Atom-Pocket Manual

[**1. Setup**](#_l2v9xrr8f6l) **2**

[1.1. Requirement](#_mo2g5itgctff) 2

[1.2. Config](#_3b20w45putuu) 2

[1.3. Install NPM packages](#_hi9nd43m40f0) 2

[1.4. Create Database For Application](#_8j8ncqnna7dn) 2

[**2. Run**](#_dg9nt8xrje5t) **3**

[**3. Routing**](#_huws1z3zy86g) **3**

[3.1. Dompet](#_hcr5png55lbz) 3

[3.1.1. Create](#_nx0s0uo1op2y) 3

[3.1.2. Update](#_2w638cf4sibs) 4

[3.1.3. Update Status](#_ubxxaxxnaslh) 5

[3.1.4. Get](#_ugx5c4nt77jy) 6

[3.2. Kategori](#_q1zipn5vdx24) 7

[3.2.1. Create](#_9jfnzxj5rgk0) 7

[3.2.2. Update](#_33bspwu6jry0) 8

[3.2.3. Update Status](#_7hl5pji2o3) 8

[3.2.4. Get](#_9aq5clmiaewp) 9

[3.3. Transaksi](#_zc5ifmtusihy) 10

[3.3.1. Create](#_wp49cukwwumt) 10

[3.3.2. Get](#_p97j8kwu2ayw) 11

[3.4. Laporan](#_bjis7our4n6y) 13

[3.4.1. Transaksi](#_87j4swp7nmbe) 13

# 1. Setup

## 1.1. Requirement

Software :

* PostgreSQL v.10+
* NodeJS v12+

## 1.2. Config

Ubah konfigurasi aplikasi di “*./Atom-Pocket/config.json*”

|  |  |
| --- | --- |
| Nama | Default Value |
| HOST\_Port | 3000 |
| DB\_User | postgres |
| DB\_Host | localhost |
| DB\_Database\_Name | atom\_pocket |
| DB\_User\_Password | postgres |
| DB\_Port | 5432 |

## 1.3. Install NPM packages

Install Package yang dibutuhkan aplikasi dengan menjalankan shell command “npm install” di dalam directory “*./Atom-Pocker*”

Untuk melakukan langkah ini node harus sudah terinstal di mesin.

## 1.4. Create Database For Application

Jalankan file “*./Atom-Pocker/Atom\_Pocket\_PostgresDB\_init.sql*” di console postgresql untuk membuat database.

# 2. Run

Untuk menjalankan aplikasi jalankan shell command “npm start” di dalam directory “*./Atom-Pocket*”.

# 3. Routing

http://{*host}{port}*

http://*localhost:3000*

## 3.1. Dompet

/dompet

### 3.1.1. Create

HTTP Method : POST

/create

http://{*host}{port}/dompet/create*

http://*localhost:3000/dompet/create*

Membuat data *Dompet* baru

Query Parameter:

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Parameter | Mandatory | Validasi | Deskripsi |
| nama | Y | length>=5 | Nama dompet |
| referensi |  |  | Referensi dompet |
| deskripsi |  | length<=100 | deskripsi |

Expected respone :

{

"Message": "success"

}

### 3.1.2. Update

HTTP Method : POST

/update

http://{*host}{port}/dompet/update*

http://*localhost:3000/dompet/update*

Mengupdate data *Dompet* yang sudah ada

Query Parameter:

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Parameter | Mandatory | Validasi | Deskripsi |
| id | Y | Number | Id dari data dompet |
| nama | Y | length>=5 | Nama dompet |
| referensi |  |  | Referensi dompet |
| deskripsi |  | length<=100 | deskripsi |

Expected respone :

{

"Message": "success"

}

### 

### 3.1.3. Update Status

HTTP Method : POST

/updatestatus

http://{*host}{port}/dompet/updatestatus*

http://*localhost:3000/dompet/updatestatus*

Mengupdate data *Dompet* yang sudah ada

Query Parameter:

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Parameter | Mandatory | Validasi | Deskripsi |
| id | Y | Number | Id dari data dompet |
| isactive | Y | “true” or “false” | Status baru dompet |

Expected respone :

{

"Message": "success"

}

### 

### 3.1.4. Get

HTTP Method : POST

/update

http://{*host}{port}/dompet/get*

http://*localhost:3000/dompet/get*

Membaca data *Dompet* yang sudah ada

Query Parameter:

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Parameter | Mandatory | Validasi | Deskripsi |
| id |  | Number | Id dari data dompet yang dibaca |
| isactive |  | “true” or “false” | Status dompet yang dibaca, default ‘true’ |
| keyword |  |  | Mencari data dengan kata tertentu |
| orderbycolumn |  | columnname | Urutkan data yang dibaca menggunakan column |

Expected respone :

{

"message": "success",

"data": [

{

"id": 1,

"nama": "Dompet Utama",

"referensi": "52700072502",

"deskripsi": "Bank BCA",

"isactive": true

},

{

"id": 4,

"nama": "Dompet Utama",

"referensi": "52700072502",

"deskripsi": "Bank BCA",

"isactive": true

}

]

}

## 3.2. Kategori

/dompet

### 3.2.1. Create

HTTP Method : POST

/create

http://{*host}{port}/kategori/create*

http://*localhost:3000/kategori/create*

Membuat data *Dompet* baru

Query Parameter:

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Parameter | Mandatory | Validasi | Deskripsi |
| nama | Y | length>=5 | Nama kategori |
| deskripsi |  | length<=100 | deskripsi |

Expected respone :

{

"Message": "success"

}

### 

### 3.2.2. Update

HTTP Method : POST

/update

http://{*host}{port}/kategori/update*

http://*localhost:3000/kategori/update*

Mengupdate data *Dompet* yang sudah ada

Query Parameter:

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Parameter | Mandatory | Validasi | Deskripsi |
| id | Y | Number | Id dari data kategori |
| nama | Y | length>=5 | Nama kategori |
| deskripsi |  | length<=100 | deskripsi |

Expected respone :

{

"Message": "success"

}

### 3.2.3. Update Status

HTTP Method : POST

/updatestatus

http://{*host}{port}/domkategoripet/updatestatus*

http://*localhost:3000/kategori/updatestatus*

Mengupdate data *Dompet* yang sudah ada

Query Parameter:

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Parameter | Mandatory | Validasi | Deskripsi |
| id | Y | Number | Id dari data kategori |
| isactive | Y | “true” or “false” | Status baru kategori |

Expected respone :

{

"Message": "success"

}

### 3.2.4. Get

HTTP Method : POST

/update

http://{*host}{port}/kategori/get*

http://*localhost:3000/kategori/get*

Membaca data *Kategori* yang sudah ada

Query Parameter:

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Parameter | Mandatory | Validasi | Deskripsi |
| id |  | Number | Id dari data kategori yang dibaca |
| isactive |  | “true” or “false” | Status kategori yang dibaca, default ‘true’ |
| keyword |  |  | Mencari data dengan kata tertentu |
| orderbycolumn |  | columnname | Urutkan data yang dibaca menggunakan column |

Expected respone :

{

"message": "success",

"data": [

{

"id": 2,

"nama": "Masuk",

"deskripsi": "",

"isactive": true

},

{

"id": 5,

"nama": "Masuk",

"deskripsi": "kategori untuk uang masuk",

"isactive": true

}

]

}

## 

## 3.3. Transaksi

/transaksi

### 3.3.1. Create

HTTP Method : POST

/create

http://{*host}{port}/*transaksi*/create*

http://*localhost:3000/*transaksi*/create*

Membuat data Transaksi baru

Query Parameter:

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Parameter | Mandatory | Validasi | Deskripsi |
| tanggal | Y | format (DD-MM-YYYY)  ex. 31-12-2020 | Tanggal transaksi |
| deskripsi |  | length<=100 | deskripsi |
| nilai | Y | nilai > 0 | Nilai transaksi |
| dompet\_id | Y |  | Id *dompet* |
| kategori\_id | Y |  | Id *kategori* |
| istransaksimasuk | Y | “true” or “false” | true : transaksi uang masuk  false : transaksi uang keluar |

Expected respone :

{

"Message": "success"

}

### 

### 3.3.2. Get

HTTP Method : POST

/update

http://{*host}{port}/*transaksi*/get*

http://*localhost:3000/*transaksi*/get*

Membaca dataTransaksi yang sudah ada

Query Parameter:

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Parameter | Mandatory | Validasi | Deskripsi |
| id |  | Number | Id dari data transaksi yang dibaca |
| datefrom |  | format (DD-MM-YYYY)  ex. 31-12-2020 | Baca data transaksi setelah tanggal parameter |
| dateto |  | format (DD-MM-YYYY)  ex. 31-12-2020 | Baca data transaksi sebelum tanggal parameter |
| keyword |  |  | Mencari data dengan kata tertentu |
| orderbycolumn |  | columnname | Urutkan data yang dibaca menggunakan column |

Expected respone :

{

"message": "success",

"data": [

{

"id": 1,

"kode": "WIN100000",

"deskripsi": "transaksi uang masuk",

"nilai": "100000",

"dompet\_id": 1,

"kategori\_id": 2,

"istransaksimasuk": true,

"tanggal": "01-10-2020",

"dompet\_nama": "Dompet Utama",

"kategori\_nama": "Masuk"

},

{

"id": 5,

"kode": "WOUT100000",

"deskripsi": "transaksi uang keluar",

"nilai": "-30000",

"dompet\_id": 1,

"kategori\_id": 1,

"istransaksimasuk": false,

"tanggal": "01-10-2020",

"dompet\_nama": "Dompet Utama",

"kategori\_nama": "Keluar"

}

]

}

## 

## 3.4. Laporan

/laporan

### 3.4.1. Transaksi

HTTP Method : POST

/update

http://{*host}{port}/laporan/transaksi*

http://*localhost:3000/laporan/transaksi*

Membaca datakompilasi transaksi yang sudah ada

Query Parameter:

|  |  |  |  |
| --- | --- | --- | --- |
| Nama Parameter | Mandatory | Validasi | Deskripsi |
| datefrom |  | format (DD-MM-YYYY)  ex. 31-12-2020 | Baca data transaksi setelah tanggal parameter |
| dateto |  | format (DD-MM-YYYY)  ex. 31-12-2020 | Baca data transaksi sebelum tanggal parameter |
| dompet\_id |  |  | Baca data transaksi dengan dompet\_id parameter |
| kategori\_id |  |  | Baca data transaksi dengan kategori\_id parameter |

Expected respone :

{

"message": "success",

"data": {

"transaksi": [

{

"id": 1,

"kode": "WIN100000",

"deskripsi": "transaksi uang masuk",

"nilai": "100000",

"dompet\_id": 1,

"kategori\_id": 2,

"istransaksimasuk": true,

"tanggal": "01-10-2020",

"dompet\_nama": "Dompet Utama",

"kategori\_nama": "Masuk"

},

{

"id": 5,

"kode": "WOUT100000",

"deskripsi": "transaksi uang keluar",

"nilai": "-30000",

"dompet\_id": 1,

"kategori\_id": 1,

"istransaksimasuk": false,

"tanggal": "01-10-2020",

"dompet\_nama": "Dompet Utama",

"kategori\_nama": "Keluar"

},

{

"id": 4,

"kode": "WIN100001",

"deskripsi": "transaksi uang masuk",

"nilai": "200000",

"dompet\_id": 1,

"kategori\_id": 2,

"istransaksimasuk": true,

"tanggal": "02-10-2020",

"dompet\_nama": "Dompet Utama",

"kategori\_nama": "Masuk"

}

],

"sum": [

{

"istransaksimasuk": false,

"sum": "-30000"

},

{

"istransaksimasuk": true,

"sum": "300000"

}

]

}

}