
PyNuance Documentation

Release 0.1

Thibault Cohen

Aug 13, 2017

Contents:

1	Installation	1
1.1	Dependencies	1
1.2	From Pypi	1
1.3	From Github	1
2	Getting Started	3
2.1	Account Creation	3
2.2	Usage	3
3	Development	7
3.1	Create developer environment	7
3.2	Run tests	7
4	References	9
5	PyNuance developer documentation	11
5.1	pynuance package	11
6	Indices and tables	15
	Python Module Index	17

CHAPTER 1

Installation

Dependencies

```
sudo apt-get install libsamplerate0-dev portaudio19-dev libspeexdsp-dev
```

Warning: Need validation

From Pypi

```
pip install pynuance
```

From Github

```
pip install -e https://github.com/titilambert/pynuance.git#egg=pynuance
```


CHAPTER 2

Getting Started

Account Creation

1. Create a Nuance Developer Account here: <https://developer.nuance.com/public/index.php?task=register>
2. Check the box “Yes - request access” in the “Want Mix ?” box
3. Go to your email and activate it
4. Get your credentials (generate credentials.json file):

```
pynuance credentials -u USERNAME -p PASSWORD -c credentials.json
```

5. Then you can use TTS and STT features
6. To use NLU feature, you have to wait maximum 24 hours after the account creation

Usage

Download your cookies and credentials

Get Cookies

To speed up Nuance websites requests, it's recommended to download your cookies

```
pynuance cookies -u USERNAME -p PASSWORD -C cookies.json
```

This will store your cookies in cookies.json file

Get Credentials

Then download your credentials needed to use Nuance services

```
pynuance credentials -C cookies.json -c credentials.json
```

This will store your credentials in credentials.json file

Use Nuance services

Text To Speech

```
pynuance tts -c credentials.json -l en_US -v Allison -d speex -t "Hello World"
```

Speech To Text

```
pynuance stt -c credentials.json -l en_US
```

Then say something in your microphone

NLU

1. Check if you have access to Nuance Mix

```
pynuance mix -C cookies.json check
```

If you got *Your Mix account is activated*, you can use NLU, you can use it !

2. Create a new model

```
pynuance mix -C cookies.json model -m mymodel create -l en_US
```

3. Upload your data

```
pynuance mix -C cookies.json model -m mymodel upload -M examples/example1_en-US.trsx
```

4. Train your model

```
pynuance mix -C cookies.json model -m mymodel train
```

5. Create a build

```
pynuance mix -C cookies.json model -m mymodel build create -N "My first Version"
```

6. List builds

```
pynuance mix -C cookies.json model -m mymodel build list
```

Version	Status	Created at	Notes
1	COMPLETED	2017-07-30T19:17:55	My first Version

7. Attach build to the Sandbox app

```
pynuance mix -C cookies.json model -m mymodel build attach -T mytag
```

8. Run NLU command


```
pynuanance nlu -c credentials.json -l en_US -T mytag text -t "What time is it ?"
```

Note: For next NLU commands, only step 8 is required

Create developer environment

```
virtualenv -p /usr/bin/python3 env
source env/bin/activate
pip install -r requirements.txt
pip install -r test_requirements.txt
python setup.py develop
```

Run tests

```
:: source env/bin/activate pip install -r test_requirements.txt PYNUANCE_USERNAME=yourusername
PYNUANCE_PASSWORD=yourpassword pytest -html=pytest/report.html -junit-
xml=pytest/junit.xml -cov=pynuance/ -cov-report=term -cov-report=html:pytest/coverage/html -cov-
report=xml:pytest/coverage/xml tests/
```


CHAPTER 4

References

- <https://github.com/NuanceDev/ndev-python-http-cli/issues/7>
- <https://github.com/Fadyazmy/harrawr/blob/master/wsclient.py>
- https://developer.nuance.com/public/Help/SpeechKitFrameworkReference_Android/com/nuance/speechkit/RecognitionType.html
- <https://github.com/lemonzi/VoCoMi/blob/master/nuance.py>
- https://developer.nuance.com/public/Help/HttpInterface/HTTP_web_services_for_NCS_clients_1.0_programmer_s_guide.pdf
- <http://nuancedev.github.io/samples/http/python/#overview>

pynuance package

Submodules

pynuance.__main__ module

pynuance.cli module

pynuance.credentials module

Manage Nuance credentials and cookies

`pynuance.credentials.get_credentials` (*username=None, password=None, cookies_file=None*)
Get credentials from Nuance dev page

`pynuance.credentials.save_cookies` (*cookies_file, username=None, password=None*)
Login Dev and Mix Nuance web sites and save cookies to the disk

pynuance.libs namespace

Submodules

pynuance.libs.common module

`pynuance.libs.common.parse_credentials` (*file_path*)

pynuance.libs.error module

exception `pynuance.libs.error.PyNuanceError`

Bases: `Exception`

PyNuance Base Error

pynuance.libs.languages module

pynuance.libs.nuance_http module

`pynuance.libs.nuance_http.get_cookies` (*cookies_file*, *website*)

Get cookies from saved file

`pynuance.libs.nuance_http.nuance_login` (*website*)

pynuance.logger module

pynuance.mix module

Provides functions to interacte with Nuance Mix Website

`pynuance.mix.create_model` (*name*, *language*, *username=None*, *password=None*, *cookies_file=None*)

Create a new model in Nuance Mix.

`pynuance.mix.delete_model` (*name*, *username=None*, *password=None*, *cookies_file=None*)

Delete a model from model name or model ID

`pynuance.mix.get_model` (*name*, *username=None*, *password=None*, *cookies_file=None*)

Get model data from Nuance Mix Website

`pynuance.mix.get_model_id` (*name*, *username=None*, *password=None*, *cookies_file=None*)

Get model ID from model name.

If name is already a model ID, this function can help to validate the existance of the model.

Raise if there are 2 or more models with the same name.

`pynuance.mix.list_models` (*username=None*, *password=None*, *cookies_file=None*)

Get list of models/project from Nuance Mix.

`pynuance.mix.mix_activated` (*username=None*, *password=None*, *cookies_file=None*)

Check if the account has access to Nuance Mix.

URL: <https://developer.nuance.com/mix/nlu/#!/models/>

Returns

- True means Mix account activated.
- False means Mix is being created or not requested

Return type `bool`

`pynuance.mix.model_build_attach` (*name*, *build_version=None*, *context_tag='latest'*, *username=None*, *password=None*, *cookies_file=None*)

Attach model version to a Nuance App

For now, only SandBoxApp is supported by pynuance


```
pynuance.mix.model_build_create(name, notes='', username=None, password=None, cookies_file=None)
```

Create a new model build.

```
pynuance.mix.model_build_list(name, username=None, password=None, cookies_file=None)
```

Return the list of all builds for a given model

```
pynuance.mix.train_model(name, username=None, password=None, cookies_file=None)
```

Train a given Mix Model

```
pynuance.mix.upload_model(name, model_file, username=None, password=None, cookies_file=None)
```

Upload intent file into a Mix model.

pynuance.nlu module

pynuance.recorder module

pynuance.stt module

pynuance.tts module

pynuance.websocket module

Module defining abstractWebsocket class

```
class pynuance.websocket.AbstractWebsocketConnection(url, logger)
```

Bases: `object`

WebSocket connection object to handle Nuance server communications

MSG_AUDIO = 2

MSG_JSON = 1

close()

Close WebSocket connection

connect(app_id, app_key, use_plaintext=True)

Connect to the websocket

receive()

Handle server response

send_audio(audio)

Send audio to the server

send_message(msg)

Send json message to the server

```
class pynuance.websocket.BadWebsocketConnection(url, logger)
```

Bases: `pynuance.websocket.AbstractWebsocketConnection`

WebSocket connection object to handle Nuance server communications

connect(app_id, app_key, use_plaintext=True)

Connect to the server

```
class pynuance.websocket.WebsocketConnection(url, logger)
```

Bases: `pynuance.websocket.AbstractWebsocketConnection`

Websocket client

connect (*app_id, app_key, use_plaintext=True*)
Connect to the websocket

static sign_credentials (*datestr, app_key, app_id*)
Handle credentials

`pynuance.websocket.connection_handshake` (*client*)
Nuance connection handshake.
Use for STT and NLU audio.

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`

p

- `pynuance`, [11](#)
- `pynuance.credentials`, [11](#)
- `pynuance.libs.common`, [11](#)
- `pynuance.libs.error`, [12](#)
- `pynuance.libs.languages`, [12](#)
- `pynuance.libs.nuance_http`, [12](#)
- `pynuance.mix`, [12](#)
- `pynuance.websocket`, [13](#)

A

AbstractWebsocketConnection (class in pynu-
ance.websocket), 13

B

BadWebsocketConnection (class in pynu-
ance.websocket), 13

C

close() (pynuance.websocket.AbstractWebsocketConnection
method), 13

connect() (pynuance.websocket.AbstractWebsocketConnection
method), 13

connect() (pynuance.websocket.BadWebsocketConnection
method), 13

connect() (pynuance.websocket.WebsocketConnection
method), 13

connection_handshake() (in module pynu-
ance.websocket), 14

create_model() (in module pynuance.mix), 12

D

delete_model() (in module pynuance.mix), 12

G

get_cookies() (in module pynuance.libs.nuance_http), 12

get_credentials() (in module pynuance.credentials), 11

get_model() (in module pynuance.mix), 12

get_model_id() (in module pynuance.mix), 12

L

list_models() (in module pynuance.mix), 12

M

mix_activated() (in module pynuance.mix), 12

model_build_attach() (in module pynuance.mix), 12

model_build_create() (in module pynuance.mix), 12

model_build_list() (in module pynuance.mix), 13

MSG_AUDIO (pynuance.websocket.AbstractWebsocketConnection
attribute), 13

MSG_JSON (pynuance.websocket.AbstractWebsocketConnection
attribute), 13

N

nuance_login() (in module pynuance.libs.nuance_http),
12

P

parse_credentials() (in module pynuance.libs.common),
11

pynuance (module), 11

pynuance.credentials (module), 11

pynuance.libs.common (module), 11

pynuance.libs.error (module), 12

pynuance.libs.languages (module), 12

pynuance.libs.nuance_http (module), 12

pynuance.mix (module), 12

pynuance.websocket (module), 13

PyNuanceError, 12

R

receive() (pynuance.websocket.AbstractWebsocketConnection
method), 13

S

save_cookies() (in module pynuance.credentials), 11

send_audio() (pynuance.websocket.AbstractWebsocketConnection
method), 13

send_message() (pynuance.websocket.AbstractWebsocketConnection
method), 13

sign_credentials() (pynu-
ance.websocket.WebsocketConnection static
method), 14

T

train_model() (in module pynuance.mix), 13

U

`upload_model()` (in module `pynuance.mix`), [13](#)

W

`WebsocketConnection` (class in `pynuance.websocket`), [13](#)