



Seller Web Service

Version 1.1

Jobg8 Limited

Seller Web Service		10 October 2014
	Version 1.1	1



Document Version History

Date	Version	Description	Author
24 Sept 2013	1.0	Initial Version	Brad O'Connor
10 Oct 2014	1.1	Added GetClicks Method	Brad O'Connor

Seller Web Service		10 October 2014
	Version 1.1	2



1. Contents

DOCUMENT VERSION HISTORY	2
1. CONTENTS	3
2. OVERVIEW	4
3. SELLER WEB SERVICE	4
3.1 FAILURE RESPONSES	4
4. GETAPPLICATIONS METHOD	5
4.1 APPLICATION DETAILS	6
4.2 APPLICATION EXAMPLE XML.....	7
5. GETAPPLICATIONSEX METHOD	8
6. GETCLICKS METHOD	9
6.1 CLICK DETAILS	10
6.2 CLICK EXAMPLE XML.....	10
7. EXAMPLE .NET CODE	11

Seller Web Service		10 October 2014
	Version 1.1	3



2. Overview

This web service is available to allow a seller to download data from Jobg8 and to assist with content management and customer support.

This is a technical document and assumes the reader has working knowledge of XML and web services.

3. Seller Web Service

The web service's URI is:

<http://www.jobg8.com/WebService/Seller.asmx>

and exposes the following methods:

- GetApplications
- GetApplicationsEx
- GetClicks

A username and password is expected by this method as access credentials. These are the same login details used to access the Jobg8 website. If desired, an additional user can be created through the website for the specific purpose of using this web service.

3.1 Failure Responses

When a problem is encountered after successfully calling one of the web service methods, a failure response will be returned. Below is an example of the failure XML:

```
<Error xmlns="">  
  <Code>2</Code>  
  <Type>InvalidCredentials</Type>  
  <Message>Invalid Credentials</Message>  
</Error>
```

This XML will contain one of the following errors:

Code	Type	Message
1	Exception	Exception - please contact Jobg8 support
2	InvalidCredentials	Invalid Credentials
3	AccountAccessDenied	Account Access Denied
4	InvalidApplicationID	The supplied applicationID is invalid

Seller Web Service		10 October 2014
	Version 1.1	4



4. GetApplications Method

This method can be used to return the details of any applications from the last 14 days which have not already been downloaded. It can be polled at any interval to synchronise the applications, however, we recommend it is used regularly to reduce the size of the responses generated. The maximum number of applications that can be returned in one response is capped at 100.

The following parameters are expected:

Parameter	Type	Description
username	String	Access credential.
password	String	Access credential.
accountID	Integer	Indicates the account ID for which to download applications. If an invoice account ID is supplied, the applications for the entire invoice account group will be returned.

XML returned from this method will conform to the following schema (available upon request):

seller web service - getapplications.xsd

Seller Web Service		10 October 2014
	Version 1.1	5



4.1 Application Details

These details will be supplied for each application that is downloaded.

Detail	Description
JobReference	The reference that you use for the job. This is supplied to Jobg8 when we initially posted you the job.
JobID	The Jobg8 identifier for the job.
AccountID	The account ID that provided the application.
AccountName	The account name that provided the application.
Sector	The job's sector.
ApplicationID	The Jobg8 identifier for the application.
UTCDateApplied	The Coordinated Universal Time (UTC) date/time the applicant applied for the job. Formatted as a SortableDateTimePattern.
LocalDateApplied	The local date/time relevant to the account that the applicant applied for the job. Formatted as a SortableDateTimePattern.
ApplicationRevenue	The payment received for the application in the accounts handling currency.
Qualified	Whether or not the candidate qualified.
Duplicate	Whether or not the application was a duplicate.
CustomText	Custom data supplied by yourself when redirecting an applicant to Jobg8 e.g. your applicant identifier.
CandidateName	The name of the candidate.
EmailAddress	The email address of the candidate.
CandidateSold	Whether or not the candidate has been sold.
CandidateRevenue	The payment received for the candidate in the accounts handling currency.
Questions	A question element that is repeated for each question the applicant answered. As application question forms are dynamic this could contain any number and order of questions.
CV and CoverLetter	The CV and Cover Letter elements contain binary Base64-encoded content, which can be used to restore the original files according to the filetype specified in the Filename attribute.



4.2 Application Example XML

```
<?xml version="1.0" encoding="utf-8"?>
<Applications>
  <Application>
    <JobReference />
    <JobID>10001</JobID>
    <AccountID>1030810003</AccountID>
    <AccountName>TestAccount</AccountName>
    <Sector>Accounting</Sector>
    <ApplicationID>4</ApplicationID>
    <UTCDateApplied>2010-11-04T12:23:05</UTCDateApplied>
    <LocalDateApplied>2010-11-04T12:23:05</LocalDateApplied>
    <ApplicationRevenue>0.00</ApplicationRevenue>
    <Qualified>false</Qualified>
    <Duplicate>false</Duplicate>
    <CustomText />
    <CandidateName>John Smith</CandidateName>
    <EmailAddress>example@example.com</EmailAddress>
    <CandidateSold>true</CandidateSold>
    <CandidateRevenue>0.00</CandidateRevenue>
    <CV Filename="cv.txt">VGVzdA==</CV>
    <ApplicationDetails>
      <Questions>
        <Question id="2" formatID="2">
          <QuestionText>First Name</QuestionText>
          <AnswerText>John</AnswerText>
        </Question>
        <Question id="3" formatID="5">
          <QuestionText>Surname</QuestionText>
          <AnswerText>Smith</AnswerText>
        </Question>
        <Question id="11" formatID="15">
          <QuestionText>Email Address</QuestionText>
          <AnswerText>example@example.com</AnswerText>
        </Question>
        <Question id="41" formatID="38" variableID="754">
          <QuestionText>Please select your level of fluency in Chinese</QuestionText>
          <AnswerText>Basic</AnswerText>
        </Question>
        <Question id="154" formatID="191">
          <QuestionText>Please indicate your typing speed?</QuestionText>
          <AnswerText>&lt; 30 wpm</AnswerText>
        </Question>
      </Questions>
    </ApplicationDetails>
  </Application>
</Applications>
```

Seller Web Service		10 October 2014
	Version 1.1	7



5. GetApplicationsEx Method

This method is an extended version of the GetApplications method (see section 4). The extended version is aimed at fully integrated sellers which may require less information to be returned. Its usage and restrictions are exactly the same as the GetApplications method.

The following parameters are expected:

Parameter	Type	Description
username	String	Access credential.
password	String	Access credential.
accountID	Integer	Indicates the account ID for which to download applications. If an invoice account ID is supplied, the applications for the entire invoice account group will be returned.
includeQuestions	Boolean	Indicated whether the Question elements are included in the response.
includeFiles	Boolean	Indicated whether the CV and Cover Letter elements are included in the response.

XML returned from this method will conform to the following schema (available upon request):

seller web service - getapplicationsex.xsd

The schema and returned XML are the same as the GetApplication method apart from the Questions, CV and Cover Letter elements are optional.

Seller Web Service		10 October 2014
	Version 1.1	8



6. GetClicks Method

This method can be used to return the details of any clicks from the last 14 days which have not already been downloaded. It can be polled at any interval to synchronise the clicks, however, we recommend it is used regularly to reduce the size of the responses generated. The maximum number of clicks that can be returned in one response is capped at 100.

The following parameters are expected:

Parameter	Type	Description
username	String	Access credential.
password	String	Access credential.
accountID	Integer	Indicates the account ID for which to download clicks. If an invoice account ID is supplied, the clicks for the entire invoice account group will be returned.

XML returned from this method will conform to the following schema (available upon request):

seller web service - getclicks.xsd

Seller Web Service		10 October 2014
	Version 1.1	9



6.1 Click Details

These details will be supplied for each click that is downloaded.

Detail	Description
JobID	The Jobg8 identifier for the job.
AccountID	The account ID that provided the click.
AccountName	The account name that provided the click.
ClickID	The Jobg8 identifier for the click.
UTCDate	The Coordinated Universal Time (UTC) date/time the click was made. Formatted as a SortableDateTimePattern.
LocalDate	The local date/time relevant to the account that the click was made. Formatted as a SortableDateTimePattern.
Revenue	The payment received for the click in the accounts handling currency.
IPAddress	The IPAddress of the user making the click
CustomText	Custom data supplied by yourself when redirecting an applicant to Jobg8 e.g. your applicant identifier.

6.2 Click Example XML

```
<?xml version="1.0" encoding="utf-8"?>
<Clicks>
  <Click>
    <JobID>10001</JobID>
    <AccountID>1030810003</AccountID>
    <AccountName>TestAccount</AccountName>
    <ClickID>123456</ClickID>
    <UTCDate>2014-10-10T09:40:13</UTCDate>
    <LocalDate>2014-10-10T10:40:13</LocalDate>
    <Revenue>0.06</Revenue>
    <IPAddress>111.170.68.215</IPAddress>
    <CustomText />
  </Click>
</Clicks>
```

Seller Web Service		10 October 2014
	Version 1.1	10



7. Example .NET Code

The code example below demonstrates how to use the GetApplications method:

```
// Create the soap XML
string soapXML = "<soap:Envelope xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\">"
    + "<soap:Body>"
    + "<GetApplications xmlns=\"http://jobg8.com/\">"
    + "<username>testlogin@jobg8.com</username>"
    + "<password>test</password>"
    + "<accountID>123456</accountID>"
    + "</GetApplications>"
    + "</soap:Body>"
    + "</soap:Envelope>";

// Create the request and add the soap XML
System.Net.WebRequest request =
    System.Net.WebRequest.Create("http://www.jobg8.com/WebService/Seller.asmx");
request.Method = "POST";
request.ContentType = "text/xml; charset=utf-8";
request.Headers.Add("SOAPAction", "http://jobg8.com/GetApplications");

using (System.IO.StreamWriter streamWriter = new System.IO.StreamWriter(request.GetRequestStream()))
{
    streamWriter.Write(soapXML);
}

// Get the response and create an XML document from it
System.Net.WebResponse response = request.GetResponse();
System.Xml.XmlDocument xmlResponse = new System.Xml.XmlDocument();
xmlResponse.Load(response.GetResponseStream());

// Parse the response XML
System.Xml.XmlNamespaceManager nsmgr = new System.Xml.XmlNamespaceManager(xmlResponse.NameTable);
nsmgr.AddNamespace("SOAP", "http://schemas.xmlsoap.org/soap/envelope/");
nsmgr.AddNamespace("JOBG8", "http://jobg8.com/");

foreach (System.Xml.XmlNode application in
xmlResponse.SelectNodes("//SOAP:Body/JOBG8:GetApplicationsResponse/JOBG8:GetApplicationsResult/Applications", nsmgr))
{
    // Work through each application
}
```

Alternatively an easy way to access a web service is by using a Web Reference in Visual Studio. The following link details a good example of how to use Web References: [http://msdn.microsoft.com/en-us/library/xb5th1ba\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/xb5th1ba(VS.80).aspx)

Seller Web Service		10 October 2014
	Version 1.1	11