echokit Documentation

Release 0.4

Edward Wells

Feb 26, 2018

Contents:

CHAPTER 1

echokit 0.4

A light(er)weight toolkit to create Alexa skills (Python 3.6)

1.1 Installation

Requirements:

- Python 3.6 or newer

From GitHub:

Clone/download this repo and run this from the `echokit/` directory:

```
python setup.py install
```

1.2 Getting Started

Sample skills can be found at https://github.com/arcward/echokit under the *samples/* directory.

More comprehensive documentation can be found on ReadTheDocs

1.3 Application ID/Handler

When you're configuring your skill in the Alexa developer console, you'll be given an ID for your skill. This can be specified in EchoKit, as well as whether it should verify incoming application IDs.

When you configure your Lambda function, you'll need to specify a handler for incoming requests as well. To set these:

```
from echokit import EchoKit

app = EchoKit("your_app_id")
handler = app.handler
```

If, for example, your skill's module is `sample.py`, then you would define `sample.handler` as the handler in your Lambda function.

1.4 Handling Requests

Responses to requests are created via `echokit.response.Response`

There are three decorators to handle each basic request type, as well as one to handle slots:

```
from echokit import EchoKit

app = EchoKit("your_app_id")
handler = app.handler

# LaunchRequest
@app.launch
def on_launch(request, session):
    pass

# SessionEndedRequest
@app.session_ended
def on_session_ended(request, session):
    pass

# IntentRequest
@app.intent("MyIntentName")
def on_my_intent(request, session):
    pass

# IntentRequest (with slots)
@app.intent("MyIntentWithSlots")
@app.slot("first_slot", "second_slot")
def on_intent_with_slots(request, session, first_slot, second_slot):
    pass
```

1.5 Creating a ZIP file for upload to AWS Lambda

1.6 EchoZip

`echozip` is a script included to help create ZIP deployment packages. If you installed via *setup.py*, you can run it from the command line (try `echozip --help`).

Specify your top-level package directory. For example, if your `__init__.py` is located at `example/__init__.py` you would run `echozip example`

This would create a ZIP file in your current directory, named like *example_{YY-MM-DD-HHMMSS}.zip*. If you extract it again, you'll see that it includes the full subtree of the directory you specified, with an additional *echokit/* directory at the top level.

1.7 Manually

Your ZIP file should be created from within your top-level package (don't just zip the enclosing directory). You'll need to download/clone echokit and include `echokit/` in in that same top-level directory. So if your `__init__.py` is in `~/my_project/` you should have `~/my_project/echokit`.

See the official docs for more info.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search