

Microsite Stores API

Untuk menjaga backward compatibility dengan chat-commerce-backend API, maka path, request method, request format, response format, dll pada microsite stores API ini masih mempertahankan eksisting, kecuali pada bagian yang dipastikan tidak dikonsumsi oleh FE.

Select Store

User story

Sebagai user, saya ingin mengetahui state store yang sedang berjalan pada store tersebut dan memastikan bahwa store tersebut bukan merupakan store sendiri.

Path

/m/select-store

Method

POST

Request

Request Mengandung Token pada header. Contoh:

```
curl --location 'localhost:8686/select-store' \  
--header 'X-Token: 93_547ff3bc-592d-464c-9cef-88fcb68b9d52' \  
--header 'Content-Type: application/json' \  
--data '{  
  "id":5  
'
```

Request tersebut adalah penyederhanaan dari request eksisting pada chat-commerce-backend dengan pertimbangan bahwa data yang diproses memang hanya field “id” saja. Hal ini juga sudah dikoordinasikan dengan tim FE untuk disesuaikan, sehingga dapat menghemat resource. Sebagai referensi, request eksisting adalah sebagai berikut

Request lama (field yang ditandai warna kuning adalah field yang tidak diproses oleh BE)

```
curl --location 'localhost:8686/select-store' \  
--header 'X-Token: 93_547ff3bc-592d-464c-9cef-88fcb68b9d52' \  
--header 'Content-Type: application/json' \  
--data '{
```

```
"id": 2,  
  "name": "dummy-store-name",  
  "address": "dummy-store-address",  
  "city": "dummy-city",  
  "country": "dummy-country",  
  "max_radius": 2,  
  "admin": "dummy-admin",  
  "is_active": true,  
  "location_lat": 0.0,  
  "location_long": 0.0,  
  "picture": "dummy-picture",  
  "zipcode": "15414",  
  "category_id": 12  
'
```

Response

Sukses dengan header status code 200

Berikut response dalam bentuk json ketika request berhasil.

```
{  
  "is_expired": true,  
  "state": "initial",  
  "token": "93_2a1b194d-3020-4830-a3f7-20587b8e8c04"  
}
```

Gagal dengan header status code 400

Response ini terjadi ketika payload invalid (tidak dapat dibaca oleh server)

```
{  
  "error": "Bad Request",  
  "message": "can't decode request"  
}
```

Gagal dengan header status code 401

Response ini terjadi ketika request tidak menyertakan token atau token invalid.

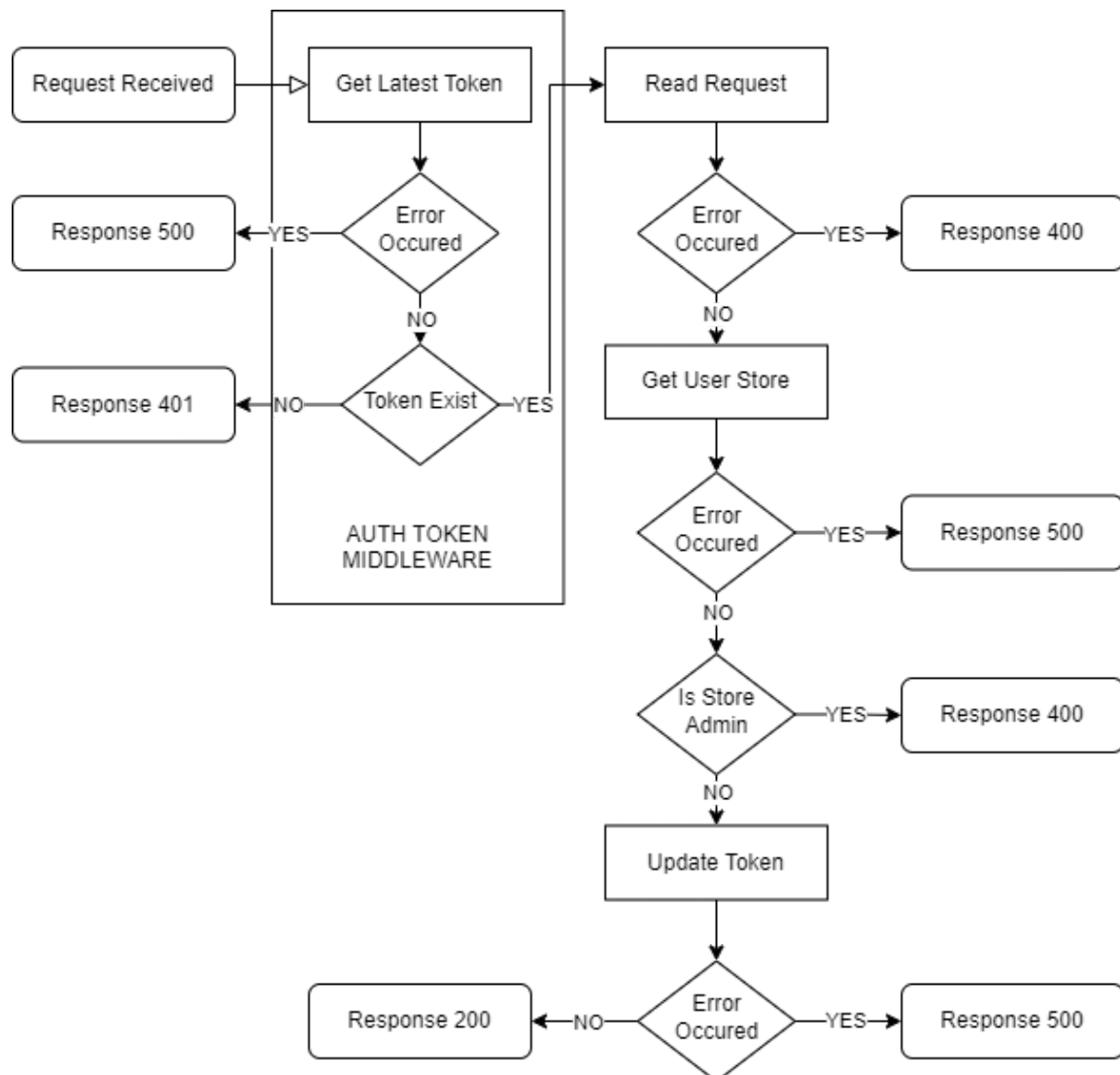
```
{  
  "error": "invalid_token",  
  "message": "Invalid token, please use the correct url from whatsapp  
message"  
}
```

Gagal dengan header status code 500

Response ini terjadi ketika terdapat kesalahan dalam server

```
{
  "error": "Internal Server Error",
  "message": "Something went wrong or server is under maintenance. Please contact
app support"
}
```

Application flow



Database related

1. Proses getUserStore eksisting pada chat-commerce backend mengambil seluruh field pada tabel user lalu dilakukan pengecekan isAdmin menggunakan for loops. Proses tersebut dapat diefisienkan dengan mengubah query untuk langsung mengembalikan boolean apakah user adalah admin store atau bukan

Query baru
<pre>SELECT true AS is_admin FROM users WHERE id = \$1 AND store_admin = \$2 AND deleted_at IS NULL</pre>
Query lama
<pre>SELECT id, name, phone_number, is_loyal_user, store_admin, created_at, updated_at, deleted_at FROM users WHERE store_admin = \$1 AND deleted_at IS NULL</pre>
Query updateUserTokens
<pre>UPDATE user_tokens SET address_id = \$1 WHERE id = \$2</pre>

user_tokens	
123	id
ABC	token
123	user_id
🕒	expire_at
🕒	created_at
🕒	updated_at
🕒	deleted_at
ABC	state
123	store_id
123	merchant_id
123	address_id

users	
123	id
ABC	name
ABC	phone_number
🕒	created_at
🕒	updated_at
🕒	deleted_at
<input checked="" type="checkbox"/>	is_loyal_user
123	store_admin
123	preferred_address_id

Test scenario

1. Saat request ke endpoint select-store dengan mengirimkan token dan request body yang benar, maka service akan mengembalikan response code 200
2. Saat request ke endpoint select-store dan request tidak mengirimkan token, maka service akan mengembalikan response code 401
3. Saat request ke endpoint select-store dan terjadi kesalahan pada server, maka service akan mengembalikan response code 500

Store State

User story

Sebagai user, saya ingin mengetahui state store yang sedang berjalan pada store tersebut dan memastikan bahwa store tersebut bukan merupakan store sendiri.

Path

/m/state-store

Method

POST

Request

Request mengandung token pada header. Contoh,

```
curl --location 'localhost:8686/store-state' \  
--header 'X-Token: 93_547ff3bc-592d-464c-9cef-88fcb68b9d52' \  
--header 'Content-Type: application/json' \  
--data '{  
    "reference_id": "9d292699d",  
    "type": "items"  
}'
```

Response

Sukses dengan header status code 200

Berikut response dalam bentuk json ketika request berhasil.

```
{  
  "data": {  
    "store": {  
      "id": 1,  
      "name": "Jessie Grosir Fatmawati",  
      "address": "Jl. RS. Fatmawati Komp. Golden Fatmawati No. 15, Gandaria Selatan, Cilandak, RT.8/RW.6",  
      "city": "Jakarta Selatan",  
      "country": "",  
    },  
  },  
}
```

```

        "max_radius": 500,
        "is_active": true,
        "location_lat": -6.261014610959956,
        "location_long": 106.52486774404345,
        "merchant_id": 1,
        "is_open": false,
        "picture":
"https://jcc-file.jatismobile.com/api-prod/assets/1683860684824423921Jessie
Logo_Orange Icon (1).png",
        "zip_code": "15710",
        "category_id": 12,
        "min_stock": null,
        "max_sales_days": null
    },
    "message": "Store Updated",
    "status": "success"
}

```

Gagal dengan header status code 400

Response ini terjadi ketika terdapat kesalahan dalam server

```

{
  "error": "Bad Request",
  "message": "can't decode request"
}

```

Gagal dengan header status code 401

Response ini terjadi ketika request tidak menyertakan token atau token invalid.

```

{
  "error": "invalid_token",
  "message": "Invalid token, please use the correct url from whatsapp
message"
}

```

Gagal dengan header status code 404

Response ini terjadi ketika store tidak ditemukan

```

{
  "error": "Not Found",
  "message": "Store not found"
}

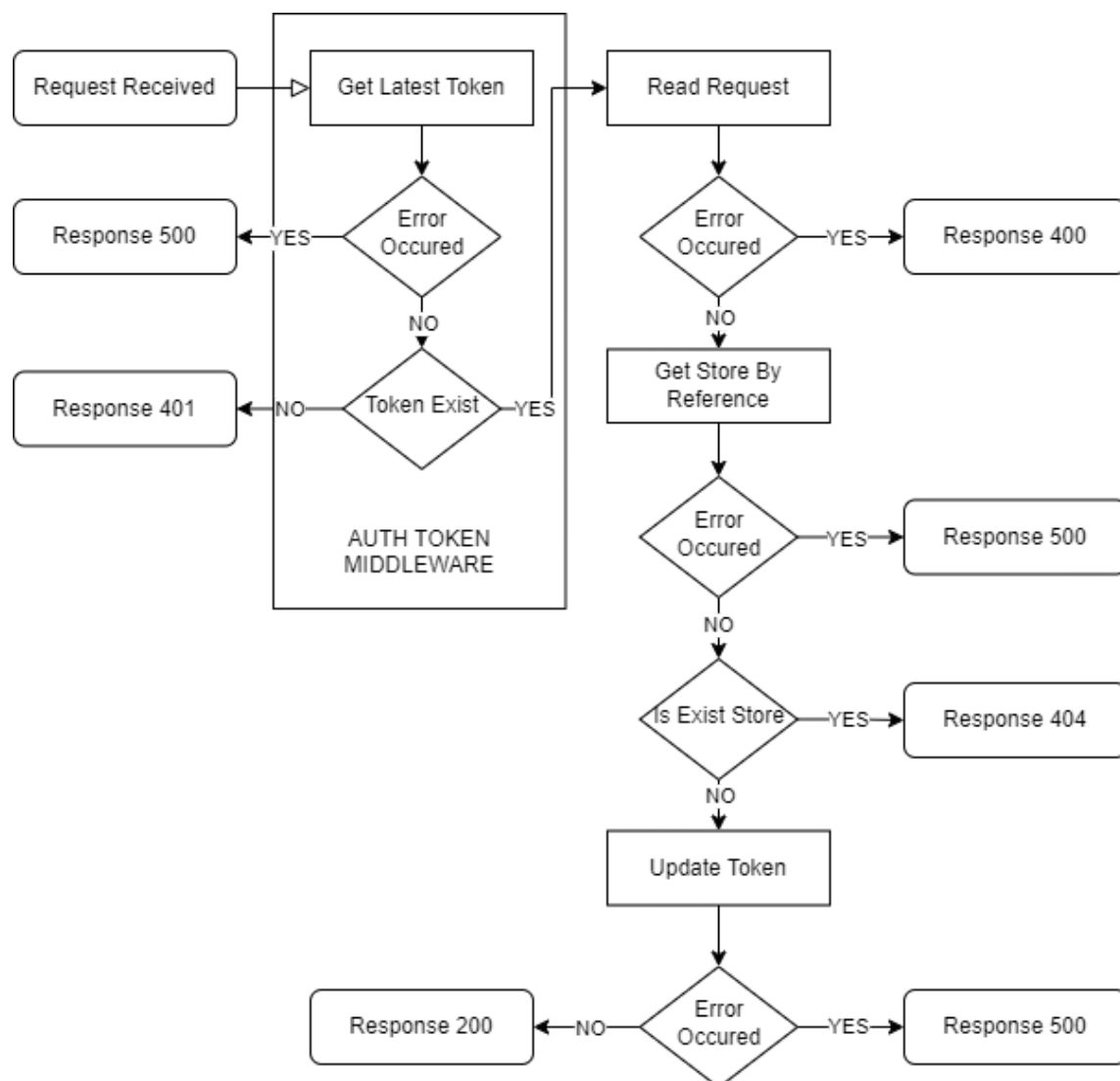
```

Gagal dengan header status code 500

Response ini terjadi ketika terdapat kesalahan dalam server

```
{
  "error": "Internal Server Error",
  "message": "Something went wrong or server is under maintenance. Please contact app support"
}
```

Application flow



Database related

query

```
SELECT
    s.id,
    s.merchant_id,
    s.name,
    s.address,
    s.city,
    s.country,
    s.max_radius,
    s.location_lat,
    s.location_long,
    s.pic,
    s.is_active,
    s.zip_code,
    s.category_id,
    s.created_at,
    s.updated_at,
    s.deleted_at
FROM
    stores s
    LEFT JOIN items i on i.store_id = s.id
    LEFT JOIN deep_links d on d.table_id = i.id
WHERE
    d.reference_id = $1
    AND d.table_name = 'items'
    AND s.is_active = true
    AND s.deleted_at IS NULL
```

```
SELECT
    s.id,
    s.merchant_id,
    s.name,
    s.address,
    s.city,
    s.country,
    s.max_radius,
    s.location_lat,
    s.location_long,
    s.pic,
    s.is_active,
    s.zip_code,
    s.category_id,
    s.created_at,
    s.updated_at,
    s.deleted_at
```

```
FROM
    stores s
    LEFT JOIN categories c on c.store_id = s.id
    LEFT JOIN deep_links d on d.table_id = c.id
WHERE
    d.reference_id = $1
    AND d.table_name = 'categories'
    AND s.is_active = true
    AND s.deleted_at IS NULL
```

```
SELECT
    s.id,
    s.merchant_id,
    s.name,
    s.address,
    s.city,
    s.country,
    s.max_radius,
    s.location_lat,
    s.location_long,
    s.pic,
    s.is_active,
    s.zip_code,
    s.category_id,
    s.created_at,
    s.updated_at,
    s.deleted_at
FROM
    stores s
    LEFT JOIN deep_links d on d.table_id = s.id
WHERE
    d.reference_id = $1
    AND d.table_name = 'stores'
    AND s.is_active = true
    AND s.deleted_at IS NULL
```

stores	items	categories	deep_links
123 id	123 id	123 id	ABC table_name
ABC name	123 category_id	ABC name	123 table_id
ABC address	ABC name	ABC picture	123 id
ABC city	ABC short_desc	<input checked="" type="checkbox"/> is_active	ABC reference_id
ABC country	ABC description	⌚ created_at	
123 max_radius	ABC picture	⌚ updated_at	
ABC pic	<input checked="" type="checkbox"/> is_active	⌚ deleted_at	
<input checked="" type="checkbox"/> is_active	⌚ created_at	123 store_id	
⌚ created_at	⌚ updated_at	123 price	
⌚ updated_at	⌚ deleted_at	123 stock	
⌚ deleted_at	123 store_id	123 width	
123 location_lat	123 price	123 height	
123 location_long	123 stock	123 length	
123 merchant_id	123 width	123 weight	
ABC zip_code	123 height		
123 category_id	123 length		
123 min_stock	123 weight		
123 max_sales_days			

Test scenario

1. Saat request ke endpoint store-state dengan mengirimkan token dan request body yang benar, maka service akan mengembalikan response code 200
2. Saat request ke endpoint store-state dan request tidak mengirimkan token, maka service akan mengembalikan response code 401
3. Saat request ke endpoint store-state dan request reference_id yang tidak terdaftar di database, maka service akan mengembalikan response code 404
4. Saat request ke endpoint store-state dan terjadi kesalahan pada server, maka service akan mengembalikan response code 500

Nearest Store

User story

Sebagai user, saya ingin mengetahui informasi store aktif terdekat, berdasarkan kriteria filter tertentu, yaitu name, category_type, category_id, dan limit.

Path

/m/v4/nearest-store

Method

GET

Request

Request mengandung token pada header. Request menyertakan query param yang bersifat opsional, yaitu

1. name
Dapat merupakan nama produk, nama kategori produk, nama store, atau nama kategori store.
2. category_type
Merupakan nama kategori store. Default value adalah "ecommerce". Jika user mengisi kriteria category_type, value yang diizinkan hanyalah "ecommerce" (selain string kosong).
Catatan: kriteria ini menjadi tidak begitu berguna karena hanya mengizinkan satu value valid. Namun untuk menjaga backwad compatibility dan dengan asumsi penggunaan yang lebih luas di masa depan, maka kriteria ini tetap akan dipertahankan.
3. category_id
Merupakan id kategori store.
Catatan: kriteria ini tumpang tindih dengan category_type karena memiliki maksud yang sama, yaitu untuk memilih kategori store, di mana pada saat dokumen ini ditulis terdapat 11 kategori yang dapat dipilih yaitu "2" untuk "Toko Kelontong", "3" untuk "Bahan Makanan Segar", "4" untuk "Makanan Olahan", "5" untuk "Pakaian dan Busana", "6" untuk "Peralatan Rumah Tangga", "7" untuk "Kerajinan Tangan", "8" untuk "Elektronik", "9" untuk "Toko Kesehatan", "10" untuk "Perlengkapan Bayi", 11 untuk "Perlengkapan Sekolah", dan 12 "Lain-lain". Terlihat bahwa parameter category_id ini lebih berguna daripada parameter category_type karena user dapat memilih lebih dari satu pilihan valid.
4. limit
Merupakan batas banyaknya store terdekat yang perlu dikembalikan.

Contoh request

```
curl --location  
'localhost:8181/m/v4/nearest-store?name=lego&limit=10&category_type=e-commerce&category_id=1' \  
--header 'x-token: 7_ef34d500-b1ef-4fb3-aa42-3f72979b2d1f'
```

Response

Sukses dengan header status code 200

Berikut response dalam bentuk json ketika request berhasil.

```
{  
  "data": [  
    {  
      "id": 7,  
      "name": "Jessie Grosir Pasar Rebo",  
      "picture":  
"https://jcc-file.jatismobile.com/api-prod/assets/1683860654828092732Jessie  
Logo_Orange Icon (1).png",  
      "address": "Jl. TB Simatupang No.6, RT.7/RW.3, Susukan, Kec.  
Ciracas, Kota Jakarta Timur, Daerah Khusus Ibukota Jakarta 13750",  
      "distance": "11.73 km",  
      "location_lat": -6.3079969,  
      "location_long": 106.8697533,  
      "is_open": true  
    }  
  ],  
  "is_expired": true,  
  "state": "completed",  
  "token": "7_ef34d500-b1ef-4fb3-aa42-3f72979b2d1f"  
}
```

Response di atas adalah penyederhanaan dari response eksisting (di chat-commerce-backend) yang banyak memiliki field tak terpakai (tidak dikonsumsi oleh FE). Sebagai perbandingan, response eksisting adalah sebagai berikut

Response eksisting (chat-commerce-backend). Field yang diberi tanda kuning adalah field yang tidak dikonsumsi oleh FE.

```
{  
  "data": [  
    {  
      "id": 7,
```

```

        "name": "Jessie Grosir Pasar Rebo",
        "address": "Jl. TB Simatupang No.6, RT.7/RW.3, Susukan, Kec.
        Ciracas, Kota Jakarta Timur, Daerah Khusus Ibukota Jakarta 13750",
        "city": "Jakarta Timur",
        "country": "",
        "max_radius": 30,
        "is_active": false,
        "location_lat": -6.3079969,
        "location_long": 106.8697533,
        "merchant_id": 1,
        "distance": "11.73 km",
        "is_open": true,
        "picture":
        "https://jcc-file.jatismobile.com/api-prod/assets/1683860654828092732Jessie
        Logo_Orange Icon (1).png",
        "zip_code": "13750",
        "category_id": 12,
        "category_name": "Lain-lain",
        "min_stock": null,
        "max_sales_days": null,
        "dist": 11.734386316864878
    }
},
    "is_expired": true,
    "state": "completed",
    "token": "7_ef34d500-b1ef-4fb3-aa42-3f72979b2d1f"
}

```

Gagal dengan header status code 400

Response ini terjadi ketika alamat tidak ditemukan

```

{
  "error": "Bad Request",
  "message": "user not choose address yet"
}

```

Response ini terjadi ketika merchant ID tidak valid

```

{
  "error": "Bad Request",
  "message": "invalid merchantID from user token"
}

```

Gagal dengan header status code 401

Response ini terjadi ketika request tidak menyertakan token atau token invalid.

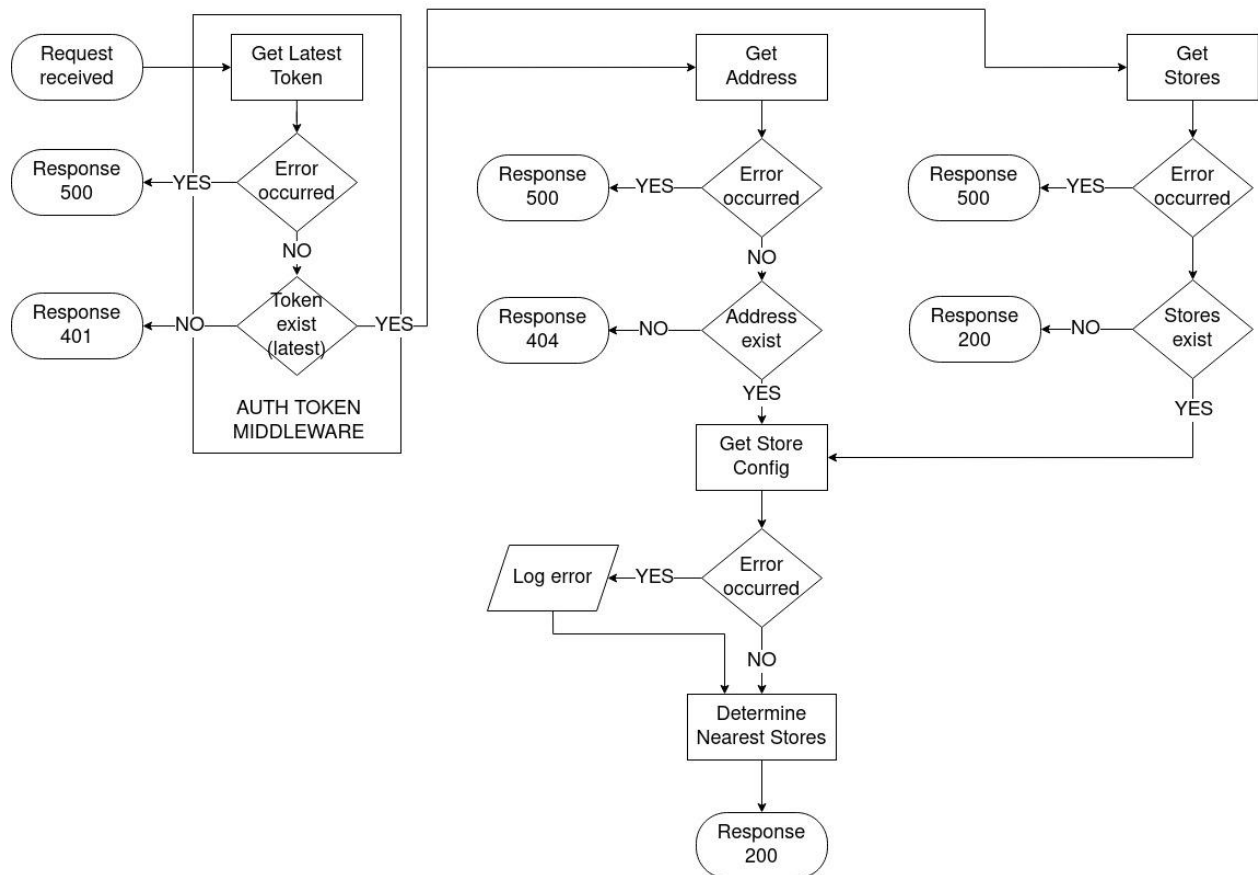
```
{
  "error": "invalid_token",
  "message": "Invalid token, please use the correct url from whatsapp
message"
}
```

Gagal dengan header status code 500

Response ini terjadi ketika terdapat kesalahan dalam server

```
{
  "error": "Internal Server Error",
  "message": "Something went wrong or server is under maintenance. Please contact
app support"
}
```

Application flow



Database related

1. Pada proses getAddress, query eksisting pada chat-commerce-backend mengembalikan semua field dari tabel address, sedangkan kebutuhan untuk nearest store hanyalah field location_lat dan location_long saja. Untuk itu, query yang akan diterapkan akan disederhanakan untuk mengembalikan kedua field tersebut saja. Order by clause pada query lama juga tidak diperlukan lagi karena sudah difilter oleh where clause yaitu id, yang merupakan primary key (unique)

Query baru:

```
SELECT
    location_lat,
    location_long
FROM
    address
WHERE
    id = $1
    AND deleted_at IS NULL
```

Query lama:

```
SELECT
    id,
    user_id,
    name,
    billing_name,
    address,
    notes,
    location_lat,
    location_long,
    is_default,
    created_at,
    updated_at,
    deleted_at
FROM
    address
WHERE
    id = $1
    AND deleted_at IS NULL
ORDER BY
    id
```

2. Pada proses getStores, query eksisting pada chat-commerce-backend terdiri dari 2 subproses, yaitu getStores itu sendiri dan dropStoreWhereUserIsAdmin. Kedua subproses tersebut dapat digabung dalam satu transaksi, supaya menghemat resource dan memperbaiki execution time. Di sisi lain, terdapat perubahan proses query building yang tadinya menggunakan penggabungan (concatenation) string, menjadi

menggunakan parameter placeholder untuk mencegah sql injection. Contoh sql injection adalah dengan menyematkan query pada parameter name sebagai berikut
name=grosir%5C%27%29%3Bdelete%20from%20store_categories%3B%2D%2D

Query baru

```
WITH sa_cte AS (
    SELECT
        COALESCE(store_admin, 0) AS store_admin
    FROM
        users
    WHERE
        id = $1
)
SELECT
    DISTINCT s.id,
    s.name,
    s.address,
    s.city,
    s.country,
    s.max_radius,
    s.pic,
    s.location_lat,
    s.location_long,
    s.merchant_id,
    s.zip_code,
    s.category_id,
    sc.name AS category_name
FROM
    stores s
    JOIN sa_cte ON sa_cte.store_admin != s.id
    JOIN items i ON i.store_id = s.id
    JOIN categories ic ON i.category_id = ic.id
    JOIN store_categories sc ON sc.id = s.category_id
WHERE
    s.merchant_id = $2
    AND i.is_active = true
    AND s.is_active = true
    AND ic.is_active = true
    AND i.deleted_at IS NULL
    AND s.deleted_at IS NULL
    AND (
        LOWER(i.name) ILIKE $3
        OR LOWER(ic.name) ILIKE $3
        OR LOWER(s.name) ILIKE $3
        OR LOWER(sc.name) ILIKE $3
    )
    AND sc.name != 'Makanan dan Minuman'
    AND s.category_id = $4
```

Query lama

- query get stores

```
SELECT
    DISTINCT s.id,
    s.name,
    s.address,
    s.city,
    s.country,
    s.max_radius,
    s.pic,
    s.location_lat,
    s.location_long,
    s.merchant_id,
    s.zip_code,
    s.category_id,
    sc.name AS category_name
FROM
    stores s
    JOIN sa_cte ON sa_cte.store_admin != s.id
    JOIN items i ON i.store_id = s.id
    JOIN categories ic ON i.category_id = ic.id
    JOIN store_categories sc ON sc.id = s.category_id
WHERE
    s.merchant_id = %d
    AND i.is_active = true
    AND s.is_active = true
    AND ic.is_active = true
    AND i.deleted_at IS NULL
    AND s.deleted_at IS NULL
    AND (
        LOWER(i.name) ILIKE '%%%s%%'
        OR LOWER(ic.name) ILIKE '%%%s%%'
        OR LOWER(s.name) ILIKE '%%%s%%'
        OR LOWER(sc.name) ILIKE '%%%s%%'
    )
    AND sc.name != 'Makanan dan Minuman'
    AND s.category_id = %d
ORDER BY
    s.id DESC
```

- query getStoreAdmin untuk proses dropStoreWhereUserIsAdmin

```
SELECT
    id,
    name,
    phone_number,
    is_loyal_user,
    store_admin,
    created_at,
```

```

        updated_at,
        deleted_at
FROM
    users
WHERE
    store_admin IN (?)
    AND deleted_at IS NULL

```

3. Pada proses `getStoreConfig`, query eksisting pada `chat-commerce-backend` dijalankan 2 kali untuk setiap store, masing-masing untuk key `closed` dan key `service`. Misalnya, ada 4 store dari proses `getStores` maka query `getStoreConfig` menjalankan 8 query. Untuk menghemat resource dan memperbaiki execution time, query yang dijalankan berulang-ulang akan diubah supaya dapat dijalankan dalam satu transaksi.

Catatan: query masih bisa dipercepat dengan menambahkan unique index (key, deleted_at), sesuai business logic bahwa tidak mungkin ada key yang duplikat. Namun, penerapan unique index ini akan mempengaruhi service lain pada `chat-commerce-backend` (misalnya saat insert new config), sehingga tidak dilanjutkan.

Query baru (dijalankan 1 kali untuk semua kembalian `getStores`)

```

WITH cte AS (
    SELECT
        SPLIT_PART(key, '_', 2) AS store_id,
        JSON_OBJECT_AGG(
            SPLIT_PART(key, '_', 1), value
        ) AS configs
    FROM
        config
    WHERE
        key IN ($1, $2, ...)
        AND deleted_at IS NULL
    GROUP BY
        store_id
)
SELECT
    JSON_OBJECT_AGG(
        store_id, configs
    ) AS result
FROM
    cte

```

Query lama (dijalankan 2 kali untuk setiap store)

```

SELECT
    id,
    key,
    value,

```

```
        created_at,  
        updated_at,  
        deleted_at  
FROM  
    config  
WHERE  
    key = $1  
    and deleted_at IS NULL  
LIMIT  
    1
```

Test scenario

1. Saat request ke endpoint nearest-store dengan mengirimkan token dan request body yang benar, maka service akan mengembalikan response code 200
2. Saat request ke endpoint nearest-store dan request tidak mengirimkan token, maka service akan mengembalikan response code 401
3. Saat request ke endpoint nearest-store dan address yang dicari yang tidak terdaftar di database, maka service akan mengembalikan response code 404
4. Saat request ke endpoint nearest-store dan terjadi kesalahan pada server, maka service akan mengembalikan response code 500