alexandra Documentation

Release 0.3.0

Author

Oct 11, 2018

Contents

Python support for Alexa applications.

Because like everything Amazon it involves a ton of tedious boilerplate.

CHAPTER 1

alexandra.app

class alexandra.app.**Application**

create_wsgi_app(*validate_requests=True*)
Return an object that can be run by any WSGI server (uWSGI, etc.) to serve this Alexa application.

dispatch_request(*body*)
Given a parsed JSON request object, call the correct Intent, Launch, or SessionEnded function.

This function is called after request parsing and validaion and will raise a *ValueError* if an unknown request type comes in.

Parameters **body** – JSON object loaded from incoming request's POST data.

intent(*intent_name*)
Decorator to register a handler for the given intent.

The decorated function can either take 0 or 2 arguments. If two are specified, it will be provided a dictionary of *{slot_name: value}* and a *alexandra.session.Session* instance.

If no session was provided in the request, the session object will be *None*.

```
@alexa_app.intent('FooBarBaz')
def foo_bar_baz_intent(slots, session):
    pass

@alexa_app.intent('NoArgs')
def noargs_intent():
    pass
```

launch(*func*)
Decorator to register a function to be called whenever the app receives a LaunchRequest (which happens when someone invokes your skill without specifying an intent).

```
@alexa_app.launch
def launch_handler(session):
    pass
```

run(*host*, *port*, *debug=True*, *validate_requests=True*)
Utility method to quickly get a server up and running.

Parameters

- **debug** – turns on Werkzeug debugger, code reloading, and full logging.
- **validate_requests** – whether or not to ensure that requests are sent by Amazon. This can be usefulfor manually testing the server.

session_end(*func*)
Decorator to register a function to be called when a SessionEndedRequest is received.

unknown_intent(*func*)
Decorator to register a function to be called when an unknown intent is received. This should only happen when the intents/utterance file are malformed.

CHAPTER 2

alexandra.session

class alexandra.session.**Session**(*session_body*)

Provides easier access to session objects sent along with requests.

The object parsed from the request can be accessed directly through the `body` member.

application_id

get(*attr*, *default=None*)

Get an attribute defined by this session

is_new

Is this a new session or a previously open one?

session_id

user_access_token

user_id

CHAPTER 3

alexandra.util

Utility functionality for Alexandra

alexandra.util.**reprompt**(*text=None*, *ssml=None*, *attributes=None*)

Convenience method to save a little bit of typing for the common case of reprompting the user. Simply calls *alexandra.util.respond()* with the given arguments and holds the session open.

One of either the *text* or *ssml* should be provided if any speech output is desired.

Parameters

- **text** – Plain text speech output
- **ssml** – Speech output in SSML format
- **attributes** – Dictionary of attributes to store in the current session

alexandra.util.**respond**(*text=None*, *ssml=None*, *attributes=None*, *reprompt_text=None*, *reprompt_ssml=None*, *end_session=True*)

Build a dict containing a valid response to an Alexa request.

If speech output is desired, either of *text* or *ssml* should be specified.

Parameters

- **text** – Plain text speech output to be said by Alexa device.
- **ssml** – Speech output in SSML form.
- **attributes** – Dictionary of attributes to store in the session.
- **end_session** – Should the session be terminated after this response?
- **reprompt_ssml** (*reprompt_text,*) – Works the same as *text*/*ssml*, but instead sets the reprompting speech output.

alexandra.util.**validate_request_certificate**(*headers*, *data*)

Ensure that the certificate and signature specified in the request headers are truely from Amazon and correctly verify.

Returns True if certificate verification succeeds, False otherwise.

Parameters

- **headers** – Dictionary (or sufficiently dictionary-like) map of request headers.
- **data** – Raw POST data attached to this request.

alexandra.util.**validate_request_timestamp**(*req_body*, *max_diff=150*)

Ensure the request's timestamp doesn't fall outside of the app's specified tolerance.

Returns True if this request is valid, False otherwise.

Parameters

- **req_body** – JSON object parsed out of the raw POST data of a request.
- **max_diff** – Maximum allowable difference in seconds between request timestamp and system clock. Amazon requires <= 150 seconds for published skills.

CHAPTER 4

alexandra.wsgi

class alexandra.wsgi.**WsgiApp**(*alexa*, *validate_requests=True*)

This class is wraps *alexandra.app.Application* object and handles all the gory details of parsing Alexa requests and validating that they were actually sent by Amazon.

This class implements the WSGI interface, so an instance of can be passed to standard WSGI servers (uWSGI, gunicorn, ...)

wsgi_app(*request*)

Incoming request handler.

Parameters **request** – Werkzeug request object

Python Module Index

a

Index