Alasan Documentation

Release 0.0.1

Lucas Hild

Dec 31, 2018

First Steps

CHAPTER 1

Installation

Alasan uses serverless, a toolkit for deploying serverless application, to upload your skill to AWS Lambda. You can install it using the package manager npm.

```
npm install -g serverless
```

Next we will create a project using serverless

```
serverless create --template aws-python3 --path <name-of-your-skill>
cd <name-of-your-skill>
```

Next we will install Alasan using pip. Moreover, put *alasan* into *requirements.txt*.

```
pip install alasan
echo alasan > requirements.txt
```

Serverless needs a plugin for managing Python requirements. We can install it like that.

```
sls plugin install -n serverless-python-requirements
```

Next we will have to configure serverless. Therefore we can use the a configuration file called `serverless.yml`. Put the following into the file.

```
service: <name-of-your-skill>

provider:
  name: aws
  runtime: python3.6

functions:
  skill:
    handler: myskill.skill

package:
  individually: true
```

(continues on next page)

(continued from previous page)

```
  exclude:
    - ./**
  include:
    - myskill.py

plugins:
  - serverless-python-requirements
```

Replace `<name-of-your-skill>` with the name your AWS Lambda function should be called. Moreover, we have to change the handler, the entrypoint for our skill. Replace `myskill` with the name of your file / module and `skill` with the instance of Alasan (`skill = Alasan()`).

For example if you are using one file …

```
- awesome_skill.py [ my_skill = Alasan() ]
- README.md
...
```

… your configuration should look like that: `handler: awesome_skill.my_skill`.

But if you are using a module …

```
- fun_skill
  - __init__.py [ skill = Alasan() ]
  ...
- README.md
...
```

… your configuration should look like that: `handler: fun_skill.skill`.

CHAPTER 2

Your first skill

In this section you are going to create your first skill. You will be able to say hello and your skill will greet you back. Therefore we need one custom intent. Create a custom intent called `HelloIntent` in the Alexa Developer Console and add *Hello* as a Sample Utterance.

A conversation with the skill could look like that:

- **User**: Alexa, start awesome skill.
- **Alexa**: Welcome from Alasan. Say hello to me!
- **User**: Hello.
- **Alexa**: Hello, thanks for chatting with me.

First of all we need to import Alasan and create an instance of it.

```
from alasan import Alasan, Response

skill = Alasan()
```

2.1 LaunchIntent

Next we can listen to *LaunchIntent*s. They are called, when the user starts the skill (Alexa, open . . . / Alexa, start . . .).

```
@skill.launch()
def launch():
    ...
```

2.2 Response

Now we want to build a response for the intent. Therefore we use the class `Response` and return it. The response is built using method cascading.

```
@skill.launch()
def launch():
    return Response.speak("Welcome to Alasan. Say hello to me!")
```

2.3 Custom Intent

After that Alexa waits for a new intent. If the user says *Hello*, a *HelloIntent* will be fired.

```
@skill.intent("HelloIntent")
def hello_intent():
    ...
```

This time we want to respond, but then the skill should end. Therefore we have to call the method `end_session()`.

```
@skill.intent("HelloIntent")
def hello_intent():
    return Response \
        .speak("Hello, thanks for chatting with me.") \
        .end_session()
```

2.4 Deployment

That's it. Now we can test our skill. To do so, we have to upload it to AWS Lambda. We will be using serverless for that.

```
serverless deploy
```

Now you should be able to talk to your skill. Just say: *Alexa, start awesome skill.*

Alasan helps you build Alexa skills on AWS Lambda using Python.