

HW1: Mid-term assignment report

Fábio Matias [108011], v2022-04-07

1 Introduction¹

- 1.1 Overview of the work¹
- 1.2 Current limitations¹

2 Product specification²

- 2.1 Functional scope and supported interactions²
- 2.2 System architecture²
- 2.3 API for developers³

3 Quality assurance⁴

- 3.1 Overall strategy for testing⁴
- 3.2 Unit and integration testing⁴
- 3.3 Functional testing⁴
- 3.4 Code quality analysis⁴
- 3.5 Continuous integration pipeline [optional]⁴

4 References & resources⁵

1 Introduction

1.1 Overview of the work

Este relatório apresenta o trabalho individual de meio de semestre para TQS, cobre as características do produto e a estratégia de controlo de qualidade usada. O Travel Planner, tem como principal objetivo pesquisar e reservar viagens.

1.2 Current limitations

De forma a simplificar a implementação, fiz com que os IDs fossem gerados de forma sequencial, o que constitui um grave problema de segurança num cenário de utilização mais realista. Atualmente as partes funcionais da aplicação incluem apenas um conjunto de viagens muito limitado, que dificilmente poderia ser escalável, até porque não existe nenhuma forma, para além de dar query diretamente na Base de Dados, de adicionar ou eliminar viagens.

As reservas podem apenas ser feitas uma a uma, e a pesquisa só pode ser feita 30 segundos depois da reserva, implementei assim a cache, com um TTL limitado para que seja enfatizado neste cenário de teste.

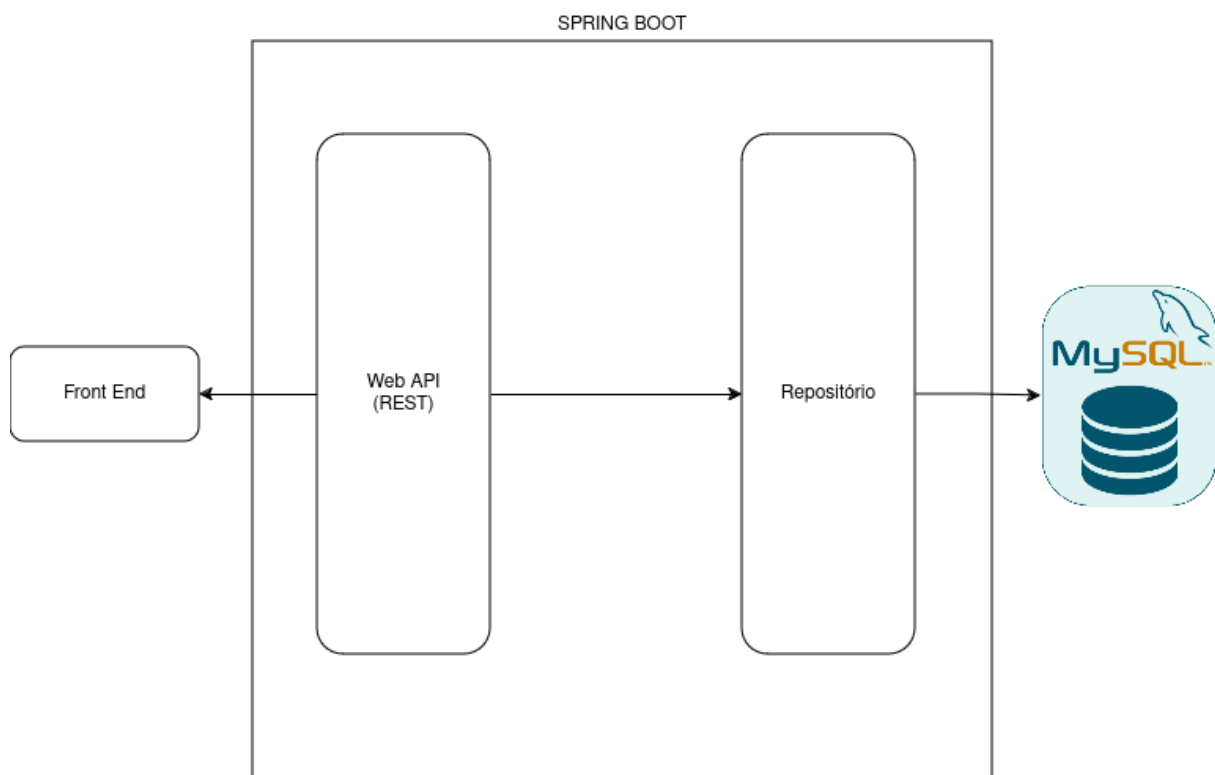
Poderia ainda haver suporte para mais moedas,

2Product specification

2.1 Functional scope and supported interactions

Os passageiros, abrem o site, pesquisam o seu destino, e a sua origem, bem como a data da viagem. Quando os resultados aparecem, carregam no botão de reserva, é-lhes atribuído um ID de reserva que podem usar para pesquisar a reserva feita. Podem ainda a qualquer altura seleccionar a moeda desejada.

2.2 System architecture



2.3 API for developers

trip-controller

POST /api/trips/reserve

Parameters

No parameters

Request body required

application/json

Example Value | Schema

```
{
  "id": 0,
  "origin": "string",
  "destination": "string",
  "date": "string",
  "priceEuro": 0
}
```

GET /api/trips/search

GET /api/trips/reservations/{id}

Parameters

Try it out

| Name | Description |
|-----------------------------------|-------------|
| Id <small>required</small> | |
| string | id |
| (path) | |

Responses

| Code | Description | Links |
|------|-------------|----------|
| 200 | OK | No links |

Media type
/

Controls Accept header.

Example Value | Schema

```
{
  "id": 0,
  "trip": {
    "id": 0,
    "origin": "string",
    "destination": "string",
    "date": "string",
    "priceEuro": 0
  },
  "timestamp": "2024-04-09T23:00:37.429Z"
}
```

Schemas

| Code | Description | Links |
|------|-------------|----------|
| 200 | OK | No links |

Media type
/

Controls Accept header.

Example Value | Schema

```
{
  "id": 0,
  "origin": "string",
  "destination": "string",
  "date": "string",
  "priceEuro": 0
}
```

GET /api/trips/reservations/{id}

3Quality assurance

3.1 Overall strategy for testing

Todos os componentes testados, foram testados de forma isolada, foi usado o Junit, “test containers”, usei ainda mocks. Comecei por fazer a aplicação, e só depois comecei a implementar os testes.

3.2 Unit and integration testing

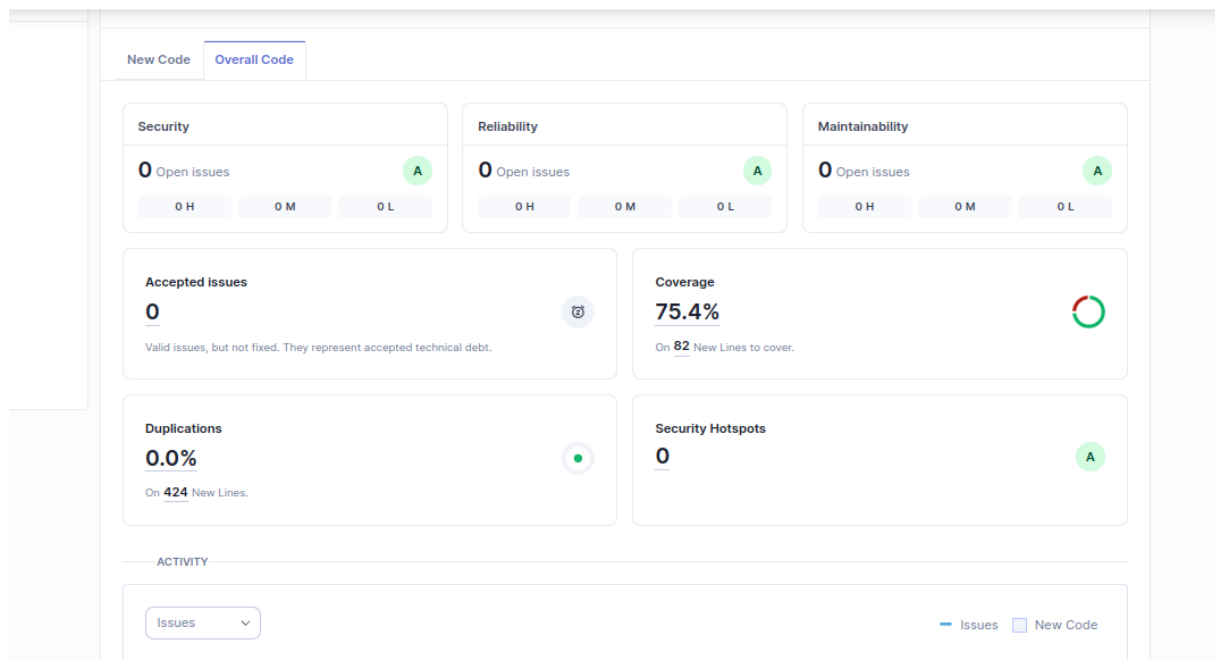
Testei os componentes individuais, como métodos e construtores usando testes unitários, nos testes de integração, implementei testes ao nível do REST API, testando com valores fictícios o funcionamento dos controladores.

3.3 Functional testing

Não foram implementados

3.4 Code quality analysis

Usei o SonarQube para corrigir erros, e aumentar a qualidade dos testes feitos.



3.5 Continuous integration pipeline [optional]

Não foi implementado.

4References & resources

Project resources

| Resource: | URL/location: |
|-------------------------|--|
| Git repository | https://github.com/SlicF/TQS_108011 |
| Video demo | homework_assignment/demo.mp4 |
| QA dashboard (online) | [optional ; if you have a quality dashboard available online (e.g.: sonarcloud), place the URL here] |
| CI/CD pipeline | [optional ; if you have th CI pipeline definition in a server, place the URL here] |
| Deployment ready to use | [optional ; if you have the solution deployed and running in a server, place the URL here] |

Reference materials