PROJECT-BASED VIRTUAL INTERNSHIP

SOFTWARE QUALITY ASSURANCE

EVERMOS X RAKAMIN ACADEMY

Created by

Mohammad Ibadul Haqqi

Rakamin Academy evermos

TABLE OF CONTENTS

INTRODUCTION

This Program Project-Based Internship will provide me with virtual internship experience as a Quality Assurance at Evermos. I will be positioned as a Quality Assurance Intern who will be exposed to issues, case studies, and projects that are part of Evermos' daily operations.

Quality Assurance is a series of systematic processes to determine whether a product or service meets specified requirements. QA determines and sets requirements for creating or developing specific products to ensure they have good quality.

QA can be integrated into all stages of development and used even after the post-release stage. QA specialists create and implement various strategies for software quality improvement using SDLC methods. They implement the necessary types of testing to ensure that the software will function correctly.

INTEGRATION TESTING

Create test scenarios for integration tests from these 2 APIs and implement them in K6 and add assertions from each test carried out in 1 test file.

- API Create
  - Base url : https://reqres.in
  - Path url : /api/users
  - Method : POST
  - Header : application/json
  - Request body :

```
{
    "name": "morpheus",
    "job": "leader"
}
```

- API Update
  - Base url : https://reqres.in
  - Path url : /api/users/2
  - Method : PUT
  - Header : application/json
  - Request body :

```
{
    "name": "morpheus",
    "job": "zion resident"
}
```

REPORT
INTEGRATION TESTING

```
const BASE_URL = 'https://reqres.in';


export default function () {
  const name = 'morpheus'
  const job = 'zion resident'
```

BASE_URL, name, dan **job** adalah paramater yang digunakan untuk mengirim data

```
group('Create with valid request should success', function () {
  const FULL_URL = BASE_URL + '/api/users';
  const payload = JSON.stringify({
    name : name,
    job: job
  })
  const params = {
    headers: {
      'Content-Type': 'application/json',
    },
  };
```

Untuk request Create User

```
group('Update with valid request should success', function () {
  const FULL_URL = BASE_URL + '/api/users/2';
  const payload = JSON.stringify({
    name : name,
    job: job
  })
  const params = {
    headers: {
      'Content-Type': 'application/json',
    },
  };
```

Untuk request Update User

- **FULL_URL:** endpoint dari API yang akan diuji
- Data **payload** yang terdiri dari **name** dan **job** yang telah dibuat sebelumnya diubah menjadi string JSON sebelum dikirimkan sesuai request user yang diminta.

REPORT
INTEGRATION TESTING

```
check(res, {
    'response code was 201': (res) => res.status == 201,
});
check(res, {
    'response name should same with request': (res) => {
        const response = JSON.parse(res.body);
        return response.name === name
    },
});
check(res, {
    'response job should same with request': (res) => {
        const response = JSON.parse(res.body);
        return response.job === job
    },
});
```

Assertion untuk check Pengujian API Create User

- **Response code was 201:** Memeriksa apakah status respons adalah 201 Created, yang menunjukkan bahwa permintaan berhasil.
- **Response name should same with request:** Memeriksa apakah status response pada name sesuai yang diinginkan
- **Response job should same with request:** Memeriksa apakah status response pada job sesuai yang diinginkan

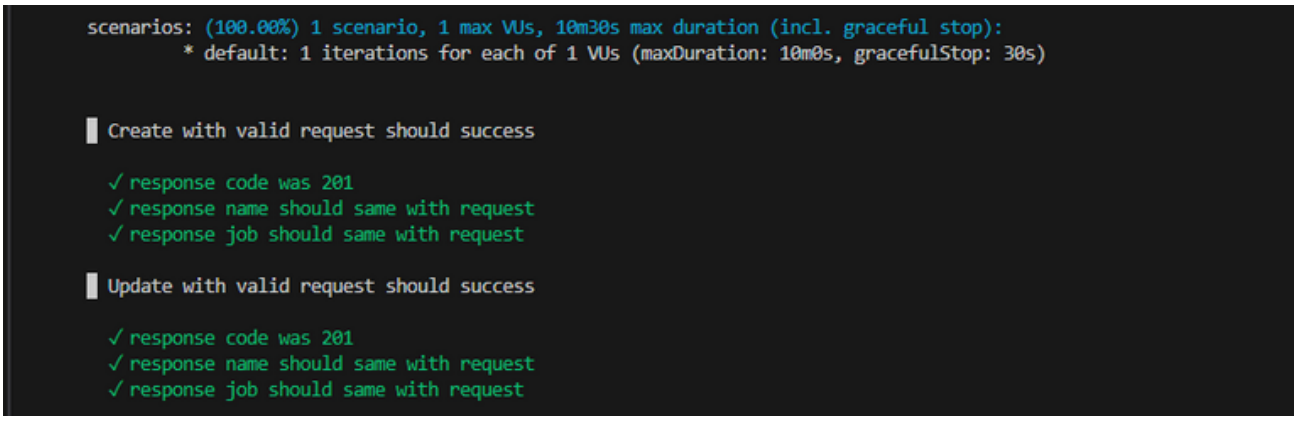
```
check(res, {
    'response code was 201': (res) => res.status == 200,
});
check(res, {
    'response name should same with request': (res) => {
        const response = JSON.parse(res.body);
        return response.name === name
    },
});
check(res, {
    'response job should same with request': (res) => {
        const response = JSON.parse(res.body);
        return response.job === job
    },
});
});
```

Assertion untuk check Pengujian API Update User

- **Response code was 200:** Memeriksa apakah status respons adalah 200 OK, yang menunjukkan bahwa permintaan untuk memperbarui pengguna berhasil.
- **Response name should same with request:** Memeriksa apakah status response pada job sesuai yang diinginkan
- **Response job should same with request:** Memeriksa apakah status response pada job sesuai yang diinginkan

Mohammad Ibadul Haqqi | Project Based Internship

RESULT
INTEGRATION TESTING



Dengan demikian, hasil pengujian menunjukkan bahwa sistem yang diuji berkinerja baik dan memenuhi semua kriteria yang diharapkan selama pengujian. Semua permintaan berhasil, tidak ada metrik yang melebihi ambang batas, dan semua pemeriksaan dilakukan dengan sukses.

PERFORMANCE TESTING

Create a test scenario to test the performance of these 2 APIs with a total of 1000 virtual users, 3500 iterations, and a maximum API response tolerance limit of 2 seconds and add assertions from each test carried out in 1 test file.

- API Create
  - Base url : https://reqres.in
  - Path url : /api/users
  - Method : POST
  - Header : application/json
  - Request body :

```
{
    "name": "morpheus",
    "job": "leader"
}
```

- API Update
  - Base url : https://reqres.in
  - Path url : /api/users/2
  - Method : PUT
  - Header : application/json
  - Request body :

```
{
    "name": "morpheus",
    "job": "zion resident"
}
```

```
export const options = {
    vus: 1000,
    iterations: 3500,
    thresholds: {
        http_req_duration: ['avg < 2000'], //reponse API max 2s
    },
};
```

- **vus: 1000:** Ini menetapkan jumlah pengguna virtual (Virtual Users - VUs) yang akan digunakan selama pengujian. Dalam hal ini, jumlahnya adalah 1000.
- **iterations: 3500:** Ini menetapkan jumlah iterasi atau siklus yang akan dieksekusi selama pengujian. Dalam hal ini, ada 3500 iterasi.
- **thresholds:** Ini adalah kriteria atau ambang batas yang digunakan untuk mengevaluasi kinerja aplikasi selama pengujian. Dalam hal ini, kita memiliki satu ambang batas yang dinamai **http_req_duration**. Ambang batas ini menetapkan bahwa durasi maksimum dari permintaan HTTP (http_req_duration) tidak boleh melebihi **2000** milidetik (2 detik).

REPORT
PERFORMANCE TESTING

```
const BASE_URL = 'https://reqres.in';

export default function () {
  const name = 'morpheus'
  const job = 'zion resident'
```

BASE_URL, name, dan **job** adalah paramater yang digunakan untuk mengirim data

```
group('Create with valid request should success', function () {
  const FULL_URL = BASE_URL + '/api/users';
  const payload = JSON.stringify({
    name : name,
    job: job
  })
  const params = {
    headers: {
      'Content-Type': 'application/json',
    },
  };
```

Untuk request Create User

```
group('Update with valid request should success', function () {
  const FULL_URL = BASE_URL + '/api/users/2';
  const payload = JSON.stringify({
    name : name,
    job: job
  })
  const params = {
    headers: {
      'Content-Type': 'application/json',
    },
  };
```

Untuk request Update User

- **FULL_URL:** endpoint dari API yang akan diuji
- Data **payload** yang terdiri dari **name** dan **job** yang telah dibuat sebelumnya diubah menjadi string JSON sebelum dikirimkan sesuai request user yang diminta.

```
import { htmlReport } from "https://raw.githubusercontent.com/benc-uk/k6-reporter/main/dist/bundle.js";
```

```
export function handleSummary(data) {
  return {
    "./report.html": htmlReport(data),
  };
}
```

- Ikuti referensi https://github.com/benc-uk/k6-reporter atau dengan tambahkan code seperti diatas untuk dapat menampilkan report dalam bentuk html

REPORT
PERFORMANCE TESTING

```
check(res, {
    'response code was 201': (res) => res.status == 201,
});
check(res, {
    'response name should same with request': (res) => {
        const response = JSON.parse(res.body);
        return response.name === name
    },
});
check(res, {
    'response job should same with request': (res) => {
        const response = JSON.parse(res.body);
        return response.job === job
    },
});
```

Assertion untuk check Pengujian API Create User

- **Response code was 201:** Memeriksa apakah status respons adalah 201 Created, yang menunjukkan bahwa permintaan berhasil.
- **Response name should same with request:** Memeriksa apakah status response pada name sesuai yang diinginkan
- **Response job should same with request:** Memeriksa apakah status response pada job sesuai yang diinginkan

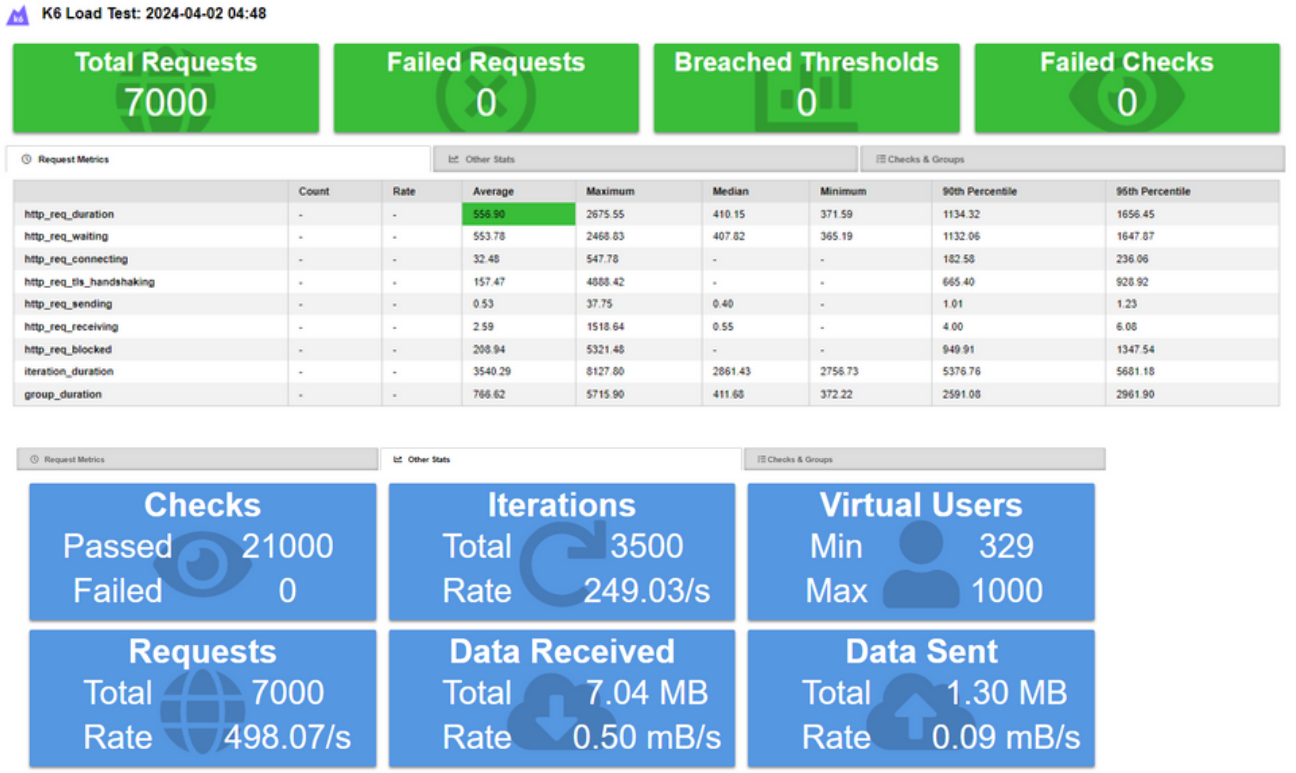
```
check(res, {
    'response code was 201': (res) => res.status == 200,
});
check(res, {
    'response name should same with request': (res) => {
        const response = JSON.parse(res.body);
        return response.name === name
    },
});
check(res, {
    'response job should same with request': (res) => {
        const response = JSON.parse(res.body);
        return response.job === job
    },
});
});
```

Assertion untuk check Pengujian API Update User

- **Response code was 200:** Memeriksa apakah status respons adalah 200 OK, yang menunjukkan bahwa permintaan untuk memperbarui pengguna berhasil.
- **Response name should same with request:** Memeriksa apakah status response pada job sesuai yang diinginkan
- **Response job should same with request:** Memeriksa apakah status response pada job sesuai yang diinginkan

Mohammad Ibadul Haqqi | Project Based Internship

RESULT
PERFORMANCE TESTING



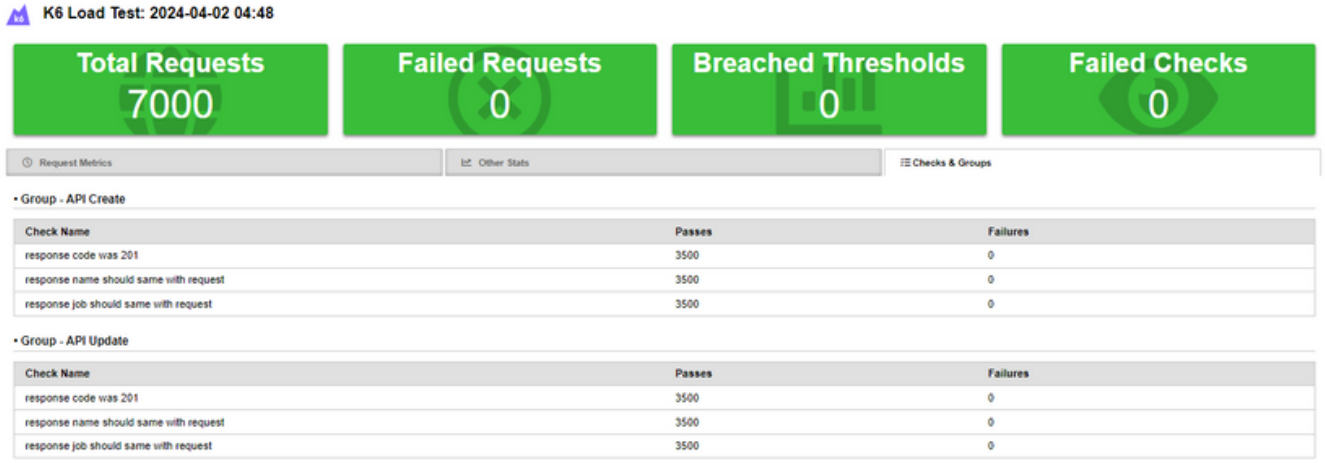
	Count	Rate	Average	Maximum	Median	Minimum	90th Percentile	95th Percentile
http_req_duration	-	-	556.90	2675.55	410.15	371.59	1134.32	1656.45
http_req_waiting	-	-	553.78	2468.83	407.82	365.19	1132.06	1647.87
http_req_connecting	-	-	32.48	547.78	-	-	182.58	236.06
http_req_tls_handshaking	-	-	157.47	4888.42	-	-	665.40	928.92
http_req_sending	-	-	0.53	37.75	0.40	-	1.01	1.23
http_req_receiving	-	-	2.59	1518.64	0.55	-	4.00	6.08
http_req_blocked	-	-	208.94	5321.48	-	-	949.91	1347.54
iteration_duration	-	-	3540.29	8127.80	2861.43	2756.73	5376.76	5681.18
group_duration	-	-	766.62	5715.90	411.68	372.22	2591.08	2961.90

- Tidak ada permintaan yang gagal selama pengujian. Semua 7000 permintaan berhasil dilakukan tanpa kegagalan.
- Tidak ada metrik yang melebihi ambang batas yang ditetapkan selama pengujian. Semua metrik berada dalam batas yang diharapkan.
- Tidak ada pemeriksaan (checks) yang gagal selama pengujian. Semua pemeriksaan berhasil dilakukan tanpa masalah.
- Rata-rata yang dihasilkan pada **http_req_duration** sesuai yang diharapkan dengan tidak boleh melebihi 2000 milidetik (2 detik).

Detail iterasi:

- Total Permintaan: 7000 dengan kecepatan 498.07/s.
- Virtual Users: 3500 dalam tes.
- Durasi Iterasi: Rata-rata 766.62s, maksimum 572s. Persentil ke-90 dan ke95: 259s dan 2965s.

RESULT
PERFORMANCE TESTING



Assertion yang ditampilkan ke dalam website menunjukkan bahwa:

- **Request API Create** dengan response **code was 201, name should same, job was should same** berstatus **passed** dengan nilai request masing-masing adalah 3500 yang artinya masing-masing dari response tersebut sesuai yang diharapkan.
- **Request API Update** dengan response **code was 200, name should same, job was should same** berstatus **passed** dengan nilai request masing-masing adalah 3500 yang artinya masing-masing dari response tersebut sesuai yang diharapkan.

CONCLUSIONS

Dalam task ini, kami mengembangkan skenario pengujian **integration testing** dan **performance testing** untuk dua API yang disediakan oleh **reqres.in**. Skenario integrasi mencakup **CREATE** user baru dan **UPDATE** user yang telah ada sebelumnya, dengan pengecekan respons untuk memverifikasi status, tipe konten, dan data yang benar. Di sisi lain, skenario performa mengevaluasi respons API dalam situasi beban tinggi dengan 1000 pengguna virtual selama 30 detik, dengan request 3500 setiap respons harus diterima dalam waktu kurang dari 2 detik. Melalui pengujian ini, kami mendapatkan hasil bahwa API yang digunakan bekerja sesuai harapan dan memenuhi standar kinerja yang ditetapkan, yang merupakan langkah kunci untuk memastikan kehandalan dan responsivitas sistem secara keseluruhan.