Hello World

Table of Contents

1.	. 仕様	1
2.	設計	1
	2.1. TODOリスト	1
	2.2. ユースケース図	1
	2.3. クラス図	1
	2.4. シーケンス図	1
3.	.実装	2
4.	- 参照	3

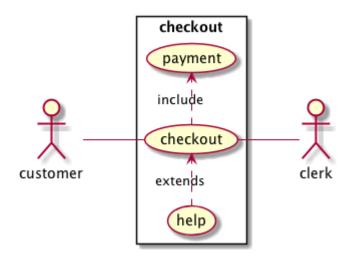
1. 仕様

2. 設計

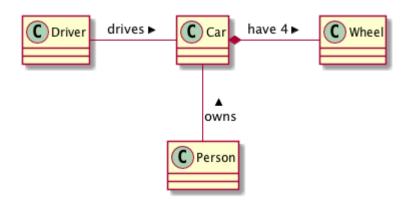
2.1. TODO リスト

- □ TODO
- **▼** TODO DONE

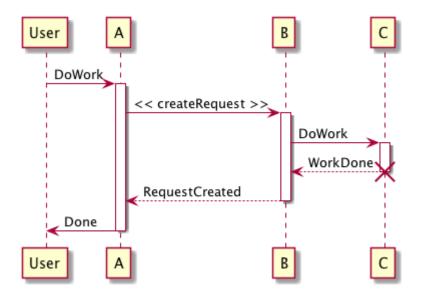
2.2. ユースケース図



2.3. クラス図



2.4. シーケンス図



3. 実装

```
/* eslint-disable no-console */
const axios = require('axios');
const url = 'http://checkip.amazonaws.com/';
let response;
exports.lambda_handler = async (event, context, callback) => {
    try {
        const ret = await axios(url);
        response = {
            statusCode: 200,
            body: JSON.stringify({
                message: 'hello world',
                location: ret.data.trim(),
            }),
        };
    } catch (err) {
        console.log(err);
        callback(err, null);
    }
    callback(null, response);
};
```

```
/* eslint-disable prefer-destructuring,no-unused-vars,no-shadow */
const chai = require('chai');
const app = require('../../app.js');
const expect = chai.expect;
let event; let
    context;
describe('Tests index', () => {
    it('verifies successful response', async () => {
        const result = await app.lambda_handler(event, context, (err, result) => {
            expect(result).to.be.an('object');
            expect(result.statusCode).to.equal(200);
            expect(result.body).to.be.an('string');
            const response = JSON.parse(result.body);
            expect(response).to.be.an('object');
            expect(response.message).to.be.equal('hello world');
            expect(response.location).to.be.an('string');
        });
    });
});
```

4. 参照

• PlantUML[http://plantuml.com]