Local Skills - HelloWorldpart5_RecordAudio

In part 5 of our *Hello World SDK* tutorial we'll discover how to record audio and play it back to the user. This involves two very simple commands: `StartRecordingAudio()` and `StopRecordingAudio()`.

HelloWorldpart5_RecordAudio.json

```
{
      "Name": "HelloWorldpart5_RecordAudio",
      "UniqueId": "2697a7f3-0cc5-4180-bb03-84bae827e751",
      "Description": "Local 'Hello, World!' tutorial series, part 5.",
      "StartupRules": [ "Manual", "Robot" ],
      "Language": "javascript",
      "BroadcastMode": "verbose",
      "TimeoutInSeconds": 300,
      "CleanupOnCancel": false,
      "WriteToLog": false
}
```

HelloWorldpart5_RecordAudio.js

Start by calling `StartRecordingAudio()` to tell the microphone to start recording (pass in the name you want to assign to the resulting audio clip). Use `Pause()` to establish a duration for how long you want the recording to last, then call `StopRecordingAudio()` to halt the recording process and save the clip. Call `Pause` again for `2000ms` to give Misty time to save the recording.

```
misty.StartRecordingAudio("RecordingExample.wav");
misty.Pause(5000);
misty.StopRecordingAudio();
misty.Pause(2000);
```

Once the clip has been saved we want to check that the recording was saved correctly. To do this, call `GetListOfAudioFiles()`. As mentioned in previous tutorials, the callback function (automatically named `_<command>`) will run once the data is ready to be received.

```
misty.GetListOfAudioFiles();
function _GetListOfAudioFiles(data) {}
```

Now that we have the data, we want to check that our recording shows up on the list. Once way to do this is to create a `boolean` variable to designate if the list contains the file, then loop through the list and check if the name of any of the audio files match the one for our recording. If it does, change the boolean from `false` to `true`. **Note:** This logic also needs to be contained within the callback as we are using the data received from our GET request.

```
let containsNewFile = false;
for (let i = 0; i < audioArr.length; i++) {
   if (audioArr[i].Name === "RecordingExample.wav") {
      containsNewFile = true;
   }
}
```

If the list contains the recording, we can call `PlayAudioClip()` to play the recording and end the program. Otherwise, use `Debug()` to display an error message, as something went wrong with the process.

```
if (containsNewFile) {
   misty.PlayAudioClip("RecordingExample.wav", 500, 500);
} else {
   misty.Debug("file was not found");
}
```

See the complete file below for reference.

```
// Debug message to indicate the skill has started
misty.Debug("starting skill helloworld part5");

// Send commands to start recording audio, pause for  five seconds
// to record, then stop recording audio
misty.StartRecordingAudio("RecordingExample.wav");
misty.Pause(5000);
misty.StopRecordingAudio();

// give Misty time to save the recording
misty.Pause(2000);

// Send request to fetch list of audio files
misty.GetListOfAudioFiles();
```

```
// Define the callback for request
function _GetListOfAudioFiles(data) {
      // Grab array of audio files
      let audioArr = data.Result;

      // Initialize a variable to tell us if the list contained the audio file
      let containsNewFile = false;

      // Loop through list, compare file names to name specified for recording
      for (let i = 0; i < audioArr.length; i++) {
            if (audioArr[i].Name === "RecordingExample.wav") {
                  // If there's a match, track it by updating boolean
                  containsNewFile = true;
            }
      }

      // If list contains recording, send command to play recording
      if (containsNewFile) {
            misty.PlayAudioClip("RecordingExample.wav", 500, 500);
      } else {
            // Log error message
            misty.Debug("file was not found");
      }
}
```