

SwipeRight

Testing

Luka Spaninks
Semester 6
RB03

Version 1

Preface

The complete testing phase of the project SwipeRight can be found in this document. This file will evolve over time and can always be expanded upon.

Inhoud

Testing Strategy.....	3
Unit Testing	3
Static Code Analysis.....	3
Load Testing	4
Integration Testing & e2e testing.....	4

Testing Strategy

All tests will be automatically ran in a pipeline, specifically GitHub actions.

Unit Testing

For every service I will be writing unit tests using the native go test packages. These tests will be executed in a pipeline and the coverage is uploaded to Codecov. I am aware that coverage doesn't say a lot about the quality of the code but it does motivate you to think of more test cases.

Static Code Analysis

Every time changes to a service are pushed to the remote repository, a static code analysis will be executed. I am using the [Staticcheck](https://staticcheck.io/docs/checks/) package for go which has a lot of checks:
<https://staticcheck.io/docs/checks/>.

I am also using go vet which is natively build into Go. Its purpose is to check for and report suspicious constructs.



Load Testing

For performance/load testing I am using k6 from Grafana labs. The idea is to run these automatically in the pipeline. The tests can be defined by JavaScript and can easily be stored in scripts. The tests do not run in Node but in Go which makes it way more efficient.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  GITLENS

  A K6 .io

execution: local
  script: .\load\script.js
  output: -

scenarios: (100.00%) 1 scenario, 20 max VUs, 3m30s max duration (incl. graceful stop):
    * default: Up to 20 looping VUs for 3m0s over 3 stages (gracefulRampDown: 30s, gracefulStop: 30s)

[
running (0m14.4s), 05/20 VUs, 28 complete and 0 interrupted iterations
default  [==>-----] 05/20 VUs  0m14.4s/3m00.0s
```

Integration Testing & e2e testing

If there is time left it would be nice to add integration tests and later on end-to-end tests.