Hello World

Table of Contents

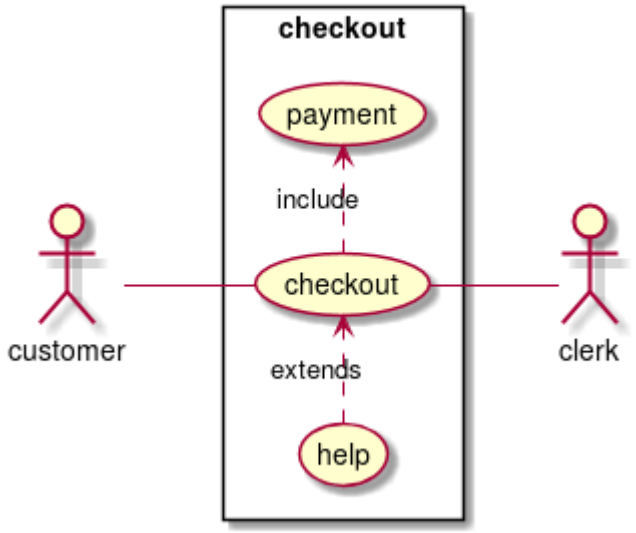
1. 仕様

2. 設計

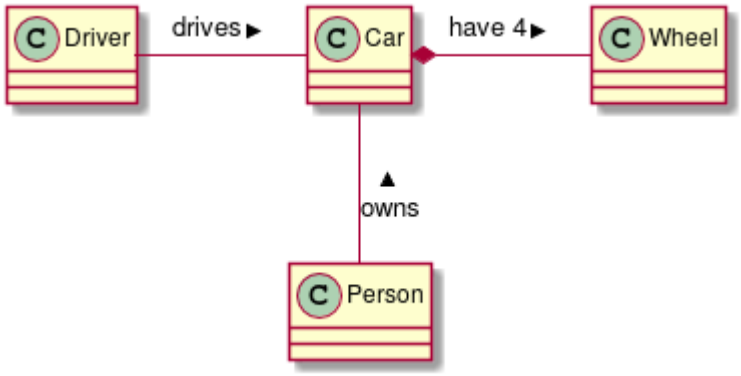
2.1. TODOリスト

- [ ] TODO
- [x] ~~TODO DONE~~

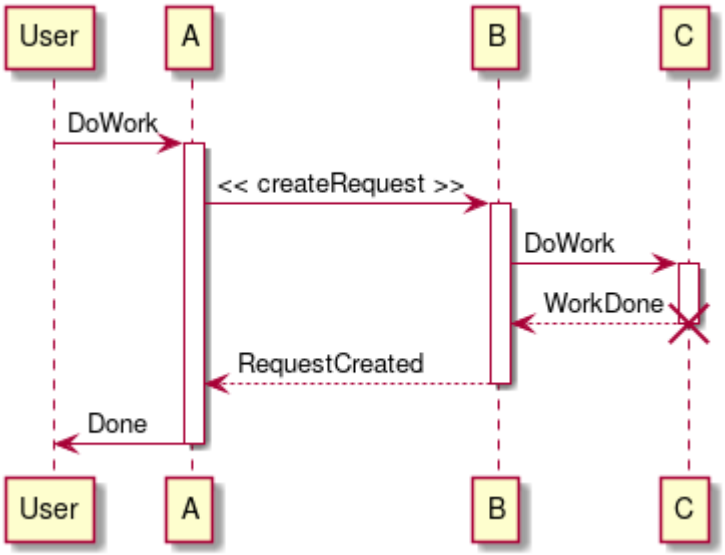
2.2. ユースケース図



2.3. クラス図



2.4. シーケンス図



3. 実装

```
import json

import requests


def lambda_handler(event, context):
    """Sample pure Lambda function

    Arguments:
        event LambdaEvent -- Lambda Event received from Invoke API
        context LambdaContext -- Lambda Context runtime methods and attributes

    Returns:
        dict -- {'statusCode': int, 'body': dict}
    """

    ip = requests.get('http://checkip.amazonaws.com/')

    return {
        "statusCode": 200,
        "body": json.dumps({
            'message': 'Hello Python lambda world',
            'location': ip.text.replace('\n', ''),
        }),
        "headers": {
            'Content-Type': 'application/json',
            'Access-Control-Allow-Origin': '*',
        }
    }
```

```
import json
```

```
import pytest
from hello_world import app


@pytest.fixture()
def apigw_event():
    """ Generates API GW Event"""

    return {
        "body": "{ \"test\": \"body\"}",
        "resource": "/{proxy+}",
        "requestContext": {
            "resourceId": "123456",
            "apiId": "1234567890",
            "resourcePath": "/{proxy+}",
            "httpMethod": "POST",
            "requestId": "c6af9ac6-7b61-11e6-9a41-93e8deadbeef",
            "accountId": "123456789012",
            "identity": {
                "apiKey": "",
                "userArn": "",
                "cognitoAuthenticationType": "",
                "caller": "",
                "userAgent": "Custom User Agent String",
                "user": "",
                "cognitoIdentityPoolId": "",
                "cognitoIdentityId": "",
                "cognitoAuthenticationProvider": "",
                "sourceIp": "127.0.0.1",
                "accountId": ""
            },
            "stage": "prod"
        },
        "queryStringParameters": {
            "foo": "bar"
        },
        "headers": {
            "Via":
            "1.1 08f323deadbeefa7af34d5feb414ce27.cloudfront.net (CloudFront)",
            "Accept-Language":
            "en-US,en;q=0.8",
            "CloudFront-Is-Desktop-Viewer":
            "true",
            "CloudFront-Is-SmartTV-Viewer":
            "false",
            "CloudFront-Is-Mobile-Viewer":
            "false",
            "X-Forwarded-For":
            "127.0.0.1, 127.0.0.2",
            "CloudFront-Viewer-Country":
            "US",
```

```
            "Accept":

"text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8",
            "Upgrade-Insecure-Requests":
            "1",
            "X-Forwarded-Port":
            "443",
            "Host":
            "1234567890.execute-api.us-east-1.amazonaws.com",
            "X-Forwarded-Proto":
            "https",
            "X-Amz-Cf-Id":
            "aaaaaaaaaae3VYQb9jd-nvCd-de396Uhbp027Y2JvkCPNLmGJHqlaA==",
            "CloudFront-Is-Tablet-Viewer":
            "false",
            "Cache-Control":
            "max-age=0",
            "User-Agent":
            "Custom User Agent String",
            "CloudFront-Forwarded-Proto":
            "https",
            "Accept-Encoding":
            "gzip, deflate, sdch"
        },
        "pathParameters": {
            "proxy": "/examplepath"
        },
        "httpMethod": "POST",
        "stageVariables": {
            "baz": "qux"
        },
        "path": "/examplepath"
    }


def test_lambda_handler(apigw_event):

    ret = app.lambda_handler(apigw_event, "")
    assert ret['statusCode'] == 200

    for key in ('message', 'location'):
        assert key in ret['body']

    data = json.loads(ret['body'])
    assert data['message'] == 'Hello Python lambda world'
```

4. 参照

- PlantUML[http://plantuml.com]