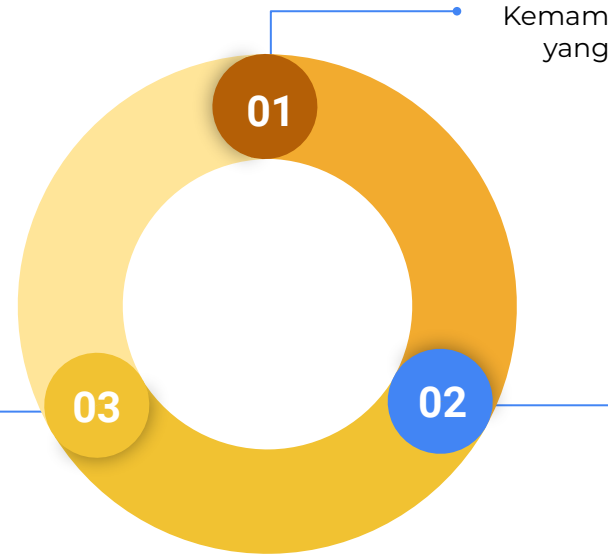


Fullstack Web Challenge

Gold Chapter 6

ACCEPTANCE CRITERIA

Kriteria pengumpulan
challenge yang harus
kamu penuhi untuk
dapat melewati chapter



SKILL METRICS

Kemampuan teknis
yang akan kamu
pelajari

DELIVERY

Hal yang akan kamu
lakukan untuk dapat
melewati chapter

SKILL METRICS

(Kemampuan teknis yang akan kamu pelajari)

1. Design Pattern
 - Design Pattern
 - MVC (Model View Controller)
 - Service Repository Pattern
 - Microservice Pattern
2. Asynchronous Process
 - Process
 - Synchronous Process & Asynchronous Process
 - Callback
 - Promises
3. Authentication
 - Authentication
 - Encryption
 - Authorization
 - Session Based Authentication
 - Token Based Authentication
4. Open API
 - Dokumentasi API
 - Struktur dokumen Open API
 - Open API Tools
 - Swagger UI & Redoc in Express
 - Mock UP API

DELIVERY

(Hal yang akan kamu lakukan untuk dapat melewati chapter)

1. Design Pattern

- 1.1. Memahami konsep dan kegunaan Design Pattern
- 1.2. Merancang struktur project dengan menggunakan Design Pattern
- 1.3. Membuat sebuah aplikasi dengan menggunakan MVC
- 1.4. Membuat aplikasi dengan menggunakan Service Repository Pattern
- 1.5. Memahami Architectural Pattern, Microservice & Monolith

2. Asynchronous Process

- 2.1. Memahami apa itu Asynchronous Process
- 2.2. Menggunakan callback dalam handle asynchronous process
- 2.3. Mampu membuat sebuah Promise dan menggunakannya dalam handle asynchronous process
- 2.4. Mampu membuat Promise dengan menggunakan `async/await` syntax

DELIVERY

(Hal yang akan kamu lakukan untuk dapat melewati chapter)

3. Authentication

- 1.1. Memahami konsep Authentication
- 1.2. Membuat sebuah fungsi register yang mana menggunakan enkripsi untuk menyimpan password dari user
- 1.3. Membuat fungsi login yang melakukan komparasi plain password yang dikirim oleh user melalui dengan encrypted password yang disimpan di dalam database
- 1.4. Memahami konsep authorization
- 1.5. Mengimplementasikan Session Based Authentication menggunakan express-session
- 1.6. Membuat fungsi login untuk REST API yang merespon dengan JWT
- 1.7. Membuat middleware untuk memverifikasi JWT

DELIVERY

(Hal yang akan kamu lakukan untuk dapat melewati chapter)

4. Open API

- 1.1. Memahami guna mendokumentasikan API
- 1.2. Mengenal struktur dokumen Open API
- 1.3. Memahami cara kerja Open API dan Swagger UI
- 1.4. Membuat dokumentasi API untuk sebuah CRUD operation
- 1.5. Menampilkan dokumentasi API di dalam express.js
- 1.6. Membuat Mock API berdasarkan Open API yang sudah dibuat
- 1.7. Memahami fungsi Mock Up API

CRITERIA

(Kriteria pengumpulan challenge yang harus kamu penuhi untuk dapat melewati chapter)

- Mampu menerapkan Service Repository Pattern di dalam sebuah Project **(1 point)**
- Mampu membuat asynchronous function dan menjalankannya **(1 point)**
- Mampu menerapkan Token Based Authentication sebagai metode autentikasi di dalam REST API **(4 point)**
- Mampu membuat Open API Documentation dari REST API yang akan dibuat **(4 point)**

Buatlah sebuah REST API yang dapat digunakan untuk melakukan manajemen data mobil dengan fitur authentication~



Definition of Done 🤖

- Terdapat endpoint untuk login sebagai superadmin, admin, maupun member.
- Terdapat endpoint untuk menambahkan admin yang mana hanya boleh dilakukan oleh superadmin.
- Terdapat endpoint untuk registrasi sebagai member.
- Terdapat 4 endpoint untuk **melakukan CRUD** terhadap data mobil, dan hanya admin dan superadmin saja yang dapat melakukan operasi tersebut.
- Terdapat endpoint untuk melihat daftar mobil yang tersedia.
- Terdapat endpoint untuk melihat **current user** dari token yang dimiliki.



Definition of Done 🤖

- Setiap data mobil mempunyai informasi berikut:
 - Siapa yang membuat data tersebut
 - Siapa yang menghapus data tersebut
 - Siapa yang terakhir kali mengupdate data tersebut
- Menggunakan Service Repository Pattern dalam membangun project ini.
- Terdapat halaman yang menampilkan dokumentasi API, baik itu menggunakan Swagger UI, Redoc atau Library lain di dalam HTTP Server tersebut.
- Terdapat endpoint yang merespon dengan Open API document dari REST API yang dibangun dalam bentuk JSON.



Quick Brief

1. Gunakan **express** dan **sequelize**, untuk memenuhi kebutuhan **CRUD**.
2. Gunakan **bcrypt** atau **bcryptjs** untuk mengenkripsi password user.
3. Gunakan **passport** atau **jsonwebtoken**, untuk memenuhi kebutuhan **Token Based Authentication**.
4. Gunakan **Swagger Editor**, **Stoplight Studio**, atau tools lain dalam menulis **Open API Documentation**.
5. Gunakan **Swagger**, **Redoc**, **Stoplight Element**, atau Tools yang memvisualisasikan **Open API documentation** lainnya.
6. Gunakan **seeders** untuk **menginsert data superadmin**.



Submission Guidelines 🔥

1. Buatlah sebuah repository di GitHub dengan nama yang representatif dari challenge ini, sebagai contoh: **Car Management API**
2. Jadikan repository tersebut sebagai **node.js** project dan kerjakan project tersebut sesuai dengan requirement dari challenge ini.
3. Cantumkan halaman ke Open API Documentation dan endpoint untuk mendapatkan Open API Document dalam bentuk JSON, di dalam **README**.
4. Cantumkan **email** dan **password** dari **superadmin** di dalam **README**.
5. Submit form pengumpulan challenge, dan kirim email ke fasilitator-mu dengan subject **Binar: Full-stack Web Engineering - Chapter 06 / Challenge Submission**.



Happy Hacking