Onebox Challenge - Miguel Coronel

Setup:

- Development Language: Java 8 (Amazon Corretto)
- Main Framework: Spring Boot Framework
- Database: H2 Database (in-memory to facilitate challenge checking and testing)
- Project Management: Maven
- Documentation: Swagger
- Testing: Junit Jupiter

Workflow:

1. To start the workflow, the user should obtain a new empty cart with id and expiration information inside.

2. After the cart is created, the id could be use for the following requests:
    - GET cart information (including product list and expiration). *
    - UPDATE product list inside the cart. *
    - DELETE cart.

    * This operation will update expiration time (+10min).

3. Every minute, a worker runs to obtain a list of expired carts and then delete each one of them.

Demonstration:

- **GET NEW CART:**

RQ:
GET {{service-url}}/api/v1/carts/new

RS:

```
{
    "uuid": "8b557eba-409a-4990-b817-99f6e3fe711b",
    "expireAt": "2023-05-02T15:51:08.208",
    "productList": []
}
```

- **GET CART INFORMATION BY ID:**

  RQ:
  GET {{service-url}}/api/v1/carts/8b557eba-409a-4990-b817-99f6e3fe711b

  RS:

```
{
    "uuid": "8b557eba-409a-4990-b817-99f6e3fe711b",
    "expireAt": "2023-05-02T15:52:58.385",
    "productList": []
}
```

- **UPDATE PRODUCT LIST:**

  RQ:
  PUT {{service-url}}/api/v1/carts/8b557eba-409a-4990-b817-99f6e3fe711b

```
[
    {
        "id": 1,
        "description":"A new product description",
        "amount":12.00
    },
    {
        "id": 2,
        "description":"Another product description",
        "amount": 23.00
    },
    {
        "id": 3,
        "description": "One more product description",
        "amount": 4.00
    }
]
```

  RS:

```
{
    "uuid": "297f4e27-f4e4-48f8-8822-6e341c200f14",
    "expireAt": "2023-05-02T13:46:47.946",
    "productList": [
        {
            "id": 2,
            "description": "product description 2",
            "amount": 2.00
        },
        {
            "id": 3,
            "description": "product description 3",
            "amount": 3.00
        },
        {
            "id": 4,
            "description": "product description 4",
            "amount": 4.00
        }
    ]
}
```

- **DELETE CART:**

  RQ:
  DELETE {{service-url}}/api/v1/carts/8b557eba-409a-4990-b817-99f6e3fe711b

  RS:
  CODE 200 - OK

- **GET ALL CARTS (FOR DEMO PURPOSE)**

  RQ:
  GET {{service-url}}/api/v1/carts

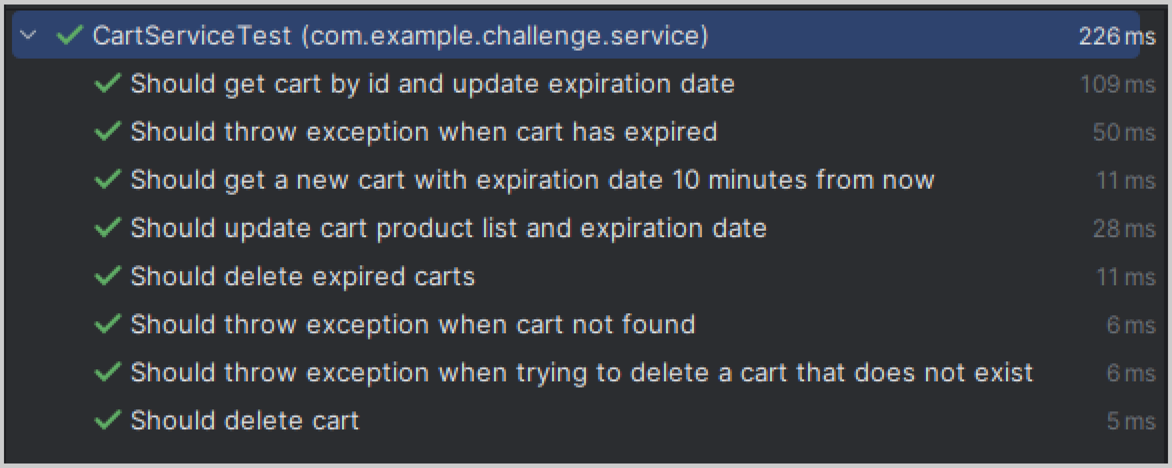
  RS:

```
[
    {
        "uuid": "c5ffc86c-7d67-4656-bbd1-81d22e8da67b",
        "expireAt": "2023-05-02T17:06:35.958",
        "productList": []
    },
    {
        "uuid": "e7f1bd98-50ea-4390-9192-74b7fa144876",
        "expireAt": "2023-05-02T17:06:38.234",
        "productList": []
    },
    {
        "uuid": "f8b5ff7d-c85c-4193-9fa4-5ca25e63a1c0",
        "expireAt": "2023-05-02T17:06:49.075",
        "productList": [
            {
                "id": 1,
                "description": "A new product description",
                "amount": 12.00
            },
            {
                "id": 2,
                "description": "Another product description",
                "amount": 23.00
            },
            {
                "id": 3,
                "description": "One more product description",
                "amount": 4.00
            }
        ]
    },
    {
        "uuid": "1b3496f2-7df8-4b7b-8c10-ec5ce24d3bcb",
        "expireAt": "2023-05-02T17:07:27.747",
        "productList": [
            {
                "id": 4,
                "description": "A new product description",
                "amount": 12.00
            },
            {
                "id": 5,
                "description": "Another product description",
                "amount": 23.00
            },
            {
                "id": 6,
                "description": "One more product description",
                "amount": 4.00
            }
        ]
    }
]
```

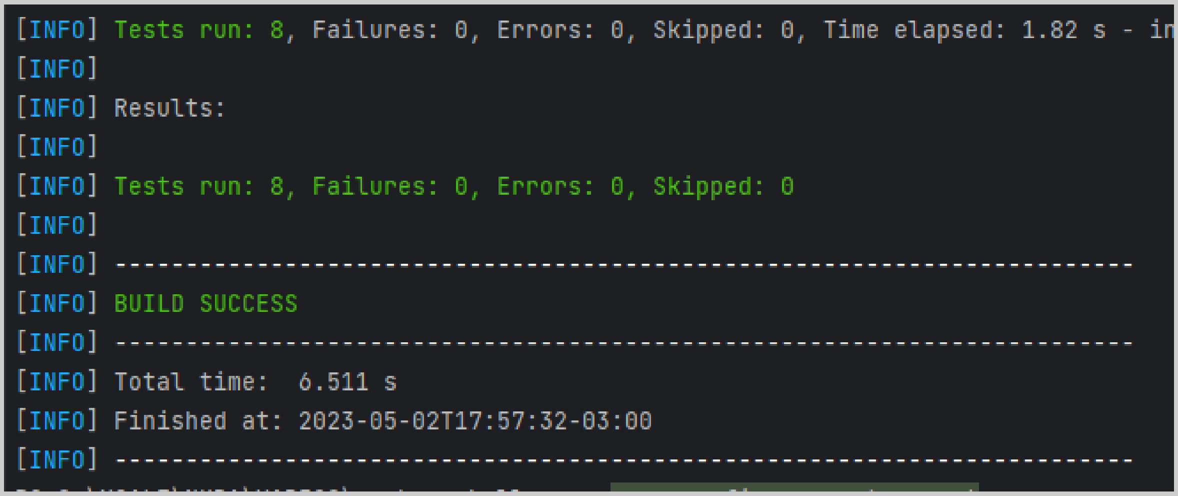
- **INFO:** A postman collection would be attached along with this document to support api check/testing.

Unit Testing:

RUNNING **mvn test**

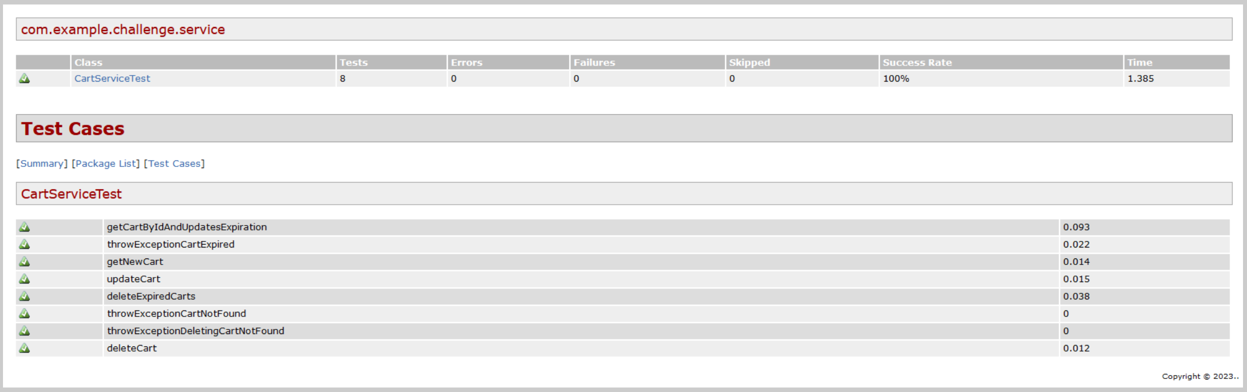


```
⌄ ✓ CartServiceTest (com.example.challenge.service)                              226 ms
    ✓ Should get cart by id and update expiration date                           109 ms
    ✓ Should throw exception when cart has expired                                50 ms
    ✓ Should get a new cart with expiration date 10 minutes from now              11 ms
    ✓ Should update cart product list and expiration date                         28 ms
    ✓ Should delete expired carts                                                 11 ms
    ✓ Should throw exception when cart not found                                   6 ms
    ✓ Should throw exception when trying to delete a cart that does not exist      6 ms
    ✓ Should delete cart                                                           5 ms
```



```
[INFO] Tests run: 8, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.82 s - in
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 8, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  6.511 s
[INFO] Finished at: 2023-05-02T17:57:32-03:00
[INFO] ------------------------------------------------------------------------
```

RUNNING **mvn surefire-report:report**



com.example.challenge.service

	Class	Tests	Errors	Failures	Skipped	Success Rate	Time
✓	CartServiceTest	8	0	0	0	100%	1.385

Test Cases

[Summary] [Package List] [Test Cases]

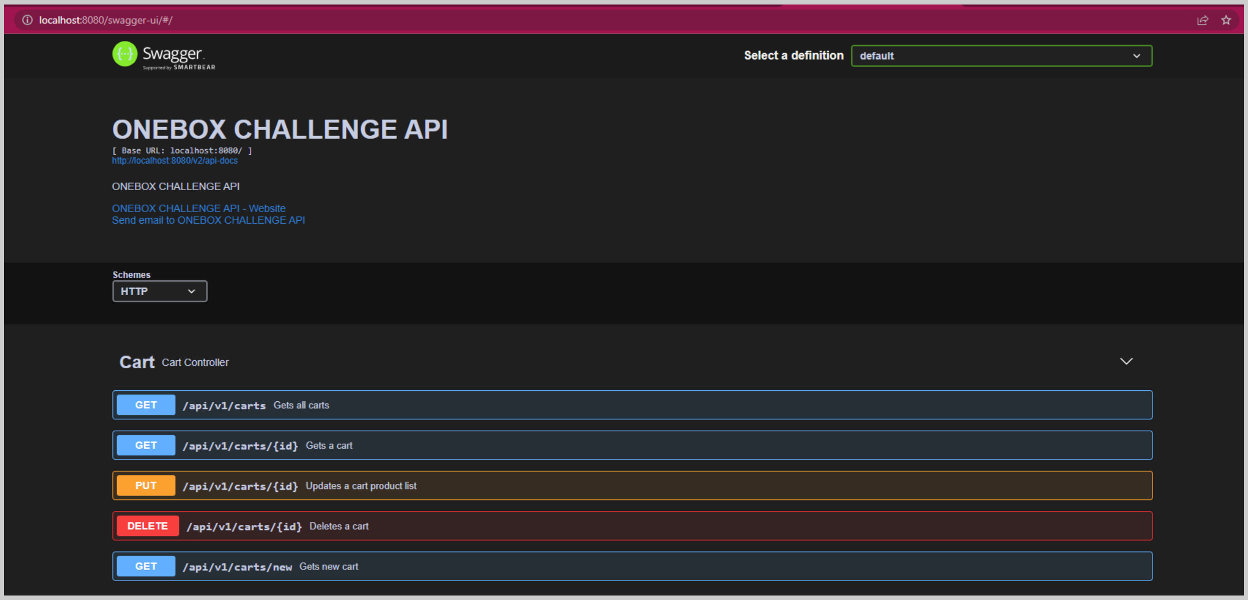
CartServiceTest

✓	getCartByIdAndUpdatesExpiration	0.093
✓	throwExceptionCartExpired	0.022
✓	getNewCart	0.014
✓	updateCart	0.015
✓	deleteExpiredCarts	0.038
✓	throwExceptionCartNotFound	0
✓	throwExceptionDeletingCartNotFound	0
✓	deleteCart	0.012

Copyright © 2023..

Documentation:

URL: {{service-url}}//swagger-ui/



localhost:8080/swagger-ui/#/

Select a definition: default

ONEBOX CHALLENGE API

[Base URL: localhost:8080/]
http://localhost:8080/v2/api-docs

ONEBOX CHALLENGE API

ONEBOX CHALLENGE API - Website
Send email to ONEBOX CHALLENGE API

Schemes: HTTP

Cart Cart Controller

GET /api/v1/carts Gets all carts

GET /api/v1/carts/{id} Gets a cart

PUT /api/v1/carts/{id} Updates a cart product list

DELETE /api/v1/carts/{id} Deletes a cart

GET /api/v1/carts/new Gets new cart