

Google TTS – Verwendung der Skripte

(Anleitung für Linux-Systeme)

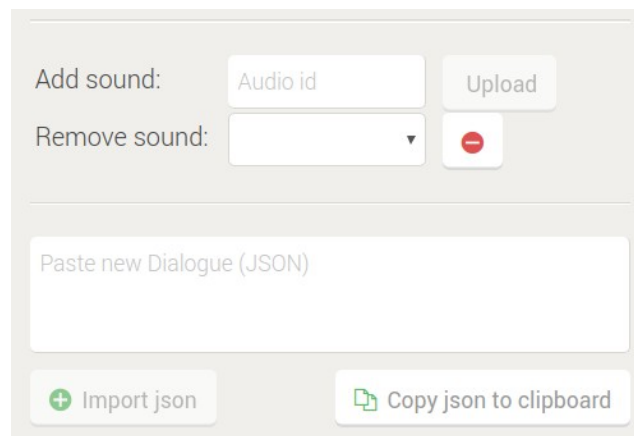
Audio-Dateien können aus dem Locandy-System in 4 Schritten erstellt werden. Dazu generiert ein Skript aus einer JSON-Datei die gesprochenen SSML-Dateien. Diese können dann nach belieben angepasst werden, um Sprechpausen, Betonungen, etc. anzupassen. Mit der google-tts API kann dann ein weiteres Skript die Audio-Dateien aus den SSML-Dateien generieren.

1. Exportieren der JSON-Datei

Für die Dialog-Texte und Effekt-Texte muss die JSON-Datei jeweils unterschiedlich exportiert werden:

Dialog-Texte:

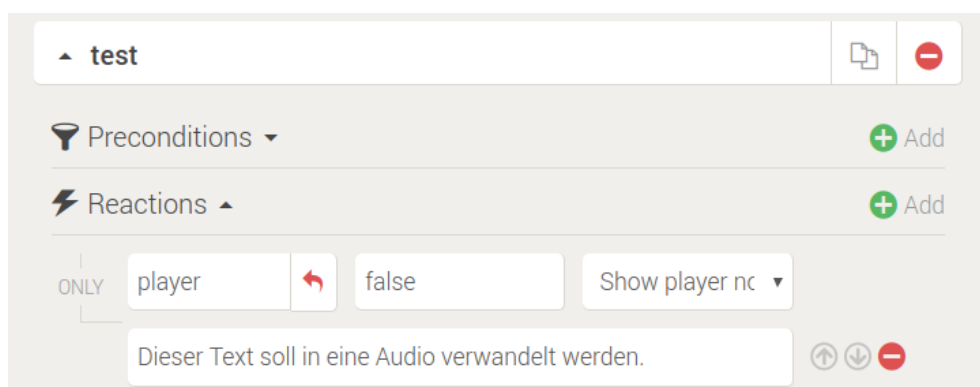
Mit dem Button „Copy json to clipboard“ kann die JSON-Datei des gesamten Dialoges kopiert und über einen beliebigen Editor in eine .json-Datei gespeichert werden.



The screenshot shows a web interface for managing JSON files. It includes a section for adding and removing sounds with input fields for 'Audio id' and 'Upload', and a 'Remove sound' button with a red minus icon. Below this is a large text area labeled 'Paste new Dialogue (JSON)'. At the bottom, there are two buttons: 'Import json' with a green plus icon and 'Copy json to clipboard' with a green document icon.

Effekt-Texte:

Die zu übersetzenden Texte müssen als Effekte „Show Player Notification and STOP! (no value)“ im Locandy-System zuvor erstellt worden sein.



The screenshot shows a web interface for editing a reaction effect. At the top, there is a header bar with a dropdown menu set to 'test' and two icons (a document and a red minus). Below this, there are sections for 'Preconditions' and 'Reactions', each with a green plus icon and the word 'Add'. Under the 'Reactions' section, there is a row of controls: a dropdown menu set to 'ONLY', a text input field containing 'player', a red circular arrow icon, a text input field containing 'false', and a dropdown menu set to 'Show player nc'. At the bottom, there is a large text area containing the text 'Dieser Text soll in eine Audio verwandelt werden.' and three icons (a green up arrow, a green down arrow, and a red minus).

Die JSON-Datei für die Effekte einer gesamten Quest-Station kann nun über die URL-Leiste aufgerufen werden.

Dazu einfach die Quest zum Bearbeiten aufrufen und in der URL-Leiste „/mrm.json“ anhängen. Danach kann der gesamte Text kopiert und über einen beliebigen Editor in eine .json-Datei gespeichert werden.

Bsp.:

```
https://www.locandy.com/q/at-enterbionics/edit/mrm.json
```

2. Extrahieren der SSML-Dateien aus der JSON-Datei

Die JSON-Datei aus Schritt 1 kann nun in einem Skript verwendet werden, um die SSML-Dateien zu generieren. Auch hier muss die JSON-Datei der Dialog-Texte und die der Effekt-Texte jeweils separat bearbeitet werden:

Bevor das Skript ausgeführt werden kann, muss im Terminal der Ordner geöffnet werden. Danach kann das Skript mit „./name.sh“ ausgeführt werden. Als Argument muss die JSON-Datei übergeben werden.

Falls der Befehl jq noch nicht installiert ist, kann er folgendermaßen installiert werden:

```
sudo apt-get install jq
```

Dialog-Texte:

Das Skript „dialogueJsonToSsml.sh“ wird für die json-Datei der Dialoge verwendet.

Bsp.:

```
qe@QE-T450s:~/Desktop/Manual$ ./dialogueJsonToSsml.sh station2.json
ENDE
START
experiment_materials
lab_journal
leaf_bud
miura_beispiele_aus_der_natur
miura_email
miura_ende
miura_folding
miura_folding_help
miura_folding_mad
nasa_message
nasa_message_questions
normal_folding
origami
prototyp_falten
prototyp_finale
prototyp_lösung
prototyp_quadrat
```

In diesem Beispiel wird im Ordner „Desktop/Manual“ das Skript „dialogueJsonToSsml.sh“ mit dem Argument „station2.json“ ausgeführt. Nach erfolgreichem Ausführen sind die generierten SSML-Dateien im gleichen Ordner erstellt worden.

Effekt-Texte:

Das Skript „mrmJsonToSsml.sh“ wird für die json-Datei der Effekte verwendet.

Bsp.:

```
qe@QE-T450s:~/Desktop/Manual$ ./mrmJsonToSsml.sh effects.json
jq: error: syntax error, unexpected INVALID_CHARACTER, expecting $end (Unix shell quoting issues?) at <top-level>, line 1:
.effects.S2_prototyp_lösung_aw1.effects[0][4]
jq: 1 compile error
jq: error: Invalid numeric literal at EOF at line 1, column 3 (while parsing '.0.') at <top-level>, line 1:
.effects.0.effects[0][4]
jq: error: syntax error, unexpected LITERAL, expecting $end (Unix shell quoting issues?) at <top-level>, line 1:
.effects.0.effects[0][4]
jq: 2 compile errors
jq: error: Invalid numeric literal at EOF at line 1, column 4 (while parsing '.20.') at <top-level>, line 1:
.effects.20.effects[0][4]
jq: error: syntax error, unexpected LITERAL, expecting $end (Unix shell quoting issues?) at <top-level>, line 1:
.effects.20.effects[0][4]
jq: 2 compile errors
jq: error: syntax error, unexpected '+' (Unix shell quoting issues?) at <top-level>, line 1:
.effects.scene++.effects[0][4]
jq: 1 compile error
jq: error: syntax error, unexpected '+' (Unix shell quoting issues?) at <top-level>, line 1:
.effects.scene++.effects[0][4]
jq: 1 compile error
jq: error: syntax error, unexpected '+' (Unix shell quoting issues?) at <top-level>, line 1:
.effects.scene++.effects[0][4]
jq: 1 compile error
jq: error (at effects.json:0): Cannot index object with number
```

In diesem Beispiel wird im Ordner „Desktop/Manual“ das Skript „mrmJsonToSsml.sh“ mit dem Argument „effects.json“ ausgeführt. Nach erfolgreichem Ausführen sind die generierten SSML-Dateien im gleichen Ordner erstellt worden.

Die Fehlermeldungen stammen von anderen Effekten, die nicht übersetzt werden sollen. (zum Beispiel scene++ oder openURL) Diese können ignoriert werden.

3. Konvertieren der SSML-Dateien in MP3-Dateien

Die SSML-Dateien können nun mit dem Skript „gtts.sh“ in MP3-Dateien übersetzt werden.

Als Argumente müssen folgende Informationen übergeben werden:

- -k GOOGLE_API_KEY
Mit -k muss der Api-Key übergeben werden
- -p -5
Mit -p kann der pitch-Wert definiert werden. Standardwert: -5
- -r 0.9
Mit -r kann die Speed-Rate definiert werden. Standardwert: 0.9
- -i nameDerDatei.ssml
Mit -i muss die input-Datei (SSML-Datei) übergeben werden.
- Weitere Argumente wie Gender, Language, Voice, ... können übergeben werden. Siehe Code für weitere Informationen

Das Skript kann mit den entsprechenden Flags folgendermaßen verwendet werden:

```
./gtts.sh -k GOOGLE_API_KEY -p -5 -r 0.9 -i nameDerDatei.ssml
```

Beispiel:

```
qe@QE-T450s:~/Desktop/Manual$ ./gtts.sh -k API-KEY -p -5 -r 0.9 -i S4_START_aw1.ssml
Reading SSML from file: S4_START_aw1.ssml
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   Dload  Upload  Total    Spent    Left   Speed
100 812k    0 812k  100  425  1030k    539  --:--:-- --:--:-- --:--:-- 1031k
```

Hier wurde aus der Datei „S4_START_aw1.ssml“ die Datei „S4_START_aw1.mp3“ erstellt.

Falls die Befehle curl und ffmpeg noch nicht installiert sind, können sie folgendermaßen installiert werden:

```
sudo apt-get install curl
sudo apt-get install ffmpeg
```

4. Überprüfen der MP3-Dateien

Nach dem Erstellen einer Audio-Datei muss überprüft werden, ob alles richtig ausgesprochen und betont wurde. Man kann die ssml-Datei im Nachhinein mit bestimmten Keywords angepasst werden.

Hier einige Beispiele zum Bearbeiten der Aussprache:

```
<break time="300ms"/>
<break strength="weak"/>
<audio src="source">text</audio>
<say-as interpret-as="cardinal">10</say-as>
<say-as interpret-as="ordinal">10</say-as>
<say-as interpret-as="characters">10</say-as>
<sub alias="World Wide Web Consortium">W3C</sub>
<p><s>This is sentence one.</s><s>This is sentence two.</s></p>
<prosody rate="slow" pitch="-2st">Can you hear me now?</prosody>
<emphasis level="moderate">This is an important announcement</emphasis>
```

Beispiel in ssml-Datei:

```
<say>
  Here are <say-as interpret-as="characters">SSML</say-as> samples.
  I can pause <break time="3s"/>.
  I can play a sound
  <audio src="https://www.example.com/MY_MP3_FILE.mp3">didn't get your MP3 audio
file</audio>.
  I can speak in cardinals. Your number is <say-as interpret-as="cardinal">10</say-as>.
  Or I can speak in ordinals. You are <say-as interpret-as="ordinal">10</say-as> in line.
  Or I can even speak in digits. The digits for ten are <say-as
interpret-as="characters">10</say-as>.
  I can also substitute phrases, like the <sub alias="World Wide Web
Consortium">W3C</sub>.
  Finally, I can speak a paragraph with two sentences.
  <p><s>This is sentence one.</s><s>This is sentence two.</s></p>
</say>
```

Unter folgendem Link kann die Bedeutung der Aktionen und Attribute in ssml-Dateien genauer nachgelesen werden:

<https://developers.google.com/assistant/actions/reference/ssml>

Nach dem Anpassen der ssml-Dateien muss Schritt 3 wiederholt werden. Die Schritte können solange wiederholt werden, bis man mit der Audio-Datei zufrieden ist.

Erstellt von: Manuel Kuß, am 19.06.2020