



RECRUITMENT TEST for Backend Software Engineer

! Baca baik baik petunjuk dibawah ini hingga selesai sebelum memulai test ini. !

Anda punya maksimum 24 jam untuk menyelesaikan tes. Waktu pengerjaan dimulai dari saat dokumen ini dikirim kepada Anda. Anda diminta untuk membuat sebuah REST API berdasarkan requirement dokumen ini.

Buatlah sebuah service yang terdiri dari berbagai endpoint yang berdasarkan REST API, yang memiliki kriteria sebagai berikut : (contoh ada di halaman terakhir)

- 1) Buatlah sebuah public API endpoint, sehingga user dapat melakukan proses registrasi & melakukan proses login.

Untuk autentikasi endpoint, silahkan merancang struktur database yang menurut anda terbaik. Anda juga dapat menambahkan fitur yang dirasa baik untuk di implementasi, seperti rate limiting, fail2ban, dll.

- 2) Buatlah sebuah public API endpoint, sehingga user dapat menampilkan random quotes/facts dari API yang sudah disiapkan.

Anda dapat mengambil data random quotes/facts dari API sebagai berikut:

- Chuck Norris Jokes API (<https://api.chucknorris.io/jokes/random>)
- Dog Facts API (<https://dog-facts-api.herokuapp.com/api/v1/resources/dogs?number>)
- Cat Facts API (<https://catfact.ninja/fact>)

- 3) Buatlah sebuah **private API** menggunakan data soal nomor 1. Terapkan database locking dan concurrency dengan requirement dibawah

pseudocode:

1. If input.amount = 0.00000001, decline
2. find balance by *user_id* from table balance,
3. if balance.amount_available < input.amount, decline (*insufficient*)
4. Insert table transaction
5. add sleep/delay 30 seconds
6. If *trx_id* exists, decline & rollback transaction
7. update balance.amount_available, (*balance.amount_available - input.amount*)
8. Return data transaction & balance by *user_id*. Trim (not rounding) to 6 digit decimal and not scientific e notation (ex: 3.456e11)

database:

1. silahkan running query (dump) dibawah
2. alter table transaction set *trx_id* = unique
3. alter table balance set *user_id* = index

```
CREATE TABLE `transaction` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `trx_id` varchar(255) NOT NULL DEFAULT '',  
  `user_id` bigint(20) NOT NULL DEFAULT '0',  
  `amount` decimal(10,0) NOT NULL DEFAULT '0',  
  `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
INSERT INTO `transaction` (`id`, `trx_id`, `user_id`, `amount`,  
  `created_at`, `updated_at`) VALUES  
(1, 'a', 1, 0.01000000, '2022-03-07 09:55:44', '2022-03-07 09:55:44'),  
(2, 'B', 1, 0.02000000, '2022-03-07 09:55:44', '2022-03-07 09:55:44');  
  
CREATE TABLE `balance` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `user_id` bigint(20) NOT NULL DEFAULT '0',  
  `amount_available` decimal(10,0) NOT NULL DEFAULT '0',  
  `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=utf8;  
  
INSERT INTO `balance` (`id`, `user_id`, `amount`, `created_at`,  
  `updated_at`) VALUES  
(1, 1, 0.00674223, '2022-03-07 09:57:13', '2022-03-07 09:57:13'),  
(2, 2, 1.00000000, '2022-03-07 09:57:13', '2022-03-07 09:57:13'),  
(3, 3, 0.00000001, '2022-03-07 09:57:13', '2022-03-07 09:57:13'),  
(4, 4, 21.00000000, '2022-03-07 09:57:13', '2022-03-07 09:57:13');
```

note:

1. *test race condition, silahkan call API kurang dari waktu delay*
2. **user_id = id dari table user yang register** dari task/soal nomor 1.

- 4) Buatlah sebuah private API endpoint, sehingga user yang sudah login, dapat melakukan proses upload CSV. Data CSV ini, nantinya akan di baca, dan di insert ke dalam database, skema table nya, bebas. Currency yang digunakan adalah IDR & USD.

File CSV, akan diberikan, ukuran file tersebut sekitar 80MB (zipped), dan jumlah row adalah sekitar 4.667.920 rows

File CSV dapat di unduh [disini](#)

- 5) Buatlah dua buah private endpoint, sehingga user dapat memperoleh data, data history (data yang di upload melalui CSV, dari soal nomor 4), untuk mendapatkan:
 - a) Mengambil harga terendah dan tertinggi dari periode yang dipilih, berdasarkan parameter yang dikirimkan.
 - b) Mengambil harga historikal, dari jangka waktu, berdasarkan parameter yang dikirimkan.

Dari data yang yang di upload di soal nomor 4, buatlah dua buah endpoint, dengan speks di bawah.



RECRUITMENT TEST for Backend Software Engineer

Spesifikasi Teknis

Bangun sebuah backend sederhana berbasis REST API sesuai dengan spesifikasi & ketentuan dibawah ini.

Pilihan bahasa pemrograman

- PHP menggunakan Laravel
- Node.JS menggunakan express / Koa (Javascript/Typescript)

Mengenai Database

- Implementasi menggunakan varian database MySQL / Postgres
- Desain database bebas
- Setiap tabel harus memiliki kolom ID yang unique (auto increment) & menjadi primary key
- Boleh juga **menambahkan** database jenis lain, seperti noSQL, jika dibutuhkan

Penilaian & submission

- Untuk kemudahan penilaian, buatlah Postman API List dari API yang dikembangkan. Export Postman API List, sertakan di submission.
- Task nomor 1,2,3 **wajib** untuk dikerjakan (Estimasi waktu pengerjaan adalah 4 jam)
- Task nomor 4,5 **tidak wajib** untuk dikerjakan, tetapi jika dikerjakan, highly considered.
- Yang perlu disertakan di submission:
 - MySQL Database dump / migrations
 - Source code
 - README yang berisikan dokumentasi & instruksi build
- Pilih salah satu metode submission:
 - **Kirim Email** ke putri.sri@usenobi.com dengan subject "Nama Anda - Backend Software Engineer". Pastikan semua submission sudah Anda attach di email.
 - **Github** share github repository ke user GitHub **anwar-github**, **ace3** dan **ipan.ardian**

Faktor Penilaian

- Pemahaman requirements
- Feature completion
- Logika dan Validasi
- Negative Case
- Error Handling
- Code cleanliness
- Creativity
- Dokumentasi
- Point extra untuk extra effort (silahkan dijelaskan di README)



RECRUITMENT TEST for Backend Software Engineer

REST API Endpoint yang perlu dibangun:

#1a POST /api/v1/auth/register	
Input	email, password
Output	user_id, status, message
Logic	Menambahkan user. email sifatnya unique jadi tidak boleh sama

#1b POST /api/v1/auth/login	
Input	email, password
Output	token
Logic	Mengecek, apakah email ada, dan password sesuai

#2 GET /api/v1/quote	
Input	-
Output	quote, status, source
Logic	Tampilkan quotes/facts, berdasarkan sumber data yang di set di backend. Dan dapat diganti sewaktu-waktu jika dibutuhkan.

#3 POST /api/v1/transaction	
Input	<string>trx_id, <float>amount, <integer>user_id
Output	<string>trx_id, <string>amount, <string>message

#4 POST /api/v1/price/upload	
Input	file
Output	status, message
Logic	Hanya dokumen CSV yang boleh diolah, filesize maximal adalah 100Mb

#5a POST /api/v1/price/low-high	
Input	week, year, ticker, currency
Output	data: Array of { min, max, week, year, pair }, status , message



nobi

RECRUITMENT TEST

for Backend Software Engineer

Logic	
-------	--

#5b POST /api/v1/price/history	
Input	timeframe, ticker, currency
Output	data: Array of { datetime, price }, status , message
Logic	