

測試報告

- 110213041
- 陳家輝

1. 簡介

本公司受托為 貴公司 web 後台進行黑箱測試. 經測試后發現 貴公司 web 後台一些執行結果與預期并不相符.

2. 目錄

- 1. 簡介
- 2. 目錄
- 3. 背景
- 4. 環境設定
 - 4.1. 環境規格
 - 4.2. 使用插件
 - 4.3. 協助函式/參數
 - 4.3.1. `class subjectURL`
 - 4.3.2. `function createSession`
 - 4.3.3. `function fetchHelper`
 - 4.3.4. 測試執行方法
- 5. 測試過程
 - 5.1. 基礎測試
 - 5.1.1. Test action getTotal
 - 5.1.2. Test action addItem
 - 5.1.3. Test action discount
 - 5.1.4. Test action reset
 - 5.2. 測試增加產品
 - 5.2.1. 負數處理
 - 5.2.1.1. One item negative price
 - 5.2.1.2. Negative arithmetic
 - 5.2.2. Edge Value Check
 - 5.3. 測試折扣
 - 5.3.1. 協助資料形態
 - 5.3.2. Normal Case (Under 5k)
 - 5.3.3. Bonus Case
 - 5.3.4. Extreme Value
- 6. 測試結果
 - 6.1. 增加產品允許負數
 - 6.2. 商品加總
 - 6.3. 折扣價格
- 7. 修改建議
- 8. 附錄
 - 8.1. 輸出結果

3. 背景

本次測試預期各 API回傳結果如下.

- `act=reset`
 - 沒有回傳
- `act=getTotal`
 - 傳回不含折扣總價.
- `act=getDiscount`
 - 傳回純折扣金額.
 - **只有超過 5000 (> 5000) 才會額外獎勵, 該獎勵只會執行一次.**
- `act=actItem`
 - 沒有回傳

4. 環境設定

本公司選用 **Deno** 執行環境, 撰寫 **TypeScript** 進行測試.

4.1. 環境規格

作業系統	Window 10 Pro	22H2
執行環境	Deno	1.38.5
語言版本	TypeScript	5.2.2

4.2. 使用插件

由於 **Deno** 本身沒有提供 Cookies 管理, 因此需要額外下載 **CookieJar** 管理 **Cookies**.

deno-another-cookiejar 5.0.3

4.3. 協助函式/參數

為了方便測試, 將會設立以下函數/參數.

4.3.1. **class subjectURL**

static class 儲存後端鏈接.

- Properties
 - (static) **base: string**
 - 測試鏈接
 - (static) **reset: URL**
 - action reset
 - (static) **getTotal: URL**
 - action getTotal
 - (static) **getDiscount: URL**
 - action getDiscount
- Methods
 - (static) **addItem(name: string, price: number | string): URL**
 - action addItem query

4.3.2. **function createSession**

方便建立 session.

- Parameters
 - **url: URL | string**

- `cookie: CookieJar | undefined`

- Returns `CookieJar`

4.3.3. `function fetchHelper`

fetch with session.

- Parameters
 - `url: URL | string`
 - `cookie: CookieJar`
 - `init: RequestInit | undefined`
- Returns `Promise<Response>`

4.3.4. 測試執行方法

1. cd 進本資料夾根目錄 PS `.../110213041_04_blackbox >`
2. 執行以下指令 `deno task test`

```
# PowerShell

C:\User> ls

    Directory: C:\User

Mode                LastWriteTime         Length Name
----                -
d-----          XX/XX/XXXX         XX:XX           110213041_04_blackbox

C:\User> cd 110213041_04_blackbox
C:\User\110213041_04_blackbox>
C:\User\110213041_04_blackbox> deno task test
```

5. 測試過程

測試分為 3 大部分.

1. 基礎測試. 簡單測試個 API 是否可以簡單運作.
2. 測試增加產品, 對增加商品進行特殊測試.
3. 測試折扣. 測試折扣價格.

以下大致描述測試 pseudo code.

5.1. 基礎測試

5.1.1. Test action getTotal

```
create Session
GET getTotal
assert total == 0
```

5.1.2. Test action addItem

```
create Session
POST addItem name=foo price=50
GET getTotal
assert total == 50
```

5.1.3. Test action discount

```
create Session
POST addItem name=foo price=1000
GET getDiscount
assert discount == 100
```

5.1.4. Test action reset

```
create Session
POST addItem name=foo price=1000
GET getTotal
assert total == 1000
GET reset
GET getTotal
assert total == 0
```

5.2. 測試增加產品

5.2.1. 負數處理

本次測試預期. 本次測試預期後台會無視負數數值.

5.2.1.1. One item negative price

```
create Session
POST addItem name=foo price=-1
GET getTotal
assert total == 0
```

5.2.1.2. Negative arithmetic

```
create Session
POST addItem name=foo price=50
POST addItem name=foo price=-500
GET getTotal
assert total == 50
```

5.2.2. Edge Value Check

測試數值如下:

```
[
  0,
  1,
  2 ** 31 - 1, // 2147483647          max int32
  2 ** 32 - 1, // 4294967295          max unsigned int32
  2 ** 63 - 1, // 9223372036854776000 max int64
  2 ** 64 - 1, // 18446744073709552000 max unsigned int64
  25000,
  25001,
]
```

以上測試資料會使用以下方式進行測試.

```
create Session
POST addItem name=foo price=(Test Value)
GET getTotal
assert total == (Test Value)
```

5.3. 測試折扣

為加速測試時間, 此測試將會以非同步方式進行.

5.3.1. 協助資料形態

為方便管理, 在此區域中會定義一個資料結構 `testCase_t`.

- definition
 - `price: number`
 - `discount: number`

本測試資料會使用以下方式進行測試.

```
create Session
POST addItem name=foo price=(TestCase price)
GET getDiscount
assert discount == (TestCase discount)
```

5.3.2. Normal Case (Under 5k)

本範圍測資如下:

```
[
  { price: 999, discount: 0 },
  { price: 1000, discount: 100 },
  { price: 1001, discount: 100 },
  { price: 2000, discount: 200 },
  { price: 4999, discount: 400 },
  { price: 5000, discount: 500 },
]
```

5.3.3. Bonus Case

本範圍測資如下:

```
[
  { price: 5001, discount: 600 },
  { price: 10000, discount: 1100 },
  { price: 10001, discount: 1100 },
  { price: 15000, discount: 1600 },
  { price: 15001, discount: 1600 },
  { price: 25000, discount: 2600 },
]
```

5.3.4. Extreme Value

本範圍測資如下:

```
[
  {
    price: (1_000_000 - 1000) / 2, // (990000) exclude bonus
    discount: 10_000,
  },
  {
    price: (1_001_000 - 1000) / 2, // (1000000) exclude bonus
    discount: 10_000,
  },
  {
    price: (1_002_000 - 1000) / 2, // (1001000) exclude bonus
    discount: 10_000,
  },
]
```


6. 測試結果

根據上述所定義測試方式, 大部分項目均通過測試. 但於 5.2. 測試項目中發現疑似程式邏輯錯誤.

6.1. 增加產品允許負數

在 5.2.1. 針對負數測試中, 傳入負數數值 **-1** 時沒有任何錯誤回饋. 在檢查數時得到結果 **-1**. 見得目標後台並沒有檢查數值範圍.

由於新增產品允許負數, 同時計算總額方式推測為 `map((price, quantity) => price * quantity).reduce((total, price) => total + price)`. 因此總額會出錯.

6.2. 商品加總

在 5.2.2. 針對極端值測試中, 當價格 **> 25000**, 總額會自動乘以 **2**.

6.3. 折扣價格

在 5.3. 針對負數測試中, 只有金額超過 **5000** 才會一次額外加成. 由於未知 貴公司折扣方案, 因此假設每到 **5000 倍數并未折**. 在此前提下本次此項測試沒有任何問題.

7. 修改建議

我們建議 貴公司於新增產品時驗證數量, 確保輸入不會出現負數. 同時需要修正 > 25000 總額出錯問題.

8. 附錄

8.1. 輸出結果

```
PS > deno task test
Task test deno test --allow-net ./src/test.ts
running 3 tests from ./src/test.ts
Basic Testing ...
  Test action getTotal ... ok (715ms)
  Test action addItem ... ok (502ms)
  Test action discount ... ok (502ms)
  Test action reset ... ok (732ms)
Basic Testing ... ok (2s)
AddItem Edge Case ...
  Prevent negative price ...
    One item negative price ... FAILED (503ms)
    Negative arithmetic ... FAILED (665ms)
  Prevent negative price ... FAILED (due to 2 failed steps) (1s)
Edge Value Check ...
  Value: 0 ... ok (500ms)
  Value: 1 ... ok (501ms)
  Value: 2147483647 ... FAILED (503ms)
  Value: 4294967295 ... FAILED (498ms)
  Value: 9223372036854776000 ... FAILED (497ms)
  Value: 18446744073709552000 ... FAILED (500ms)
  Value: 25000 ... ok (502ms)
  Value: 25001 ... FAILED (556ms)
Edge Value Check ... FAILED (due to 5 failed steps) (4s)
AddItem Edge Case ... FAILED (due to 2 failed steps) (5s)
Discount Edge Case ...
  Normal Case (Under 5k) ...
    Price: 999, Discount: 0 ... ok (518ms)
    Price: 1000, Discount: 100 ... ok (506ms)
    Price: 4999, Discount: 400 ... ok (505ms)
    Price: 2000, Discount: 200 ... ok (509ms)
    Price: 5000, Discount: 500 ... ok (512ms)
    Price: 1001, Discount: 100 ... ok (518ms)
  Normal Case (Under 5k) ... ok (521ms)
  Bonus Case ...
    Price: 5001, Discount: 600 ... ok (303ms)
    Price: 10000, Discount: 1100 ... ok (233ms)
    Price: 25000, Discount: 2600 ... ok (297ms)
    Price: 15001, Discount: 1600 ... ok (297ms)
    Price: 15000, Discount: 1600 ... ok (300ms)
    Price: 10001, Discount: 1100 ... ok (363ms)
  Bonus Case ... ok (364ms)
  Extreme Value ...
    Price: 499500, Discount: 10000 ... ok (235ms)
    Price: 500500, Discount: 10000 ... ok (231ms)
    Price: 500000, Discount: 10000 ... ok (234ms)
  Extreme Value ... ok (235ms)
```

Discount Edge Case ... ok (1s)

.
.
.

FAILURES

AddItem Edge Case ... Prevent negative price ... One item negative price =>
./src/test.ts:77:13

AddItem Edge Case ... Prevent negative price ... Negative arithmetic =>
./src/test.ts:90:13

AddItem Edge Case ... Edge Value Check ... Value: 2147483647 =>
./src/test.ts:134:15

AddItem Edge Case ... Edge Value Check ... Value: 4294967295 =>
./src/test.ts:134:15

AddItem Edge Case ... Edge Value Check ... Value: 9223372036854776000 =>
./src/test.ts:134:15

AddItem Edge Case ... Edge Value Check ... Value: 18446744073709552000 =>
./src/test.ts:134:15

AddItem Edge Case ... Edge Value Check ... Value: 25001 =>
./src/test.ts:134:15

FAILED | 2 passed (25 steps) | 1 failed (9 steps) (8s)

error: Test failed