Voices.Com - WebService API Documentation

Wesley Leonard

April 22, 2008

Contents

1 Introduction

Voices.Com Web Service API - Spec Version 0.1a (pre-alpha) - Refers to implementation versions 0.0 - 1.0a

1.1

-
-
-
-
-
-
-

2 Changelog

2.1 Changes from version 0.1a

- Expanded ”Conventions” section
- Added description of how to accept an audition
- .

2.2 Changes from version 0

This is the initial spec, no changes.

3 TODO

- Complete attributes list
- Add more examples
- Finalize xml format and tag names
- Add support for ”compound” objects - ex. a Job + all auditions, Audition + User details, etc
- Figure out how to encode and upload files

4 Conventions

This section covers both conventions used in this document as well as conventions (eg. names, structures) used throughout the API.

Objects refer to types of information that can be created, updated, or queried through the web service. In programming terminology, an object is a clearly defined entity with properties (or attributes) and methods (actions which can be performed on that entity. For example, a *user* object has a username, email, first name, last name, and many more attributes.

User can refer to both talent or clients. *Talent* and *member* refer to voice talent users who will be viewing a job and auditioning for jobs. *Client* refers to users who post jobs, hire talent, and make use of the webservice API. *Talent* information queried through the API is read-only. A *client* may only update their own user information.

Query results are returned in XML format.

- ”item” is a generic term for query results - boolean values returned as ”true” or ”false” - a ”key” field is a unique identifier for an object - a numeric ”id” for most things and a text ”username” for Users. - file info consists of meta data plus a URL - not the actual file itself

Web service URLs - Testing: https://www.voices.com/webservice_sandbox.php - Live: (to be determined)

5 Main Objects

This section describes the main objects available through the web service API. Attributes listed for each object are not complete and may not match actual attribute names. (See appendix [forthcoming] for complete attribute list for each object.)

5.1 Users

Users are the voice talent or clients in the voices.com database. Users can be queried but not created or updated. (Clients can update their own user info only.)

Attributes:

- contact info
- description

- voices samples
- misc documents

5.2 Jobs

(jobs a client has posted) Attributes:

- title
- description
- approved (boolean)
- status (open,pending,closed)
- closing date
- number of auditions

5.3 Auditions

(responses to job postings)

- user information (of member auditioning)
- sound sample (either custom for this job or pre-existing)
- bid price
- notes from talent

5.4 Surepay Transactions

(status of transaction between talent and client after accepting conditions of job offer)

- client and talent information
- price
- final agreement between talent and client

- status of:
  - offer/counter-offer process
  - payment deposit from client
  - files uploaded by talent
  - files approved by client
  - final payment by client
  - feedback between parties

5.5 Files

(files uploaded for job offers, auditions, or any other purpose)

- file URL
- type and size of file
- owner of file

6 Actions

This section describes the actions available and which objects they can be applied to.

6.1 Query

This action retrieves all available information for an object. (Sensitive or internal information such as passwords, notes, or tracking information will not be returned from a query.) All objects (Jobs, Users, Auditions, Surepay Transactions, Files) can be queried. A client can query for any User or File but can only query thier own Jobs, Auditions for their jobs, and Surepay Transactions pertaining to their own jobs.

Query action searches for objects matching the specified criteria. If successful, returns one or more ”items” in an xml list.

6.2 Create

Create inserts a new object into the database. Only Jobs and Files can be explicitly created.

Job Files Surepay Transactions (created by accepting a member audition)

Create action attempts to create a new object with the information provided. If successful, creates object and returns information

6.3 Update

Jobs User (your client info only) Surepay Transactions Files

Update requires the key field be indicated (in most cases this will be a numberic ”id” but for Users it is ”username”. All other fields provided will be

Form ”actions” by attaching verb to noun (object): QueryJob (or queryjob or queryJob or QUERYJOB - not case sensitive) CreateJob UpdateJob QueryUser UpdateFile etc...

6.4 Special Action: AcceptAudition

AcceptAudition is a rare action which is unique to the Audition object. By simply supplying the audition id, an audition by a member is accepted and a Surepay Transaction is generated. The response includes the Surepay Transaction information.

6.5 Special Action: AcceptSurepayOffer

7 Examples

Examples are shown as html forms but do not need to be submitted this way. This is simply to illustrate what fields should be sent and how they should be sent (POST action to the web service URL).

¡!– Returns job with id ”123” (only if it belongs to the client making the query) –¿ ¡form name=”job_query_form” method=”post” action=”https://www.voices.com/vwebservice.php”¿ ¡input type=”hidden” name=”websrvicekeything” value=”2469216769653e97de768361698d483d7d129b ¡input type=”hidden” name=”action” value=”QueryJob”¿ ¡input type=”hidden” name=”id” value=”123”¿ ¡/form¿

¡!– Returns all jobs belonging to the client making the query –¿ ¡form name=”job_query_form” method=”post” action=”https://www.voices.com/vwebservice.php”¿ ¡input type=”hidden”

name=”websrvicekeything” value=”2469216769653e97de768361698d483d7d129b38ce59fb0a07bea7c8
¡input type=”hidden” name=”action” value=”QueryJob”¿ ¡/form¿

¡!– Returns all jobs with title ”Radio Commercial” (not case sensitive) –¿
¡form name=”job_query_form” method=”post” action=”https://www.voices.com/vwebservice.php”¿
¡input type=”hidden” name=”websrvicekeything” value=”2469216769653e97de768361698d483d7d129b
¡input type=”hidden” name=”action” value=”QueryJob”¿ ¡input type=”hidden”
name=”title” value=”Radio Commercial”¿ ¡/form¿

¡!– Returns all jobs with title beginning with ”Radio” (wildcard * matches any
text) –¿ ¡form name=”job_query_form” method=”post” action=”https://www.voices.com/vwebservice.php
¡input type=”hidden” name=”websrvicekeything” value=”2469216769653e97de768361698d483d7d129b
¡input type=”hidden” name=”action” value=”QueryJob”¿ ¡input type=”hidden”
name=”title” value=”Radio*”¿ ¡/form¿

¡response¿ ¡action¿QueryJob¡/action¿ ¡result¿Success¡/result¿ ¡datetime¿2009/01/30
17:23:06¡/datetime¿ ¡cputime¿529¡/cputime¿ ¡numitems¿3¡/numitems¿ ¡items¿ ¡item¿
¡id¿123¡/id¿ ¡title¿Radio Commercial¡/title¿ ¡etc¿...¡/etc¿ ¡/item¿ ¡item¿ ¡id¿345¡/id¿
¡title¿Radio Jingle¡/title¿ ¡etc¿...¡/etc¿ ¡/item¿ ¡item¿ ¡id¿789¡/id¿ ¡title¿Radioactive
Marketing, Inc.¡/title¿ ¡etc¿...¡/etc¿ ¡/item¿ ¡/items¿ ¡/response¿

¡!– Create new job –¿ ¡form name=”job_create_form” method=”post” action=”https://www.voices.com
¡input type=”hidden” name=”websrvicekeything” value=”2469216769653e97de768361698d483d7d129b
¡input type=”hidden” name=”action” value=”CreateJob”¿ ¡input type=”hidden”
name=”title” value=”Radio Commercial”¿ ¡input type=”hidden” name=”description”
value=”Need a youthful sounding female for...”¿ ¡!– Recommended date format:
YYYY/MM/DD Other formats will work but the above is the safest –¿ ¡input
type=”hidden” name=”deadline” value=”2010/01/01”¿ ¡/form¿

Results:

¡response¿ ¡action¿CreateJob¡/action¿ ¡result¿Success¡/result¿ ¡datetime¿2009/01/30
17:23:06¡/datetime¿ ¡cputime¿230¡/cputime¿ ¡numitems¿1¡/numitems¿ ¡items¿ ¡item¿
¡id¿123¡/id¿ ¡title¿123¡/title¿ ¡etc¿...¡/etc¿ ¡/item¿ ¡/items¿ ¡/response¿

OR

¡response¿ ¡action¿CreateJob¡/action¿ ¡result¿Failure¡/result¿ ¡message¿Missing
required information...¡/message¿ ¡datetime¿2009/01/30 17:23:06¡/datetime¿ ¡cputime¿105.00¡/cputime¿
¡/response¿

¡!– Update a job –¿ ¡form name=”job_create_form” method=”post” action=”https://www.voices.com/v
¡input type=”hidden” name=”websrvicekeything” value=”2469216769653e97de768361698d483d7d129b
¡input type=”hidden” name=”action” value=”UpdateJob”¿ ¡input type=”hidden”
name=”id” value=”123”¿ ¡input type=”hidden” name=”title” value=”Radio Com-
mercial”¿ ¡input type=”hidden” name=”description” value=”Need a youthful sound-
ing female for...”¿ ¡/form¿

Result:

¡response¿ ¡action¿UpdateJob¡/action¿ ¡result¿Success¡/result¿ ¡datetime¿2009/01/30 17:23:06¡/datetime¿ ¡cputime¿310.30¡/cputime¿ ¡numitems¿1¡/numitems¿ ¡items¿ ¡item¿ ¡id¿123¡/id¿ ¡title¿Radio Commercial¡/title¿ ¡etc¿...¡/etc¿ ¡/item¿ ¡/items¿ ¡/response¿

¡!– Create a new file –¿ ¡form name=”file_create_form” method=”post” action=”https://www.voices.com/vwebservice.php”¿ ¡input type=”hidden” name=”websrvicekeything” value=”2469216769653e97de768361698d483d7d129b38ce59fb0a07bea7c825d3f85b”¿ ¡input type=”hidden” name=”action” value=”CreateFile”¿ ¡input type=”hidden” name=”title” value=”Script for Radio Commercial”¿ ¡input type=”hidden” name=”description” value=””¿ ¡input type=”file” name=”filename”¿ ¡/form¿

Response:

¡response¿ ¡action¿CreateFile¡/action¿ ¡result¿Success¡/result¿ ¡datetime¿2009/01/30 17:23:06¡/datetime¿ ¡cputime¿610.21¡/cputime¿ ¡numitems¿1¡/numitems¿ ¡items¿ ¡item¿ ¡id¿321¡/id¿ ¡title¿Script for Radio Commercial¡/title¿ ¡filename¿http://www.voices.com/uploads/f ¡etc¿...¡/etc¿ ¡/item¿ ¡/items¿ ¡/response¿

All xml responses will contain the following tags: action - name of action requested (text) result - status (Success/Failure) of request (text/boolean) message - on failure, reason for failure (text) datetime - date and time (YYYY/MM/DD HH:MN:SS) result was returned cputime - time (in milliseconds) required to process request numitems - number of items returned (integer, 0 - infinity) items - list of items (list)

8 Notes

This is an initial spec which is subject to change. Any changes will be reflected in future documentation and explicitly declared in the ”Changelog” section.