```
"""
How does AWS Lambda cheer up Amazon Lex? By saying, "Don't worry, I've got your back(end)!"

- NextWork :)
"""
import json
import random
import decimal

def random_num():
    return(decimal.Decimal(random.randrange(1000, 50000))/100)

def get_slots(intent_request):
    return intent_request['sessionState']['intent']['slots']

def get_slot(intent_request, slotName):
    slots = get_slots(intent_request)
    if slots is not None and slotName in slots and slots[slotName] is not None:
        return slots[slotName]['value']['interpretedValue']
    else:
        return None

def get_session_attributes(intent_request):
    sessionState = intent_request['sessionState']
    if 'sessionAttributes' in sessionState:
        return sessionState['sessionAttributes']

    return {}

def elicit_intent(intent_request, session_attributes, message):
    return {
        'sessionState': {
            'dialogAction': {
                'type': 'ElicitIntent'
            },
            'sessionAttributes': session_attributes
        },
        'messages': [ message ] if message != None else None,
        'requestAttributes': intent_request['requestAttributes'] if 'requestAttributes' in
intent_request else None
    }


def close(intent_request, session_attributes, fulfillment_state, message):
```

```
    intent_request['sessionState']['intent']['state'] = fulfillment_state
    return {
        'sessionState': {
            'sessionAttributes': session_attributes,
            'dialogAction': {
                'type': 'Close'
            },
            'intent': intent_request['sessionState']['intent']
        },
        'messages': [message],
        'sessionId': intent_request['sessionId'],
        'requestAttributes': intent_request['requestAttributes'] if 'requestAttributes' in
intent_request else None
    }

def CheckBalance(intent_request):
    session_attributes = get_session_attributes(intent_request)
    slots = get_slots(intent_request)
    account = get_slot(intent_request, 'accountType')
    #The account balance in this case is a random number
    #Here is where you could query a system to get this information
    balance = str(random_num())
    text = "Thank you. The balance on your "+account+" account is $"+balance+"
dollars."
    message =  {
            'contentType': 'PlainText',
            'content': text
        }
    fulfillment_state = "Fulfilled"
    return close(intent_request, session_attributes, fulfillment_state, message)

def FollowupCheckBalance(intent_request):
    session_attributes = get_session_attributes(intent_request)
    slots = get_slots(intent_request)
    account = get_slot(intent_request, 'accountType')
    #The account balance in this case is a random number
    #Here is where you could query a system to get this information
    balance = str(random_num())
    text = "Thank you. The balance on your "+account+" account is $"+balance+"
dollars."
    message =  {
            'contentType': 'PlainText',
            'content': text
        }
    fulfillment_state = "Fulfilled"
```

```
    return close(intent_request, session_attributes, fulfillment_state, message)


def dispatch(intent_request):
    intent_name = intent_request['sessionState']['intent']['name']
    response = None
    # Dispatch to your bot's intent handlers
    if intent_name == 'CheckBalance':
        return CheckBalance(intent_request)
    elif intent_name == 'FollowupCheckBalance':
        return FollowupCheckBalance(intent_request)

    raise Exception('Intent with name ' + intent_name + ' not supported')

def lambda_handler(event, context):
    response = dispatch(event)
    return response
```