

Guile3d API functions

which you can use in VoxCommando with the Scrape action

Important: *Green* will show you the part which you can change for your purposes.

Ask for authentication key to get access to the guile3d-bot:

To get access to sending commands from Voxcommando to the 3d-bot, you need an Authentication Key which looks like: ABCD1234EFKJ5678XYZ - this key is changing each time when the guile3d software was terminated - for example when you do a full shutdown of your PC.

```
http://127.0.0.1:8000/Kernel?function=AskAuthKey&Login=YourGuile3dUsername&Password=YourGuile3dPassword
```

response: **Authentication Key** - if success
FALSE - if no success

After sending this command, you will get as response the KEY, it will showing up in the history window in VoxCommando and it is also stored in {LastResult}. If you get FALSE as response, check the IP-address, username and password again and make sure that everything is correctly written.

I've automated this process by creating a „connect“ command which overwrite the key with the new value when you execute it (by an event or speech). This command contain the following 3 actions:

1. Map.CreateTable

parameter: **botpass** (this action create a MapTable with name **botpass**)

2. Scrape

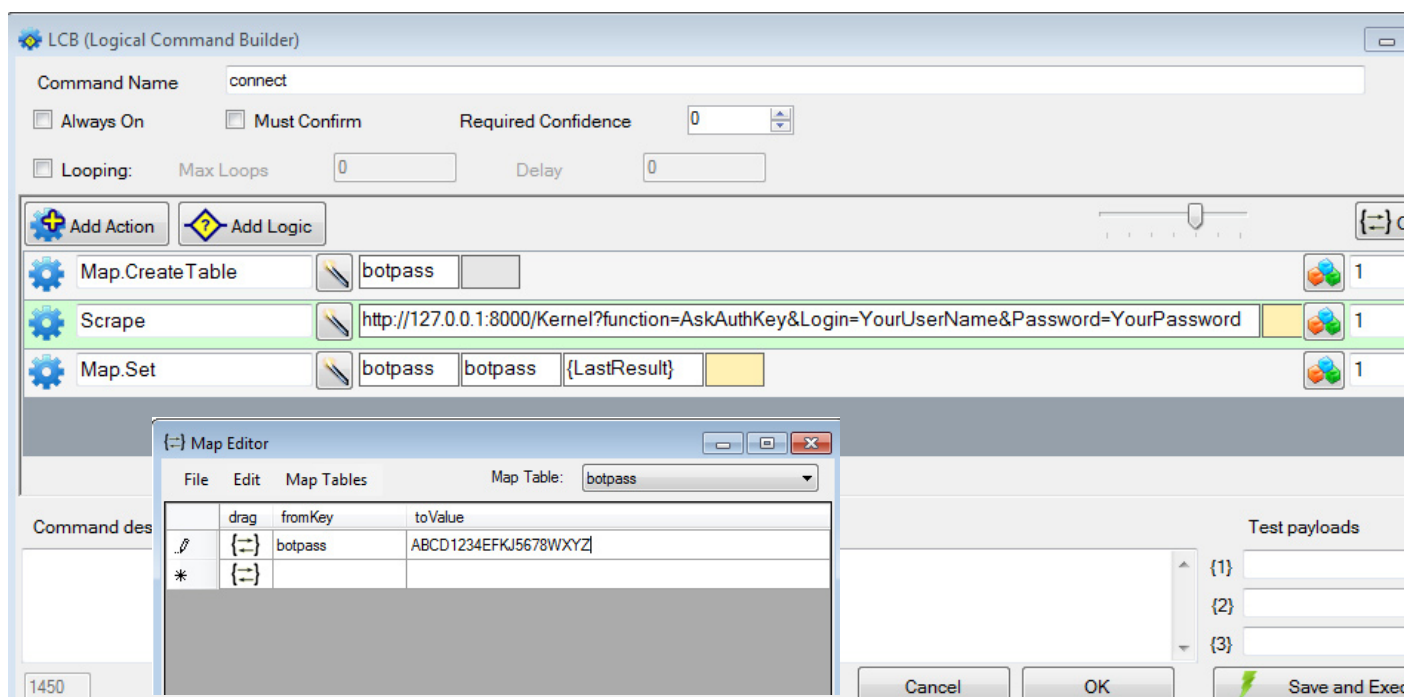
parameter: **http://127.0.0.1:8000/Kernel?function=AskAuthKey&Login=YourGuile3dUsername&Password=YourGuile3dPassword**

3. Map.Set

parameter 1: name of the MapTable from above

parameter 2: name of the key in the Maptable (you can give the key any name it does not matter, because we have only one key stored in the table).

parameter 3: the value of the key (this is the KEY which you get as response from the Scrape-action).



Send data to the bot, which speaks it aloud:

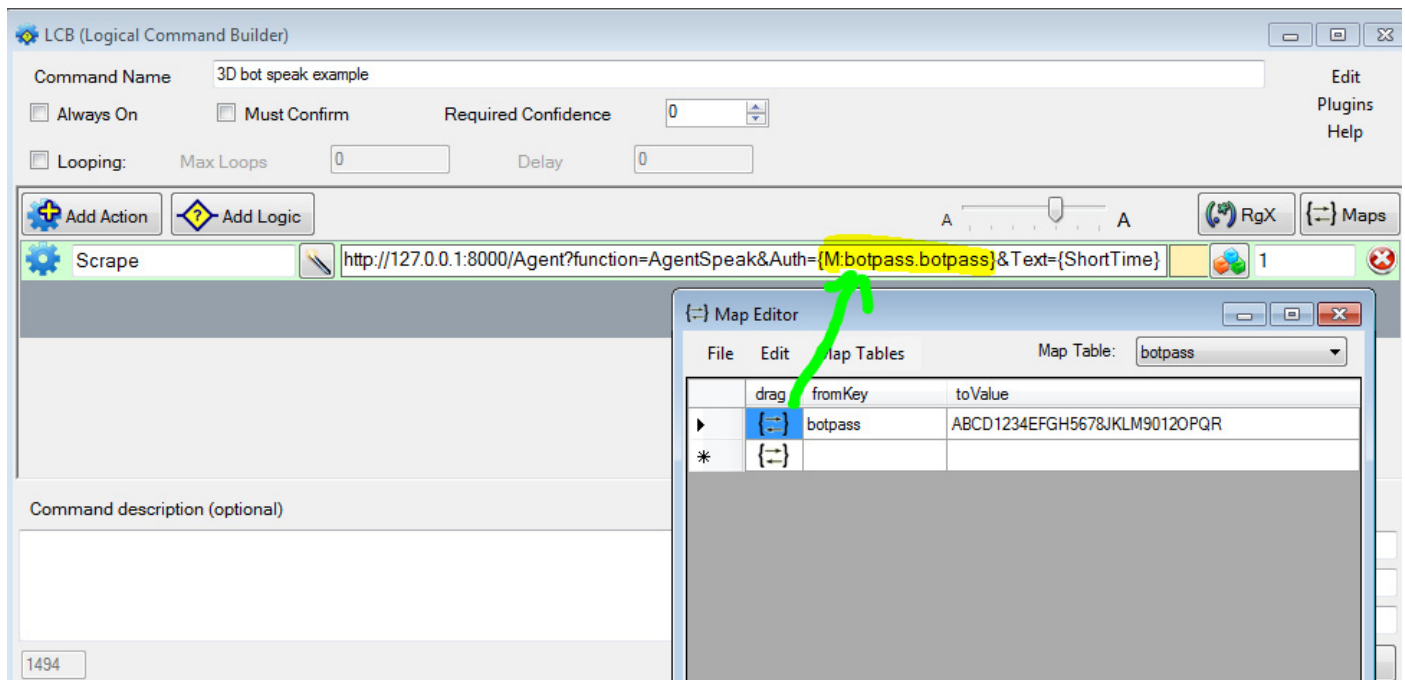
This scrape action send the text or value from VoxCommando to the bot.

http://127.0.0.1:8000/Agent?function=AgentSpeak&Auth={M:botpass.botpass}&Text=text which is speak aloud

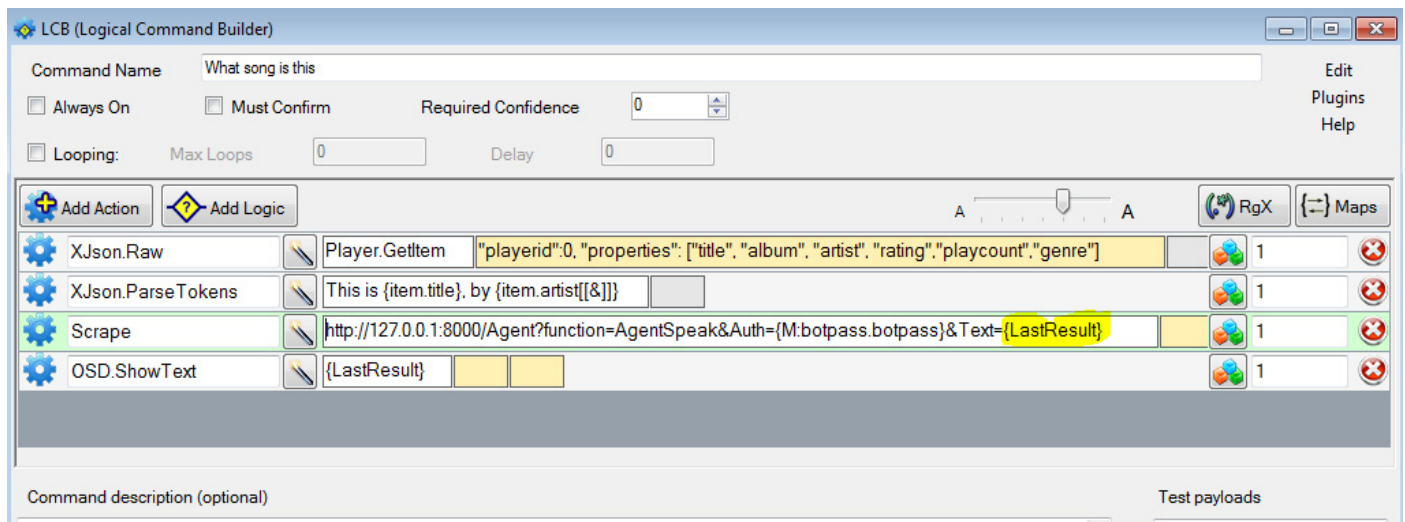
How you can see, this command use the **MapTable** that contain the Authentication Key from the command which we have created before. You can insert the **Authentication Key** by drag'n drop it directly from the MapTable. Put at the end of the action parameter what you will hear speaking aloud from the bot (after the **&Text=**).

This can be for example: **text**, **{LastResult}**, **{ShortTime}** or **any other data**

The example command show you, how the bot speak aloud the current time by insert the {ShortTime} variable at the end of the action parameter.



If you want use this command as example for XBMC to get the current playing song name, you can use this scrape action as following:



The data from the XJson.Raw action is stored in {LastResult}, so we must only insert the variable {LastResult} at the end and after this command is executed, the bot speak the song title and artist aloud. You can use this command exact in this way for all other „get now playing infos“ in VoxCommando for XBMC.

Conclusion: This scrape action can be used instead of the TTS.Speak action in VoxCommando, but not as TTS.SpeakSync, because VoxCommando doesn't wait to continuing next actions which following after the scrape command - **so it makes not sense to use two or more of this scrape actions in one command**. With the TTS.SpeakSync action, VoxCommando will wait until speaking is complete before continuing to the next action.

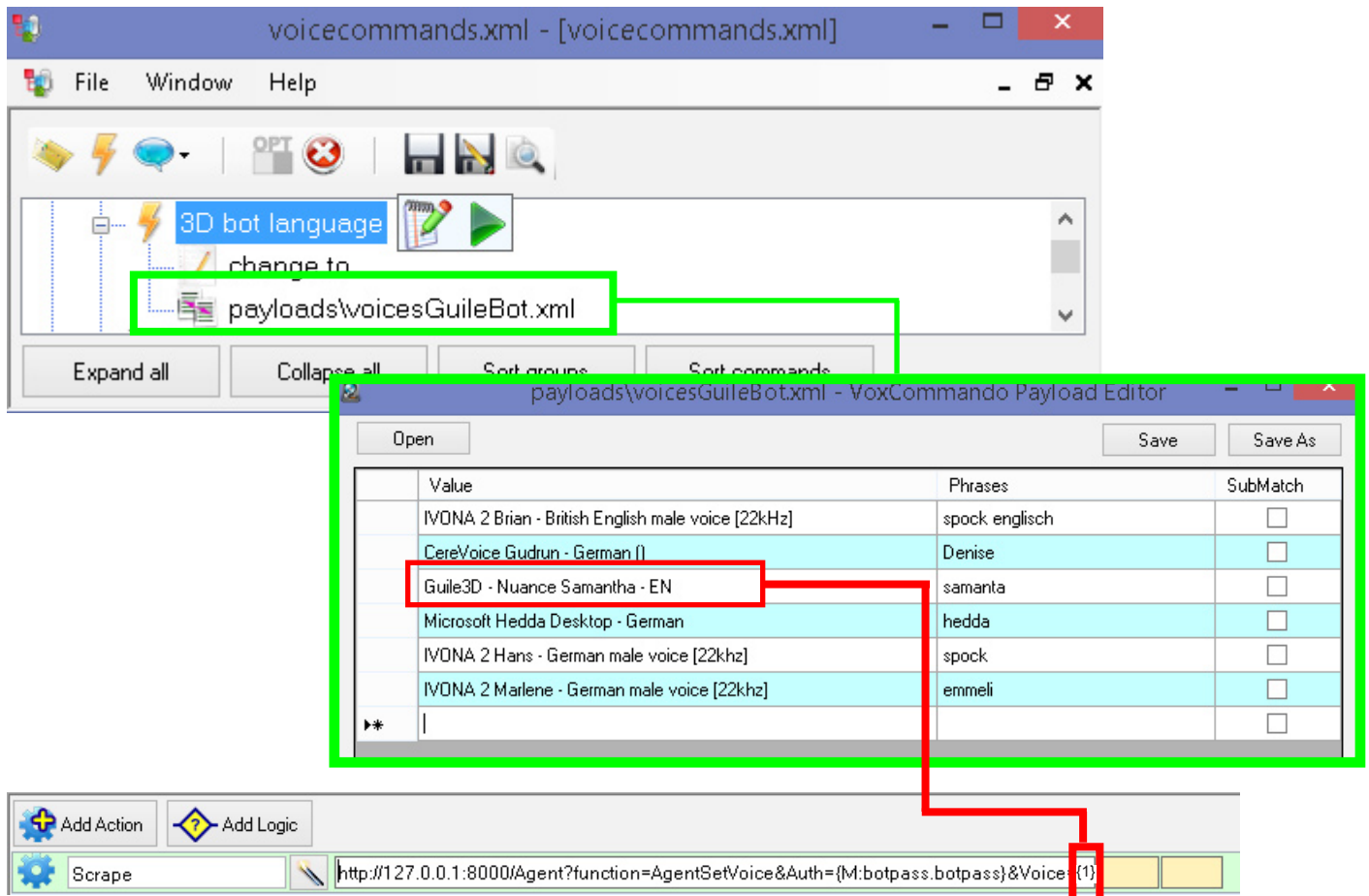
Changing the bot voice:

This scrape action in VoxCommando change the voice of the bot (not the TTS voice in VoxCommando)

http://127.0.0.1:8000/Agent?function=AgentSetVoice&Auth={M:botpass.botpass}&Voice=existing TTS Voice name

Get safe that you type the voice name exactly as is written in your system, otherwise it will not work!

You can use this action perfect with a payload or MapTable in VoxCommando which has stored the voice names.



In this way you need only to say the phrase which is stored and affiliated with the value in the payload. Here in the example above „change to samanta“ is samanta affiliated to value **Guile3D - Nuance Samantha - EN**. This action makes sense to use it with a other bot face like „Spock“, so you can change (for example) to male voice when you have changed the bot to a male character and this is the next step.

Changing the bot character:

This scrape action in VoxCommando change the character of the bot (if you have others then Denise)

http://127.0.0.1:8000/Agent?function=AgentSetCharacter&Auth={M:botpass.botpass}&Character=character name

You will find the character name in the Guile3D Software settings in „Assistant“

Set the bot position on the screen:

This scrape action in VoxCommando move the bot to a specified position on the screen.

http://127.0.0.1:8000/Agent?function=AgentPosition&Auth={M:botpass.botpass}&left=number&top=number

For example: &left=300&top=500 move the bot to position 300px from left and 500px from top of your screen

Set the bot size on your screen:

This scrape action in VoxCommando set the bot to a specified size.

http://127.0.0.1:8000/Agent?function=AgentResize&Auth={M:botpass.botpass}&width=number&height=number

For example: &width=300&height=300 resize the bot to 300x300px on your screen.

Show/hide the bot on your screen:

This scrape action in VoxCommando show or hide the bot on your screen.

http://127.0.0.1:8000/Agent?function=AgentShow&Auth={M:botpass.botpass}&STATE=STATE parameter

Parameters for **STATE**:

WSMAXIMIZED show the bot on the screen.

wsMinimized hide the bot

wsNormal - not tested yet -

Set the bot resolution:

This scrape action in VoxCommando set the bot to a specified resolution (0,1,2).

http://127.0.0.1:8000/Agent?function=AgentResolution&Auth={M:botpass.botpass}&Resolution=parameter

Parameters for **Resolution**:

0 correspond to 800x600px

1 correspond to 1024x768px

2 correspond to 1280x800px

Set the bot quality:

This scrape action in VoxCommando set the bot quality.

http://127.0.0.1:8000/Agent?function=AgentQuality&Auth={M:botpass.botpass}&Quality=parameter

Parameters for **Quality**:

LOW correspond to *poor quality*

HIGH correspond to *good quality*

SUPERHIGH correspond to *best quality*

Lock/unlock the bot at the screen position:

This scrape action in VoxCommando can lock or unlock the bot position at the screen. If you lock the bot, you can't move the bot with the mouse on the screen.

http://127.0.0.1:8000/Agent?function=AgentLockPosition&Auth={M:botpass.botpass}&lock=parameter

Parameters for **lock**:

True lock the bot

False unlock the bot

Bot stay on top:

This scrape action in VoxCommando will let stay your bot on the top over a window at any time.

http://127.0.0.1:8000/Agent?function=AgentStayonTop&Auth={M:botpass.botpass}&onTop=parameter

Parameters for **onTop**:

True let stay the bot in front of a window at any time.

False the bot will stay in the background at any window which has focus.

Chat with the bot:

This scrape action in VoxCommando let chat with the bot in the same way as when you type text in the chatbox of your bot. You can use this action in a command together with payload dictation.

```
http://127.0.0.1:8000/Chat?function=ChatInput&Auth={M:botpass.botpass}&Text=text to talk with your bot
```

Parameters for Text:

If you use it with payload dictation in VoxCommando, set as parameter: {1}

In this way you can have the following command structure in VoxCommando:

Phrase: *please, tell me*

payloadDictation: Regular <- this is then payload {1} in the parameter above.

So you can speak to your bot: „tell me“ - „how far is it from earth to the moon“

bot stop to talk:

This scrape action in VoxCommando let the bot stop talking.

```
http://127.0.0.1:8000/Agent?function=AgentStop&Auth={M:botpass.botpass}
```

There are 5 other functions in the Guile3D-API available, but not significant to use it with VoxCommando.