

Buatlah REST API untuk fitur-fitur di bawah ini:

1. Register
2. Login
3. Top Up
4. Payment
5. Transfer
6. Report Transactions
7. Update Profile

Notes:

Transferring money should be executed in background server.

Several hints how you can do this:

- A script that can run in background server.
- Having a queue task that will transfer money to another user (bonus point).

Optional Criteria:

- Using an ORM and migrations script to manage your database
- Building a unit test (bonus point)
- Having a dashboard to monitor all the background process (bonus point)

Code Consideration:

- Source control familiarity
- Code quality and maintainability
- Error handling

Tech Specs:

- Must be built in Python 3+, Flask Microframework
- Using MongoDB as database

Task Submission:

- Source code must be uploaded to a source control environment (Github, Gitlab, or Bitbucket)
- Please dump example database in a file called `*example_data*` for better testing

1. Register

Digunakan untuk mendaftarkan user baru.

user_id menggunakan format **uuid**.

Phone Number adalah unique key.

URL	/register
METHOD	POST
Request JSON	{ "first_name": "Guntur", "last_name": "Saputro", "phone_number": "0811255501", "address": "Jl. Kebon Sirih No. 1", "pin": "123456" }
Response JSON SUCCESS	{ "status": "SUCCESS" "result": { "user_id": "bc1c823e-b0fb-4b20-88c0-dff25e283252", "first_name": "Guntur", "last_name": "Saputro", "phone_number": "0811255501", "address": "Jl. Kebon Sirih No. 1", "created_date": "2021-4-1 22:21:20" } }
Response JSON FAILED	{ "message": "Phone Number already registered" }

2. Login

Digunakan untuk melakukan login.

Gunakan JWT sebagai autentikasi.

URL	/login
METHOD	POST
Request JSON	<pre>{ "phone_number": "0811255501", "pin": "123456" }</pre>
Response JSON SUCCESS	<pre>{ "status": "SUCCESS" "result": { "access_token": "{access_token}", "refresh_token": "{refresh_token}" } }</pre>
Response JSON FAILED	<pre>{ "message": "Phone number and pin doesn't match." }</pre>

3. Top Up

Digunakan untuk menambahkan saldo user.

top_up_id menggunakan format **uuid**.

URL	/topup
HEADERS	Authorization: Bearer {jwt_token}
METHOD	POST
Request JSON	{ "amount": 500000 }
Response JSON SUCCESS	{ "status": "SUCCESS" "result": { "top_up_id": "201ddde1-f797-484b-b1a0-07d1190e790a", "amount_top_up": 500000, "balance_before": 0, "balance_after": 500000, "created_date": "2021-04-01 22:21:21" } }
Response JSON FAILED	{ "message": "Unauthenticated" }

4. Payment

Digunakan untuk melakukan pembelian.

payment_id menggunakan format **uuid**.

URL	/pay
HEADERS	Authorization: Bearer {jwt_token}
METHOD	POST
Request JSON	{ "amount": 100000, "remarks": "Pulsa Telkomsel 100k", }
Response JSON SUCCESS	{ "status": "SUCCESS" "result": { "payment_id": "13bcb11c-111e-4a65-9afd-90a86a01cd21", "amount": 100000, "remarks": "Pulsa Telkomsel 100k", "balance_before": 500000, "balance_after": 400000, "created_date": "2021-04-01 22:22:00" } }
Response JSON FAILED	{ "message": "Balance is not enough" } { "message": "Unauthenticated" }

5. Transfer

Digunakan untuk mengirimkan balance pribadi ke user lain.

transfer_id menggunakan format **uuid**.

URL	/transfer
HEADERS	Authorization: Bearer {jwt_token}
METHOD	POST
Request JSON	{ "target_user": "b7342e8e-e8e7-4a5d-873e-b1b1bfcdeddb", "amount": 30000, "remarks": "Hadiah Ultah", }
Response JSON SUCCESS	{ "status": "SUCCESS" "result": { "transfer_id": "a7d39cf6-44b6-41fc-b3e9-7b16df5321c5", "amount": 30000, "remarks": "Hadiah Ultah", "balance_before": 400000, "balance_after": 370000, "created_date": "2021-04-01 22:23:20" } }
Response JSON FAILED	{ "message": "Balance is not enough" } { "message": "Unauthenticated" }

6. Report Transactions

Menampilkan seluruh transaksi yang pernah dilakukan oleh user tersebut.

URL	/transactions
HEADERS	Authorization: Bearer {jwt_token}
METHOD	GET
Response JSON SUCCESS	<pre>{ "status": "SUCCESS" "result": [{ "transfer_id": "a7d39cf6-44b6-41fc-b3e9-7b16df5321c5", "status": "SUCCESS", "user_id": "bc1c823e-b0fb-4b20-88c0-dff25e283252", "transaction_type": "DEBIT", "amount": 30000, "remarks": "Hadiah Ultah", "balance_before": 400000, "balance_after": 370000, "created_date": "2021-04-01 22:23:20" }, { "payment_id": "13bcb11c-111e-4a65-9afd-90a86a01cd21", "status": "SUCCESS", "user_id": "bc1c823e-b0fb-4b20-88c0-dff25e283252", "transaction_type": "DEBIT", "amount": 10000, "remarks": "Pulsa Telkomsel 100k", "balance_before": 500000, "balance_after": 400000, "created_date": "2021-04-01 22:22:00" }, { "top_up_id": "201ddde1-f797-484b-b1a0-07d1190e790a", "status": "SUCCESS", "user_id": "bc1c823e-b0fb-4b20-88c0-dff25e283252", "transaction_type": "CREDIT", "amount": 500000, "remarks": "", "balance_before": 0, "balance_after": 500000, "created_date": "2021-04-01 22:21:21" }] }</pre>

Response JSON FAILED	{ "message": "Unauthenticated" }
-------------------------	--

7. Update Profile

Digunakan untuk update profile.

Yang dapat diganti hanyalah selain unique key.

URL	/profile
HEADERS	Authorization: Bearer {jwt_token}
METHOD	PUT
Request JSON	{ "first_name": "Tom", "last_name": "Araya", "address": "Jl. Diponegoro No. 215" }
Response JSON SUCCESS	{ "status": "SUCCESS" "result": { "user_id": "bc1c823e-b0fb-4b20-88c0-dff25e283252", "first_name": "Tom", "last_name": "Araya", "address": "Jl. Diponegoro No. 215", "updated_date": "2021-04-01 23:00:20" } }