

Table of Contents

Overview.....	1
Overall API Usage.....	2
Create meeting (with your own meeting ID) example:.....	2
Create meeting (with no meeting ID) example:.....	2
Join meeting example:.....	2
Is meeting running example:.....	2
API Security Model.....	3
Configuration.....	3
Usage.....	3
Error handling.....	4
Desired Future Features.....	4
Support for JSON / JSONP.....	4
Meeting event callbacks.....	5
API Calls.....	5
Standard parameters and responses.....	5
Parameters:.....	5
Response:.....	6
Meeting workflow calls.....	6
Create meeting (create).....	6
Parameters:.....	7
Response:.....	8
Join meeting (join).....	9
Parameters:.....	9
Response:.....	9
Back-office administration calls.....	9
Is meeting running (isMeetingRunning).....	10
Parameters:.....	10
Response:.....	10
End Meeting (endMeeting).....	10
Parameters:.....	10
Response:.....	11
List Meeting Attendees (listAttendees).....	11
Get Meeting Info (getMeetingInfo).....	11

Overview

This document outlines the API that is available for integrating BigBlueButton with other applications. It is currently in draft stage as not all features are available. Features which are not currently implemented will be clearly marked as such.

Overall API Usage

The API is implemented with the end-user in mind. To call the API, you simply make an HTTP request to a URL with the query parameters supplying the information needed for the call. The API returns an XML response. Here are some example API calls and their responses:

Create meeting (with your own meeting ID) example:

[http://yourserver.com/bigbluebutton/api/create?
name=Test+Meeting&meetingID=abcdef&attendeePW=111222&moderatorPW=333444](http://yourserver.com/bigbluebutton/api/create?name=Test+Meeting&meetingID=abcdef&attendeePW=111222&moderatorPW=333444)

```
<response>
  <returncode>SUCCESS</returncode>
  <meetingToken>6db0a605-627f-4f6c-a384-53f745ee372c</meetingToken>
  <meetingID>abcdef</meetingID>
  <attendeePW>111222</attendeePW>
  <moderatorPW>333444</moderatorPW>
  <messageKey/>
  <message/>
</response>
```

Create meeting (with no meeting ID) example:

<http://yourserver.com/bigbluebutton/api/create?name=Test+Meeting&meetingID=fedcba>

```
<response>
  <returncode>SUCCESS</returncode>
  <meetingToken>0752956f-dc4a-46f1-81a7-alf98014c723</meetingToken>
  <meetingID>fedcba</meetingID>
  <attendeePW>tverwpay</attendeePW>
  <moderatorPW>eqma84wp</moderatorPW>
  <messageKey/>
  <message/>
</response>
```

Join meeting example:

[http://yourserver.com/bigbluebutton/api/join?
fullName=Joe+User&meetingID=abcdef&password=333444](http://yourserver.com/bigbluebutton/api/join?fullName=Joe+User&meetingID=abcdef&password=333444)

This call sends the caller directly into the application. There is no XML response. You should redirect your user to a URL like this when you are done creating the meeting and are ready to direct them into the meeting.

Is meeting running example:

<http://yourserver.com/bigbluebutton/api/isMeetingRunning?meetingID=abcdef>

```
<response>
  <returncode>SUCCESS</returncode>
  <running>true</running>
</response>
```

More details on these calls can be found in the "[API Calls](#)" section below.

API Security Model

The API provides a way to be secured so that third-party apps may make calls to it, but not allow other people (end users) to make similar calls to create their own meetings, etc.

For instance, a third party application that is integrated with the API may embed a JavaScript URL that calls to the BBB API to create a meeting. An end user will be able to see this URL in their browser and should *not* be allowed to modify a parameter (for example, the meeting ID) and create their own meeting. By doing so, they could create meetings that were not tracked by the third-party app, which could present a billing problem, et cetera. They should also not be able to modify any parameter (for instance the maximum attendees) to modify the meeting that is being created by the third party app.

Those examples are clearly abuses of the system, and therefore must be protected against. The API must provide a way of enabling this protection.

Configuration

TODO: the securitySalt is currently set on the ApiController and needs to be able to be configured via bigbluebutton.properties (or similar)

IMPORTANT: DO NOT ALLOW END USERS TO KNOW YOUR SECURITY SALT OR ELSE YOUR SECURITY WILL BE COMPROMISED.

Usage

The implementation of this security model lies in ApiController.groovy. It entails creating a checksum out of the combination of the entire HTTP query string and a server-configured security token.

To use the security model, you must be able to create an SHA-1 checksum out of the query string *plus* the security salt that you configured on your server. To do so, follow these steps:

1. Create the entire query string for your API call *without* the checksum parameter.
 - (example for create meeting API call:
name=Test+Meeting&meetingID=abc123&attendeePW=111222&moderatorPW=333444)
2. Append the security salt to your string
 - example for above query string:
 - security salt is 639259d4-9dd8-4b25-bf01-95f9567eaf4b
 - String becomes:

```
name=Test+Meeting&meetingID=abc123&attendeePW=111222&moderatorPW=333444639259d4-9dd8-4b25-bf01-95f9567eaf4b
```

3. Now, find the SHA-1 sum for that string (implementation varies based on programming language).
 - The SHA-1 sum for the above string is: fe1f3f036d420997a5aa6fdadb21eba346a892ad
4. Add a checksum parameter to your query string that contains this checksum.
 - Above example becomes:
name=Test+Meeting&meetingID=abc123&attendeePW=111222&moderatorPW=333444&checksum=fe1f3f036d420997a5aa6fdadb21eba346a892ad

If you configured a security salt on your server, you ***MUST*** send this checksum with ***EVERY*** API call. Since end users do not know your security salt, they can not fake calls to the server, and they can not modify your calls – since changing a single parameter name or value by only one character will completely change the checksum required to validate the call.

Implementations of the SHA-1 functionality exist in nearly all programming languages. Here are example methods or links to example implementations for various languages:

- JavaScript (describes MD5 also): <http://pajhome.org.uk/crypt/md5/>
- Java: <http://commons.apache.org/codec/>
 - You can use `org.apache.commons.codec.digest.DigestUtils` and call `DigestUtils.shaHex(string + salt)`
- PHP: simply call `sha(string + salt)`
 - See <http://php.net/manual/en/function.sha1.php>

Error handling

- Any invalid API calls should return a properly formatted document that contains enough information for the user to be able to determine what the error is.
- It would be good if this returned a reasonable error message as well as a distinct error key for each type of error so that end users could internationalize their applications and provide custom messages based only on the key that we return.
 - i.e. - an invalid request may return an error message of "No conference exists with that token ID" but the error key returned is simply "invalidTokenID".

Desired Future Features

Support for JSON / JSONP

- It would be very nice to optionally allow JSON responses, and to support JSONP. This might allow for simpler integrations, even within static or almost-static webpages using JavaScript as the primary integration language. It should not be assumed that all users will be running custom software on a server and be able to process XML responses, etc.
 - This being said, even within JavaScript there are simple ways to make the API call and

process the returned XML (using jQuery and \$.xml2json, for example)

Meeting event callbacks

This may actually even be called a "reverse API" - where we define an interface that the third-party application can implement to be notified of events. This would not be necessary for the first version of the API, but would be a nice feature for future enhancements. More details:

- When major events happen within meetings, it would be very helpful to provide a way for third-party applications to be notified of these events. For instance, when a user joins a conference, they will presumably be joining through the third-party app. However, when they leave the conference, the app may have certain auditing that it needs to do to record their disconnect time, etc. If BBB could make some callback to the application, this would assist in such scenarios.
- For example, the application may be able to register a URL that BBB would call with status updates. BBB would define an API that such an app would be required to implement at that URL.
 - i.e. - BBB could call:
 - <http://third-party-app/bbb-integ.php?event=userLeft&userID=1234>
 - <http://third-party-app/bbb-integ.php?event=meetingEnded&meetingID=abcd>
- We are already announcing events such as this within BBB through ActiveMQ to communicate between the Red5 apps where these events originate and the API. It would probably not be very difficult to register additional listeners through the Spring Integration configuration that will make these HTTP calls for us for each event.

API Calls

Standard parameters and responses

The following response parameters are standard to every call and may be returned from any call.

Parameters:

Param Name	Required / Optional	Type	Description
checksum	Varies	String	<p>See the "Security" section for more details on the usage for this parameter.</p> <p>This is basically an SHA-1 hash of queryString + securitySalt. The security salt will be configured into the application at deploy time. If a security salt is present, all calls to the API must include the checksum parameter.</p>

Response:

Param Name	When Returned	Type	Description
returnCode	Always	String	Indicates whether the intended function was successful or not. Always one of two values: <i>FAILED</i> – there was an error of some sort – look for the message and messageKey for more information. Note that if the returnCode is FAILED, the call-specific response parameters marked as “always returned” will not be returned. They are only returned as part of successful responses. <i>SUCCESS</i> – the call succeeded – the other parameters that are normally associated with this call will be returned.
message	Sometimes	String	A message that gives additional information about the status of the call. A message parameter will always be returned if the returnCode was FAILED. A message may also be returned in some cases where returnCode was SUCCESS if additional information would be helpful.
messageKey	Sometimes	String	Provides similar functionality to the message and follows the same rules. However, a message key will be much shorter and will generally remain the same for the life of the API whereas a message may change over time. If your third party application would like to internationalize or otherwise change the standard messages returned, you can look up your own custom messages based on this messageKey.

Meeting workflow calls

These calls are the primary calls of the API. They are used for creating and joining meetings.

NOTE: Parameters marked with ** are not yet fully supported. You can pass them in to the controller, but they may not have any actual effect on the conference. This is more a limitation based on support by the actual BBB application more than the API itself.

Create meeting (create)

Example URL: [http://yourserver.com/bigbluebutton/api/create?
\[parameters\]&checksum=\[checksum\]](http://yourserver.com/bigbluebutton/api/create?parameters=&checksum=[checksum])

Parameters:

Param Name	Required / Optional	Type	Description
name	Required	String	A name for the meeting. TODO: Currently there is no way to specify a name for the meeting that is different than the meeting ID / meetme room name. This is a limitation of BBB client and not really the API.
meetingID	Optional	String	A meeting ID that can be used to identify this meeting by the third party application. This is optional, and if not supplied, BBB will generate a meeting token that can be used to identify the meeting.
attendeePW	Optional	String	The password that will be required for attendees to join the meeting. This is optional, and if not supplied, BBB will assign a random password.
moderatorPW	Optional	String	The password that will be required for moderators to join the meeting or for certain administrative actions (i.e. ending a meeting). This is optional, and if not supplied, BBB will assign a random password.
maxParticipants	Optional **	Number	The maximum number of participants to allow into the meeting (including moderators). After this number of participants have joined, BBB will return an appropriate error for other users trying to join the meeting. A negative number indicates that an unlimited number of participants should be allowed (this is the default setting). TODO: This can be supplied to the API currently, but BBB does not actually restrict access yet once the room is full.
meetmeRoom	Optional **	String	The name of the meetme room to use for audio conferencing.

Param Name	Required / Optional	Type	Description
meetmeServer	Optional **	String	<p>The name of the server that will be handling the audio portion of this conference.</p> <p>TODO: This is not currently supported by BBB at all. It would be very nice to allow this to be configurable and not assume that the Asterisk and BBB servers will have a 1:1 relationship. For example, some applications assign the web and audio portions to independent servers based on current load at the time of meeting creation. WebServerA might have two BBB conferences on it, while PhoneServerA and PhoneServerB are handling the actual Asterisk portion of each of those meetings.</p>

Response:

Param Name	When Returned	Type	Description
meetingToken	Always	String	The internal meeting token assigned by the API for this meeting. It can be used by subsequent calls for joining or otherwise modifying a meeting's status.
meetingID	Sometimes	String	The meeting ID supplied by the third party app, or null if none was supplied. If can be used in conjunction with a password in subsequent calls for joining or otherwise modifying a meeting's status.
attendeePW	Always	String	The password that will be required for attendees to join the meeting. If you did not supply one, BBB will assign a random password.
moderatorPW	Always	String	The password that will be required for moderators to join the meeting or for certain administrative actions (i.e. ending a meeting). If you did not supply one, BBB will assign a random password.

Join meeting (join)

Parameters:

Param Name	Required / Optional	Type	Description
fullName	Required	String	The full name that is to be used to identify this user to other conference attendees.
meetingToken	Optional	String	The internal meeting token assigned by the API for this meeting when it was created. Note that <i>either</i> the meetingToken or the meetingID along with one of the passwords <i>must</i> be passed into this call in order to determine which meeting to find.
meetingID	Optional	String	If you specified a meeting ID when calling create, then you can use <i>either</i> the generated meeting token <i>or</i> your specified meeting ID to find meetings. This parameter takes your meeting ID.
password	Optional	String	The password that this attendee is using. If the moderator password is supplied, he will be given moderator status (and the same for attendee password, etc)
redirectImmediately	Optional	Boolean	If this is passed as true, then BBB will not return a URL for you to redirect the user to, but will instead treat this as an entry URL and will immediately set up the client session and redirect the user into the conference. Values can be either a 1 (one) or a 0 (zero), indicating true or false respectively. Defaults to false.

Response:

Param Name	When Returned	Type	Description
entryURL	Always	String	The URL that the user can be sent to in order to join the meeting. When they go to this URL, BBB will setup their client session and redirect them into the conference.

Back-office administration calls

These calls may be used by the back-office applications to handle meeting management, etc.

Is meeting running (isMeetingRunning)

This call enables you to simply check on whether or not a meeting is running by looking it up with either the token or your ID.

Parameters:

Param Name	Required / Optional	Type	Description
meetingToken	Optional	String	The internal meeting token assigned by the API for this meeting when it was created.
meetingID	Optional	String	If you specified a meeting ID when calling create, then you can use <i>either</i> the generated meeting token <i>or</i> your specified meeting ID to find meetings. This parameter takes your meeting ID.

Response:

Param Name	When Returned	Type	Description
running	Always	String	A string of either "true" or "false" that signals whether a meeting with this ID or token is currently running.

End Meeting (endMeeting)

Use this to forcibly end a meeting and kick all participants out of the meeting.

TODO: this call is not yet implemented.

Parameters:

Param Name	Required / Optional	Type	Description
meetingToken	Optional	String	The internal meeting token assigned by the API for this meeting when it was created. Note that <i>either</i> the meetingToken or the meetingID along with one of the passwords <i>must</i> be passed into this call in order to determine which meeting to find.
meetingID	Optional	String	If you specified a meeting ID when calling create, then you can use <i>either</i> the generated meeting token <i>or</i> your specified meeting ID to find meetings. This parameter takes your meeting ID.
password	Required	String	The moderator password for this meeting. You can not end a meeting using the attendee password.

Response:

Param Name	When Returned	Type	Description
ended	Always	String	A string of either "true" or "false" that signals whether the meeting was successfully ended.

List Meeting Attendees (listAttendees)

This call will list all current attendees of a meeting.

TODO: this call is not yet implemented.

Get Meeting Info (getMeetingInfo)

This call will return all of a meeting's information, including the list of attendees as well as start and end times.

TODO: this call is not yet implemented.