TG user profiles, including the updating of Talent Records. Clients can update form field data in a candidate's Talent Record after that candidate has been authenticated. Clients can use the authentication token returned when a new user is created or use the Authenticate User API.

When the Talent Record node is set to Yes (sendToDST) a talent record is created. The stacking logic configured on that talent gateway applies. When this node is not sent or is blank, it does not create the BrassRing talent record.

> **Note:** When the node is set to YES, it ignores the TG setting for Only create BrassRing record on job apply. Furthermore, this is ignored if the resumekey was provided in the previous CreateUser operation.

When the UpdateProfile operation is called and a resume key was sent in a previous CreateUser operation, then that resume key value will be used to tie this TG profile to the existing BrassRing resume key. Profile fields that are required are based on the stacking logic configured for that talent gateway.

Additional profile element fields include:

- Language code.
- First, Middle, and Last Name fields (with pronunciation key)
- Full addresses (2 line addresses, city, state/region/province, zip, postal code, country)
- Home, Work, Other, and Fax phone fields
- Resume name and text
- Full educational information (School, Degree type, Major,End Date, GPA)
- Work Experience
- Optional node to create candidate's Talent Record
- Optional node to allow reset of password for authenticated candidates

When the UpdateProfile operation is called and a resumekey was sent in a previous CreateUser operation, then that resumekey value will be used to tie this TG profile to the existing BrassRing resumekey. Profile fields that are required are based on the stacking logic configured for that talent gateway.

For more information concerning configuration, please contact your BrassRing Support Team member.

Update User Input

Web Service Call: UpdateProfile

```
 <?xml version="1.0" encoding="utf-8" ?>
- <Envelope version="01.00">
- <Sender>
  <Id>10540</Id>
- <!-- you need to either use employeeid of the user who initiate this request. It is required, so if user
  doesn't have it - you may use some default emploeeid you need to configure per client(it has to be
  valid on our side)
  -->
  <Credential>11721</Credential>
- <!-- clientid
  -->
  </Sender>
- <TransactInfo transactType="data" transactId="075cbfe6-b9dc-49f9-8f12-8eae62b62055">
  <TransactId>10/24/2008</TransactId>
  <TimeStamp>12:00:00 AM</TimeStamp>
  </TransactInfo>
- <Unit UnitProcessor="UpdateProfile">
- <Packet>
- <PacketInfo packetType="data">
  <packetId>1</packetId>
  </PacketInfo>
- <Payload>
- <InputString>
  <ClientId />
  <SiteId />
  <AuthenticationToken>valid token</AuthenticationToken>
- <!-- Authntication token form the create user /authentication method
  -->
- <Language id="1" localeid="1033">
- <ProfileInfo>
- <firstname>
- <![CDATA[ FirstNameRs
  ]]>
```


BR ← Client

```
  </firstname>
- <middlename>
- <![CDATA[ MiddleNameRS
  ]]>
  </middlename>
- <lastname>
- <![CDATA[ LastNameRS
  ]]>
  </lastname>
- <address1>
- <![CDATA[ Address 1 Line
  ]]>
  </address1>
- <address2>
- <![CDATA[ Address 2 Line
  ]]>
  </address2>
- <country>
- <![CDATA[ Country iso value from CoutnryStateList.xml file
  ]]>
  </country>
- <city>
- <![CDATA[ Waltham
  ]]>
  </city>
- <state>
- <![CDATA[ State Id from CoutnryStateList.xml file for that country
  ]]>
  </state>
- <zip>
- <![CDATA[ 02452
  ]]>
  </zip>
- <homephone>
- <![CDATA[ 8572047837
  ]]>
  </homephone>
- <workphone>
- <![CDATA[ 7815305159
  ]]>
  </workphone>
- <mobilephone>
- <![CDATA[ 8573347722
  ]]>
  </mobilephone>
- <emailaddress>
- <![CDATA[ someemail@kenexa.com
  ]]>
  </emailaddress>
- <fax>
- <![CDATA[ 7815305000
  ]]>
  </fax>
- <webaddress>
- <![CDATA[ http://google.com
  ]]>
  </webaddress>
- <EducationInfo>
- <eduDegree>
- <schoolName>
- <![CDATA[ BE
  ]]>
  </schoolName>
- <degree>
```

```
- <![CDATA[ BACHELORS
  ]]>
  </degree>
- <major>
- <![CDATA[ Electonic
  ]]>
  </major>
- <GPA>
- <![CDATA[ 59
  ]]>
  </GPA>
- <schoolEndDate>
- <![CDATA[ 1995
  ]]>
  </schoolEndDate>
  <MostRecent />
  </eduDegree>
- <eduDegree>
- <schoolName>
- <![CDATA[ MBA
  ]]>
  </schoolName>
- <degree>
- <![CDATA[ MASTERS
  ]]>
  </degree>
- <major>
- <![CDATA[ System Mgmt
  ]]>
  </major>
- <GPA>
- <![CDATA[ 66
  ]]>
  </GPA>
- <schoolEndDate>
- <![CDATA[ 1998
  ]]>
  </schoolEndDate>
  <MostRecent>1</MostRecent>
  </eduDegree>
  </EducationInfo>
- <WorkExpInfo>
- <workExp>
- <title>
- <![CDATA[ Software Engg
  ]]>
  </title>
- <company>
- <![CDATA[ Garnet software
  ]]>
  </company>
- <Responsibilities>
- <![CDATA[ Coding
  ]]>
  </Responsibilities>
- <Skills>
- <![CDATA[ VB/ASP
  ]]>
  </Skills>
- <startDate>
- <![CDATA[ 1998
  ]]>
  </startDate>
- <endDate>
```

BR ← Client

```
- <![CDATA[ 2001
 ]]>
 </endDate>
 <MostRecent />
 </workExp>
- <workExp>
- <title>
- <![CDATA[ Senior Soft Engg
 ]]>
 </title>
- <company>
- <![CDATA[ Yojana Systems
 ]]>
 </company>
- <Responsibilities>
- <![CDATA[ Coding and Development
 ]]>
 </Responsibilities>
- <Skills>
- <![CDATA[ VB/ASp/MSMQ
 ]]>
 </Skills>
- <startDate>
- <![CDATA[ 2001
 ]]>
 </startDate>
- <endDate>
- <![CDATA[ 2003
 ]]>
 </endDate>
 <MostRecent />
 </workExp>
- <workExp>
- <title>
- <![CDATA[ Project leader
 ]]>
 </title>
- <company>
- <![CDATA[ IBM
 ]]>
 </company>
- <Responsibilities>
- <![CDATA[ Lead
 ]]>
 </Responsibilities>
- <Skills>
- <![CDATA[ Oracle/microsoft
 ]]>
 </Skills>
- <startDate>
- <![CDATA[ 2003
 ]]>
 </startDate>
- <endDate>
- <![CDATA[ 2006
 ]]>
 </endDate>
 <MostRecent>1</MostRecent>
 </workExp>
 </WorkExpInfo>
 </ProfileInfo>
 </Language>
 <sendToDST>YES/NO</sendToDST>
- <!-- will create a BrassRing Talent Record if yes
```

BR ← Client

```
-->
</InputString>
</Payload>
</Packet>
</Unit>
</Envelope>
```

BR ← **Client**

To view the XML Element Table for this API, click here.

> **Note:** When using the UpdateProfile operation, it is advisable to provide **all** profile fields.

Update User Response to Client

Web Service Call: UpdateProfile

The Update User API responds to client noting all form fields values updated and notifying client if there were any errors.

```
<?xml version="1.0" encoding="utf-8" ?>
- <Envelope version="01.00">
- <Sender>
  <Id>!@#$%^*()</Id>
  <Credential>pd3HhL1lfYDp/xqxPPjCCA==</Credential>
  </Sender>
  <Recipient />
- <TransactInfo transactType="response" transactId="2f40c370-531b-4133-aaa6-e7fd9fc135c6">
  <TransactId>8/1/2012</TransactId>
  <TimeStamp>12:00:00 AM</TimeStamp>
  </TransactInfo>
- <Unit UnitProcessor="Decryptor">
- <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element" encoding="utf-8"
  xmlns="http://www.w3.org/2001/04/xmlenc#">
  <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc" />
- <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <ds:KeyName>Integration</ds:KeyName>
  </ds:KeyInfo>
- <CipherData>
- <CipherValue>
- <Unit UnitProcessor="UpdateProfile">
- <Packet>
- <PacketInfo packetType="data" encrypted="yes">
  <packetId>1</packetId>
  </PacketInfo>
- <Payload>
- <InputString>
  <ClientId>516</ClientId>
  <SiteId>8723</SiteId>
  <AuthenticationToken>7eb45ac9-fbbf-4a1f-8a0b-0df90225b23f</AuthenticationToken>
- <ProfileInfo>
  <firstname>FirstNameush</firstname>
  <middlename>MiddleNameu</middlename>
  <lastname>LastNameush</lastname>
  <address1>Address 1 Line</address1>
  <address2>Address 2 Line</address2>
  <country>US</country>
  <city>Waltham</city>
  <state>MA</state>
  <zip>02452</zip>
  <homephone>8572047967</homephone>
  <workphone>7815305869</workphone>
  <mobilephone>8573347962</mobilephone>
  <emailaddress>uanandakumar@kenexa.com</emailaddress>
  <fax>7815305220</fax>
```

BR → **Client**

```
  <webaddress>http://google.com</webaddress>
- <EducationInfo>
- <eduDegree>
  <schoolName>BE</schoolName>
  <degree>BACHELORS</degree>
  <major>Electonic</major>
  <GPA>59</GPA>
  <schoolEndDate>1995</schoolEndDate>
  <MostRecent />
  </eduDegree>
- <eduDegree>
  <schoolName>MBA</schoolName>
  <degree>MASTERS</degree>
  <major>System Mgmt</major>
  <GPA>66</GPA>
  <schoolEndDate>1998</schoolEndDate>
  <MostRecent>1</MostRecent>
  </eduDegree>
  </EducationInfo>
- <WorkExpInfo>
- <workExp>
  <title>Software Engg</title>
  <company>Garnet software</company>
  <Responsibilities>Coding</Responsibilities>
  <Skills>VB/ASP</Skills>
  <startDate>1998</startDate>
  <endDate>2001</endDate>
  <MostRecent />
  </workExp>
- <workExp>
  <title>Senior Soft Engg</title>
  <company>Yojana Systems</company>
  <Responsibilities>Coding and Development</Responsibilities>
  <Skills>VB/ASp/MSMQ</Skills>
  <startDate>2001</startDate>
  <endDate>2003</endDate>
  <MostRecent />
  </workExp>
- <workExp>
  <title>Project leader</title>
  <company>IBM</company>
  <Responsibilities>Lead</Responsibilities>
  <Skills>Oracle/microsoft</Skills>
  <startDate>2003</startDate>
  <endDate>2006</endDate>
  <MostRecent>1</MostRecent>
  </workExp>
  </WorkExpInfo>
  </ProfileInfo>
  <sendToDST>YES</sendToDST>
  </InputString>
  <ResultSet />
  </Payload>
- <Status>
  <Code>200</Code>
  <ShortDescription>Success</ShortDescription>
  <LongDescription>Call was successfully processed without error.</LongDescription>
  </Status>
  </Packet>
- <Status>
  <Code>200</Code>
  <ShortDescription>Success</ShortDescription>
  <LongDescription>Call was successfully processed without error.</LongDescription>
  </Status>
```

BR ⟶ Client

```
    </Unit>
    </CipherValue>
    </CipherData>
    </EncryptedData>
-  <Status>
    <Code>200</Code>
    <ShortDescription>Message decrypted and processed without errors.</ShortDescription>
    <LongDescription />
    </Status>
    </Unit>
-  <Status>
    <Code>200</Code>
    <ShortDescription>Success</ShortDescription>
    <LongDescription>All units are processed with success codes</LongDescription>
    </Status>
    </Envelope>
```

BR ➜ Client

To view the XML Element Table for this API, click here.

To view the Error Codes for this API, click here.

> When using the UpdateProfile operation, it is advisable to provide **all** profile fields.

XML Element Table - Candidate APIs

The following is a list of the XML elements used in Candidate APIs.

Candidate API	Request/ Response	XML Element Tag	Functionality Details
		?xml version="1.0"/encoding+"utf-8"?	Indicates version of xml and encoding type For more information see: http://www.w3.org/standards/xml/
		Sender/ID	Client's Site ID Client's ID identifies client. . To view: Tools > Talent Gateway > TG Admin > Special Configuration...
		Credential	Client's credential ID
		TransactionInfo/transact type	Contains TransactID and Timestamp information
		TransactID/Timestamp	Date and time of transaction
		Unit Processor/Encrypted Data Type	Processor being requested. If request/response is using encryption, encryption data/information is contained within this element.
		PacketInfo	Indicates type of information
		packetID	
Access Page			
	Request		
		ClientID	ClientID identifies the client. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		SiteId	The client's site ID. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		AuthenticationToken	Cilent use their Authentication 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.
		DestinationPage	Numerical value indicates landing page for access: HomePage = 0 SearchOpenings = 1 EditProfile = 2 StatusCheck = 3 esumeManager = 4 AgentManager = 5 JobCart = 6 SavedDrafts = 7 Logout = 8 SubmitNow = 9
	Response		
		Result Set/Redirect URL	This is the URL for landing page for candidate.
		Code	200
		Short Description	Successful/Failed – used for most API responses

Candidate API	Request/ Response	XML Element Tag	Functionality Details
Access Page (cont.)		Long Description	Notifies clients if there were errors in API processing. Used for most API responses.
Assessments Pending			
	Request		
		ClientID	ClientID identifies the client. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		SiteId	The client's site ID. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		AuthenticationToken	Clients use their Authentication Token to validate the candidate when using the APIs. It is generally returned to Client after a successful login, and is used when Clients make additional API requests.
		Redirect URL	Clients can determine web page candidate views when accessing pending assessments.
	Response		
		ClientID	ClientID identifies the client. To view: Tools > Talent Gateway > TG Admin > Special Configuration
		SiteId	The client's site ID. To view: Tools > Talent Gateway > TG Admin > Special Configuration
		AuthenticationToken	Authentication token is returned in a CDATA format. Clients use their Authentication Token to validate the candidate when using the APIs. It is generally returned to Client after a successful login, and is used when Clients make additional API requests.
		Redirect URL	Clients can determine web page candidate views when accessing pending assessments. -
		Job title/Req ID	Job title associated with each Req ID
		Assessment Name	Name of Assessment associated with the specific Req ID
		Assessment URL	The Assessment link displays if candidate needs to
		Assessment Status	Not Started or In Progress.
Authenticate User			
	Request		
		ClientID	ClientID identifies the client. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		SiteId	The client's site ID. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		Username	Client's Username

Candidate API	Request/ Response	XML Element Tag	Functionality Details
Authenticate User (cont.)		Password	Client's Password
		SSOID	For SSO TGs: Client's SSO ID For SSO support, BrassRing uses existing Authenticate User API with addition of new node called SSOSessionID, contains valid SSO Session ID.
	Response		
		ClientID	ClientID identifies the client. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		SiteId	The client's site ID. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		SSOID	For SSO TGs: Client's SSO ID For SSO support, BrassRing uses existing Authenticate User API with addition of new node called SSOSessionID, contains valid SSO Session ID.
		Authentication Status/Authentication Token	Successful requests return authentication status and token.
		Code	If the request is successful, the code is 200.
Candidate Forms			
	Request		
		ClientID	ClientID identifies the client. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		SiteId	The client's site ID. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		Intervals	This value represents the date range of eLinked forms to display. If no date field is configured, all forms are returned in the response.
		FormTypeID	This value represents the form types. Clients can list multiple form types.Form types must be listed separated by commas. To view FormTypeIDs: WB > View Client settings.
	Response		
		Mode	Mode of candidate form. Modes are Add, View or Edit. .
		ResumeKey	Candidate's Numerical Resume Key ID
		Form name	Display of form name.
		Formurl	Location URL where the forms are displayed.
		Candidate Name	Candidate's name
		RequisitionID	Related Requisition ID

Candidate API	Request/ Response	XML Element Tag	Functionality Details
Candidate Forms (cont.)		Job Title/Req Status	Job Title and Req Status
		Sent Date/Added on/Edited on	Dates Form was sent, added to candidate's record, and when the form was last edited.
		Draft created on/Draft Edited on	Dates the form was saved and/or edited as a draft. Draft forms get *converted* to Forms when the candidate clicks the 'Save' button vs. the Save as draft button.
Change Security Question/Answer			
	Request		
		ClientID	ClientID identifies the client. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		SiteId	The client's site ID. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		AuthenticationToken	Cilent use their Authentication 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.
		SecurityQID	New Security Question IDs.
		Answer	New answer value.
	Response		
		SecurityQID	New Security Question IDs.
		Answer	New answer value.
Communications History			
	Request		
		Page Number/Return Message Count	Client indicates number of pages and message count to display for each candidate. If no date is indicated, the API returns the entire communications history.
		Date	Client can choose to enter a date range for the communication history.
	Response		
		Correspondence ID	Value of Client's ID.
		Job ID	Job ID related to job correspondence.
		Job Title	Job Title related to job correspondence..
		Sender	Person who sent correspondence.
		Template Name/Type	Type of correspondence sent.

Candidate API	Request/ Response	XML Element Tag	Functionality Details
Communications History (cont.)		Subject	Subject of correspondence.
		Message Link	The message hyperlink displays. Candidate can access and view original correspondence.
Event Manager			
	Request		
		Authentication Token	Client transmits their unique authentication token.
		ClientID	ClientID identifies the client. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		Criteria/Event Type	Client specifies event type.
		Location/Value	Client specifies event locations to display
		City/Value	Client specifies event city locations to display
		Region/Value	Client specifies event region to display
		Postal Code/Value	Client specifies event postal codes to display
		Country/Value	Client specifies event countries to display
		Start Date/End Date	Client input date range. Format: yyyy/mm/dd hh:mm If not date is set, the API returns events for the next 90 days.
	Response		
		Result Set includes:	
		Event ID/Event Template/Event Type	Event ID and type display. Events can be series or one instance.
		Event Date	Date of event.
		Event Time	Time of event.
		Event Location	Event location.
		Event City/Region	Event City and region.
		Postal Code	Postal code of Event.
		Event Description	Description of Event.

API Reference Guide

Candidate API	Request/ Response	XML Element Tag	Functionality Details
Documents/Docu ment Packet			
	Request		
		Client ID	Client ID?
		Credential	Client credential?
		Unit Processor	"GetCandidatePortalInfo" – this unit processor extracts documents and document packets from the candidate's portal.
		AuthenticationToken	Client's API sends candidate's encrypted authentication token to BrassRing.
		ClientID	ClientID identifies the client. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		SiteId	The client's site ID. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		Attachments	
		Document	The document element contains the following elements: - Date Posted - Name - Type - JobID - Job Title - Sent By (Last name,First) - Removal Date - URL (Link to Talent Gateway Page
		Document Packet	The document element contains the following elements: - Date Posted - Name - Type - JobID - Job Title - Sent By (Last name,First) - Removal Date - URL (Link to Talent Gateway Page
	Response		
		Sender ID/Credential	Sender and credential validates candidates identify.
		AuthenticationToken	Encrypted authentication token returned for candidate.

Candidate API	Request/ Response	XML Element Tag	Functionality Details
Documents/ Document Packet (cont.)		Result Set/Document	The document element contains the following elements: - Date Posted - ReqID - Job Title - Document Name - Document URL (Link to Talent Gateway Page) -Type (Candidate Portal or Document Packet) -Sent By (Last name,First) - Status
		Result Set/Document Packet	The document element contains the following elements: - Date Posted - ReqID - Job Title - Document Packet Name - Type (Candidate Portal or Document Packet) - Sent By (Last name,First) - Status - Date Posted - Removal Date - Document Packet URL (Link to Talent Gateway Page)
Forgot Password (Candidate)			
	Request		
		Authentication Token or User name	Candidate's authentication token or username.
		securityQuestion	Candidate's selection of security question.
		ID	Number of security question chosen. Depending on configuration there may be more than one question.
		Answer	Candidate's answer to their security question.
	Response		
		securityQuestions/securityQuestion	Candidate's selection of security question.
		ID	Number of security question chosen. Depending on configuration there may be more than one question.
		Answer	Candidate's answer to their security question.
		New Password	Candidate inputs new password.
Get Resume			
	Request		
		Authentication Token	Client transmits authentication token.
		Resume Set	Contain elements related to resume.
		Resume Language	Language of resume .
		ResumeID	Resume ID number.
		ResumeDisplayName	Name candidate chose for Resume.

Candidate API	Request/ Response	XML Element Tag	Functionality Details
Get Resume (cont.)		Resume Link	URL where resume is located.
		Resume Revised	Date of last resume revision (MM/DD/YYYY)
		Job	Contains all sub-elements related to job application.
		JobID	Job ID.
		JobTitle	Job Title
		JobLink	JobLink
		CoverLetterID	Cover Letter ID.
		CoverLetterName	Name candidate chose for cover letter.
		Date Submitted	Date resume was submitted.
	Response		
		ResultSet	
		resumename	Resume name
		resumeurl	URL of resume
		resumerevised	Revision date
		defaultresume	true (meaning there is a resume)
		jobs/job	
		jobrequisitionid	Requisition ID
		jobstatus	Staus of req (open, closed, etc.)
		jobtitle	Job title
		url	URL where job details reside.
		datesubmitted	Submission date
Get Job Cart			
	Request		
		Authentication Token	Client transmits authentication token.
		Results Set includes:	
		jobs/job	Element containing variable elements.
		jobrequisitionid	Requisition ID.
		jobstatus	Status of job
		jobtitle	Job Title
		url	URL where job is located.
		dateadded	Date job was added to job cart.
	Response		
		ResultSet includes:	
		jobrequisitionid	Requisition ID.
		jobstatus	Status of job

Candidate API	Request/ Response	XML Element Tag	Functionality Details
Get Job Cart Actions – Add or Remove			
	Request		
		Authentication Token	Client transmits authentication token.
		Action	Add or Remove
		Req ID	
		Site ID	Site ID should be provided only when Job has to be Added/Removed against non-default member TG (GTG or additional TG)
	Response		
		Req ID	Requisition ID
		Site ID	Site ID
		Status of job action	Potential responses: - Job has been saved to job cart. - Job already exists in job cart. - This job has recently been deactivated by this organization
		Status	Successful or not successful.
Get Job Status			
	Request	ResultSet includes:	
		Jobsubmissiondate	Date of job submission
		JobrequisitionID	Requisition ID
		Jobtitle	Job title
		Jobstatus	Status of Job (open, closed, etc.)
		Jobstatusupdate	Empty if there is no update
		url	URL of job posting
		HRstatus	HR status of logged in candidate
		HRstatusdate	Date of last HR status
	Response		
		Jobsubmissiondate	Date of job submission
		JobrequisitionID	Requisition ID
		Jobtitle	Job title
		Jobstatus	Status of Job (open, closed, etc.)
		Jobstatusupdate	Empty if there is no update
		url	URL of job posting
		HRstatus	HR status of logged in candidate
		HRstatusdate	Date of last HR status

Candidate API	Request/ Response	XML Element Tag	Functionality Details
New User			
	Request		
		ClientId	ClientID identifies the client. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		SiteId	The client's site ID. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		Username	Joe
		Password	password!
		SecurityQuestion	1
		SecurityQuestionAnswer	St. Mary's
		reset password	(Optional) When used, prompts a password reset for candidate's account.
		ResumeKey	(Optional) When included, the key is stored for the session. When update profile operation runs, this key is passed to it, linking the TG to the talent record.
	Response		
		ClientId	342
		SiteId	2345
		Username	Sam
		Password	Martha22
		SecurityQuestion	1 (Represents the number of the security question.)
		Additional Fields NOT SHOWN	These additional fields are referenced in Update Existing User XML Example.
		FormTypeID	This value represents the form types. Clients can list multiple form types.Form types must be listed separated by commas. To view FormTypeIDs: WB > View Client settings.
		formname	Name of Form, for example, disposition form. To view this information: To view FormTypeIDs: WB > View Client settings.
		formInput fieldid	Numerical ID
		<![CDATA[ ]]>	Input text to update TG form data. Entered as CDATA – information entered as CDATA – following guidelines here.
		Password	User's password
	Response		
		ClientId	516
		SiteId	8723
		ResultSet ORJobDetailLink	
		FormTypeID	This value represents the form types. Clients can list multiple form types. Form types must be listed separated by commas. To view FormTypeIDs: WB > View Client settings.
		formname	
		formInput fieldid	
		<![CDATA[ ]]>	Results return inputted text in this field.

Candidate API	Request/ Response	XML Element Tag	Functionality Details
Update Existing User (Profile)			
	Request		
		ClientId	ClientID identifies the client. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		SiteId	The client's site ID. To view: Tools > Talent Gateway > Edit Icon > Talent Gateway Details
		AuthenticationToken	User transmits authentication token.
		Profile Info includes:	CDATA – information entered as CDATA – following guidelines here.
		firstname	First name
		middlename	Middle Name
		lastname	Last Name
		address1	Address
		address2	Address
		country/city/state	Country/City/2 digit ISO state code. See Lang/Locale/SiteID Table for codes.
		homephone	Home Phone (input as continuous numerals)
		workphone	Work Phone (input as continuous numerals)
		mobilephone	Mobile Phone (input as continuous numerals)
		optional node to create TR	(NOT SHOWN IN XML)
		emailaddress	name@xxx.com (or net, etc.)
		fax	Fax: (input as continuous numerals)
		webaddress	input as name@provider.xxx
		Educaton Info includes:	CDATA – information entered as CDATA – following guidelines here.
		eduDegree	Educational degree.
		schoolname	School name
		degree	Degree
		major	Major
		GPA	Grade Point Average
		schoolEndDate	School End Date
		MostRecent	Most Recent if this is most recent education inpu the numeral 1
		WorkExpInfo/Work Exp	CDATA – information entered as CDATA – following guidelines here.
		title	Title held
		company	Company name
		responsibilities	Responsibilities
		skills	Skills
		startDate	State Date
		endDate	End Data

Candidate API	Request/ Response	XML Element Tag	Functionality Details
Update Existing User (Profile) (cont.)			
		sendtoDST	Yes or No. Selecting Yes creates a BrassRing Talent Record
	Response	ClientId	516
		SiteId	8723
		AuthenticationToken	7eb45ac9-fbbf-4a1f-8a0b-0df90225b23f
		Profile Info includes:	
		firstname	Joe
)			
	Response (cont.)		
		middlename	Henry
		lastname	Smith
		address1	75 Pope Lane
		address2	
		country/city/state	Danvers
		homephone	8572047967
		workphone	8572047967
		mobilephone	8572047967
		emailaddress	joesmith@gmail.com
		fax	none
		webaddress	none
		eduDegree	Bachelors
		schoolname	Boston University
		degree	Electronic Engineering
		major	Computer
		GPA	4.0
		schoolEndDate	1998
		MostRecent	1
		WorkExpInfo/Work Exp	
		title	Software Engineer
		company	Garnet Software
		responsibilities	Quality Control
		skills	VB/ASp/MSMQ
		startDate	2000
		endDate	Present
		sendtoDST	Yes
		Short Description	Success – Talent Record created.

Error Codes for Candidate APIs

Sr.No	Candidate API Name	Error Code	Error Description
	If Any unhandled Exception happens	100	An unknown error has occurred.
	AuthenticateUser	2	ClientId cannot be empty.
		3	ClientId node is missing from input.
		4	SiteId cannot be empty
		5	SiteId node is missing from input.
		8	Username cannot be empty
		9	Username node is missing from input.
		10	Password cannot be empty
		11	Password node is missing from input.
		201	The site provided does not support logins.
		202	Unable to find that email/password in the system.
		1	There was an error processing the login.
		203	The account has been locked out due to excessive invalid login attempts.
		204	The login was successful, however a password reset is required to request any pages.
		205	The site requested is inactive or does not exist.
	AuthenticateUser - If SSO ID is using for Authenticate User	6	SSO Session ID cannot be empty.
		1	There was an error processing the login.
		321	Suspected replay attack - 2 - attempting relogin with the same link.
		323	Gateway is not SSO enabled.
		2	ClientId cannot be empty.

Sr.No	Candidate API Name	Error Code	Error Description
	Create User	3	ClientId node is missing from input.
		4	SiteId cannot be empty.
		5	SiteId node is missing from input.
		8	Username cannot be empty
		12	Please remove the ' DOUBLE QUOTE (\") ' you have entered.
		13	Security Questions cannot be empty
		14	SecurityQuestions node is missing from input.
		15	Please enter an answer to your security question.
		16	The answer to your security question must be less than 50 characters.
		17	SecurityQuestionsAnswer node is missing from input.
		311	Your log in credentials cannot be validated. You may have created an account in the past, if so please click the 'forgot your password' link on the Welcome page to reset your password.
		312	Your log in credentials cannot be validated. You may have created an account in the past, if so please click the 'forgot your password' link on the Welcome page to reset your password.
		313	You can not use your email address as your password.
		314	You can not use your username as your password.
		315	Your password may not be the same as your login e-mail address.
		316	Your password may not be the same as your username.
		317	Your password must be a minimum of 6 and a maximum of 25 characters.
		318	Your password may not contain spaces.
		319	Your password must contain at least one of the following special characters: {}[],.<>;:'\"" + "?/|\\`~!@#$%^&*()_-+=.
		320	Your password must be a minimum of 8 and a maximum of 25 characters.
		322	Invalid Security Questions.

Sr.No	Candidate API Name	Error Code	Error Description
	Create User	324	Please enter a valid e-mail address.Delete all the invalid characters, including spaces, from your e-mail address.
		325	The username must be less than 100 characters.
		326	The username must not include any spaces.
		327	The username must not include the following characters: < >
		201	The site provided does not support logins.
		205	The site requested is inactive or does not exist.
	AccessPage	18	AuthenticationToken node cannot be empty.
		19	AuthenticationToken node is missing from input.
		20	DestinationPage node cannot be empty.
		21	DestinationPage node is missing from input.
		201	The token provided does not exist.
		202	The token provided has expired.
		203	The token provided must reset the password first.
		204	The page requested is not supported by the authenticated site.
		205	You have requested a page id that does not exist.
		206	The session for this authentication token has been deleted.
		207	The user has logged out of the Talent Gateway. Please request a new Authentication Token.
		208	The session for this authentication token has expired.
		207	The authentication status is not valid for this request.

Sr.No	Candidate API Name	Error Code	Error Description
61.	Assessments	405	You have completed all assessments that are required at this time! Thank you.
62.		406	Error occured while retrieving the assessment records.
63.		407	Invalid URL. Plese check the Redirect URL.
64.	GetJobCart	18	AuthenticationToken node cannot be empty.
65.		19	AuthenticationToken node is missing from input.
66.		29	Action node is missing from input.
67.		30	Action node cannot be empty.
68.		31	Invalid action specified in input.
69.		32	Req node is missing from input.
70.		217	No JobCart data found for this candidate.
71.		215	The information requested is not supported by the authenticated site.
72.		201	The token provided does not exist.
73.		202	The token provided has expired.
74.		206	The session for this authentication token has been deleted.
75.		207	The user has logged out of the Talent Gateway. Please request a new Authentication Token.
76.		208	The session for this authentication token has expired.
77.		207	The authentication status is not valid for this request.
78.		420	The job cart contains a maximum of 50 jobs. Please delete jobs from your cart in order to add new ones.
79.		421	The job cart can contain a maximum of 50 jobs. You can only add *** job(s) to your job cart.
80.	GetStatusCheck	18	AuthenticationToken node cannot be empty.
81.		19	AuthenticationToken node is missing from input.
82.		216	No StatusCheck data found for this candidate.
83.		215	The information requested is not supported by the authenticated site.

84.		**Err or Cod e**	**Error Description**
85.		201	The token provided does not exist.
86.		206	The session for this authentication token has been deleted.
87.		207	The user has logged out of the Talent Gateway. Please request a new AuthenticationToken.
88.		208	The session for this authentication token has expired.
89.		207	The authentication status is not valid for this request.
		18	AuthenticationToken node cannot be empty.
90.	GetResumeList	19	AuthenticationToken node is missing from input.
91.		201	The token provided does not exist.
92.		202	The token provided has expired.
93.		206	The session for this authentication token has been deleted.
94.		207	The user has logged out of the Talent Gateway. Please request a new AuthenticationToken.
95.		208	The session for this authentication token has expired.
96.		207	The authentication status is not valid for this request.
97.		18	AuthenticationToken node cannot be empty.
98.		19	AuthenticationToken node is missing from input.
99.	GetCommunication History	201	The token provided does not exist.
100.		202	The token provided has expired.
101.		206	The session for this authentication token has been deleted.
102.		207	The user has logged out of the Talent Gateway. Please request a new Authentication Token.
103.		208	The session for this authentication token has expired.
104.		207	The authentication status is not valid for this request.

105.		Error Code	Error Description
106.		18	AuthenticationToken node cannot be empty.
107.		19	AuthenticationToken node is missing from input.
108.		230	Authentication status is not valid for this request.
109.		322	Invalid Security Questions.
110.		422	SecurityInfo node is missing.
111.	Change Security Question/Answer	423	Security questions passed in request does not match the required number of nodes or question node names.
112.		424	SecurityQID node is missing.
113.		425	Answer node is missing.
114.		426	A value was not supplied for the answer.
115.		427	A security question cannot be configured again.
116.		18	AuthenticationToken node cannot be empty.
117.		19	AuthenticationToken node is missing from input.
118.		22	OldPassword node cannot be empty.
119.		23	OldPassword node is missing from input.
120.		24	NewPassword node cannot be empty.
121.		25	NewPassword node is missing from input.
122.		251	Your password must be a minimum of 8 and a maximum of 25 characters.
123.	ResetPassword	252	Your password must contain at least one of the following special characters: {}[],.<>;:'""?/|`~!@#$%^&*()_-+=.
124.		253	Your password may not be the same as any of the 5 preceding passwords.
125.		209	Password must be less than 25 characters.
126.		210	Password must contain one of the following characters: -~`!@#$%^&*()_+={}[|<>,.:;'?\"\\]
127.		211	Password cannot be one of the recent passwords used.
128.		212	Password cannot be the same as the username.
129.		213	Password cannot contain spaces.

		Error Code	Error Description
130.		214	The old password entry is incorrect. Please try again.
131.		215	The information requested is not supported by the authenticated site.
132.		**Error Code**	**Error Description**
133.		201	The token provided does not exist.
134.		202	The token provided has expired.
135.		206	The session for this authentication token has been deleted.
136.		207	The user has logged out of the Talent Gateway. Please request a new AuthenticationToken.
137.	Reset Password (cont.)	208	The session for this authentication token has expired.
138.		207	The authentication status is not valid for this request.
139.		208	Password must be at least six characters.
140.		205	The site requested is inactive or does not exist.
141.		201	The token provided does not exist.
142.		202	The token provided has expired.
143.	GetEvents	206	The session for this authentication token has been deleted.
144.		18	Authentifcation Token node cannot be empty
145.		19	Authentifcation Token node is missing from input.
146.	GetSecurityQuestion	201	The token provided does not exist.
147.		202	The token provided has expired.
148.		206	The session for this authentication token has been deleted.
149.		207	The user has logged out of the Talent Gateway. Please request a new AuthenticationToken.
150.		208	The session for this authentication token has expired.
151.		2	ClientID canot be empty.
152.			
153.		3	ClientID node is missing from input.
154.		4	SiteID cannot be empty.

		5	SiteID node is missing from input.
		8	Username cannot be empty.
		9	Username node is missing from input.
		18	Authentifcation Token node cannot be empty
	Forgot Password	2	ClientID canot be empty.
		3	ClientID node is missing from input.
		4	SiteID cannot be empty.
		5	SiteID node is missing from input.
		8	Username cannot be empty.
		18	Authentifcation Token node cannot be empty
		19	Authentifcation Token node is missing from input.
		201	The token provided does not exist.
		202	The token provided has expired.
		206	The session for this authentication token has been deleted.
		207	The user has logged out of the Talent Gateway. Please request a new AuthenticationToken.
		208	The session for this authentication token has expired.
		322 - .	Invalid Security questions
		404 -	Invalid Security answers
		313 - .	You can not use your email address as your password.
		314	You can not use your username as your password.
		315	Your password may not be the same as your login e-mail address.
		316	Your password may not be the same as your username.
		317	Your password must be a minimum of 8 and a maximum of 25 characters.
		318	Your password may not contain spaces.
		319	Your password must contain at least one of the following special characters: {}[],.<>;:'\"" + "?/|\\`~!@#$%^&*()_-+=.

Sr.No	Candidate API Name	Error Code	Error Description
		320	Your password must be a minimum of 8 and a maximum of 25 characters.
		501	Please enter the valid value for Required field - First Name
		502	Please enter the valid value for Required field - Middle Name.
		503	Please enter the valid value for Required field - Last Name.
		504	Please enter the valid value for Required field - Email Address.
		505	Please enter the valid value for Required field – Address 1.
		506	Please enter the valid value for Required field - Address 2.
		507	Please enter the valid value for Required field – City.
		508	Please enter the valid value for Required field - Zip.•.
		509	Please enter the valid value for Required field - Country
	New User/Update Existing User	510	. Please enter the valid value for Required field - State
		511	Please enter the valid value for Required field - Home Phone.
		512	Please enter the valid value for Required field - Work Phone
		513	Please enter the valid value for Required field - Mobile Phone.
		514	Please enter the valid value for Required field - Fax.
		515	Please enter the valid value for Required field - Home Page.
		516	Please enter the valid value for Required field - Last Name Pronunciation.
		517	Please enter the valid value for Required field - First Name Pronunciation.
		518	Invalid Email Address.
		519	First Name is missing.
		520	Last Name is missing.
		521	Config file is missing.
		522	Country State xml file is missing.
		523	Country does not exist.
		524	Error getting localized path is missing.

Sr.No		Error Code	Error Description
	New User/Update Existing User	525	Profile Information saved successfully, however a password reset is required to request any pages.
		526	Error has occured while building XML.
		527	Your e-mail address already exists in our system, or is invalid. Please try again.
		528	This username has already been registered in the system. Please enter a new username.
		529	An internal error has occurred (1).
		530	Please enter a valid first name.
		531	Please enter a valid middle name.
		532	Please enter a valid last name.
		533	Please enter a valid home phone number.
		534	Please enter a valid work phone number.
		535	Please enter a valid phone number.
	New User/Update Existing User	536	Please enter a valid fax number.
		537	Please enter a valid web address. Web addresses start with http:// or https://.
	Candidate Forms	1001	Invalid Form of ID
		1002	No records found
		1003	Invalid input xml
		1004	Unspecified
	Document Packets	218	No document or document packets found for this candidate
		225	The token provided does not exist.
	Simultaneous Login	7	Additional SiteId cannot be empty.
		351	The sites requested must all be active gateways.
		352	The sites requested must all be full gateways.
		353	SSO ID is required for authentication into SSO gateways.

		354	The sites requested must all be configured for either SSO or non-SSO.
	Simultaneous Login	**Error Code**	**Error Description**
		355	The sites requested must all be configured with the same language.
		401	Profile information is missing for this candidate.
		14	formtype id [FORMTYPID] Invalid form type id
		15	fieldid [FIELDID]: Invalid fieldid
		19	Duplicate value provided for:[FIELDID]
		21	fieldid [FIELDID]: Value must be numeric
	TG Form Import	22	fieldid [FIELDID]: Max length = 10
		23	fieldid [FIELDID]: Max length = 4000
		24	fieldid [FIELDID]: Max length = 255
		25	fieldid [FIELDID]: Invalid SSN format.
		26	fieldid [FIELDID]: Invalid email format.
		27	fieldid [FIELDID]: Invalid date format.
		28	fieldid [FIELDID]: Invalid field value provided.

Language/Site/LocaleIDs

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

Language	Language ID	Site Locale ID	ISO State Code (Not US State Codes)
Arabic	6	1025	AR
Azerbaijani	9	1068	AZ
Chinese	138	2052	ZH
Chinese (Traditional)	126	1028	TW
Croatian	44	1050	HR
Czech	20	1029	CS
Danish	22	1030	DA
Dutch	82	1043	NL
Dutch-informal	82	11043	NL-IN
English	1	1033	EN
English (Intl)	140	2057	GB
English RTL	144	21033	ER
English UK Government	145	12057	GG
English US Government	143	11033	GV
Finnish	31	1035	FI
French (Canada)	141	3084	FC
French (France)	34	1036	FR
German	23	1031	DE
German-informal	23	11031	DE-IN
Greek	25	1032	EL
Hebrew	42	1037	HE

Language	Language ID	Site Locale ID	ISO State Code (Not US State Codes)
Hungarian	45	1038	HU
Hungarian-informal	45	11038	HU-IN
Italian	52	1040	IT
Japanese	54	1041	JA
Korean	61	1042	KO
Norwegian	83	1044	NO
Polish	88	1045	PL
Portuguese (Brazil)	90	1046	PT
Portuguese (Portugal)	142	2070	PP
Romanian	94	1048	RO
Russian	95	1049	RU
Serbian	108	2074	SR
Slovak	102	1051	SK
Slovenian	103	1060	SL
Spanish	27	3082	ES
Spanish-informal	27	13082	ES-IN
Swedish	112	1053	SV
Turkish	123	1055	TR
Welsh	21	1106	CY