Can I use the My Movies Remote Server API from my third party application?

Question:

Can I use the My Movies Remote Server API from my third party remote control application?

Answer:

The My Movies Remote Server API is available to professionals in My Movies. The API is available for third party companies providing remote control applications, and is not intended for individuals personal usage. End-users can with the release of My Movies 5 no longer use the API.

Before integrating the API into your application, you must contact us on support@mymovies.dk, to ensure that we can send you any notices of service or API changes or similar. If there are changes to API functions, they are made so that existing applications will be able to continue their operations.

Important:

With the release of My Movies 5, the Remote Server API have been deprecated for end-users, and is only be available for our professional customers through our installer program.

Notice:

The API is available in all installations of My Movies 5, and was originally made available in My Movies 4.03 versions or later. The API supports control of ArcSoft TotalMedia Theatre 5 when integrated in Windows Media Center, if the "SendKey" command is used for transport controls, as "SendKey" sends a keystroke that ArcSoft TotalMedia Theatre 5 responds to, where direct commands such as "Play", "Pause" or similar directly addresses the playstate in Windows Media Center's SDK, which is not the same when the ArcSoft player is foreground.

Availability:

Can I use the My Movies Remote Server API from my third party application?

The API is made available 5-15 seconds after Windows Media Center is started, either on a Host PC, or on a Windows Media Center extender connected to the Host PC. The 5-15 seconds is the time it takes for Windows Media Center to launch background services, and the time depends on the amount of softwares installed on the PC.

Once started, the API can be accessed through http://127.0.0.1:51408/, where "51408" is the port of the Host PC, and extenders will get numbers 51409 (MCX1), 51410 (MCX2) and up to 51413 (MCX5). These ports are opened in the Windows Firewall when My Movies is installed, however notice that they are registered to be opened on "Home Network" or "Work Networks" only, which means that they are not open if the machine is configured to be on a public network. Similar, if a third party firewall or security software is used on the Host PC, these ports must be opened manually.

The remote server advertises itself through Apple Bonjour services, with the type id "_mymoviesremoteserver._tcp", including the name of the machine and the port number of the service. The name of the machine can be changed through the API itself.

Commands can be sent to the API such as http://127.0.0.1:51408/?command=SendKey&key=Up&username=user&password=pass.

The following informations must be sent with all requests:

"command". This specifies the command you would like to call.

"username" (alternative to apikey). This is the users username as specified in your application settings, which must match the username used in My Movies on the Host PC. The requirement of this ensures that the remote server is not fully open.

"password" (alternative to apikey). This is the users password as specified in your application settings, which must match the password used in My Movies on the Host PC. The requirement of this ensures that the remote server is not fully open.

"apikey" (alternative to username/password). If you do not want the users to log into your remote application with their My Movies user account, you can as an alternative ask them to enter an API key. This is typically used by custom installer installations where a user account is optional. The API key, which is a Unique generated string can be found in "Collection Management", "Tools", "Options", "Remote Control Server" in My Movies for Windows, or in "Remote Control Server" settings in the My Movies Home and Essentials Server Solution software package.

The API always returns with an XML file contaning a response node, such as:

<Response status="ok"></Response>

or

<Response status="failed">Message from server.</Response>

The server can return the following statuses:

"ok". The command was received and processed as expected. This is the default response. Depending on the command, the Response node may contain either XML or clear text inside for the result.

"failed". The command failed for one or more reasons. The reason is included as clear text inside the Response node.

There can be other return values on specific commands, which will be documented with the command.

The following API commands are available (the parameters mentioned are additional to the three mentioned above):

Ping - This command returns "pong", and can be used to check if the service is available.

Version - This command returns the version of the My Movies application on the Host PC. This can be used if API commands are added in later versions, and your application requires a specific version. The version is returned in the format "W.X.Y.Z", where "W" specifies the major version, "X" the first digit of the minor version, "Y" the second digit of the minor version, and "Z" specifies the Pre Release version, or build number on released versions. A "Z" version below 100 is a "Pre Release" version, with 1 being "Pre Release 1", 2 being "Pre Release 2" and so on. A

"Z" version of "100" or higher indicates a final release version, with 100 being the initial final release, 102 being a "Build 2" and so on. As an example "4.0.2.100" specifies a final "4.02" version, where "4.0.3.2" specifies "4.03 Pre Release 2".

Rename(name) - Renames the instance name advertised through Bonjour, which means that the default machine name can be changed to "Livingroom" or similar.

ResetName - Resets the instance name advertised through Bonjour to it's default value, which is the machine name.

Mute(state) - Changes the mute state. The state parameter is optional, and if not provided, the mute state is changed from it's current value. The optional "state" parameter can be "on" or "off" specifically setting the mute value, no matter what the current state is. The command will return "Mute On" or "Mute Off" which is the value of the new mute state.

VolumeUp - Ups the volume level by one. The command will return the new value of the volume setting.

VolumeDown - Lowers the volume level by one. The command will return the new value of the volume setting.

Stop - Stops current playback.

Pause - Pauses playback.

Play - Resumes paused playback.

PlayOrPause - Changes the state of playback to be either paused or resumed, based on what the current state is.

SkipForward - Invokes the skip-forward action on the current media, moving the playback forward by 29 seconds. For music files, it moves the playback to the end of the file.

SkipBack - Invokes the skip-back action on the current media, moving the playback back by seven seconds. For music files, it moves the playback to the beginning of the file**.**

FastForward (playrate) - Invokes fast forward on the current media. The playrate must be specified from 1 to 3, where 1 is fast, 2 is faster and 3 is fastest.

Rewind(playrate) - Invokes rewind on the current media. The playrate must be specified from 1 to 3, where 1 is fast, 2 is faster and 3 is fastest.

SendKey(key) - Sends a named key as if it was pressed on a remote control. Notice that some keys are duplicates of what is found as individual commands, in which relations it is recommend to use the SendKey command as this allows control of third party integrated players that respond to standard Windows Media Center remote control commands.

Value options for "key" parameter:

"Stop"

"Play"

"Pause"

"Record"

"FastForward"

"Rewind"

"Skip" or "SkipForward"

"Replay" or "SkipBack"

"Back"

"Up"

"Down"

"Left"

"Right"

"OK"

"MoreInfo"

"VolumeUp"

"VolumeDown"

"ChanOrPageUp"

"ChanOrPageDown"

"GreenButton" - Notice! On an original IR based remote, the green button can be used to launch Windows Media Center. Since the My Movies remote server does not run until Media Center is launched, it can not be used for this purpose.

"Mute"

"RecordedTV"

"Guide"

"LiveTV"

"DVDMenu"

"0" to "9".

"#"

"Clear" - Notice! There is no way to replicate the "Clear" button. Instead, backspace is sent. This also means that if input box is not in focus, it functions as "Backspace", which a proper "Clear" button call would not.

"Backspace"

"Enter"

"Teletext" - Notice! There is no way to replicate the "Teletext" key. Instead, "Up" and "Left" is sent, to go into the popup menu as a workaround. This may or may not work, depending on the current view experience.

"Red" - Notice! Unavailable as the key cannot be replicated - it is listed to avoid confusion on why it is not available.

"Green" - Notice! Unavailable as the key cannot be replicated - it is listed to avoid confusion on why it is not available.

"Yellow" - Notice! Unavailable as the key cannot be replicated - it is listed to avoid confusion on why it is not available.

"Blue" - Notice! Unavailable as the key cannot be replicated - it is listed to avoid confusion on why it is not available.

"DVDAudio"

"DVDSubtitle"

"MovieLibrary"

"PictureLibrary"

"VideoLibrary"

"MusicLibrary"

"TV"

"Zoom"

"Start" or "Home"

TypeKey(key) - This command types a letter as if it was being typed with a keyboard.

NavigateTo(pageid, parameters) - Navigates to the page id specified in "pageid" parameter, with optional parameters specified in "parameters" parameter.

Value options for "pageid" parameter:

{ae804109-f3a3-4fe0-8379-518463da7395} - ExtensibilityEntryPoint - Parameters must contain GUIDs that identify the destination page in the format "applicationGuid\entrypointGuid" or "entrypointGuid".

{3798c7a7-c31f-46c4-9782-655f64b5920f} - ExtensibilityURL - Parameters must contain the URL of the destination page (Hoted HTML page).

{ec5a86e5-256a-4316-b294-42d5b86a5bed} - FMRadio

{7f58b0a0-e85b-4e1e-8726-5f1ba279dd3c} - InternetRadio

{6c79b00a-8e02-400f-a6f9-06380d1ddecd} - LiveTV

{5cbc749d-bad7-4dad-b302-1628882d4504} - ManageDisks

{798b6732-ebbe-4b81-b424-d93bf91b0aad} - MorePrograms

{0b2357d5-5110-4dcb-b8e7-a4bb6b48bc9e} - MovieLibrary

{fd331980-26ab-4bac-889f-188e13001a55} - MusicAlbums

{c4a709f2-4c53-4db4-95eb-d190a95fcfcf} - MusicArtists

{5eaf585f-4273-4ce1-8284-bd60288b88d9} - MusicSongs {d69fa617-1035-4875-ac6e-ca7bdef8c7b4} - MyMusic

{3b7ff955-01f6-4f68-9775-4d1216b777b7} - MyPictures - Parameter can contain path to folder.

{80e2aa59-4c45-493c-9b75-68bb9ec80617} - MyTV

{a5abe790-14f9-4075-87a8-e4a940d632ae} - MyVideos - Parameter can contain path to folder.

{fdd1b548-c454-4db9-b702-6d96c4675af2} - Photo details - Parameter can contain path to the photo.

{42c5a65e-35d4-48c8-8828-df636825e091} - RecordedTV

{dcfa3c0a-d146-4e33-995a-565f8ca1a859} - RecorderStorageSettings

{ae33979e-6334-45f6-8490-6e05f4032703} - ScheduledTVRecordings

{1436526c-329d-47dd-9afa-458aab0da6e5} - SlideShow - Parameter can contain path to folder.

{79f21c24-a3a2-4268-9caa-5e9ef241c6a5} - SlideshowSettings

{b8eac38a-7fb8-4559-84b5-42999c6864bd} - Start

{21e33f7d-c860-49ff-bd74-a94dab62b434} - TVGuide

{5b721b57-872d-4ad2-8262-4d6bb2c09c6a} - Visualizations - Parameter must contain visualizationName:preset.

{0043FBA8-91E0-4d7d-92E4-5C4C1416B21F} - WebAddIn - Parameter must contain URL of the destination page.

PlayMyMoviesTitle(webserviceid) - Starts playback request for a title in My Movies. The "webserviceid" specified the online service id for the title, which is available through the "mymovies.xml" files once decrypted, or from the internal database. The command responds with either "selectoption" as status, or "failed" is no playback possibilities where found for the title, which includes a message in the response node. The "selectoption" response will give a list of disc playback options, such as "<Response status="selectoption"><Disc Number="1" Name="Disc 1"><Disc Number="2" Name="Disc 2"></Response>". If there is only one disc this is forwarded directly to "PlayMyMoviesDisc", otherwise, the user is presented with the playback options, and the one the user selects is passed to "PlayMyMoviesDisc".

PlayMyMoviesDisc(webserviceid, discid) - Invokes playback for the title specified with the "webserviceid" parameter, with the number of the disc received by the "PlayMyMoviesTitle" call into the "discid" parameter.

GetMusicArtists (sortby, sortdirection, filterby, filtervalue, filteroperator) - Returns a list of music artist in the library. The parameters "sortby", "sortdirection", "filterby", "filtervalue" and "filteroperator" is optional. If the "filterby" or "filtervalue" parameters is defined, it must contain the exact same number of elements.

The possible values for the parameters are listed below:

sortby - Possible values: artist (default), name (same as artist)

sortdirection - Possible values: asc (default), desc

filterby - Possible values: genre, artist, album, title, id, tracknumber, composer. Multiple values can be added by seperating by a pipe, such as "artist|album".

filtervalue - Contains the string(s) that will be used to filter. Multiple values can be added by seperating by a pipe, such as "Michael Jackson|Bad".

filteroperator - Possible values: contains (default), notcontains, notequals, notbeginswith, equals, beginswith, lessthanorequals, lessthan, greaterthanorequals, greaterthan.

GetMusicAlbums (sortby, sortdirection, filterby, filtervalue, filteroperator) - Returns a list of music albums in the library. The parameters "sortby", "sortdirection", "filterby", "filtervalue" and "filteroperator" is optional. If the "filterby" or "filtervalue" parameters is defined, it must contain the exact same number of elements.

The possible values for the parameters are listed below:

sortby - Possible values: album (default), title (same as album)

sortdirection - Possible values: asc (default), desc

filterby - Possible values: genre, artist, album, title, id, tracknumber, composer. Multiple values can be added by seperating by a pipe, such as "artist|album".

filtervalue - Contains the string(s) that will be used to filter. Multiple values can be added by seperating by a pipe, such as "Michael Jackson|Bad".

filteroperator - Possible values: contains (default), notcontains, notequals, notbeginswith, equals, beginswith, lessthanorequals, lessthan, greaterthanorequals, greaterthan.

GetMusicTracks (sortby, sortdirection, filterby, filtervalue, filteroperator) - Returns a list of music tracks in the library. The parameters "sortby", "sortdirection", "filterby", "filtervalue" and "filteroperator" is optional. If the "filterby" or "filtervalue" parameters is defined, it must contain the exact same number of elements.

The possible values for the parameters are listed below:

sortby - Possible values: genre, artist, album, title, id, tracknumber (default), composer

sortdirection - Possible values: asc (default), desc

filterby - Possible values: genre, artist, album, title, id, tracknumber, composer. Multiple values can be added by seperating by a pipe, such as "artist|album".

filtervalue - Contains the string(s) that will be used to filter. Multiple values can be added by seperating by a pipe, such as "Michael Jackson|Bad".

filteroperator - Possible values: contains (default), notcontains, notequals, notbeginswith, equals, beginswith, lessthanorequals, lessthan, greaterthanorequals, greaterthan.

GetMusicTrackArtwork (id) - Returns a JPEG file for the track defined by the "id" parameter. The "id" parameter is returned from the GetMusicTracks call. If artwork is not found for the track, an empty response is returned.

CurrentMediaInformation - Returns information about the current media playing.

PlayMusicTrack (id) - Plays the music track defined by the "id" parameter. The "id" parameter is returned from the GetMusicTracks call. The PlayMusicTrack command cleares any current playback queue.

QueueMusicTrack (id) - Adds the music track defined by the "id" parameter to the playback queue. The "id" parameter is returned from the GetMusicTracks call. If no media is currently played, this command will start playback of the track specified by "id".