



KEMENTERIAN
KESEHATAN
REPUBLIK
INDONESIA

Dokumentasi Teknis

SATUSEHAT

Master Wilayah ReST API

Pusat Data dan Teknologi Informasi - Digital Transformation Office

Versi 1.1, 13 Maret 2024
(master-b7fa2c9, 2024-03-14)

KERAHASIAAN INFORMASI

Informasi yang terkandung dalam dokumen ini bersifat:

PUBLIK

Yang berarti bahwa semua pihak/entitas yang mendapatkan dokumen ini dan/atau sebagian/keseluruhan informasi di dalam dokumen ini, **DIPERBOLEHKAN** untuk menggunakan, menerbitkan, menyebarluaskan, menduplikasi, atau mendistribusikan kembali dokumen ini dan/atau informasi di dalamnya dalam bentuk apapun, baik secara elektronik maupun mekanik tanpa izin tertulis dari **Pusat Data dan Teknologi Informasi – Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia**.

Pusat Data dan Teknologi Informasi – Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia tidak membuat pernyataan dan tidak memberikan jaminan dalam bentuk apapun sehubungan dengan informasi di dalam dokumen ini, termasuk tetapi tidak terbatas pada keakuratan atau kelengkapan informasi, fakta dan/atau pendapat yang terkandung di dalam dokumen ini.

Pusat Data dan Teknologi Informasi – Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia, para direktur, para karyawan, dan unit-unit di bawahnya tidak dapat dimintai pertanggungjawaban atas penggunaan dan kepercayaan atas opini, perkiraan, dan temuan dalam dokumen ini.

RIWAYAT PERUBAHAN

Versi	Tanggal Pembaruan	Deskripsi Perubahan	Author/Editor	Versi Aplikasi
1.1	13 Maret 2024	Update dan Rilis Dokumentasi Teknis SATUSEHAT: Master Wilayah ReST API versi AsciiDoc pdf 1. <i>Update query param.</i>	Ilham	
1.0.1	06 Februari 2024	Update dan Rilis Dokumentasi Teknis SATUSEHAT: Master Wilayah ReST API versi AsciiDoc pdf 1. <i>Hide endpoint development</i> 2. Ubah <i>endpoint</i> ReST API dari <i>development</i> ke <i>production</i>	Ilham	
1.0	08 Desember	Rilis Dokumentasi Teknis SATUSEHAT: Master Wilayah ReST ReST API versi AsciiDoc pdf	Ilham	
0.0	07 Desember 2023	Penyusunan Dokumentasi Teknis SATUSEHAT: Master Wilayah ReST API versi AsciiDoc pdf	Ilham	

DAFTAR ISI

KERAHASIAAN INFORMASI	1
RIWAYAT PERUBAHAN	2
PENGANTAR TEKNIS	7
1. Kata Kunci Prioritas	7
2. Simbol dan Istilah	8
3. ReST API	8
3.1. HTTP Method	8
3.2. HTTP Status Codes	9
4. ISO 8601	10
4.1. Format Tanggal dan Waktu	10
5. Pesan Prioritas	11
5.1. Catatan	11
5.2. Tips	11
5.3. Penting	12
5.4. Perhatian	12
5.5. Peringatan	12
A. PENDAHULUAN	13
1. Latar Belakang	13
2. Tujuan	13
3. Sistematika Dokumentasi	13
B. AUTENTIKASI	15
1. Endpoint	15
2. Postman Collection	15
3. Pengajuan dan Verifikasi Akses	16
4. Autentikasi - Mendapatkan Token	22
4.1. Request	22
URL	22
HTTP Verb/Method	22
Header	22
Query String	22
Body (<i>application/x-www-form-urlencoded</i>)	22
4.2. Response	23
2xx Success	23
4xx Client Error	24
5xx Server Error (<i>Content-Type: text/plain</i>)	24
4.3. Contoh Penggunaan/Kode	24
cURL (Windows)	24

cURL (Linux)	24
Postman	25
C. MASTER WILAYAH VERSI 1	26
1. Master Wilayah - Provinsi	27
1.1. Request	27
URL	27
HTTP Verb/Method	27
Header	27
Query String	27
1.2. Response	27
2xx Success	27
4xx Client Error	28
5xx Server Error (Content-Type: text/plain)	28
1.3. Contoh Penggunaan/Kode	28
cURL (Windows)	28
cURL (Linux)	29
Postman	29
2. Master Wilayah - Kota/Kabupaten	30
2.1. Request	30
URL	30
HTTP Verb/Method	30
Header	30
Query String	30
2.2. Response	30
2xx Success	30
4xx Client Error	31
5xx Server Error (Content-Type: text/plain)	31
2.3. Contoh Penggunaan/Kode	31
cURL (Windows)	32
cURL (Linux)	32
Postman	32
3. Master Wilayah - Kecamatan	33
3.1. Request	33
URL	33
HTTP Verb/Method	33
Header	33
Query String	33
3.2. Response	33
2xx Success	33
4xx Client Error	34
5xx Server Error (Content-Type: text/plain)	34

3.3. Contoh Penggunaan/Kode	34
cURL (Windows)	35
cURL (Linux)	35
Postman	35
4. Master Wilayah - Kelurahan/Desa	36
4.1. Request	36
URL	36
HTTP Verb/Method	36
Header	36
Query String	36
4.2. Response	36
2xx Success	36
4xx Client Error	37
5xx Server Error (Content-Type: text/plain)	37
4.3. Contoh Penggunaan/Kode	37
cURL (Windows)	38
cURL (Linux)	38
Postman	38
D. MASTER WILAYAH VERSI 2	39
1. Master Wilayah - Provinsi	40
1.1. Request	40
URL	40
HTTP Verb/Method	40
Header	40
Query String	40
1.2. Response	41
2xx Success	41
4xx Client Error	41
5xx Server Error (Content-Type: text/plain)	42
1.3. Contoh Penggunaan/Kode	42
cURL (Windows)	42
cURL (Linux)	42
Postman	42
2. Master Wilayah - Kota/Kabupaten	44
2.1. Request	44
URL	44
HTTP Verb/Method	44
Header	44
Query String	44
2.2. Response	45
2xx Success	45

4xx Client Error	46
5xx Server Error (Content-Type: text/plain)	46
2.3. Contoh Penggunaan/Kode	46
cURL (Windows)	46
cURL (Linux)	46
Postman	46
3. Master Wilayah - Kecamatan	48
3.1. Request	48
URL	48
HTTP Verb/Method	48
Header	48
Query String	48
3.2. Response	49
2xx Success	49
4xx Client Error	50
5xx Server Error (Content-Type: text/plain)	50
3.3. Contoh Penggunaan/Kode	50
cURL (Windows)	50
cURL (Linux)	50
Postman	50
4. Master Wilayah - Kelurahan/Desa	52
4.1. Request	52
URL	52
HTTP Verb/Method	52
Header	52
Query String	52
4.2. Response	53
2xx Success	53
4xx Client Error	54
5xx Server Error (Content-Type: text/plain)	54
4.3. Contoh Penggunaan/Kode	54
cURL (Windows)	54
cURL (Linux)	54
Postman	54
DAFTAR ISTILAH	56
REFERENSI	57

PENGANTAR TEKNIS

Beberapa simbol, istilah, dan konvensi digunakan pada pedoman ini agar penyampaian konsep, struktur data, baik ke tim *developer* maupun non *developer* menjadi lebih jelas. Pada bagian ini akan dijelaskan lebih lanjut terkait hal tersebut.



Setiap teks yang berwarna **biru muda**, dapat diklik untuk melompat ke bagian yang direferensikan.

1. Kata Kunci Prioritas

Dalam banyak dokumen terkait standar, beberapa kata digunakan untuk menandakan persyaratan dalam spesifikasi. Kata-kata ini sering dikapitalisasi dan/atau ditebalkan. Pada bagian ini akan didefinisikan kata-kata ini seperti yang dilakukan pada banyak dokumen RFC (*Request for Comment*), dengan beracuan pada standar di **RFC 2119**[<https://www.ietf.org/rfc/rfc2119.txt>] beberapa kata-kata tersebut dialihbahasakan ke bahasa Indonesia.

- **WAJIB**, berarti definisi tersebut merupakan persyaratan mutlak dari spesifikasi.
- **TIDAK BOLEH**, berarti definisi tersebut merupakan larangan mutlak terhadap spesifikasi.
- **SEBAIKNYA**, berarti bahwa ada alasan yang sah dan kuat dalam keadaan tertentu untuk mengesampingkan hal tertentu lainnya, tetapi implikasi dari perilaku tersebut harus dipahami dan dipertimbangkan dengan cermat sebelum memilih alur yang berbeda.
- **SEBAIKNYA TIDAK**, berarti bahwa ada alasan yang sah dan kuat dalam keadaan tertentu ketika perilaku tertentu dapat diterima atau bahkan berguna, tetapi implikasi dari perilaku harus dipahami dan kasusnya dipertimbangkan dengan cermat sebelum menerapkan perilaku apa pun yang dijelaskan dengan label ini.
- **DIREKOMENDASIKAN**, sama definisinya dengan **SEBAIKNYA**.
- **TIDAK DIREKOMENDASIKAN**, sama definisinya dengan **SEBAIKNYA TIDAK**.
- **BOLEH**, berarti bahwa suatu spesifikasi benar-benar opsional. Satu vendor dapat memilih untuk memasukkan spesifikasi tersebut karena kondisi tertentu membutuhkannya atau karena vendor merasa bahwa itu meningkatkan performa tertentu sementara vendor lain mungkin menghilangkan spesifikasi yang sama. Implementasi yang tidak menyertakan opsi tertentu **PERLU** siap untuk beroperasi dengan implementasi lain yang menyertakan opsi, meskipun mungkin dengan fungsionalitas yang berkurang. Dalam hal yang sama, implementasi yang menyertakan opsi tertentu **PERLU** siap untuk beroperasi dengan implementasi lain yang tidak menyertakan opsi (kecuali, tentu saja, untuk fitur yang disediakan opsi.)
- **OPSIONAL**, sama definisinya dengan **BOLEH**.

Penggunaan kata-kata tadi biasanya untuk panduan dari suatu spesifikasi agar berjalan sesuai yang diharapkan. Bisa juga terkait pertimbangan keamanan, dimana kata-kata tersebut sering digunakan untuk menentukan perilaku dengan implikasi keamanan. Efek pada keamanan karena tidak menerapkan **WAJIB** atau **SEBAIKNYA**, atau melakukan sesuatu yang menurut spesifikasi **TIDAK BOLEH** atau **SEBAIKNYA TIDAK** dilakukan mungkin sangat bijak.

2. Simbol dan Istilah

Berikut beberapa simbol dan istilah yang digunakan dalam dokumentasi ini.

Tabel 1. Penamaan dan Simbol

Penamaan	Simbol/Nilai	Penjelasan
string	"" atau ''	Representasi dari kumpulan karakter/huruf/symbol.
number	"1", "-1", "1.0", '1', '-1', '1.0',	Representasi dari kumpulan angka baik dari bilangan bulat atau desimal, dalam bentuk string .
uuid	uuid, uuid1, uuid2, uuid3, uuid4	Merepresentasikan ID universal unik dari suatu entitas, yang terdiri dari 36 karakter (32 karakter heksadesimal dan 4 tanda minus), lebih lengkapnya silakan lihat di https://en.wikipedia.org/wiki/Universally_unique_identifier . Contoh data uuid : 123e4567-e89b-12d3-a456-426614174000

3. ReST API

ReST adalah akronim dari *Representational State Transfer* dan salah satu tipe arsitektur untuk sistem hypermedia terdistribusi. Roy Fielding pertama kali mempresentasikannya pada tahun 2000 dalam disertasinya yang terkenal [\[restfulapi.net\]](https://restfulapi.net) [\[wikipedia-rest\]](https://wikipedia-rest).

ReST melakukan pertukaran data sesuai standar yang ada pada *HTTP Specification*, sehingga perlu dijelaskan secara singkat terkait beberapa *HTTP method* dan *HTTP status code* yang diantaranya digunakan dalam dokumentasi teknis ini.

3.1. HTTP Method

HTTP Method mendefinisikan metode untuk menunjukkan aksi yang sesuai yang akan dilakukan pada *resource* yang telah teridentifikasi. *Resource* sendiri merepresentasikan data yang sudah ada atau data dinamis yang tergantung dari apa yang sudah disepakati dan diterapkan pada *server* yang dituju [\[wikipedia-http-method\]](https://wikipedia-http-method).

Tabel 2. HTTP Method

Method	Penjelasan
GET	Meminta representasi <i>resource</i> dari sumber tertentu untuk di READ oleh peminta.
POST	Meminta server menerima entitas yang terlampir dalam <i>request</i> sesuai struktur yang disepakati sebelumnya dan memprosesnya sebagai proses CREATE .
PUT	Meminta server menerima entitas yang terlampir dalam <i>request</i> sesuai struktur yang disepakati sebelumnya dan memprosesnya sebagai proses UPDATE dari data yang dimaksud.

Method	Penjelasan
PATCH	Meminta server menerima sebagian entitas yang terlampir dalam <i>request</i> sesuai struktur yang disepakati sebelumnya dan memprosesnya sebagai proses PARTIAL UPDATE dari data yang dimaksud.
DELETE	Meminta agar <i>resource</i> yang dimaksud di- DELETE .

3.2. HTTP Status Codes

HTTP Status Codes dikeluarkan dari *server* sebagai bentuk *response* dari *request* yang berasal dari *client* [wikipedia-http-status-code]. Kode-kode ini dikategorikan menjadi 5 kelas yang dibedakan dari digit pertamanya yaitu: **1xx** *informational response*, **2xx** *successful*, **3xx** *redirection*, **4xx** *client error*, dan **5xx** *server error*. Pada bagian ini, hanya akan dijelaskan 3 *HTTP Status Codes* yang sering digunakan di ReST yaitu **2xx**, **4xx**, dan **5xx**.

Tabel 3. HTTP Method

Method	Penjelasan
200 OK	Respon standar untuk permintaan yang berhasil.
201 Created	Permintaan berhasil dan <i>resource</i> yang dimaksud telah dibuat. Biasanya akan ada header location yang menunjukkan letak <i>resource</i> yang sudah dibuat.
202 Accepted	Permintaan berhasil dan <i>resource</i> yang dimaksud telah dijadwalkan akan diproses. Biasanya akan ada header location yang menunjukkan letak <i>resource</i> yang akan menunjukkan status atau hasil proses yang sudah atau akan dikerjakan.
204 No Content	Permintaan berhasil dan <i>resource</i> yang dimaksud telah dibuat, tetapi server tidak menyediakan data kembalian/response apapun (pesan di body kosong).
206 Partial Content	Biasanya digunakan dalam proses <i>download</i> data secara partial (<i>byte serving</i> [https://en.wikipedia.org/wiki/Byte_serving]) sesuai rentang yang didefinisikan di- <i>header</i> .
400 Bad Request	Permintaan gagal dikarenakan ada kesalahan di bagian <i>client</i> , seperti struktur data yang tidak sesuai, masukan data yang tidak sah, atau ukuran <i>resource</i> yang dikirim terlalu besar.
401 Unauthorized	Permintaan gagal dikarenakan perlu ada proses autentikasi (<i>authentication</i>) tertentu untuk mengaksesnya atau hasil autentikasi tidak sah.
403 Forbidden	Permintaan gagal dikarenakan perlu ada proses otorisasi (<i>authorization</i>) tertentu untuk mengaksesnya atau otorisasi tidak ditolak.
404 Not Found	Permintaan gagal dikarenakan <i>resource</i> yang dimaksud tidak ditemukan.
405 Method Not Allowed	Permintaan gagal dikarenakan <i>HTTP Method</i> yang digunakan untuk meminta <i>resource</i> tidak sesuai atau tidak diperbolehkan.

Method	Penjelasan
409 Conflict	Permintaan gagal dikarenakan adanya konflik terkait <i>resource</i> yang diakses, misal pembaruan <i>resource</i> yang sama secara bersamaan.
413 Payload Too Large	Permintaan gagal dikarenakan ukuran <i>payload</i> yang dikirimkan melebihi batas yang bisa diproses oleh <i>server</i> .
415 Unsupported Media Type	Permintaan gagal dikarenakan <i>server</i> tidak mendukung jenis <i>resource</i> yang dikirimkan.
422 Unprocessable Entity	Permintaan gagal dikarenakan adanya galat terkait semantik dari data atau <i>resource</i> yang dikirimkan saat diproses.
429 Too Many Requests	Permintaan gagal dikarenakan <i>client</i> terlalu cepat atau banyak mengakses <i>server</i> , biasanya galat ini terkait dengan ketentuan dari skema <i>rate-limiter</i> .
500 Internal Server Error	Permintaan gagal dikarenakan ada kesalahan dibagian <i>server</i> .
501 Not Implemented	Target permintaan yang dimaksud belum diimplementasikan di <i>server</i> .
503 Service Unavailable	<i>Server</i> dalam kondisi mati atau dalam pemeliharaan (<i>maintenance</i>).
504 Gateway Timeout	<i>Server</i> hanya bertindak sebagai <i>gateway</i> atau <i>proxy</i> dan menerima <i>response</i> yang salah dari <i>server</i> sumber.

4. ISO 8601

ISO 8601 adalah sebuah standar internasional yang digunakan pada pertukaran data terkait tanggal dan waktu [wikipedia-iso8601]. Format tanggal yang digunakan pada dokumentasi teknis ini sebagian besar mengacu pada standar ISO 8601 untuk merepresentasikan tanggal dan waktu.

4.1. Format Tanggal dan Waktu

Tabel 4. Format ISO8601 untuk tanggal dan waktu

Simbol	Penjelasan
YYYY	Merepresentasikan 4 digit tahun, <i>zero-padded</i> dibagian kiri, mulai dari 0000 hingga 9999. Tahun 0000 sama dengan tahun 1 Sebelum Masehi (SM). Representasi tahun Sebelum Masehi harus menggunakan tanda minus -0000 = 1 SM, -0001 = 2 SM, dst.
MM	Merepresentasikan 2 digit bulan, <i>zero-padded</i> dibagian kiri, dalam satu tahun mulai dari 01 hingga 12. Secara berurutan merepresentasikan bulan: 01 Januari, 02 Februari, 03 Maret, 04 April, 05 Mei, 06 Juni, 07 Juli, 08 Agustus, 09 September, 10 Oktober, 11 November, 12 Desember.
DD	Merepresentasikan 2 digit hari, <i>zero-padded</i> dibagian kiri, dalam satu bulan mulai dari 01 hingga 31 tergantung dari bulan yang dimaksud.

Simbol	Penjelasan
hh	Merepresentasikan 2 digit jam, <i>zero-padded</i> dibagian kiri, dari 00 hingga 23.
mm	Merepresentasikan 2 digit menit, <i>zero-padded</i> dibagian kiri, dari 00 hingga 59.
ss	Merepresentasikan 2 digit detik, <i>zero-padded</i> dibagian kiri, dari 00 hingga 60.
.sss	Merepresentasikan 3 digit milidetik, <i>zero-padded</i> dibagian kiri.
.sssss	Merepresentasikan 6 digit nanodetik, <i>zero-padded</i> dibagian kiri.

Contoh penggunaan dari format yang telah dijelaskan sebelumnya:

- YYYY-MM-DD untuk merepresentasikan tanggal, contoh: 2021-01-02 (2 Januari 2021).
- hh:mm:ss untuk merepresentasikan waktu lokal, contoh: 22:10:58 (pukul 22 lewat 10 menit 58 detik).
- YYYY-MM-DD hh:mm:ss untuk merepresentasikan tanggal dan waktu lokal, contoh: 2021-01-02 22:10:58 (2 Januari 2021, pukul 22 lewat 10 menit 58 detik).
- hh:mm:ssZ atau hh:mm:ss+0000 untuk merepresentasikan waktu UTC/GMT, contoh: 10:10:58Z (pukul 10 lewat 10 menit 58 detik UTC) bila dikonversi ke zona Asia/Jakarta (UTC+7) menjadi 17:10:58+0700.
- YYYY-MM-DD hh:mm:ssZ untuk merepresentasikan tanggal dan waktu lokal, contoh: 2021-01-02 10:10:58Z (2 Januari 2021, pukul 10 lewat 10 menit 58 detik UTC) bila dikonversi ke zona Asia/Jakarta (UTC+7) menjadi 2 Januari 2021, 17:10:58+0700.
- YYYY-MM-DD hh:mm:ss.sssss untuk merepresentasikan tanggal dan waktu lokal hingga ke 3 digit mili detik, contoh: 2021-01-02 10:10:58.353 (2 Januari 2021, pukul 10 lewat 10 menit 58 detik 353 mili detik).

5. Pesan Prioritas

Beberapa pesan prioritas berisi pernyataan, peringatan, perhatian, atau teguran tertentu sebagai salah satu cara untuk menekankan beberapa penjelasan yang dianggap penting, digunakan pada dokumentasi ini. Tipe-tipe dari kotak prioritas tersebut antara lain:

5.1. Catatan

Memberikan catatan singkat terkait informasi tertentu. Contoh:



Postman adalah salah satu aplikasi untuk melakukan *testing* terkait ReST API yang *user-friendly* dibandingkan cURL.

5.2. Tips

Memberikan saran kepada pembaca yang terkait *best practices* atau tips tertentu. Contoh:



Untuk versi cURL terbaru, gunakan parameter `--oauth2-bearer` untuk melakukan *request* yang membutuhkan *token bearer*.

5.3. Penting

Memberikan saran kepada pembaca bahwa anjuran yang disebutkan penting untuk dibaca. Contoh:



Selalu pakai *token bearer* yang didapatkan setiap melakukan *request* API yang bukan untuk autentikasi.

5.4. Perhatian

Memberikan saran kepada pembaca agar anjuran yang telah dijelaskan dilakukan dengan hati-hati. Contoh:



Mohon diperhatikan saat melakukan *request* data, dikarenakan ada perbedaan target URL untuk versi *production* dan *development*.

5.5. Peringatan

Memberikan tekanan kepada pembaca agar anjuran yang telah dijelaskan dilakukan dengan serius dan hati-hati, dikarenakan kemungkinan akan ada efek samping. Contoh:



Cek apakah *payload* yang akan dikirimkan sudah sesuai standar yang disepakati sebelumnya, karena bila tidak sesuai *payload* tersebut tidak akan dapat diproses.

A. PENDAHULUAN

1. Latar Belakang

Di samping adanya dokumentasi secara umum terkait Master Wilayah, maka diperlukan satu dokumentasi khusus yang membahas semua ReST API yang disediakan oleh Master Wilayah yang dikembangkan oleh tim *developer* **Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia**. Dokumentasi ReST API Master Wilayah ini berisi spesifikasi teknis ReST API tersebut mencakup parameter yang tersedia, contoh *request*, contoh hasil *response*, penjelasan dari kode status HTTP yang mungkin diterima, serta cara penggunaannya menggunakan cURL dan Postman.

2. Tujuan

Dokumentasi **ReST API Wilayah** dibuat untuk kebutuhan ReST API yang tersedia dari Master Wilayah yang telah dikembangkan oleh tim *developer* SATUSEHAT. Penjelasan ReST API Master Wilayah di dokumentasi ini hanya terbatas dari spesifikasi dari ReST API itu sendiri. Penjelasan lengkap terkait Master Wilayah, akan dijelaskan pada dokumentasi tersendiri.

3. Sistematika Dokumentasi

Sistematika pembahasan ReST API Master Wilayah pada dokumentasi ini dibagi menjadi beberapa bagian yang terdiri dari:

Lembar Kerahasiaan Informasi, berisi pernyataan terkait siapa saja yang boleh memiliki akses terkait dokumentasi ini.

Lembar Riwayat Perubahan, berisi riwayat perubahan terkait isi dari dokumentasi ini dan detail singkat dari perubahan tersebut.

Daftar Isi, berisi daftar nama bab, sub bab, dan halaman terkait pada dokumentasi teknis ReST API Master Wilayah ini.

Pengantar Teknis, berisi pengantar teknis singkat terkait beberapa konvensi yang digunakan pada dokumentasi teknis ini. Di bagian ini juga terdapat penjelasan singkat mengenai ReST API serta format tanggal dan waktu sesuai ISO 8601.

Pendahuluan, berisi penjelasan mengenai latar belakang, tujuan, dan sistematika penyusunan dokumentasi teknis terkait ReST API Master Wilayah ini.

Autentikasi, berisi penjelasan dan spesifikasi teknis terkait ReST API yang digunakan untuk proses autentikasi ke Master Wilayah melalui ReST API yang disediakan.

Daftar Istilah, berisi istilah-istilah yang mungkin ada dan perlu penjelasan lebih lanjut yang sebagian atau selalu digunakan pada dokumentasi teknis ReST API Master Wilayah ini.

Referensi, berisi daftar referensi yang digunakan atau untuk penjelasan lebih lanjut dari suatu topik

yang dibahas pada dokumentasi teknis ReST API Master Wilayah ini.

B. AUTENTIKASI

Untuk melakukan transaksi data dari Master Sarana Index (MSI), perlu dilakukan proses autentikasi terlebih dahulu agar mendapatkan akses yang tersedia. Autentikasi yang digunakan oleh MSI mengikuti standar protokol OAuth 2 dengan tipe pemberian akses (*grant type*) adalah **client_credentials**.

Autentikasi menggunakan *grant type* **client_credentials** adalah proses autentikasi yang dilakukan antara *server to server*, sehingga tidak ada proses registrasi atau *log in* di sini. Autentikasi dengan tipe tersebut hanya memerlukan data berupa **client_id** dan **client_server**, dimana nilai tersebut didapatkan ketika pihak yang ingin menggunakan atau mengakses MSI ini telah melakukan pengajuan, terdaftar, serta mendapatkan persetujuan dari **Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia**.

Cara Mendapatkan Nilai dari **client_id** dan **client_secret**

Silakan terlebih dahulu melakukan **Pengajuan dan Verifikasi Akses** pada website <https://satusehat.kemkes.go.id/platform> melalui web *browser* Anda. Untuk informasi lebih lanjut dapat dilihat pada [Pengajuan dan Verifikasi Akses](#)



Setiap teks yang berwarna **biru muda**, dapat diklik untuk melompat ke bagian yang direferensikan.

1. Endpoint

Pada bagian ini akan dijelaskan spesifikasi untuk **ReST API SATUSEHAT Terkait Autentikasi**, yang mempunyai *endpoint* berdasarkan jenis lingkungan pengembangannya (*development environment*) yaitu:

- *staging* <https://api-satusehat-stg.dto.kemkes.go.id/oauth2/v1>
- *production* <https://api-satusehat.kemkes.go.id/oauth2/v1>



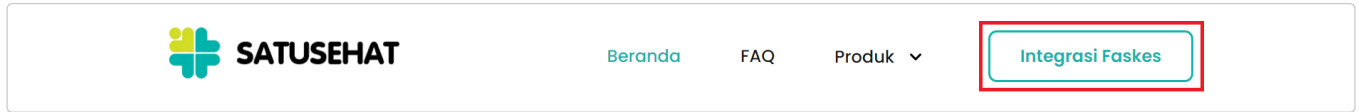
Semua penerapan, penjelasan, dan contoh yang akan dibahas akan menggunakan *endpoint production*.

2. Postman Collection

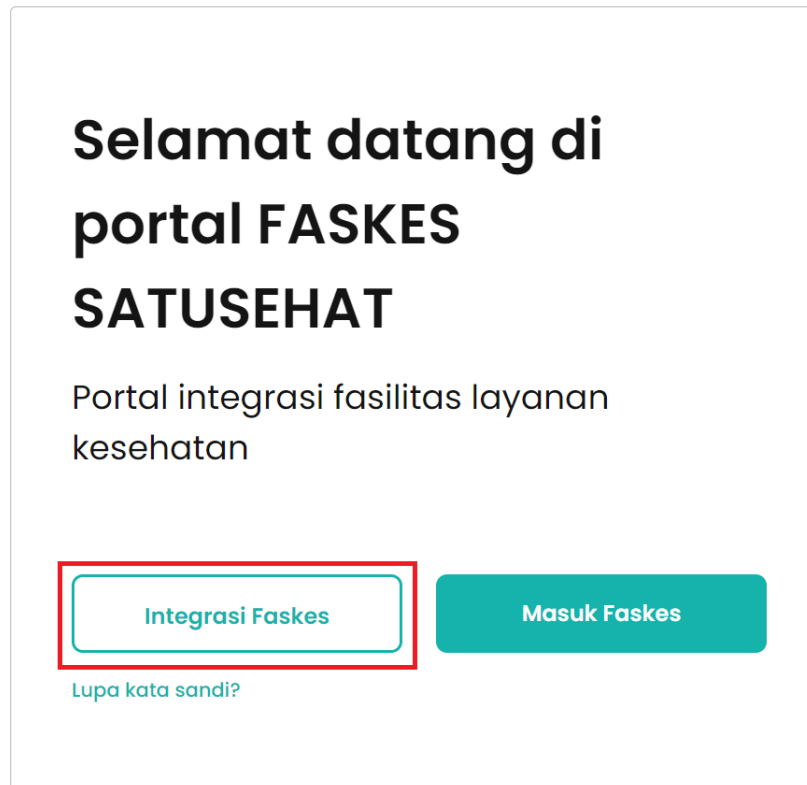
Silakan mengunduh Postman Collection untuk melihat contoh/melakukan workshop secara mandiri pada website [Postman Collection Wilayah](https://link.kemkes.go.id/MasterDataAPI) [<https://link.kemkes.go.id/MasterDataAPI>] melalui web *browser* Anda.

3. Pengajuan dan Verifikasi Akses

1. Untuk bergabung dengan Ekosistem FASKES SATUSEHAT, akses laman <https://satusehat.kemkes.go.id/platform> melalui web *browser* Anda, kemudian klik **Integrasi Faskes**.



2. Pada laman <https://satusehat.kemkes.go.id/platform/welcome> klik **Integrasi Faskes** untuk membuat akun SATUSEHAT.



3. Lengkapi data identitas berdasarkan *form* yang disediakan, kemudian klik **Buat Akun** untuk mendapatkan konfirmasi aktivasi akun pada *e-mail* yang didaftarkan.

Buat Akun FASKES SATUSEHAT Anda

Nama Pengelola Faskes

Email

Nomor telepon

Kata sandi

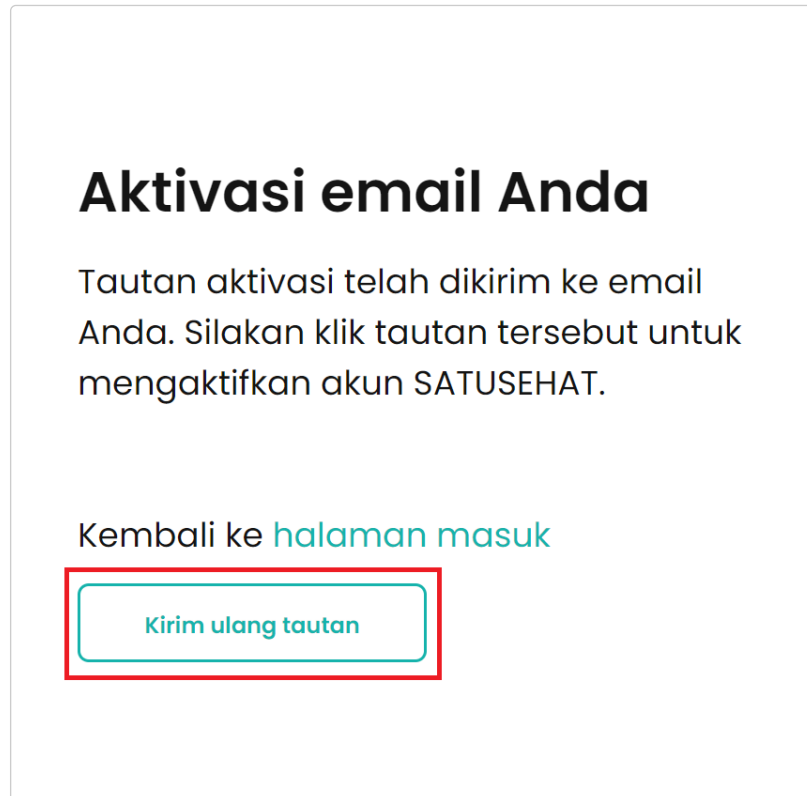
Minimal 8 karakter termasuk huruf kapital (A-Z), huruf kecil (a-z) dan angka (0-9)

Konfirmasi kata sandi

Minimal 8 karakter termasuk huruf kapital (A-Z), huruf kecil (a-z) dan angka (0-9)

Buat akun

4. Selanjutnya akan muncul laman **Aktivasi e-mail Anda**, apabila aktivasi akun SATUSEHAT tidak terkirim pada e-mail yang telah didaftarkan, klik **Kirim ulang tautan**.



5. Buka e-mail aktivasi akun SATUSEHAT, pada detail/badan e-mail klik **Aktifkan akun** atau dengan menyalin tautan pada browser Anda.



6. Aktivasi akun berhasil, Anda dapat melanjutkan Integrasi Faskes dengan klik **Masuk**.

Aktivasi email Anda berhasil

Klik tombol dibawah untuk login.

Masuk

7. Pada laman *login* masukan alamat *e-mail* dan kata sandi yang telah terdaftar, kemudian klik **Masuk**.

Masuk ke SATUSEHAT

Alamat email

Masukkan alamat email

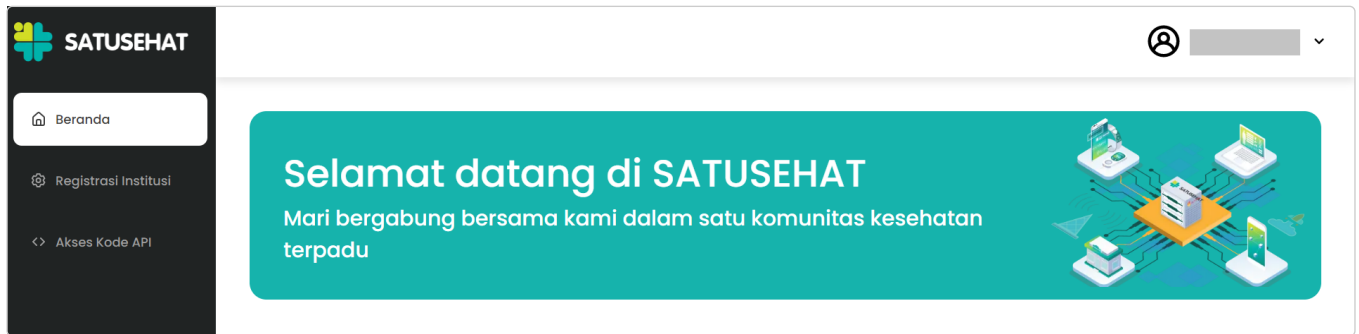
Kata sandi [Lupa kata sandi?](#)

Masukkan kata sandi

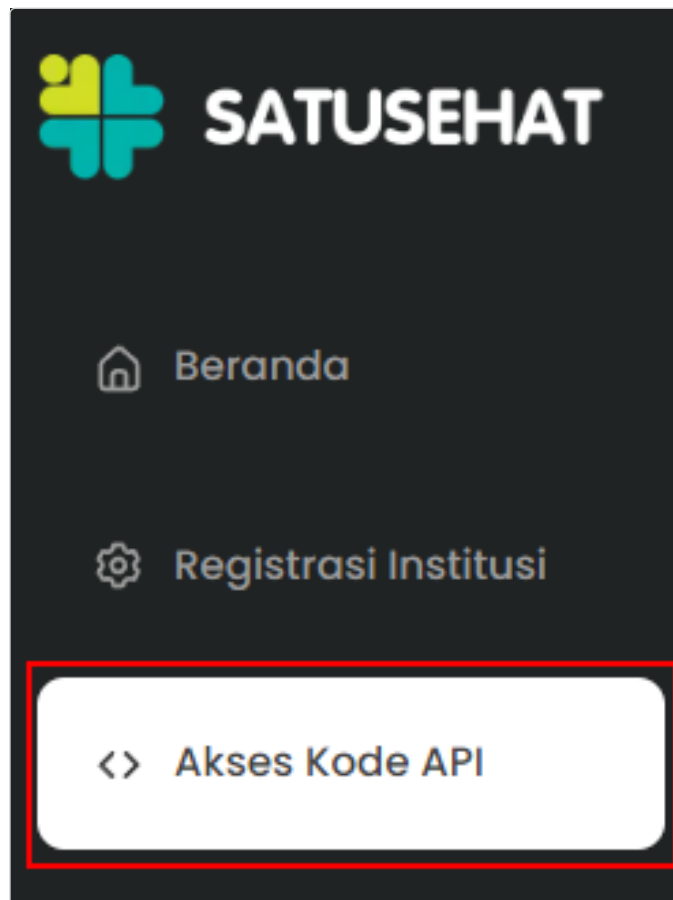
Minimal 8 karakter termasuk huruf kapital (A-Z), huruf kecil (a-z) dan angka (0-9)

Masuk

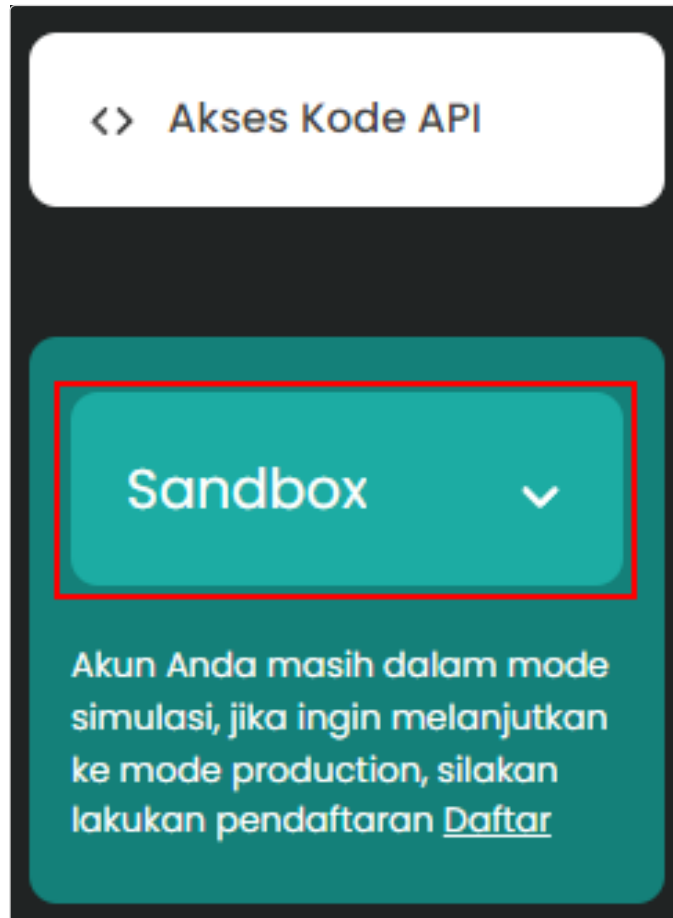
8. Setelah *login*, Anda akan masuk ke laman beranda.



9. Pada *sidebar* menu, klik menu <> **Akses Kode API**.



10. Pada *sidebar* menu, dibawah menu <> **Akses Kode API**. Pilih *endpoint* API yang diinginkan, diantaranya **Sandbox(Development)**, **Staging**, atau **Production**.



11. Anda dapat melihat **Organization ID**, **Client Key**, dan **Secret Key** yang dapat digunakan pada API SATUSEHAT sesuai dengan *endpoint* API yang diinginkan.

Akses Kode API

i Kode API sandbox
Akses mode sandbox dengan menggunakan kode berikut untuk terintegrasi langsung dengan SATUSEHAT

Organization ID

[Salin](#)

Client Key

[Salin](#)

Secret Key

[Salin](#)

4. Autentikasi - Mendapatkan Token

Melakukan proses autentikasi untuk mendapatkan akses token yang akan dipakai pada setiap *request* ReST API Master Sarana Index (MSI) selanjutnya.



Setiap terdapat simbol asterik ***** sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: ***variabel**.

4.1. Request

URL

`https://api-satusehat.kemkes.go.id/oauth2/v1/accesstoken`

HTTP Verb/Method

POST

Header

Nama Parameter	Tipe Data	Keterangan
*Content-Type	string	Mime type dari <i>payload</i> data yang akan dikirimkan di dalam <i>body</i> dalam bentuk <i>URL Encoded</i> , WAJIB diisi dengan <code>application/x-www-form-urlencoded</code>

Query String

Nama Parameter	Tipe Data	Keterangan
*grant_type	string	Tipe permintaan akses (<i>grant</i>) Oauth2, WAJIB diisi dengan <code>client_credentials</code> .

Body (`application/x-www-form-urlencoded`)

Nama Parameter	Tipe Data	Keterangan
*client_id	string	Nilai <i>client ID</i> yang telah didapatkan dari Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia setelah melakukan Pengajuan dan Verifikasi Akses pada website https://satusehat.kemkes.go.id/platform , WAJIB diisi. Nilai ini bisa disamakan seperti <i>username</i> yang akan digunakan untuk akses aplikasi.

Nama Parameter	Tipe Data	Keterangan
* <code>client_secret</code>	<code>string</code>	Nilai <i>client secret</i> yang telah didapatkan dari Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia setelah melakukan <i>Pengajuan dan Verifikasi Akses</i> pada website https://satusehat.kemkes.go.id/platform , WAJIB diisi. Nilai ini bisa disamakan seperti kata sandi (<i>password</i>) yang akan digunakan untuk akses aplikasi.

Contoh Data



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

```
client_id: <client-id>
client_secret: <client-secret>
```

4.2. Response

Hasil *response*, dengan HTTP *Status Code* berpola `2xx` atau `4xx`, yang dikembalikan dari server mempunyai parameter *Content-Type* dengan nilai `application/json` di salah satu parameter *header*-nya.

2xx Success

Dari hasil *response* ini, **PERLU** disimpan nilai akses token yang didapat dari properti `access_token`, dimana tipe token (lihat properti `token_type`) tersebut adalah `BearerToken`. Nilai akses token tersebut **WAJIB** selalu digunakan sebagai nilai dari *header* `Authorization: Bearer <access_token>` saat melakukan *request* lainnya dari ReST API SATUSEHAT.

Contoh Data



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

```
{
  "refresh_token_expires_in": "0",
  "api_product_list": "[api-sandbox]",
  "api_product_list_json": [
    "api-sandbox"
  ],
  "organization_name": "ihs-prod-1",
  "developer.email": "<developer-email>",
  "token_type": "BearerToken",
  "issued_at": "1688958601590",
  "client_id": "<client-id>",
  "access_token": "<access-token>",
  "application_name": "0b1dc094-62cc-4bb1-85a5-616562305ed2",
  "scope": "",
}
```

```
{
  "expires_in": "14399",
  "refresh_count": "0",
  "status": "approved"
}
```

4xx Client Error

Sistem akan mengembalikan pesan *error* bila *client* belum melakukan autentikasi, tidak memiliki akses, menggunakan HTTP *method* yang tidak tepat, atau mengirimkan data dengan format atau ketentuan yang tidak sesuai.

Contoh Data

```
{
  "resourceType": "OperationOutcome",
  //data.terkait.resource.OperationOutcome
}
```

5xx Server Error (Content-Type: text/plain)

Sistem akan mengembalikan pesan *error* bila terjadi kesalahan pada sisi server saat memproses data yang telah dikirimkan.

Contoh Data

Gateway Timeout

4.3. Contoh Penggunaan/Kode



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

cURL (Windows)

```
curl --insecure --location \
  --header "Content-Type: application/x-www-form-urlencoded" ^
  --data-urlencode "client_id=<client-id>" ^
  --data-urlencode "client_secret=<client-secret>" ^
  --request POST ^
  "https://api-satusehat.kemkes.go.id/oauth2/v1/accesstoken?grant_type=client_credentials"
```

cURL (Linux)

```
curl --insecure --location \
  --header 'Content-Type: application/x-www-form-urlencoded' \
  --data-urlencode 'client_id=<client-id>' \
  --data-urlencode 'client_secret=<client-secret>' \
  --request POST \
  'https://api-satusehat.kemkes.go.id/oauth2/v1/accesstoken?grant_type=client_credentials'
```

Postman

1. Buat *request* baru menggunakan **New** › **HTTP Request**, atau klik  untuk buat tab *request* baru.
2. Masukkan *request* URL

```
https://api-satusehat.kemkes.go.id/oauth2/v1/accesstoken
```

3. Lalu pilih *request method* **POST**.
4. Pada tab **Params**, di bagian **Query Params**:
 - a. masukan nilai **grant_type** pada kotak masukan pada kolom **KEY**,
 - b. lalu masukan nilai **client_credentials** pada kotak masukan pada kolom **VALUE**.
5. Pada tab **Body**:
 - a. pilih **x-www-form-urlencoded**,
 - b. masukan nilai **client_id** pada kotak masukan pada kolom **KEY**,
 - c. lalu masukan nilai *client ID* yang sudah didapatkan dari **Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia** pada kotak masukan pada kolom **VALUE**,
 - d. selanjutnya masukan nilai **client_secret** pada kotak masukan pada kolom **KEY**,
 - e. terakhir masukan nilai *client secret* yang sudah didapatkan dari **Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia** pada kotak masukan pada kolom **VALUE**.
6. Klik tombol .
7. Hasil *response* akan ditampilkan di bagian **Response**.

C. MASTER WILAYAH VERSI 1

Untuk melakukan transaksi data dari Master Wilayah, perlu dilakukan proses autentikasi terlebih dahulu agar mendapatkan akses yang tersedia. Autentikasi yang digunakan oleh Master Wilayah mengikuti standar protokol OAuth 2 dengan tipe pemberian akses (*grant type*) adalah `client_credentials`.

Autentikasi menggunakan *grant type* `client_credentials` adalah proses autentikasi yang dilakukan antara *server to server*, sehingga tidak ada proses registrasi atau *log in* di sini. Autentikasi dengan tipe tersebut hanya memerlukan data berupa `client_id` dan `client_secret`, dimana nilai tersebut didapatkan ketika pihak yang ingin menggunakan atau mengakses Wilayah ini telah melakukan pengajuan, terdaftar, serta mendapatkan persetujuan dari **Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia**.

Cara Mendapatkan Nilai dari `client_id` dan `client_secret`

Silakan terlebih dahulu melakukan **Pengajuan dan Verifikasi Akses** pada website <https://satusehat.kemkes.go.id/platform> melalui web *browser* Anda. Untuk informasi lebih lanjut dapat dilihat pada **Pengajuan dan Verifikasi Akses**



Setiap teks yang berwarna **biru muda**, dapat diklik untuk melompat ke bagian yang direferensikan.

Pada bagian ini akan dijelaskan spesifikasi untuk **ReST API Master Wilayah**, yang mempunyai *endpoint* berdasarkan jenis lingkungan pengembangannya (*development environment*) yaitu:

API Master Wilayah Versi 1

- *staging* <https://api-stg.dto.kemkes.go.id/masterdata/v1>
- *production* <https://api.kemkes.go.id/masterdata/v1>



Semua penerapan, penjelasan, dan contoh yang akan dibahas akan menggunakan *endpoint production*.



Versi 1 belum menerapkan *pagination* dan penggunaan filter kode harus berdasarkan kode dari Kemendagri.

1. Master Wilayah - Provinsi



Setiap terdapat simbol asterik ***** sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: ***variabel**.

1.1. Request

URL

```
https://api-stg.dto.kemkes.go.id/masterdata/v1/provinces?codes
```

HTTP Verb/Method

GET

Header

Nama Parameter	Tipe Data	Keterangan
*Authorization	string	Header ini WAJIB diisi dengan nilai sesuai format: Bearer <access_token> . Nilai dari variabel <access_token> didapatkan dari properti access_token pada object dari hasil response JSON setelah proses autentikasi.

Query String

Nama Parameter	Tipe Data	Keterangan
codes	number	Isi dengan kode provinsi berdasarkan kode wilayah dari Kemendagri. (dapat di- <i>input</i> multi dengan menyertakan koma) Contoh: 11, 12 .
search	text	Isi dengan nama provinsi yang ingin dicari

1.2. Response

Hasil *response*, dengan HTTP *Status Code* berpola **2xx** atau **4xx**, yang dikembalikan dari server mempunyai parameter **Content-Type** dengan nilai **application/json** di salah satu parameter *header*-nya.

2xx Success

Contoh Data



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

```
{
```

```
"status": 200,
"error": false,
"message": "success",
"data": [
  {
    "code": "11",
    "parent_code": "",
    "bps_code": "11",
    "name": "Aceh"
  },
  /* lompat beberapa data */
  {
    "code": "96",
    "parent_code": "",
    "bps_code": "",
    "name": "Papua Barat Daya"
  }
]
```

4xx Client Error

Sistem akan mengembalikan pesan *error* bila *client* belum melakukan autentikasi, tidak memiliki akses, menggunakan HTTP *method* yang tidak tepat, atau mengirimkan data dengan format atau ketentuan yang tidak sesuai.

Contoh Data

```
{
  "status_code": 400,
  "message": "limit cannot be more than 2000",
  "data": null
}
```

5xx Server Error (Content-Type: text/plain)

Sistem akan mengembalikan pesan *error* bila terjadi kesalahan pada sisi server saat memproses data yang telah dikirimkan.

Contoh Data

Gateway Timeout

1.3. Contoh Penggunaan/Kode



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

cURL (Windows)

```
curl --location ^
```

```
--request GET ^  
"https://api-stg.dto.kemkes.go.id/masterdata/v1/provinces?codes"
```

cURL (Linux)

```
curl --location \  
--request GET \  
'https://api-stg.dto.kemkes.go.id/masterdata/v1/provinces?codes'
```

Postman

1. Buat *request* baru menggunakan **New** > **HTTP Request**, atau klik  untuk buat tab *request* baru.
2. Masukkan *request* URL

```
https://api-stg.dto.kemkes.go.id/masterdata/v1/provinces?codes
```

3. Lalu pilih *request method* **GET**.
4. Pada tab **Auth**:
 - a. pada pilihan **Type**, pilih **Bearer Token**,
 - b. lalu masukan nilai akses token yang sudah didapatkan pada saat autentikasi pada kotak inputan **Token**.
5. Pada tab **Headers**:
 - a. Pada kolom **KEY** isi **Accept**, lalu pada kolom **VALUE** isi dengan **application/json**.
6. Pada tab **Params**, di bagian **Query Params**:
 - a. silakan masukan parameter untuk melakukan pencarian sesuai dengan yang sudah dijelaskan pada bagian *query string* terkait ReST API Master Sarana Indeks (MSI) ini,
 - b. masukan satu atau lebih nama dari parameter tersebut pada kotak masukan pada kolom **KEY**,
 - c. sedangkan untuk nilainya, masukan pada kotak masukan pada kolom **VALUE**.
7. Klik tombol .
8. Hasil *response* akan ditampilkan di bagian **Response**.

2. Master Wilayah - Kota/Kabupaten



Setiap terdapat simbol asterik ***** sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: ***variabel**.

2.1. Request

URL

```
https://api-stg.dto.kemkes.go.id/masterdata/v1/cities?province_codes
```

HTTP Verb/Method

GET

Header

Nama Parameter	Tipe Data	Keterangan
*Authorization	string	Header ini WAJIB diisi dengan nilai sesuai format: Bearer <access_token> . Nilai dari variabel <access_token> didapatkan dari properti access_token pada object dari hasil response JSON setelah proses autentikasi.

Query String

Nama Parameter	Tipe Data	Keterangan
*province_codes	number	Isi dengan Kode provinsi berdasarkan Kode Kemendagri. (dapat di- <i>input</i> multi dengan menyertakan koma) Contoh: 11, 12.
codes	number	Isi dengan Kode kabupaten/kota berdasarkan Kode Kemendagri. (dapat di- <i>input</i> multi dengan menyertakan koma) Contoh: 1103,1104, 1210.
search	text	Isi dengan nama provinsi yang ingin dicari

2.2. Response

Hasil *response*, dengan HTTP *Status Code* berpola **2xx** atau **4xx**, yang dikembalikan dari server mempunyai parameter **Content-Type** dengan nilai **application/json** di salah satu parameter *header*-nya.

2xx Success

Contoh Data



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

```
{
  "status": 200,
  "error": false,
  "message": "success",
  "data": [
    {
      "code": "1103",
      "parent_code": "11",
      "bps_code": "1105",
      "name": "Kab. Aceh Timur"
    },
    /* lompat beberapa data */
    {
      "code": "1210",
      "parent_code": "12",
      "bps_code": "1207",
      "name": "Kab. Labuhanbatu"
    }
  ]
}
```

4xx Client Error

Sistem akan mengembalikan pesan *error* bila *client* belum melakukan autentikasi, tidak memiliki akses, menggunakan HTTP *method* yang tidak tepat, atau mengirimkan data dengan format atau ketentuan yang tidak sesuai.

Contoh Data

```
{
  "status_code": 400,
  "message": "limit cannot be more than 2000",
  "data": null
}
```

5xx Server Error (Content-Type: text/plain)

Sistem akan mengembalikan pesan *error* bila terjadi kesalahan pada sisi server saat memproses data yang telah dikirimkan.

Contoh Data

Gateway Timeout

2.3. Contoh Penggunaan/Kode



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan

contoh saja, tidak untuk dipakai.

cURL (Windows)

```
curl --location ^  
--request GET ^  
"https://api-stg.dto.kemkes.go.id/masterdata/v1/cities?province_codes"
```

cURL (Linux)

```
curl --location \  
--request GET \  
'https://api-stg.dto.kemkes.go.id/masterdata/v1/cities?province_codes'
```

Postman

1. Buat *request* baru menggunakan **New > HTTP Request**, atau klik  untuk buat tab *request* baru.
2. Masukkan *request* URL

```
https://api-stg.dto.kemkes.go.id/masterdata/v1/cities?province_codes
```

3. Lalu pilih *request method* **GET**.
4. Pada tab **Auth**:
 - a. pada pilihan **Type**, pilih **Bearer Token**,
 - b. lalu masukan nilai akses token yang sudah didapatkan pada saat autentikasi pada kotak inputan **Token**.
5. Pada tab **Headers**:
 - a. Pada kolom **KEY** isi **Accept**, lalu pada kolom **VALUE** isi dengan **application/json**.
6. Pada tab **Params**, di bagian **Query Params**:
 - a. silakan masukan parameter untuk melakukan pencarian sesuai dengan yang sudah dijelaskan pada bagian *query string* terkait ReST API Master Sarana Indeks (MSI) ini,
 - b. masukan satu atau lebih nama dari parameter tersebut pada kotak masukan pada kolom **KEY**,
 - c. sedangkan untuk nilainya, masukan pada kotak masukan pada kolom **VALUE**.
7. Klik tombol .
8. Hasil *response* akan ditampilkan di bagian **Response**.

3. Master Wilayah - Kecamatan



Setiap terdapat simbol asterik ***** sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: ***variabel**.

3.1. Request

URL

```
https://api-stg.dto.kemkes.go.id/masterdata/v1/districts?city_codes
```

HTTP Verb/Method

```
GET
```

Header

Nama Parameter	Tipe Data	Keterangan
*Authorization	string	Header ini WAJIB diisi dengan nilai sesuai format: Bearer <access_token> . Nilai dari variabel <access_token> didapatkan dari properti access_token pada object dari hasil response JSON setelah proses autentikasi.

Query String

Nama Parameter	Tipe Data	Keterangan
*city_codes	number	Isi dengan Kode kabupaten/kota berdasarkan Kode Kemendagri. (dapat di- <i>input</i> multi dengan menyertakan koma) Contoh: 1103, 1104 .
codes	number	Isi dengan Kode kecamatan berdasarkan Kode Kemendagri. (dapat di- <i>input</i> multi dengan menyertakan koma) Contoh: 110302, 110301 .
search	text	Isi dengan nama provinsi yang ingin dicari

3.2. Response

Hasil *response*, dengan HTTP *Status Code* berpola **2xx** atau **4xx**, yang dikembalikan dari server mempunyai parameter **Content-Type** dengan nilai **application/json** di salah satu parameter *header*-nya.

2xx Success

Contoh Data



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

```
{
  "status": 200,
  "error": false,
  "message": "success",
  "data": [
    {
      "code": "110301",
      "parent_code": "1103",
      "bps_code": "1105140",
      "name": "Darul Aman"
    },
    /* lompat beberapa data */
    {
      "code": "110421",
      "parent_code": "1104",
      "bps_code": "1106063",
      "name": "Rusip Antara"
    }
  ]
}
```

4xx Client Error

Sistem akan mengembalikan pesan *error* bila *client* belum melakukan autentikasi, tidak memiliki akses, menggunakan HTTP *method* yang tidak tepat, atau mengirimkan data dengan format atau ketentuan yang tidak sesuai.

Contoh Data

```
{
  "status_code": 400,
  "message": "limit cannot be more than 2000",
  "data": null
}
```

5xx Server Error (Content-Type: text/plain)

Sistem akan mengembalikan pesan *error* bila terjadi kesalahan pada sisi server saat memproses data yang telah dikirimkan.

Contoh Data

Gateway Timeout

3.3. Contoh Penggunaan/Kode



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan

contoh saja, tidak untuk dipakai.

cURL (Windows)

```
curl --location ^  
--request GET ^  
"https://api-stg.dto.kemkes.go.id/masterdata/v1/districts?city_codes"
```

cURL (Linux)

```
curl --location \  
--request GET \  
'https://api-stg.dto.kemkes.go.id/masterdata/v1/districts?city_codes'
```

Postman

1. Buat *request* baru menggunakan **New > HTTP Request**, atau klik  untuk buat tab *request* baru.
2. Masukkan *request* URL

```
https://api-stg.dto.kemkes.go.id/masterdata/v1/districts?city_codes
```

3. Lalu pilih *request method* **GET**.
4. Pada tab **Auth**:
 - a. pada pilihan **Type**, pilih **Bearer Token**,
 - b. lalu masukan nilai akses token yang sudah didapatkan pada saat autentikasi pada kotak inputan **Token**.
5. Pada tab **Headers**:
 - a. Pada kolom **KEY** isi **Accept**, lalu pada kolom **VALUE** isi dengan **application/json**.
6. Pada tab **Params**, di bagian **Query Params**:
 - a. silakan masukan parameter untuk melakukan pencarian sesuai dengan yang sudah dijelaskan pada bagian *query string* terkait ReST API Master Sarana Indeks (MSI) ini,
 - b. masukan satu atau lebih nama dari parameter tersebut pada kotak masukan pada kolom **KEY**,
 - c. sedangkan untuk nilainya, masukan pada kotak masukan pada kolom **VALUE**.
7. Klik tombol .
8. Hasil *response* akan ditampilkan di bagian **Response**.

4. Master Wilayah - Kelurahan/Desa



Setiap terdapat simbol asterik ***** sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: ***variabel**.

4.1. Request

URL

```
https://api-stg.dto.kemkes.go.id/masterdata/v1/sub-districts?district_codes
```

HTTP Verb/Method

GET

Header

Nama Parameter	Tipe Data	Keterangan
*Authorization	string	Header ini WAJIB diisi dengan nilai sesuai format: Bearer <access_token> . Nilai dari variabel <access_token> didapatkan dari properti access_token pada object dari hasil response JSON setelah proses autentikasi.

Query String

Nama Parameter	Tipe Data	Keterangan
*district_codes	number	Isi dengan Kode kecamatan berdasarkan Kode Kemendagri. (dapat di- <i>input</i> multi dengan menyertakan koma) Contoh: 110301, 110302.
codes	number	Isi dengan Kode kelurahan/desa berdasarkan Kode Kemendagri. (dapat di- <i>input</i> multi dengan menyertakan koma) Contoh: 1103012002, 1103012004 ,1103022004.
search	text	Isi dengan nama provinsi yang ingin dicari

4.2. Response

Hasil *response*, dengan HTTP *Status Code* berpola **2xx** atau **4xx**, yang dikembalikan dari server mempunyai parameter **Content-Type** dengan nilai **application/json** di salah satu parameter *header*-nya.

2xx Success

Contoh Data



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang

tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

```
{
  "status": 200,
  "error": false,
  "message": "success",
  "data": [
    {
      "code": "1103012002",
      "parent_code": "110301",
      "bps_code": "1105140007",
      "name": "Alue Luddin Dua"
    },
    /* lompat beberapa data */
    {
      "code": "1103022035",
      "parent_code": "110302",
      "bps_code": "1105160013",
      "name": "Buket Makmu"
    }
  ]
}
```

4xx Client Error

Sistem akan mengembalikan pesan *error* bila *client* belum melakukan autentikasi, tidak memiliki akses, menggunakan HTTP *method* yang tidak tepat, atau mengirimkan data dengan format atau ketentuan yang tidak sesuai.

Contoh Data

```
{
  "status_code": 400,
  "message": "limit cannot be more than 2000",
  "data": null
}
```

5xx Server Error (Content-Type: text/plain)

Sistem akan mengembalikan pesan *error* bila terjadi kesalahan pada sisi server saat memproses data yang telah dikirimkan.

Contoh Data

Gateway Timeout

4.3. Contoh Penggunaan/Kode



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

cURL (Windows)

```
curl --location ^  
--request GET ^  
"https://api-stg.dto.kemkes.go.id/masterdata/v1/sub-districts?district_codes"
```

cURL (Linux)

```
curl --location \  
--request GET \  
'https://api-stg.dto.kemkes.go.id/masterdata/v1/sub-districts?district_codes'
```

Postman

1. Buat *request* baru menggunakan **New > HTTP Request**, atau klik  untuk buat tab *request* baru.
2. Masukkan *request* URL

```
https://api-stg.dto.kemkes.go.id/masterdata/v1/sub-districts?district_codes
```

3. Lalu pilih *request method* **GET**.
4. Pada tab **Auth**:
 - a. pada pilihan **Type**, pilih **Bearer Token**,
 - b. lalu masukan nilai akses token yang sudah didapatkan pada saat autentikasi pada kotak inputan **Token**.
5. Pada tab **Headers**:
 - a. Pada kolom **KEY** isi **Accept**, lalu pada kolom **VALUE** isi dengan **application/json**.
6. Pada tab **Params**, di bagian **Query Params**:
 - a. silakan masukan parameter untuk melakukan pencarian sesuai dengan yang sudah dijelaskan pada bagian *query string* terkait ReST API Master Sarana Indeks (MSI) ini,
 - b. masukan satu atau lebih nama dari parameter tersebut pada kotak masukan pada kolom **KEY**,
 - c. sedangkan untuk nilainya, masukan pada kotak masukan pada kolom **VALUE**.
7. Klik tombol .
8. Hasil *response* akan ditampilkan di bagian **Response**.

D. MASTER WILAYAH VERSI 2

Untuk melakukan transaksi data dari Master Wilayah, perlu dilakukan proses autentikasi terlebih dahulu agar mendapatkan akses yang tersedia. Autentikasi yang digunakan oleh Master Wilayah mengikuti standar protokol OAuth 2 dengan tipe pemberian akses (*grant type*) adalah `client_credentials`.

Autentikasi menggunakan *grant type* `client_credentials` adalah proses autentikasi yang dilakukan antara *server to server*, sehingga tidak ada proses registrasi atau *log in* di sini. Autentikasi dengan tipe tersebut hanya memerlukan data berupa `client_id` dan `client_secret`, dimana nilai tersebut didapatkan ketika pihak yang ingin menggunakan atau mengakses Wilayah ini telah melakukan pengajuan, terdaftar, serta mendapatkan persetujuan dari **Pusat Data dan Teknologi Informasi - Digital Transformation Office (DTO) Kementerian Kesehatan Republik Indonesia**.

Cara Mendapatkan Nilai dari `client_id` dan `client_secret`

Silakan terlebih dahulu melakukan **Pengajuan dan Verifikasi Akses** pada website <https://satusehat.kemkes.go.id/platform> melalui web *browser* Anda. Untuk informasi lebih lanjut dapat dilihat pada **Pengajuan dan Verifikasi Akses**



Setiap teks yang berwarna **biru muda**, dapat diklik untuk melompat ke bagian yang direferensikan.

Pada bagian ini akan dijelaskan spesifikasi untuk **ReST API Master Wilayah**, yang mempunyai *endpoint* berdasarkan jenis lingkungan pengembangannya (*development environment*) yaitu:

API Master Wilayah Versi 2

- *staging* <https://api-stg.dto.kemkes.go.id/masterdata/v2>
- *production* <https://api.kemkes.go.id/masterdata/v2>



Semua penerapan, penjelasan, dan contoh yang akan dibahas akan menggunakan *endpoint production*.



Versi 2 sudah menerapkan *pagination* dan data bisa di filter tanpa menggunakan kode Kemendagri

1. Master Wilayah - Provinsi



Setiap terdapat simbol asterik * sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: ***variabel**.

1.1. Request

URL

`https://api-stg.dto.kemkes.go.id/masterdata/v2/provinces?current_page`

HTTP Verb/Method

GET

Header

Nama Parameter	Tipe Data	Keterangan
*Authorization	string	Header ini WAJIB diisi dengan nilai sesuai format: Bearer <access_token> . Nilai dari variabel <access_token> didapatkan dari properti access_token pada object dari hasil response JSON setelah proses autentikasi.

Query String

Nama Parameter	Tipe Data	Keterangan
codes	number	Isi dengan Kode provinsi berdasarkan Kode Kemendagri. (dapat di-input multi dengan menyertakan koma) Contoh: 11.
search	text	Isi dengan nama provinsi yang ingin dicari
*current_page	number	Isi dengan nilai sesuai respon. Dapat dilihat pada response di dalam objek meta. Contoh: 1.
next	encrypted (base64)	Isi dengan halaman selanjutnya. Dapat dilihat pada response di dalam objek meta.
prev	encrypted (base64)	Isi dengan halaman sebelumnya. Dapat dilihat pada response di dalam objek meta.
page_size	number	Isi dengan jumlah baris yang ingin ditampilkan (Maksimal 10 baris).



Parameter **next** dan **prev** hanya dapat digunakan satu filter. Tidak dapat digunakan secara bersamaan.

1.2. Response

Hasil *response*, dengan HTTP *Status Code* berpola **2xx** atau **4xx**, yang dikembalikan dari server mempunyai parameter **Content-Type** dengan nilai **application/json** di salah satu parameter *header*-nya.

2xx Success

Contoh Data



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

```
{
  "status": 200,
  "error": false,
  "message": "success",
  "data": [
    {
      "code": "11",
      "parent_code": "",
      "bps_code": "11",
      "name": "Aceh"
    },
    {
      "code": "12",
      "parent_code": "",
      "bps_code": "12",
      "name": "Sumatera Utara"
    }
  ],
  "meta": {
    "item_count": 2,
    "page": {
      "is_cursor": true,
      "current": 1,
      "previous": 0,
      "next": 2,
      "limit": 10,
      "total_page": 1,
      "total": 2
    },
    "cursors": {
      "next": "MTI=",
      "previous": "MTE="
    }
  }
}
```

4xx Client Error

Sistem akan mengembalikan pesan *error* bila *client* belum melakukan autentikasi, tidak memiliki akses, menggunakan HTTP *method* yang tidak tepat, atau mengirimkan data dengan format atau ketentuan yang tidak sesuai.

Contoh Data

```
{  
  "status_code": 400,  
  "message": "limit cannot be more than 2000",  
  "data": null  
}
```

5xx Server Error (Content-Type: text/plain)

Sistem akan mengembalikan pesan *error* bila terjadi kesalahan pada sisi server saat memproses data yang telah dikirimkan.

Contoh Data

Gateway Timeout

1.3. Contoh Penggunaan/Kode



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

cURL (Windows)

```
curl --location ^  
--request GET ^  
"https://api-stg.dto.kemkes.go.id/masterdata/v2/provinces?current_page"
```

cURL (Linux)

```
curl --location \  
--request GET \  
'https://api-stg.dto.kemkes.go.id/masterdata/v2/provinces?current_page'
```

Postman

1. Buat *request* baru menggunakan **New > HTTP Request**, atau klik  untuk buat tab *request* baru.
2. Masukkan *request* URL

```
https://api-stg.dto.kemkes.go.id/masterdata/v2/provinces?current_page
```

3. Lalu pilih *request method* **GET**.
4. Pada tab **Auth**:
 - a. pada pilihan **Type**, pilih **Bearer Token**,
 - b. lalu masukan nilai akses token yang sudah didapatkan pada saat autentikasi pada kotak inputan **Token**.
5. Pada tab **Headers**:
 - a. Pada kolom **KEY** isi **Accept**, lalu pada kolom **VALUE** isi dengan **application/json**.

6. Pada tab **Params**, di bagian **Query Params**:

- silakan masukan parameter untuk melakukan pencarian sesuai dengan yang sudah dijelaskan pada bagian *query string* terkait ReST API Master Sarana Indeks (MSI) ini,
- masukan satu atau lebih nama dari parameter tersebut pada kotak masukan pada kolom **KEY**,
- sedangkan untuk nilainya, masukan pada kotak masukan pada kolom **VALUE**.

7. Klik tombol **Send**.

8. Hasil *response* akan ditampilkan di bagian **Response**.

2. Master Wilayah - Kota/Kabupaten



Setiap terdapat simbol asterik ***** sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: ***variabel**.

2.1. Request

URL

`https://api-stg.dto.kemkes.go.id/masterdata/v2/cities?current_page`

HTTP Verb/Method

GET

Header

Nama Parameter	Tipe Data	Keterangan
*Authorization	string	Header ini WAJIB diisi dengan nilai sesuai format: Bearer <access_token> . Nilai dari variabel <access_token> didapatkan dari properti access_token pada object dari hasil response JSON setelah proses autentikasi.

Query String

Nama Parameter	Tipe Data	Keterangan
province_codes	number	Isi dengan Kode provinsi berdasarkan Kode Kemendagri. (dapat di-input multi dengan menyertakan koma) Contoh: 11 .
codes	number	Isi dengan Kode provinsi berdasarkan Kode Kemendagri. (dapat di-input multi dengan menyertakan koma) Contoh: 1103,1104, 1210 .
search	text	Isi dengan nama provinsi yang ingin dicari
*current_page	number	Isi dengan nilai sesuai respon. Dapat dilihat pada response di dalam objek meta. Contoh: 1 .
next	encrypted (base64)	Isi dengan halaman selanjutnya. Dapat dilihat pada response di dalam objek meta
prev	encrypted (base64)	Isi dengan halaman sebelumnya. Dapat dilihat pada response di dalam objek meta

Nama Parameter	Tipe Data	Keterangan
<code>page_size</code>	<code>number</code>	Isi dengan jumlah baris yang ingin ditampilkan (Maksimal 10 baris).



Parameter `next` dan `prev` hanya dapat digunakan satu filter. Tidak dapat digunakan secara bersamaan.

2.2. Response

Hasil *response*, dengan HTTP *Status Code* berpola `2xx` atau `4xx`, yang dikembalikan dari server mempunyai parameter *Content-Type* dengan nilai `application/json` di salah satu parameter *header*-nya.

2xx Success

Contoh Data



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

```
{
  "status": 200,
  "error": false,
  "message": "success",
  "data": [
    {
      "code": "1105",
      "parent_code": "11",
      "bps_code": "1107",
      "name": "Kab. Aceh Barat"
    },
    /* lompat beberapa data */
    {
      "code": "1109",
      "parent_code": "11",
      "bps_code": "1101",
      "name": "Kab. Simeulue"
    }
  ],
  "meta": {
    "item_count": 10,
    "page": {
      "is_cursor": true,
      "current": 1,
      "previous": 0,
      "next": 2,
      "limit": 10,
      "total_page": 51,
      "total": 514
    },
    "cursors": {
      "next": "MTExMA==",

```

```
"previous": "MTEwMQ=="
}
}
}
```

4xx Client Error

Sistem akan mengembalikan pesan *error* bila *client* belum melakukan autentikasi, tidak memiliki akses, menggunakan HTTP *method* yang tidak tepat, atau mengirimkan data dengan format atau ketentuan yang tidak sesuai.

Contoh Data

```
{
  "status_code": 400,
  "message": "limit cannot be more than 2000",
  "data": null
}
```

5xx Server Error (Content-Type: text/plain)

Sistem akan mengembalikan pesan *error* bila terjadi kesalahan pada sisi server saat memproses data yang telah dikirimkan.

Contoh Data

Gateway Timeout

2.3. Contoh Penggunaan/Kode



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

cURL (Windows)

```
curl --location ^
--request GET ^
"https://api-stg.dto.kemkes.go.id/masterdata/v2/cities?current_page"
```

cURL (Linux)

```
curl --location \
--request GET \
'https://api-stg.dto.kemkes.go.id/masterdata/v2/cities?current_page'
```

Postman

1. Buat *request* baru menggunakan **New > HTTP Request**, atau klik  untuk buat tab *request* baru.
2. Masukkan *request* URL

https://api-stg.dto.kemkes.go.id/masterdata/v2/cities?current_page

3. Lalu pilih *request method* **GET**.
4. Pada tab **Auth**:
 - a. pada pilihan **Type**, pilih **Bearer Token**,
 - b. lalu masukan nilai akses token yang sudah didapatkan pada saat autentikasi pada kotak inputan **Token**.
5. Pada tab **Headers**:
 - a. Pada kolom **KEY** isi **Accept**, lalu pada kolom **VALUE** isi dengan **application/json**.
6. Pada tab **Params**, di bagian **Query Params**:
 - a. silakan masukan parameter untuk melakukan pencarian sesuai dengan yang sudah dijelaskan pada bagian *query string* terkait ReST API Master Sarana Indeks (MSI) ini,
 - b. masukan satu atau lebih nama dari parameter tersebut pada kotak masukan pada kolom **KEY**,
 - c. sedangkan untuk nilainya, masukan pada kotak masukan pada kolom **VALUE**.
7. Klik tombol **Send**.
8. Hasil *response* akan ditampilkan di bagian **Response**.

3. Master Wilayah - Kecamatan



Setiap terdapat simbol asterik ***** sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: ***variabel**.

3.1. Request

URL

`https://api-stg.dto.kemkes.go.id/masterdata/v2/districts?current_page`

HTTP Verb/Method

GET

Header

Nama Parameter	Tipe Data	Keterangan
*Authorization	string	Header ini WAJIB diisi dengan nilai sesuai format: Bearer <access_token> . Nilai dari variabel <access_token> didapatkan dari properti access_token pada object dari hasil response JSON setelah proses autentikasi.

Query String

Nama Parameter	Tipe Data	Keterangan
city_codes	number	Isi dengan Kode kabupaten/kota berdasarkan Kode Kemendagri. (dapat di- <i>input</i> multi dengan menyertakan koma) Contoh: 1103, 1104.
codes	number	Isi dengan Kode kecamatan berdasarkan Kode Kemendagri. (dapat di- <i>input</i> multi dengan menyertakan koma) Contoh: 110302, 110301.
search	text	Isi dengan nama provinsi yang ingin dicari
*current_page	number	Isi dengan halaman yang dituju. Contoh: 110302, 110301.
next	encrypted (base64)	Isi dengan halaman selanjutnya. Dapat dilihat pada response di dalam objek meta
prev	encrypted (base64)	Isi dengan halaman sebelumnya. Dapat dilihat pada response di dalam objek meta
page_size	number	Isi dengan jumlah baris yang ingin ditampilkan (Maksimal 10 baris).



Parameter **next** dan **prev** hanya dapat digunakan satu filter. Tidak dapat digunakan secara bersamaan.

3.2. Response

Hasil *response*, dengan HTTP *Status Code* berpola **2xx** atau **4xx**, yang dikembalikan dari server mempunyai parameter **Content-Type** dengan nilai **application/json** di salah satu parameter *header*-nya.

2xx Success

Contoh Data



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

```
{
  "status": 200,
  "error": false,
  "message": "success",
  "data": [
    {
      "code": "110101",
      "parent_code": "1101",
      "bps_code": "1103020",
      "name": "Bakongan"
    },
    /* lompat beberapa data */
    {
      "code": "110109",
      "parent_code": "1101",
      "bps_code": "1103010",
      "name": "Trumon"
    }
  ],
  "meta": {
    "item_count": 10,
    "page": {
      "is_cursor": true,
      "current": 1,
      "previous": 0,
      "next": 2,
      "limit": 10,
      "total_page": 727,
      "total": 7277
    },
    "cursors": {
      "next": "MTEwMTEw",
      "previous": "MTEwMTAx"
    }
  }
}
```

4xx Client Error

Sistem akan mengembalikan pesan *error* bila *client* belum melakukan autentikasi, tidak memiliki akses, menggunakan HTTP *method* yang tidak tepat, atau mengirimkan data dengan format atau ketentuan yang tidak sesuai.

Contoh Data

```
{
  "status_code": 400,
  "message": "limit cannot be more than 2000",
  "data": null
}
```

5xx Server Error (Content-Type: text/plain)

Sistem akan mengembalikan pesan *error* bila terjadi kesalahan pada sisi server saat memproses data yang telah dikirimkan.

Contoh Data

Gateway Timeout

3.3. Contoh Penggunaan/Kode



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

cURL (Windows)

```
curl --location ^
--request GET ^
"https://api-stg.dto.kemkes.go.id/masterdata/v2/districts?current_page"
```

cURL (Linux)

```
curl --location \
--request GET \
'https://api-stg.dto.kemkes.go.id/masterdata/v2/districts?current_page'
```

Postman

1. Buat *request* baru menggunakan **New > HTTP Request**, atau klik  untuk buat tab *request* baru.
2. Masukkan *request* URL

`https://api-stg.dto.kemkes.go.id/masterdata/v2/districts?current_page`

3. Lalu pilih *request method* **GET**.
4. Pada tab **Auth**:

- a. pada pilihan **Type**, pilih **Bearer Token**,
 - b. lalu masukan nilai akses token yang sudah didapatkan pada saat autentikasi pada kotak inputan **Token**.
5. Pada tab **Headers**:
- a. Pada kolom **KEY** isi **Accept**, lalu pada kolom **VALUE** isi dengan **application/json**.
6. Pada tab **Params**, di bagian **Query Params**:
- a. silakan masukan parameter untuk melakukan pencarian sesuai dengan yang sudah dijelaskan pada bagian *query string* terkait ReST API Master Sarana Indeks (MSI) ini,
 - b. masukan satu atau lebih nama dari parameter tersebut pada kotak masukan pada kolom **KEY**,
 - c. sedangkan untuk nilainya, masukan pada kotak masukan pada kolom **VALUE**.
7. Klik tombol **Send**.
8. Hasil *response* akan ditampilkan di bagian **Response**.

4. Master Wilayah - Kelurahan/Desa



Setiap terdapat simbol asterik ***** sebelum nama variabel atau parameter yang disebutkan, maka variabel atau parameter tersebut bersifat **WAJIB**, **harus ada**, atau **pasti selalu ada**, contoh: ***variabel**.

4.1. Request

URL

`https://api-stg.dto.kemkes.go.id/masterdata/v2/sub-districts?current_page`

HTTP Verb/Method

GET

Header

Nama Parameter	Tipe Data	Keterangan
*Authorization	string	Header ini WAJIB diisi dengan nilai sesuai format: Bearer <access_token> . Nilai dari variabel <access_token> didapatkan dari properti access_token pada object dari hasil response JSON setelah proses autentikasi.

Query String

Nama Parameter	Tipe Data	Keterangan
district_codes	number	Isi dengan Kode kecamatan berdasarkan Kode Kemendagri (dapat di- <i>input</i> multi dengan menyertakan koma) Contoh: { <i>qs-coddistrictses</i> }.
codes	number	Isi dengan Kode kelurahan/desa berdasarkan Kode Kemendagri (dapat di- <i>input</i> multi dengan menyertakan koma) Contoh: 1103012002, 1103012004 ,1103022004.
search	text	Isi dengan nama provinsi yang ingin dicari
*current_page	number	Isi dengan halaman yang dituju. Contoh: 1.
next	number	Isi dengan halaman selanjutnya. Dapat dilihat pada response di dalam objek meta
prev	number	Isi dengan halaman sebelumnya. Dapat dilihat pada response di dalam objek meta
page_size	number	Isi dengan jumlah baris yang ingin ditampilkan (Maksimal 10 baris).



Parameter **next** dan **prev** hanya dapat digunakan satu filter. Tidak dapat digunakan secara bersamaan.

4.2. Response

Hasil *response*, dengan HTTP *Status Code* berpola **2xx** atau **4xx**, yang dikembalikan dari server mempunyai parameter **Content-Type** dengan nilai **application/json** di salah satu parameter *header*-nya.

2xx Success

Contoh Data



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

```
{
  "status": 200,
  "error": false,
  "message": "success",
  "data": [
    {
      "code": "1101012015",
      "parent_code": "110101",
      "bps_code": "1103020023",
      "name": "Darul Ikhsan"
    },
    /* lompat beberapa data */
    {
      "code": "1101012003",
      "parent_code": "110101",
      "bps_code": "1103020014",
      "name": "Ujong Padang"
    }
  ],
  "meta": {
    "item_count": 10,
    "page": {
      "is_cursor": true,
      "current": 1,
      "previous": 0,
      "next": 2,
      "limit": 10,
      "total_page": 8377,
      "total": 83772
    },
    "cursors": {
      "next": "MTEwMTAyMjAwMw==",
      "previous": "MTEwMTAxMjAwMQ=="
    }
  }
}
```

4xx Client Error

Sistem akan mengembalikan pesan *error* bila *client* belum melakukan autentikasi, tidak memiliki akses, menggunakan HTTP *method* yang tidak tepat, atau mengirimkan data dengan format atau ketentuan yang tidak sesuai.

Contoh Data

```
{  
  "status_code": 400,  
  "message": "limit cannot be more than 2000",  
  "data": null  
}
```

5xx Server Error (Content-Type: text/plain)

Sistem akan mengembalikan pesan *error* bila terjadi kesalahan pada sisi server saat memproses data yang telah dikirimkan.

Contoh Data

Gateway Timeout

4.3. Contoh Penggunaan/Kode



Setiap nilai yang dicontohkan atau ditampilkan di dokumentasi ini adalah nilai yang tidak sebenarnya dan tidak dapat dipakai. Nilai-nilai tersebut hanya untuk keperluan contoh saja, tidak untuk dipakai.

cURL (Windows)

```
curl --location ^  
--request GET ^  
"https://api-stg.dto.kemkes.go.id/masterdata/v2/sub-districts?current_page"
```

cURL (Linux)

```
curl --location \  
--request GET \  
'https://api-stg.dto.kemkes.go.id/masterdata/v2/sub-districts?current_page'
```

Postman

1. Buat *request* baru menggunakan **New > HTTP Request**, atau klik  untuk buat tab *request* baru.
2. Masukkan *request* URL

```
https://api-stg.dto.kemkes.go.id/masterdata/v2/sub-districts?current_page
```

3. Lalu pilih *request method* **GET**.
4. Pada tab **Auth**:

- a. pada pilihan **Type**, pilih **Bearer Token**,
 - b. lalu masukan nilai akses token yang sudah didapatkan pada saat autentikasi pada kotak inputan **Token**.
5. Pada tab **Headers**:
- a. Pada kolom **KEY** isi **Accept**, lalu pada kolom **VALUE** isi dengan **application/json**.
6. Pada tab **Params**, di bagian **Query Params**:
- a. silakan masukan parameter untuk melakukan pencarian sesuai dengan yang sudah dijelaskan pada bagian *query string* terkait ReST API Master Sarana Indeks (MSI) ini,
 - b. masukan satu atau lebih nama dari parameter tersebut pada kotak masukan pada kolom **KEY**,
 - c. sedangkan untuk nilainya, masukan pada kotak masukan pada kolom **VALUE**.
7. Klik tombol **Send**.
8. Hasil *response* akan ditampilkan di bagian **Response**.

DAFTAR ISTILAH

API

Singkatan dari *Application Programming Interface*, yaitu kumpulan definisi dan protokol terkait koneksi antara komputer atau aplikasi yang memungkinkan untuk melakukan pertukaran data.

JWT

Singkatan dari *JSON Web Token*, yaitu suatu standar untuk pertukaran data yang diperlukan dalam proses autentikasi.

MSI

Data Fasyankes atau Master Sarana Index (MSI), merupakan standar data 35 fasyankes di Indonesia. Disusun dari berbagai sumber, seperti Sistem Informasi Sumber Daya Manusia Kesehatan (SI-SDMK), Sistem Informasi Manajemen Data Kefarmasian (SIMADA), RS Online, dan lain sebagainya.

REFERENSI

- [restfulapi.net] "ReST API Tutorial". <https://restfulapi.net>.
- [wikipedia-rest] "Representational State Transfer". https://en.wikipedia.org/wiki/Representational_state_transfer.
- [wikipedia-http-method] "Hypertext Transfer Protocol, Request_methods". https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol#Request_methods
- [wikipedia-http-status-code] "List of HTTP status codes". https://en.wikipedia.org/wiki/List_of_HTTP_status_codes
- [wikipedia-iso8601] "ISO 8601". https://en.wikipedia.org/wiki/ISO_8601
- [wikipedia-api] "Application Programming Interface". <https://en.wikipedia.org/wiki/API>
- [wikipedia-whitespace] "Whitespace character". https://en.wikipedia.org/wiki/Whitespace_character
- [wikipedia-fhir] "Fast_Healthcare_Interoperability_Resources", <https://en.wikipedia.org/wiki/>
- [hl7-fhir] "Welcome to FHIR. FHIR is a standard for health care data exchange, published by HL7®". <https://www.hl7.org/fhir/index.html>
- [satusehat-fhir] "SATUSEHAT FHIR R4 Implementation Guide". <https://simplifier.net/guide/SATUSEHAT-FHIR-R4-Implementation-Guide/Home?version=current>
- [admonitions] "Admonitions". <https://docs.asciidoctor.org/asciidoc/latest/blocks/admonitions>
- [atc-code] "Daftar Kode ATC" https://id.wikipedia.org/wiki/Kategori:Kode_ATC
- [atc-level] "Daftar Level ATC" <https://www.who.int/tools/atc-ddd-toolkit/atc-classification>