**BAB IV**

**HASIL DAN ANALISA**

* 1. Integrasi Aplikasi

API merupakan perantara yang menghubungkan satu aplikasi dengan aplikasi lainya untuk saling berkomunikasi/terintegrasi. API berisi kumpulan prosedur, fungsi atau cara komunikasi untuk interaksi antar aplikasi.

Dalam pengembangkan aplikasi perangkat lunak sangat sering dilakukan integrasi antar aplikasi baik buatan sendiri maupun produk dari orang/perusahaan lain. Contohnya pada toko Online yang harus terintegrasi antara produk milik sendiri, sistem pembayaran hingga sistem pengiriman yang di buat oleh perusahaan lain.

Secara garis besar terdapat 4 metode yang digunakan dalam integrasi antar aplikasi (Enterprise integration pattern)

1. File sharing

Metode ini merupakan integrasi yang sangat mudah untuk dilakukan dimana file yang telah di buat akan dibagikan ke aplikasi lain, file sharing akan bermanfaat apabila integrasi aplikasi tidak terhubung secara langsung. kelemahan dalam metode ini adalah file yang dibagikan tidak dapat dibagikan secara realtime.

1. Database sharing

Metode ini merupakan integrasi antar aplikasi yang memanfaatkan system basis data yang berada dalam 1 jaringan yang sama antara aplikasi dengan jarinag basis data, sehingga aplikasi hanya perlu melakukan CRUD (Create, Read, Update, Delete) pada database dan dapat langsung diakses oleh aplikasi lain. Kelemahan dalam metode ini adalah apabila 2 aplikasi melakukan perubahan data yang sama maka akan menyebabkan tabrakan data.

1. Remote Proccedur invocation

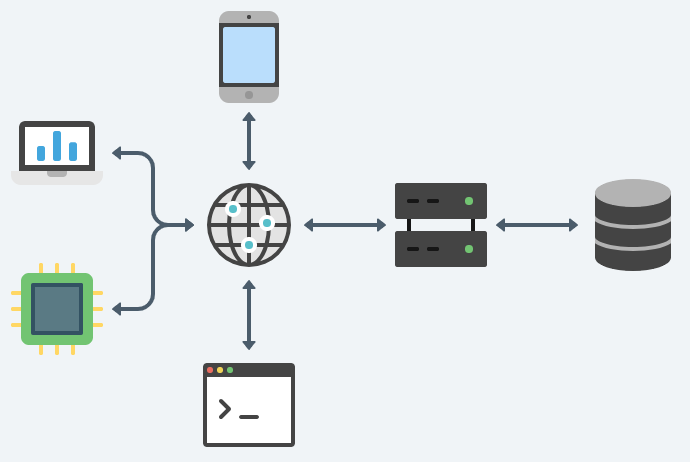
Metode ini juga dikenal dengan API (Aplication Programing Interface) dengan cara membuat yang dapat digunakan oleh aplikasi lain. Aplikasi utama yang memiliki data merpakan aplikasi utama yang akan menjadi API. Dan aplikasi yang membutuhkan data merupakan aplikasi yang akan menggunakan API. Kelebihan metode ini adalah integrasi dapat dilakukan secara realtime, dibandingkan dengan metode sharing basis data dimana jika terdapat perubahan pada data aplikasi perlu melakukan pengecekan secara manual pada database, sedangkan dengan metode ini API mengetahui semua perubahan data yang terjadi dalam system.

1. Messaging

Metode ini merupakan integrasi aplikasi yang memanfaatkan applikasi lain yang akan mengirimkan pesan yang disebut dengan message broker, metode ini cukup mirip dengan metode API dan Database sharing, namun metode ini tidak bersifat realtime dimana aplikasi menunggu pesan dari message broker untuk menerima update.

* 1. Hasil Perancangan API

RESTful API merupakan imlementasi API yang menggunakan HTTP sebagai protocol komunikasi. RESTful API dapat diadaptasi di semua Bahasa pemrograman yang dapat menggunakan HTTP seperti integrai aplikasi Web, aplikasi mobile dan aplikasi desktop. RESTful API juga dikenal sebagai Web Service dikarenakan metode ini menggunakan protocol HTTP yang biasa digunakan pada aplikasi web.



Gambar 4.1. Ilustrasi API

Pengembangan API setidaknya dilakukan dalam 4 tahapan yang akan dimulai dengan perancangan, pengembangan, pengujian dan implementasi. Dalam tahapan perancangan sangat di sarankan dengan menggunakan OpenAPI yang merupakan standarisasi untuk dokumentasi.

Dalam penelitian ini penulis akan melakukan pengujian koneksi antar teknologi dan device untuk berkomunikasi sehingga akan dibuat beberapa endpoint sesuai kebutuhan.

Kebutuhan dasar yang akan di buat

1. Mendaftarkan perangkat API
2. Menghubungkan akun WA
3. Memutuskan Akun WA
4. Cek Status API
5. Membaca pesan
6. Mengirim pesan
7. Cek status pesan

Berdasarkan spesifikasi di atas akan di buat endpoint untuk setiap kebutuhan seperti pada tabel 4.1. berikut.

Tabel 4.1. Daftar Endpoint API

|  |  |  |
| --- | --- | --- |
| METHOD | ENDPOINT | KETERANGAN |
| POST | /devices | Mendaftarkan akun ke API |
| GET | /qr | Connect akun WA |
| DELETE | /devices | Disconnect akun WA |
| GET | /devices | Cek status API |
| GET | /messages | Membaca pesan masuk |
| POST | /messages | Kirim pesan pribadi |
| GET | /messages/{MESSAGE\_ID } | Cek status pesan |

Setiap endpoint harus memiliki respons untuk agar dapat di proses oleh client yang melakukan request ke API, dalam hal ini penulis melakukan dokumentasi menggunakan standar openAPI.

* + 1. Endpoint devices (/devices)

Pada endpoint ini dapat digunakan oleh 3 http method yaitu GET, POST, DELETE yang mana akan berfungsi sesuai dengan standar yang telah di buat. Sebelum dapat mengirim pesan menggunakan API maka perlu menambahkan device terlebih dahulu. Anda dapat memiliki lebih dari satu device pada API. Sebuah device harusnya terasosiasi dengan sebuah nomor WhatsApp.

POST Endpoint ini untuk menambahkan device baru yang bisa digunakan untuk pairing ke API.

GET Endpoint ini untuk melihat semua devices yang sudah dibuat oleh akun anda. Maksimum data yang dikembalikan per page adalah 10 item. Item terkahir yang dikembalikan per halaman ada pada properti last\_key. Properti ini kemudian dapat digunakan untuk merequest halaman berikutnya dengan menggunakan query string parameter start\_key. Jika nilai dari last\_key adalah null maka data sudah habis dan ini adalah halaman terakhir.

DELETE Endpoint ini digunakan untuk menghapus sebuah device berdasarkan DEVICE\_ID. Setelah device dihapus maka request berikutnya seperti mengirim pesan yang menggunakan DEVICE\_ID akan gagal.

* + 1. Endpoint QR (/qr)

Endpoint ini digunakan untuk mendapatkan QR code yang digunakan untuk melakukan pairing device ke AP. Terdapat dua atribut yang dikembalikan yaitu qr\_code dan image\_url. Untuk menampilkan gambar QR Code kunjungi link URL yang ada terdapat pada image\_url kemudian scan QR code tersebut dengan WhatsApp.

Jika terdapar WhatsApp Web maka session WhatsApp Web anda akan terputus setelah WhatsApp dipairing ke API. Hal ini karena server WhatsApp hanya memperbolehkan satu session WhatsApp Web yang aktif.

* + 1. Endpoint messages (/messages)

Saat ini API WhatsApp mendukung pengiriman pesan teks dan gambar. Pengiriman pesan pada API bersifat asynchronous. Artinya pesan yang dikirim akan masuk ke queue (antrian) server API Proses ini sangat cepat biasanya hanya memerlukan waktu beberapa detik untuk queue diproses. API mendukung pengiriman pesan ke nomor WhatsApp individu atau sebuah grup WhatsApp. Selain itu API juga dapat mengirimkan pesan secara terjadwal. Dimana pesan akan dikirim sesuai dengan tanggal dan waktu yang ditentukan.

POST /messages Endpoint ini digunakan untuk mengirimkan pesan baik teks, gambar, list, buttons atau dokumen (segala jenis file). Beberapa batasan untuk pengiriman pesan diantaranya: Pesan teks tidak lebih dari 8000 bytes atau 8KB. Untuk pesan bertipe image harus berformat JPEG, PNG atau GIF. Ukuran file gambar atau dokumen tidak boleh lebih dari 5.000.000 bytes atau 5MB. Timeout adalah 10 detik. Jika lebih dari itu maka pengiriman dianggap gagal dan koneksi akan diputus.

GET Endpoint ini digunakan untuk mendapatkan semua pesan yang dikirim dengan menggunakan endpoint POST /messages. Maksimum data yang dikembalikan per request adalah 10 item. Untuk merequest item berikutnya gunakan nilai dari atribut last\_key dan gunakan sebagai input untuk parameter start\_key. Jika nilai dari last\_key adalah null maka data sudah habis dan ini adalah halaman terakhir.

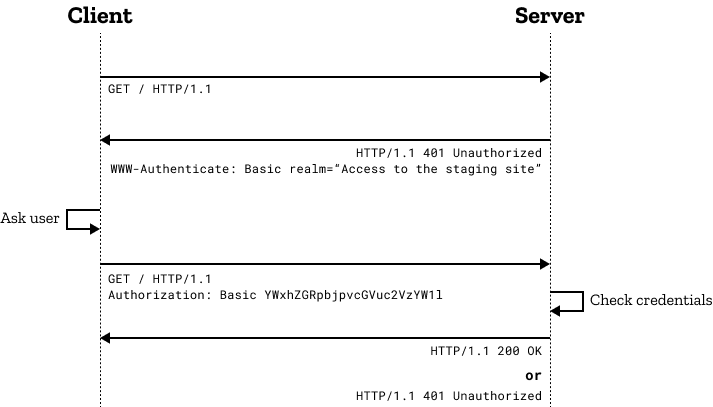
GET /messages/MESSAGE\_ID Endpoint ini digunakan untuk mendapatkan informasi sebuah message berdasarkan MESSAGE\_ID.

* + 1. Authorization

HTTP menyediakan kerangka kerja umum untuk kontrol akses dan autentikasi. Halaman ini adalah pengantar kerangka kerja HTTP untuk autentikasi, dan menunjukkan cara membatasi akses ke server Anda menggunakan skema "Basic" HTTP.

Alur tantangan dan respons bekerja seperti ini:

1. Server merespons klien dengan status respons 401 (Tidak Sah) dan memberikan informasi tentang cara mengotorisasi dengan header respons WWW-Authenticate yang berisi setidaknya satu tantangan.
2. Klien yang ingin mengautentikasi dirinya sendiri dengan server kemudian dapat melakukannya dengan menyertakan header permintaan Otorisasi dengan kredensial.
3. Biasanya klien akan memberikan permintaan kata sandi kepada pengguna dan kemudian akan mengeluarkan permintaan termasuk header Otorisasi yang benar.



Gambar 4.2. Flow HTTP Authorization

Header permintaan Otorisasi HTTP dapat digunakan untuk menyediakan kredensial yang mengautentikasi agen pengguna dengan server, memungkinkan akses ke sumber daya yang dilindungi.

Header Otorisasi biasanya, namun tidak selalu, dikirim setelah agen pengguna pertama kali mencoba meminta sumber daya yang dilindungi tanpa kredensial. Server merespons dengan pesan 401 Unauthorized yang menyertakan setidaknya satu header WWW-Authenticate. Header ini menunjukkan skema autentikasi apa yang dapat digunakan untuk mengakses sumber daya (dan informasi tambahan apa pun yang diperlukan oleh klien untuk menggunakannya). Agen pengguna harus memilih skema autentikasi paling aman yang didukungnya dari yang ditawarkan, meminta kredensial pengguna, lalu meminta ulang sumber daya (termasuk kredensial yang disandikan di header Otorisasi).

* 1. Hasil Peracangan Response API

Setiap request yang berhasil dikirimkan akan mengembalikan response berupa json object yang akan di gunakan oleh client

* + 1. Dokumentasi Spesifikasi API

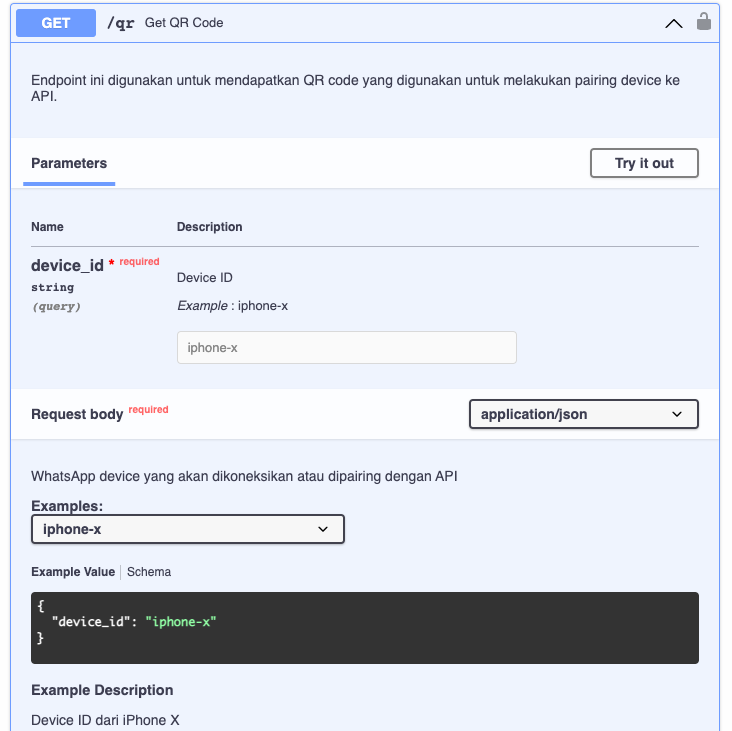
API yang akan di buat dokumentasi spesifikasi terlebih dahulu sehingga proses pengembangan akan lebih mudah dan dapat dikerjakan secara paralel.

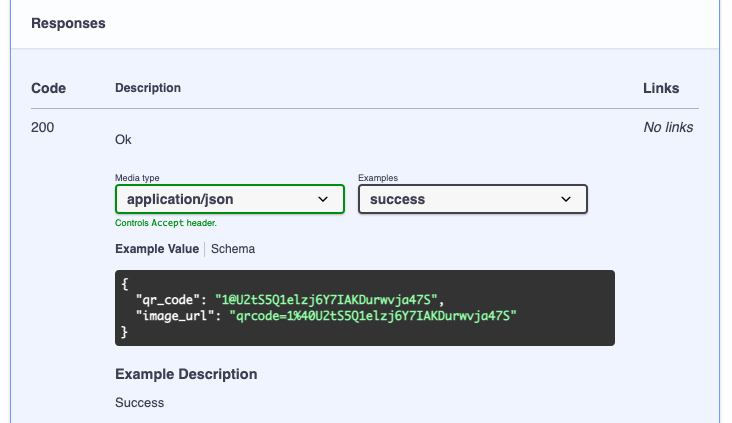
1. API SERVER

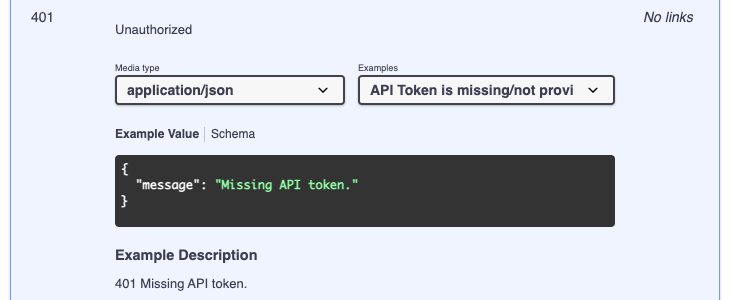
|  |
| --- |
| "servers": [  {  "description": "Unofficial WhatsApp API",  "url": "https://{environment}.restwa.my.id/v1",  "variables": {  "environment": {  "description": "Server Environment",  "default": "api",  "enum": ["dev", "test", "api"]  }  }  }  ], |

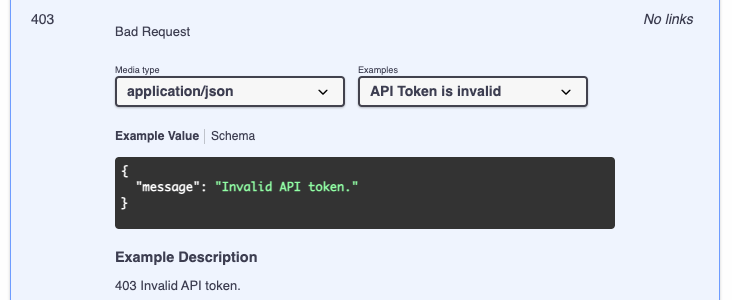


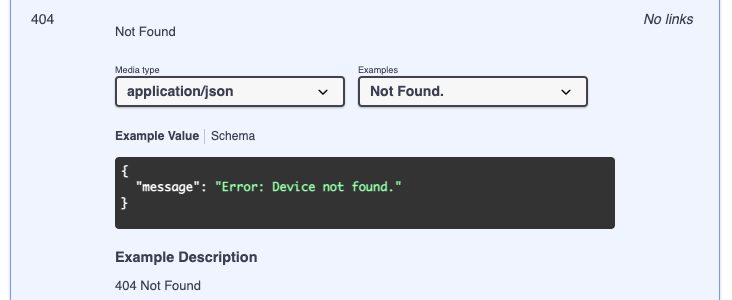
1. QR



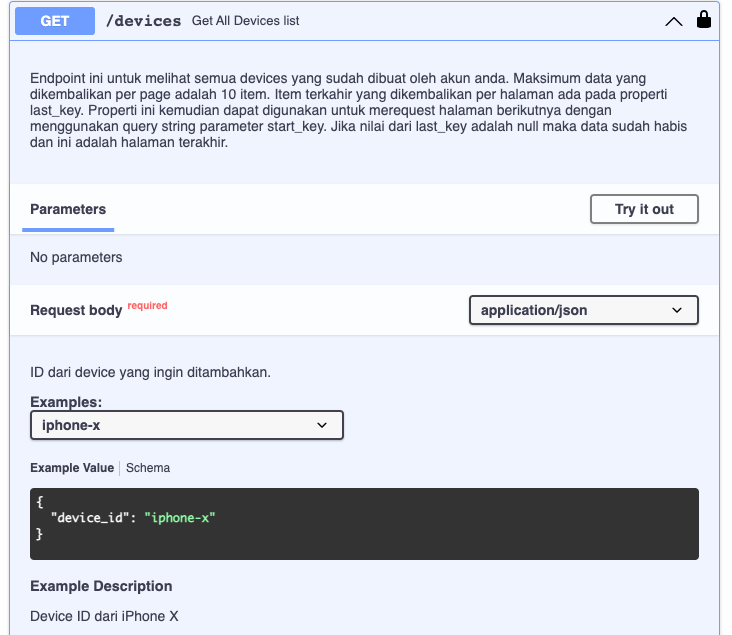


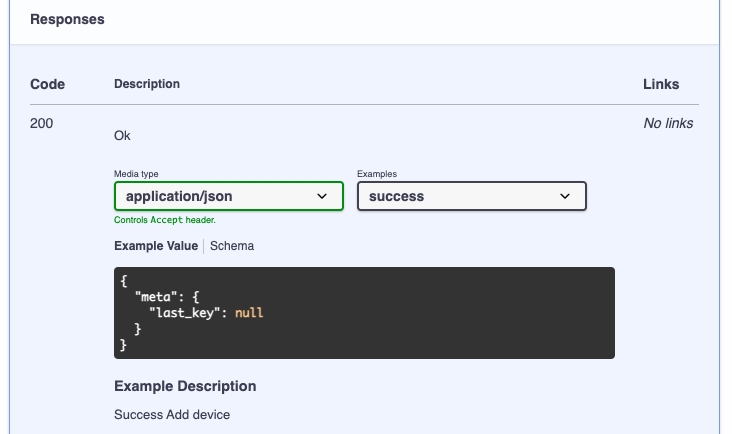




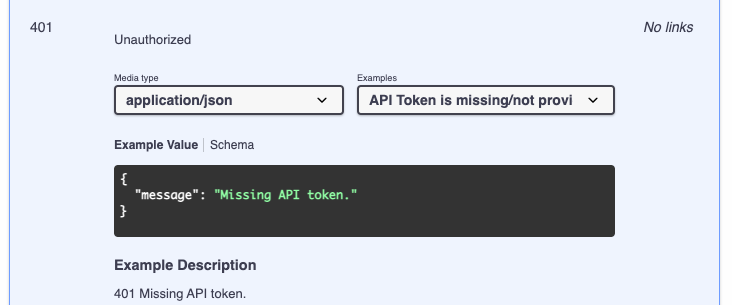


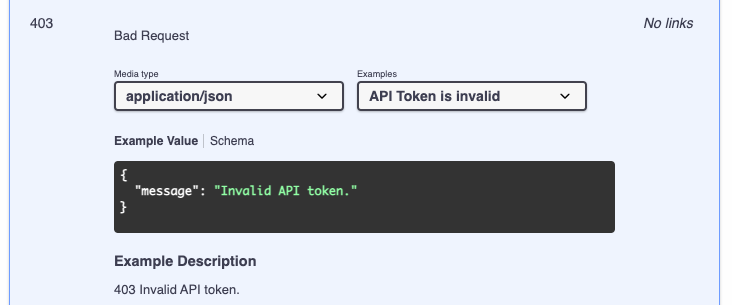
1. DEVICES –GET



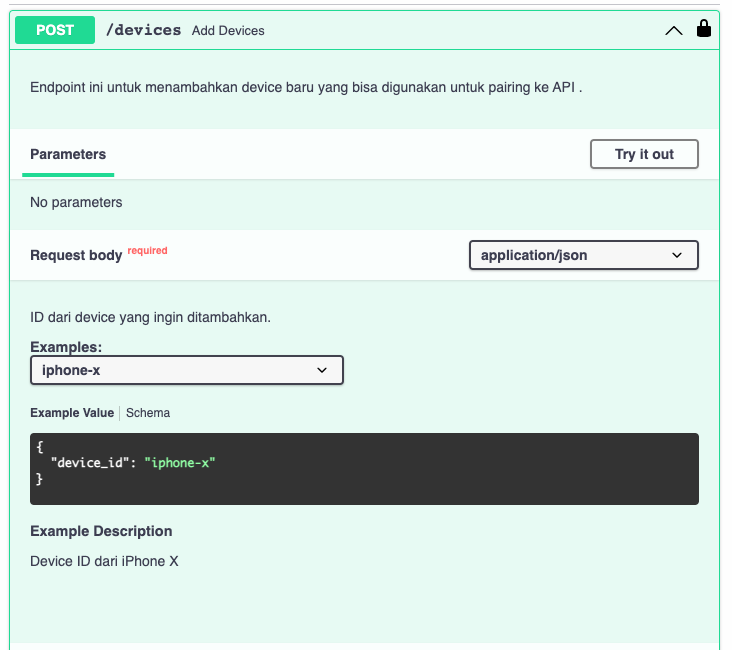


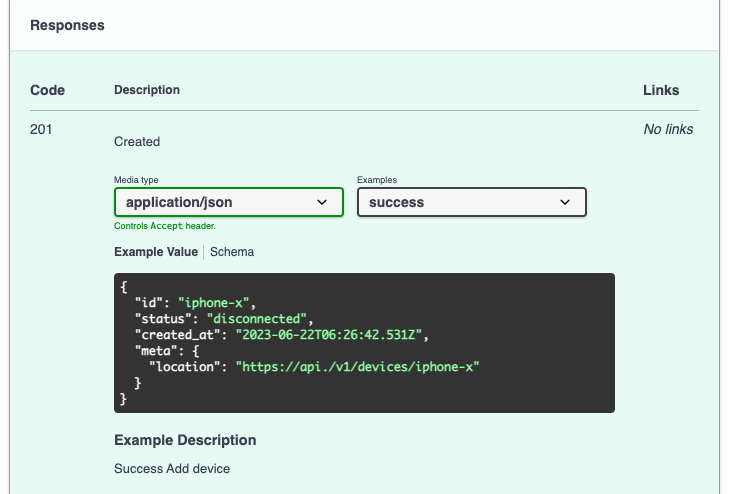


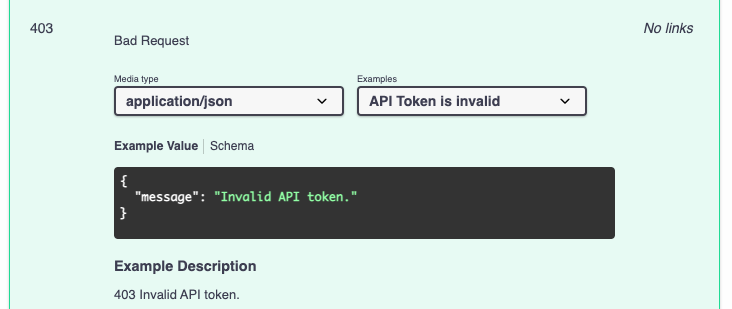


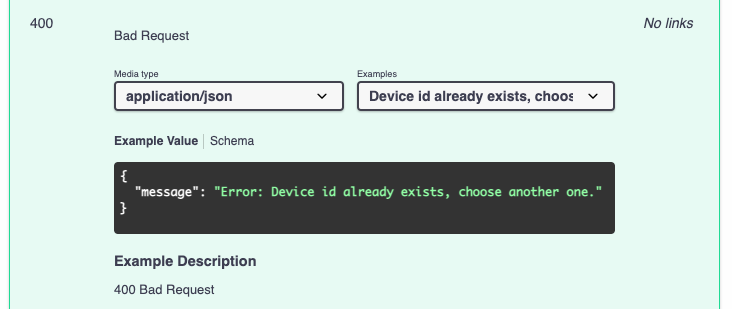


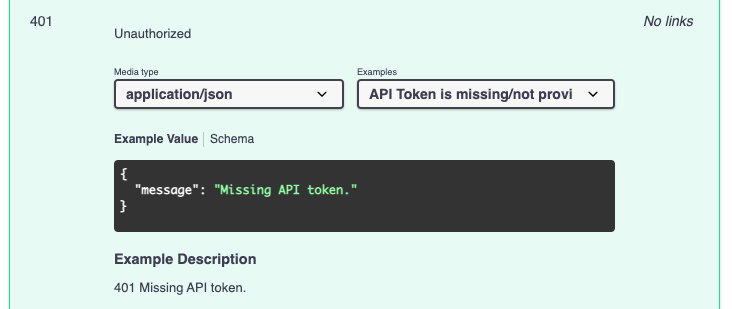
1. DEVICES – POST



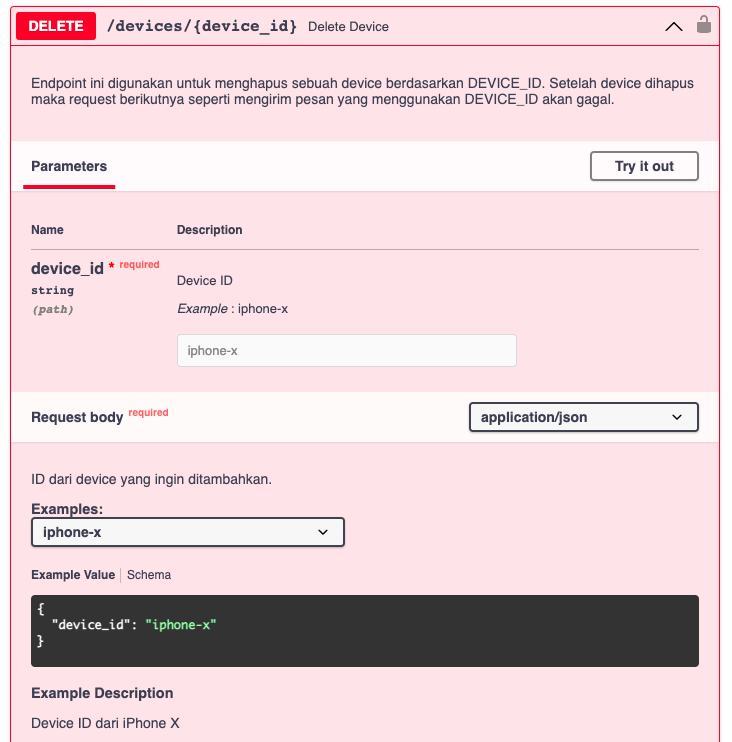


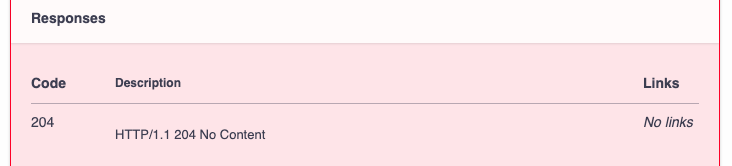


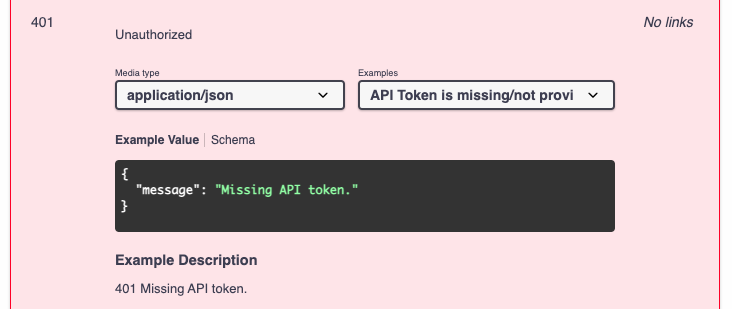


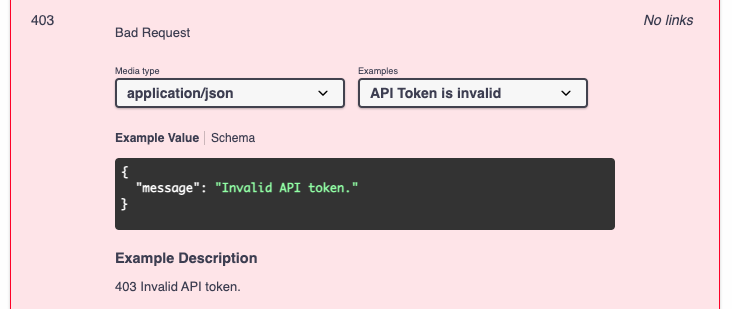


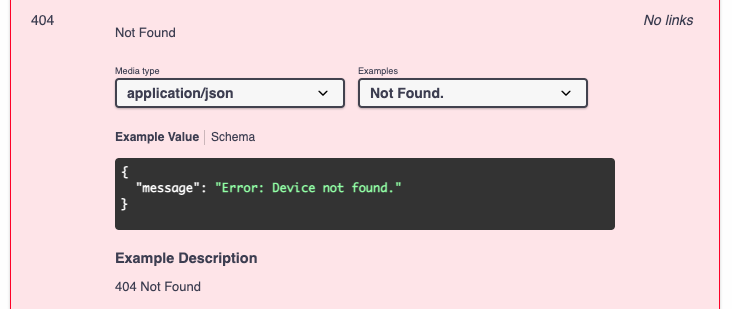
1. DEVICES – DELETE











1. MESSAGES
   * 1. Error Code

Pada API setiap request yang berhasil akan mengembalikan response HTTP code 2XX. Sedangkan jika terjadi error maka HTTP code yang dikembalikan adalah 4XX atau 5XX.

Tabel 4.1. Error Code dasar

|  |  |
| --- | --- |
| Error  Code | Keterangan |
| 400 | Bad Request/Request yang dilakukan tidak sesuai. |
| 401 | Unauthorized/API Token salah. |
| 403 | Forbidden/Tidak ada akses untuk URL yang diminta. |
| 404 | Not Found/Tidak ada halaman yang dimaksud. |
| 429 | Too Many Requests /Jumlah request yang dilakukan melebihi batas yang ditentukan. |
| 500 | Internal Server Error/Terjadi kesalahan pada server API |

* 1. Sfdssd
  2. c