BitWine API

Revision History

Date	Version	Comments
10-05-2008	1.00	Initial Version

Table of Contents

Server side REST API

Introduction

BitWine REST API allows you to get information from BitWine server and perform operations.
All API requests should be made to api.bitwine.com server. Requests made to any other server will result in an error.

Currently, only 'GET' requests are supported to retrieve information from BitWine. Future versions will use 'POST' requests to perform operations.

Each API call has its endpoint and parameters. for example user.xml API call to get information about user with user_id 5 uses endpoint http://api.bitwine.com/users/5.xml and have several optional parameters like 'page' etc.
See specific API call documentation for endpoint and parameters.

Common parameters

When accessing BitWine REST API you must provide the following parameters for every request:

Name	Value	

api_version	1	Missing or different value will result in a failed API call.
api_key	the API key that you received	

Response format

API calls can return results in 2 formats XML and JSON. The choice is done by using appropriate extension on the api call endpoint (see examples below)

Get User Profile

Endpoint: http://api.bitwine.com/users/[USER_ID].(xml|json)

USER_ID - identification for the user for which information is requested. Note that this can be both a numeric id (e.g. 17362) as well as alphanumeric username (e.g. 'vitaly'). In case the login id contains a dot ('.') it must be replaced with an underscore ('_'). e.g. login_id 'foo.bar' should be passed as foo_bar.

This API call returns XML with information about the user.

To save an additional API call it also returns the first page of feedbacks/reviews (see section 'Get Feedbacks').

Note: Only information about advisors will be returned.

Parameters:

name	**default**	**optional?**	**description**
per_page	5	yes	number of feedbacks returned

Example.

Getting information about user number 5 with 2 feedbacks:

Request	GET http://api.bitwine.com/users/5.xml?api_version=1&api_key=YOUR_API_KEY&per_page=2
Response	<?xml version="1.0" encoding="UTF-8"?> <result> <user> <languages>English Hebrew</languages> <service> <name>Digital Photography</name> <degrees_and_qualification><p>Self tought digital photographer, see my gallery at <a href="http://www.pbase.com/eladbaron/portfolio"

rel="nofollow" target="_blank">
http://www.pbase.com/eladbaron/portfolio</p>
</degrees_and_qualification>
<language>en</language>
<cents_per_minute>99</cents_per_minute>
<experience><p>I currently shoot with Nikon D200. Many Nikon
lenses. Post processing done in Nikon Capture and Photoshop
CS2</p></experience>
<video_url></video_url>
<url></url>
<special_offer>First 5 minutes are free!</special_offer>
<id>5</id>
<description><p>Help with digital photography workflow, post
processing, and equipment.
Specialized in Nikon equipment.</p></description>
<keywords>Nikon Lenses Camera DSLR, post processing,
photoshop</keywords>
<currency>USD</currency>
</service>
<feedbacks_total_pages>8</feedbacks_total_pages>
<feedbacks_total_entries>15</feedbacks_total_entries>
<last_seen_online>2007-09-17 09:56:17 -0400</last_seen_online>
<ready_to_accept_chat>true</ready_to_accept_chat>
<login>elad</login>
<time_zone>Jerusalem</time_zone>
<rating>
<final_score>100</final_score>
<total_negative>0</total_negative>
<total_neutral>0</total_neutral>
<total_positive>15</total_positive>
<unique_negative>0</unique_negative>
<unique_neutral>0</unique_neutral>
<unique_positive>9</unique_positive>
</rating>
<feedbacks>
<feedback>
<rating>1</rating>
<helpful_count>0</helpful_count>
<updated_at>2007-06-28T19:32:42Z</updated_at>
<is_anonymous>false</is_anonymous>
<unhelpful_count>0</unhelpful_count>
<comment>Perfect Advice</comment>
<left_by_user_id>7</left_by_user_id>
<created_at>2007-06-28T19:32:42Z</created_at>
</feedback>
<feedback>
<rating>1</rating>
<helpful_count>1</helpful_count>
<updated_at>2007-05-24T15:39:28Z</updated_at>
<is_anonymous>false</is_anonymous>
<unhelpful_count>0</unhelpful_count>
<comment>Finally I understood the difference between optical and digital
zoom</comment>
<left_by_user_id>20577</left_by_user_id>

<table>
<tr><td></td><td><pre>
      <created_at>2007-05-24T15:39:28Z</created_at>
    </feedback>
  </feedbacks>
  <feedbacks_page>1</feedbacks_page>
  <id>5</id>
  <feedbacks_per_page>2</feedbacks_per_page>
  <full_name>Elad Baron</full_name>
  <avatar>http://bitwine.com/images/avatars/5.jpg</avatar>
 </user>
</result>
</pre></td></tr>
</table>

<table>
<tr><td>Request</td><td>GET http://api.bitwine.com/users/
5.json?api_version=1&api_key=YOUR_API_KEY&per_page=2</td></tr>
<tr><td>Response</td><td><pre>
{
 "user": {
  "languages": "English Hebrew",
  "service": {
   "name": "Digital Photography",
   "degrees_and_qualification": "\u003Cp\u003ESelf tought digital photographer,
 see my gallery at \u003Ca href=\"http://www.pbase.com/eladbaron/portfolio\"
 rel=\"nofollow\" target=\"_blank\"
\u003Ehttp://www.pbase.com/eladbaron/portfolio\u003C/a\u003E\u003C/p\u003E",
   "language": "en",
   "cents_per_minute": 99,
   "experience": "\u003Cp\u003EI currently shoot with Nikon D200. Many Nikon lenses.
Post processing done in Nikon Capture and Photoshop CS2\u003C/p\u003E",
   "video_url": "",
   "url": "",
   "special_offer": "First 5 minutes are free!",
   "id": 5,
   "description": "\u003Cp\u003EHelp with digital photography workflow, post
processing, and equipment. \nSpecialized in Nikon equipment.\u003C/p\u003E",
   "keywords": "Nikon Lenses Camera DSLR, post processing, photoshop",
   "currency": "USD"
  },
  "feedbacks_total_pages": 8,
  "feedbacks_total_entries": 15,
  "last_seen_online": "2007/09/17 09:56:17 -0400",
  "ready_to_accept_chat": true,
  "login": "elad",
  "time_zone": "Jerusalem",
  "rating": {
   "total_positive": 15,
   "unique_negative": 0,
   "total_neutral": 0,
   "total_negative": 0,
   "unique_positive": 9,
   "unique_neutral": 0,
   "final_score": 100
  },
  "feedbacks": [
   {
    "rating": 1,
    "helpful_count": 0,
</pre></td></tr>
</table>

```
      "updated_at": "2007/06/28 19:32:42 +0000",
      "is_anonymous": false,
      "unhelpful_count": 0,
      "comment": "Perfect Advice",
      "left_by_user_id": 7,
      "created_at": "2007/06/28 19:32:42 +0000"
    }, {
      "rating": 1,
      "helpful_count": 1,
      "updated_at": "2007/05/24 15:39:28 +0000",
      "is_anonymous": false,
      "unhelpful_count": 0,
      "comment": "Finally I understood the difference between optical and digital zoom",
      "left_by_user_id": 20577,
      "created_at": "2007/05/24 15:39:28 +0000"
    }
  ],
  "feedbacks_page": 1,
  "id": 5,
  "feedbacks_per_page": 2,
  "full_name": "Elad Baron",
  "avatar": "http://bitwine.com/images/avatars/5.jpg"
 }
}
```


Get Feedbacks

Endpoint: http://api.bitwine.com/users/[USER_ID]/feedbacks.(xml|json)

USER_ID - identification for the user for which information is requested. Note that this can be both a numeric id (e.g. 17362) as well as string user login id (e.g. 'vitaly'). In case the login id contains a dot ('.') it must be replaced with an underscore ('_'). i.e. login_id 'foo.bar' should be passed as foo_bar.

This API call will return XML with feedbacks for the user.

Parameters:

name	default	optional?	description
per_page	5	yes	number of feedbacks returned
page	1	yes	which page to return
filter	none	yes	possible values: (-1, 0, 1). Only return feedbacks with the given value (i.e. -1 for only negative, 0 for only neutral and 1 for only positive)

Example.

Getting 3rd page of feedbacks for user with id 5 (2 feedbacks per page)

Request	GET http://api.bitwine.com/users/5/feedbacks.xml?api_version=1&api_key=YOUR_API_KEY&per_page=2&page=3
Response	<?xml version="1.0" encoding="UTF-8"?> <result> <feedbacks> <feedback> <rating>1</rating> <helpful_count>3</helpful_count> <updated_at>2006-11-09T13:18:20Z</updated_at> <is_anonymous>false</is_anonymous> <unhelpful_count>3</unhelpful_count> <comment>thanks for feedback on my pics</comment> <left_by_user_id>6</left_by_user_id> <created_at>2006-11-09T13:18:20Z</created_at> </feedback> <feedback> <rating>1</rating> <helpful_count>8</helpful_count> <updated_at>2006-11-09T19:40:35Z</updated_at> <is_anonymous>false</is_anonymous> <unhelpful_count>1</unhelpful_count> <comment>He knows cameras. A bit of a Nikon bias, but knowledgable about the Cannon world. Specifically helpful in discussion about lens alternatives comparing trade-offs/requirements between telephoto shots for kids sporting events vs wildlife.</comment> <left_by_user_id>2465</left_by_user_id> <created_at>2006-11-09T19:40:35Z</created_at> </feedback> </feedbacks> <total-pages>15</total-pages> <per-page>2</per-page> <page>3</page> <total-entries>30</total-entries> </result>

Same in json format:

Request	GET http://api.bitwine.com/users/5/feedbacks.json?api_version=1&api_key=YOUR_API_KEY&per_page=2&page=3
Response	{ "total_pages": 16, "per_page": 2, "total_entries": 31, "feedbacks": [{"rating": 1, "helpful_count": 3, "updated_at": "2006/11/09 13:18:20 +0000",

```
            "is_anonymous": false,
            "unhelpful_count": 3,
            "left_by_user_id": 6,
            "comment": "thanks for feedback on my pics",
            "created_at": "2006/11/09 13:18:20 +0000"
        },
        {
            "rating": 1,
            "helpful_count": 8,
            "updated_at": "2006/11/09 19:40:35 +0000",
            "is_anonymous": false,
            "unhelpful_count": 1,
            "left_by_user_id": 2465,
            "comment" : "He knows cameras. A bit of a Nikon bias, but knowledgable about
the Cannon world. Specifically helpful in discussion about lens alternatives comparing
trade-offs/requirements between telephoto shots for kids sporting events vs wildlife.",
            "created_at": "2006/11/09 19:40:35 +0000"
        }
    ],
    "page": 3
}
```


Search

Endpoint: http://api.bitwine.com/search.(xml|json)/[query]?[parameters]&[parameters]
http://api.bitwine.com/search.(xml|json)?query=[query]&[parameters]&[parameters]

This API call will return XML or JSON with search results for supplied query and parameters.
Search results include service info, user info and rating info for this user.

Note: Only information about advisors will be returned.

Program
By default the scope of search is determined by the program (Network) associated with specified API key.
If program parameter (see below) is specified, the search will be performed only within the scope of the specified program, provided that such program is visible from the program associated with API key.

Parameters:

name	default	optional?	description
query	none	yes	<link to query language document>
per_page	10	yes	number of search results returned
page	1	yes	search results offset. if per_page is 10 and page is 2 results from 11 to 20 will be returned

language	en	yes	all_lang | en | de search records written in the specified language only
category_id	none	yes	all services from specified category
affiliate_id	none	yes	search for all services that are affiliated to supplied user id
program	default	yes	search for results in this specific program
in_list	none	yes	only users from specified user list will be returned
not_in_list	none	yes	only users not in the specified user list will be returned.

Example:

Getting first 2 xml search results for search term "ruby rails":

Request	GET http://api.bitwine.com/search.xml?query=rubyonrails&api_version=1&api_key=YOUR_API_KEY&per_page=2
Response	<?xml version="1.0" encoding="UTF-8"?> <result> <page>1</page> <total-pages>1</total-pages> <total-entries>2</total-entries> <per-page>10</per-page> <users> <user> <languages>Russian Hebrew English</languages> <service> <name>Software Architecture</name> <degrees_and_qualification>...</degrees_and_qualification> <language>en</language> <cents_per_minute>200</cents_per_minute> <experience>...</experience> <video_url></video_url> <url></url> <special_offer></special_offer> <id>6</id> <description>...</description> <keywords>internet ruby rails c++ mysql rubyonrails</keywords> <currency>USD</currency> </service> <last_seen_online>2008-07-08 14:12:06 -0400</last_seen_online> <ready_to_accept_chat>true</ready_to_accept_chat> <location>Tel Aviv, Israel</location> <login>boris</login>

```
    <time_zone>Jerusalem</time_zone>
    <rating>
      <final_score>66</final_score>
      <total_negative>0</total_negative>
      <total_neutral>2</total_neutral>
      <total_positive>15</total_positive>
      <unique_negative>0</unique_negative>
      <unique_neutral>1</unique_neutral>
      <unique_positive>4</unique_positive>
    </rating>
    <id>6</id>
    <full_name>Boris  Nadion</full_name>
    <avatar>http://bitwine.com/images/avatars/6.jpg</avatar>
  </user>
  <user>
    <languages>English Hebrew Russian</languages>
    <service>
      <name>Development</name>
      <degrees_and_qualification>...</degrees_and_qualification>
      <language>en</language>
      <cents_per_minute>200</cents_per_minute>
      <experience>...</experience>
      <video_url>http://www.youtube.com/watch?v=7bLanIfR13A</video_url>
      <url>http://www.linkedin.com/in/taasaa</url>
      <special_offer>First conversation is free, to get to know each other.</special_offer>
      <id>1</id>
      <description>...</description>
      <keywords>Ruby Rails .NET C++ usability rubyonrails</keywords>
      <currency>USD</currency>
    </service>
    <last_seen_online>2007-09-14 07:51:44 -0400</last_seen_online>
    <ready_to_accept_chat>false</ready_to_accept_chat>
    <location>Tel Aviv, Israel</location>
    <login>michael</login>
    <time_zone>Jerusalem</time_zone>
    <rating>
      <final_score>100</final_score>
      <total_negative>0</total_negative>
      <total_neutral>0</total_neutral>
      <total_positive>8</total_positive>
      <unique_negative>0</unique_negative>
      <unique_neutral>0</unique_neutral>
      <unique_positive>4</unique_positive>
    </rating>
    <id>3</id>
    <full_name>Michael Mazyar</full_name>
    <avatar>http://bitwine.com/images/avatars/3.jpg</avatar>
  </user>
 </users>
</result>
```

Note: the id tag *inside user tag* should be used in subsequent calls via the Javascript API (see next section)

Example:

Getting first 2 json search results for search term "rubyonrails":

Request	GET http://api.bitwine.com/search.json?query=rubyonrails&api_version=1&api_key=YOUR_API_KEY&per_page=2

Response	{
	"page": 1,
	"total_pages": 1,
	"total_entries": 2,
	"per_page": 10,
	"users": [
	{
	"languages": "Russian Hebrew English",
	"service": {
	"name": "Software Architecture",
	"degrees_and_qualification": "...",
	"language": "en",
	"cents_per_minute": 200,
	"experience": "...",
	"video_url": "",
	"url": "",
	"special_offer": "",
	"id": 6,
	"description": "...",
	"keywords": "internet ruby rails c++ mysql rubyonrails",
	"currency": "USD"
	},
	"last_seen_online": "2008/07/08 14:12:06 -0400",
	"ready_to_accept_chat": true,
	"location": "Tel Aviv, Israel",
	"login": "boris",
	"time_zone": "Jerusalem",
	"rating": {
	"total_positive": 15,
	"unique_negative": 0,
	"total_neutral": 2,
	"total_negative": 0,
	"unique_positive": 4,
	"unique_neutral": 1,
	"final_score": 66},
	"id": 6,
	"full_name": "Boris Nadion",
	"avatar": "http://bitwine.com/images/avatars/6.jpg"
	},
	{
	"languages": "English Hebrew Russian",
	"service": {
	"name": "Development",
	"degrees_and_qualification": "...",
	"language": "en",
	"cents_per_minute": 200,
	"experience": "...",
	"video_url": "http://www.youtube.com/watch?v=7bLanIfR13A",

```
      "url": "http://www.linkedin.com/in/taasaa",
      "special_offer": "First conversation is free, to get to know each other.",
      "id": 1,
      "description": "...",
      "keywords": "Ruby Rails .NET C++ usability rubyonrails",
      "currency": "USD"
    },
    "last_seen_online": "2007/09/14 07:51:44 -0400",
    "ready_to_accept_chat": false,
    "location": "Tel Aviv, Israel",
    "login": "michael",
    "time_zone": "Jerusalem",
    "rating": {
      "total_positive": 8,
      "unique_negative": 0,
      "total_neutral": 0,
      "total_negative": 0,
      "unique_positive": 4,
      "unique_neutral": 0,
      "final_score": 100},
    "id": 3,
    "full_name": "Michael Mazyar",
    "avatar": "http://bitwine.com/images/avatars/3.jpg"
  }
 ]
}
```


Client side Javascript API

Javascript API enables client (browser) side integration.

Javascript calls open a 'virtual' floating window (implemented as an iframe) with a drop shadow over the current page. Requires flash 8 installed on the computer.

Initialization

To enable javascript API include the following lines at the top of HTML page inside of the HEAD tag.

```
<script src="http://api.bitwine.com/javascripts/bitwine_api.js" type="text/
javascript"></script>
<script type="text/javascript">
   BITWINE.api.start({auid: <YOUR_AUID>});
</script>
```

The script will load all needed components automatically. YOUR_AUID is an integer representing your BitWine Affiliate ID. You must supply your AUID if you want the traffic generated to be associated with you (for revenue sharing and tracking purposes).

Also, put the following HTML/javascript snipped somewhere in the page:

```
<script type="text/javascript">
  BITWINE.api.render_no_flash_div();
</script>
```

This javascript will render div element, which in most cases will remain invisible. Only if the user does not have Flash Player version 8.0 or later installed on the computer, a standard Adobe banner for download/upgrade Flash Player will be shown inside this div. The div should be able to grow up to 320x240 pixels. Don't give display:none or visibility:hidden CSS attributes to any parent elements of the div.

The snippet should be included only once per page.

Example:

```
<div id="left_column">
    <script type="text/javascript">
      BITWINE.api.render_no_flash_div();
    </script>
</div>
<div id="content">
    <!-- ... -->
</div>
<div id="right_column">
    <!-- ... -->
</div>
```

Some javascript API functions obtain javascript object (hash) as parameter, hash keys depend on API call and explained below.

"On available" callback

Some of the API components are loaded asynchronously. In order to know when the components are loaded you may supply a notification callback function which will be called when API is ready for usage.

To supply a notification callback function use the following:

```
BITWINE.api.on_available(func)
```

And pass your function as a parameter. Your anonymous function will be called when BitWine JS API becomes available. You may call `BITWINE.api.on_available(..)` more than once to bind several functions to this event.

Example:

```
<div id="bitwine_integration" style="display:none">
    <!-- chat html elements go here -->
</div>

<script type="text/javascript">
  BITWINE.api.on_available(function(){
     document.getElementById("bitwine_integration").style.display = "";
```

```
  });
</script>
```

Initiating a Consultation Session with an Advisor

Opens a chat window and initiates a connection with an advisor. Note that all consultation sessions start via chat (instant messaging). While in chat, the advisor may choose to invite the client to a voice conversation.

```
BITWINE.api.chat.start(opts)
```

opts is a javascript object (hash) that contains the following keys (all keys are case sensitive):

name	**type**	**default**	**optional?**	**description**
user_id	integer or string		NO	identification for the user to start chat with, supports integer user_ids as well as string usernames (see "User Profile API call" for more detailed description of user_id parameter)
first_message	string	Hello	yes	first chat message sent to advisor automatically
client_name	string	Guest	yes	name / nickname of client starting the chat
on_ringing	function	null	yes	javascript function to call when connection with BitWine servers is established and chat window is shown
on_chat_ended	function	null	yes	javascript function to call when chat ended and chat window is closed
on_chat_failed	function(reason)	null	yes	javascript function to be called when chat connection fails.

				'reason' is the error message returned.

Examples:

The following code shows "Start chat with John" link. When clicked, a floating chat window shows up, and a connection with advisor John (user_id = 12300) is established.

```
<a href="#" id="start_chat_with_john"
onclick="BITWINE.api.chat.start({user_id: 12300});return false;">Start chat
with John</a>
```

A more robust example using prototype.js (http://prototypejs.org). The user is asked to enter a client name and a first message, and then to click on "Start chat with John" link (John's username is "john"). Following the click, a "Connecting..." span is shown and the text fields are disabled. When the chat window opens the "Connecting..." span is hidden. When the chat is closed the text fields are enabled again.

```
Your name:
<input type="text" id="client_name" value="Guest" />
First message:
<input type="text" id="first_message" />
<a href="javascript:void(0)" id="chat_with_john">Start chat with john</a>
<span id="connecting" style="display:none">Connecting...</span>

<script type="text/javascript">
   Event.observe("chat_with_john", "click", function(){
        $("connecting").show();
        $("client_name").disabled = true;
        $("first_message").disabled = true;
     BITWINE.api.chat.start({
       user_id: "john",
       first_message: $("first_message").value,
       client_name: $("client_name").value,
       on_ringing: function() {
         $("connecting").hide();
       },
       on_chat_ended: function() {
         $("client_name").disabled = true;
         $("first_message").disabled = true;
       }
     });
   });
</script>
```

Leaving a message to an Advisor

Opens a window for the client to leave a message for an advisor.

```
BITWINE.api.leave_message.show(opts)
```

opts is a javascript object (hash) that contains the following keys (all keys are case sensitive):

name	type	default	optional?	description
user_id	integer or string		NO	identification for the user to start chat with, supports integer user_ids as well as string usernames (see "User Profile API call" for more detailed description of user_id parameter)
on_window_opened	function	null	yes	a javascript callback function to call when "Leave a message" window is opened
on_window_closed	function	null	yes	a javascript callback function to call when "Leave a message" window is closed

Example:

The following code shows "Leave a message with John" link. When clicked, the user is presented with a floating window for leaving a message to advisor John (user_id = 12300).

```
<!-- the link -->
  <a href="#" id="leave_a_message_to_john"
onclick="leave_a_message(12300);return false;">Leave a message to John</a>
  <!-- some message that will be shown to user when connection is being
established -->
  <span id="connecting" style="display:none">Connecting...</span>

  <script type="text/javascript">
    function leave_a_message(user_id) {
      var msg = document.getElementById("connecting");
      var lnk = document.getElementById("leave_a_message_to_john");
      // hide link and show message "Connecting..."
```

```
      msg.style.display = "";
      lnk.style.display = "none";
      BITWINE.api.leave_message.show({user_id: user_id, on_window_closed:
function() {
        // hide message and show link
        msg.style.display = "none";
        lnk.style.display = "";
      }});
    }
  </script>
```

HTML/JS Buttons API

HTML/JS Buttons API enables client-side integration with almost no effort.

It is a combination of Javascript API and pre-rendered HTML elements to start a chat with an advisor and to leave a message to an avisor. You can mix this integration approach with "Client side Javascript API". Maximum number of chat buttons allowed in one page is 50.

Initialization

To enable HTML/JS Buttons API include the following lines at the top of the HTML page inside a HEAD tag. Including this snippet will allow the functionality of HTML/JS Buttons API and Client side Javascript API.

```
<script src="http://api.bitwine.com/javascripts/bitwine_api_buttons.js"
type="text/javascript"></script>


<script type="text/javascript">
   BITWINE.api.start({auid: <YOUR_AUID>});
</script>
```

Don't include "Initialization" snippet from "Client side Javascript API" if you use HTML/JS Buttons API.

The above script will load all required components automatically.

Also, put the following HTML snipped somewhere in the page:

```
<script type="text/javascript">
  BITWINE.api.render_no_flash_div();
</script>
```

This javascript will render div element, which in most cases will remain invisible. Only if the user does not have Flash Player version 8.0 or later installed on the computer, a standard Adobe banner for download/upgrade Flash Player will be shown inside this div. The div should be able to grow up to 320x240 pixels. Don't give display:none or visibility:hidden CSS attributes to any parent elements of the div.

The snippet should be included only once per page. See example at "Client side Javascript API".

Start Chat button

Presents a button for initiating a session with an advisor. Note that all consultation sessions start via chat (instant messaging). While in chat, the advisor may choose to invite the client to a voice conversation.

Insert the following snippet, "Start Chat" button will be rendered in the place of javascript call:

```
BITWINE.api.buttons.render_start_chat(opts);
```

opts is a javascript object (hash) that contains the following keys (all keys are case sensitive):

name	**type**	**default**	**optional?**	**description**
user_id	integer or string		NO	identification for the user to start chat with, supports integer user_ids as well as string usernames (see "User Profile API call" for more detailed description of user_id parameter)
ready_to_accept_chat	boolean	null	Yes	pass true/false if you get this value from Search or Get User Profile API call

If *ready_to_accept_chat* key is not passed the user state will be detected automatically. If user is "ready_to_accept_chat" green clickable button will be rendered (as shown below). If user is not "ready_to_accept_chat" - a grayed out button will be rendered.

All the rest will be done automatically by javascript:

- rendering "Start chat" button of appropriate style depending on user status (online/offline)
- binding "onclick" events to start a chat when the button is clicked

The rendered button will be 118px wide and 48px high:

User is ready to accept chat	
User is offline or cannot accept chat at this time	
When user clicked "Start Chat"	

Examples:

User is 12300, green "Start Chat" button will be rendered inside a div with class "start_chat".

```
<div class="start_chat">
    <script type="text/javascript>
        BITWINE.api.buttons.render_start_chat({user_id: 12300,
ready_to_accept_chat: true});
    </script>
</div>
```

User is john, "Start Chat" button will be rendered inside a span with id "start_chat_john", the user state (ready_to_accept_chat) will be detected automatically.

```
<span id="start_chat_john">
    <script type="text/javascript>
        BITWINE.api.buttons.render_start_chat({user_id: "john"});
    </script>
</span>
```

Leave a Message button

Presents a button for leaving a message to an advisor.

Insert the following snippet, "Leave a Message" button will be rendered in the place of javascript call:

```
BITWINE.api.buttons.render_leave_message({user_id: USER_ID});
```

where USER_ID is the user_id as described in BITWINE.api.leave_a_message.show Javascript API call.

all the rest will be done automatically by javascript:

- rendering "Leave a message" button
- binding "onclick" events to show leave-a-message window when button is clicked

The rendered button will have be 118px wide and 48px high:

Leave a Message

Chat and Message button

A combination of the Start Chat button and the Leave a Message button. Shows both buttons if advisor is online. If offline, the "Start Chat" button is grayed out.

Insert the following snippet:

```
BITWINE.api.buttons.render_chat_and_message(opts);
```

opts are the same as Start Chat button API call.

The rendered buttons will be 256px wide and 48px high:

 Start Chat Leave a Message

Integration Example

Assuming your BitWine affiliate id is 13000, and API key is fbfcb9cac429214ca3f406def7891d03.

Flow overview:

- Make search API call
- Get and parse XML response
- Render results on your page using Client Side Javascript API

Step I:

- Make search API call searching for "ruby rails":

GET http://api.bitwine.com/search.xml?query=rubyonrails&api_version=1&api_key= fbfcb9cac429214ca3f406def7891d03&per_page=10

Step II:

- Parse API response. Note the user_id in each record found (marked in bold below). You will need these for the next step

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <page>1</page>
  <total-pages>1</total-pages>
  <total-entries>2</total-entries>
  <per-page>10</per-page>
  <users>
    <user>
      <languages>Russian Hebrew English</languages>
      <service>
        <name>Software Architecture</name>
```

```
        <degrees_and_qualification>...</degrees_and_qualification>
        <language>en</language>
        <cents_per_minute>200</cents_per_minute>
        <experience>...</experience>
        <video_url></video_url>
        <url></url>
        <special_offer></special_offer>
        <id>6</id>
        <description>...</description>
        <keywords>internet ruby rails c++ mysql rubyonrails</keywords>
        <currency>USD</currency>
      </service>
      <last_seen_online>2008-07-08 14:12:06 -0400</last_seen_online>
      <ready_to_accept_chat>true</ready_to_accept_chat>
      <location>Tel Aviv, Israel</location>
      <login>boris</login>
      <time_zone>Jerusalem</time_zone>
      <rating>
        <final_score>66</final_score>
        <total_negative>0</total_negative>
        <total_neutral>2</total_neutral>
        <total_positive>15</total_positive>
        <unique_negative>0</unique_negative>
        <unique_neutral>1</unique_neutral>
        <unique_positive>4</unique_positive>
      </rating>
      <id>6</id>
      <full_name>Boris  Nadion</full_name>
      <avatar>http://bitwine.com/images/avatars/6.jpg</avatar>
    </user>
    <user>
      <languages>English Hebrew Russian</languages>
      <service>
        <name>Development</name>
        <degrees_and_qualification>...</degrees_and_qualification>
        <language>en</language>
        <cents_per_minute>200</cents_per_minute>
        <experience>...</experience>
        <video_url>http://www.youtube.com/watch?v=7bLanIfR13A</video_url>
        <url>http://www.linkedin.com/in/taasaa</url>
        <special_offer>First conversation is free, to get to know each
other.</special_offer>
        <id>1</id>
        <description>...</description>
        <keywords>Ruby Rails .NET C++ usability rubyonrails</keywords>
        <currency>USD</currency>
      </service>
      <last_seen_online>2007-09-14 07:51:44 -0400</last_seen_online>
      <ready_to_accept_chat>false</ready_to_accept_chat>
      <location>Tel Aviv, Israel</location>
      <login>michael</login>
      <time_zone>Jerusalem</time_zone>
      <rating>
        <final_score>100</final_score>
        <total_negative>0</total_negative>
        <total_neutral>0</total_neutral>
        <total_positive>8</total_positive>
        <unique_negative>0</unique_negative>
```

```
        <unique_neutral>0</unique_neutral>
        <unique_positive>4</unique_positive>
      </rating>
      <id>3</id>
      <full_name>Michael Mazyar</full_name>
      <avatar>http://bitwine.com/images/avatars/3.jpg</avatar>
    </user>
  </users>
</result>
```

In this example we want to provide a "Start Chat" link for each online advisor and "Leave a message" for advisors that are not ready to accept chats.

- Advisor with user_id=6 from the above results is ready to accept chats:

```
<user>

    <ready_to_accept_chat>true</ready_to_accept_chat>
     ...
 <id>6</id>
     ...
</user>
```

- Advisor with user_id=3 is **not** ready to accept chats:

```
<user>
     <ready_to_accept_chat>false</ready_to_accept_chat>
     ...
     <id>3</id>

     ...
 </user>
```

Step III:

- Render html results using Client Side Javascript API:

```
<html>
    <head>
        <script src="http://api.bitwine.com/javascripts/bitwine_api.js"
type="text/javascript"></script>
        <script type="text/javascript">
          BITWINE.api.start({auid: 13000});
         BITWINE.api.on_available(function(){
            document.getElementById("bitwine_integration").style.display =
"";
         });
          function leave_a_message(user_id) {
            var connecting = document.getElementById("connecting");
            connecting.style.display = "";
            BITWINE.api.leave_message.show({user_id: user_id,
              on_window_closed: function() {
                 connecting.style.display = "none";
              }
            });
```

```
          }
          function start_chat(user_id) {
             var connecting = document.getElementById("connecting");
             connecting.style.display = "";
             BITWINE.api.chat.start({user_id: user_id,
              on_chat_ended: function(){
                 connecting.style.display = "none";
              }
             });
          }
        </script>

    </head>
       <body>
        <script type="text/javascript">
           BITWINE.api.render_no_flash_div();
        </script>
        <div id="bitwine_integration" style="display:none">
          <div id="connecting" style="display:none">Connecting...</div>
          <a href="#" onclick="start_chat(6);return false;">Start a Chat
with Boris Nadion</a>
          <a href="#" onclick="leave_a_message(3);return false;">Leave a
Message to Michael Mazyar </a>
        </div>
       <body>
</html>
```