Nama: Gusti Fahmi Fadhila

Kode Peserta: JVSB001ONL012

Link Heroku: <https://final-project-gustifahmi.herokuapp.com>

Link Github: <https://github.com/gustifahmi/FinalProjectSpringGusti>

**Panduan Final Project – Gusti Fahmi Fadhila**

**Daftar Isi**

A. Table & Columns 2

B. Database Migration 6  
C. Controller List 9

D. Authentication Instructions 11

E. API Details 12

1. User Controller 12

2. Auth Controller 15

3. Reservation Controller 16

4. Agency Controller 22

5. Bus Controller 28

6. Stop Controller 34

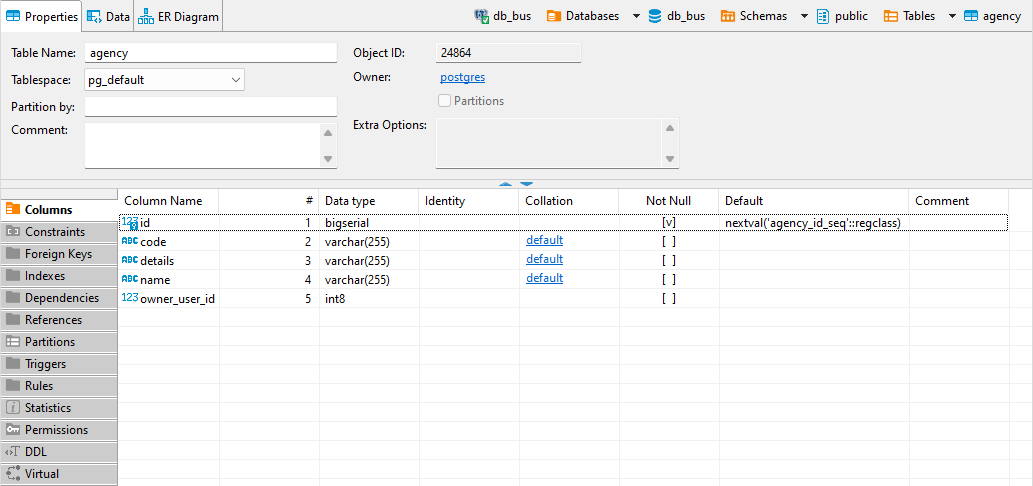
7. Ticket Controller 39

8. Trip Controller 42

9. Trip Schedule Controller 52

1. **Tables & Columns (Database menggunakan PostgreSQL, tipe data berdasarkan yang tertera di PostgreSQL)**
2. agency

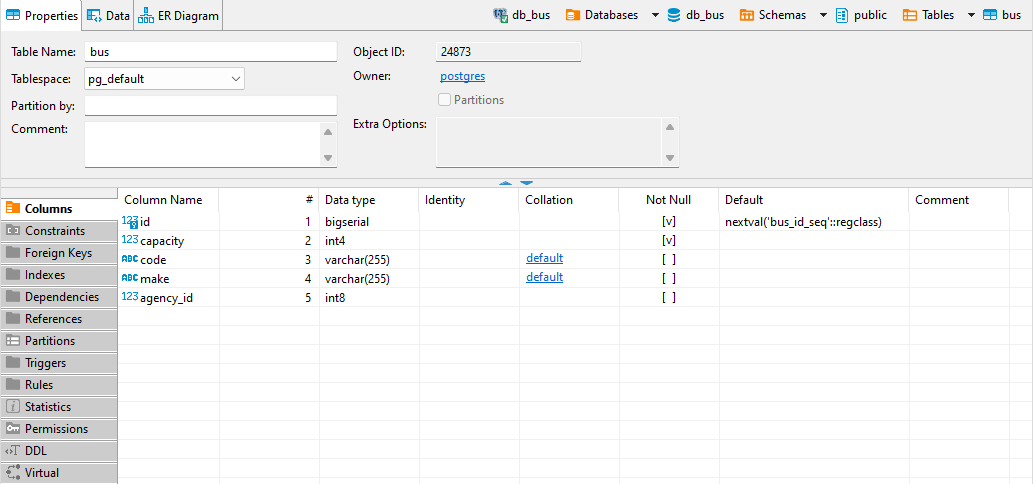
* id bigserial -> Primary Key
* code varchar(255)
* details varchar(255)
* name varchar(255)
* owner\_user\_id int8 -> Foreign Key ke table user\_info



Gambar 1. Screenshot DBeaver table agency properties

1. bus

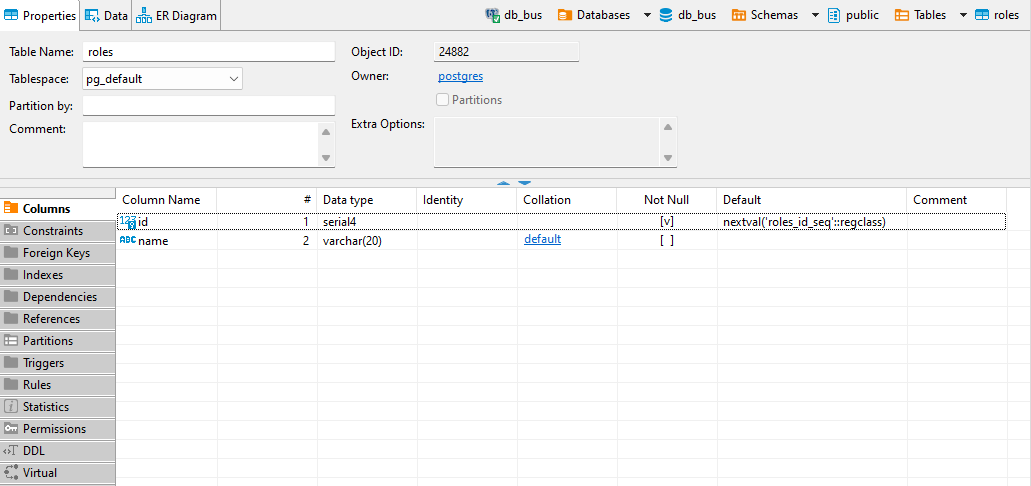
* id bigserial -> Primary Key
* capacity int4
* code varchar(255)
* make varchar(255)
* agency\_id int8 -> Foreign Key ke table agency



Gambar 2. Screenshot DBeaver table bus properties

1. roles

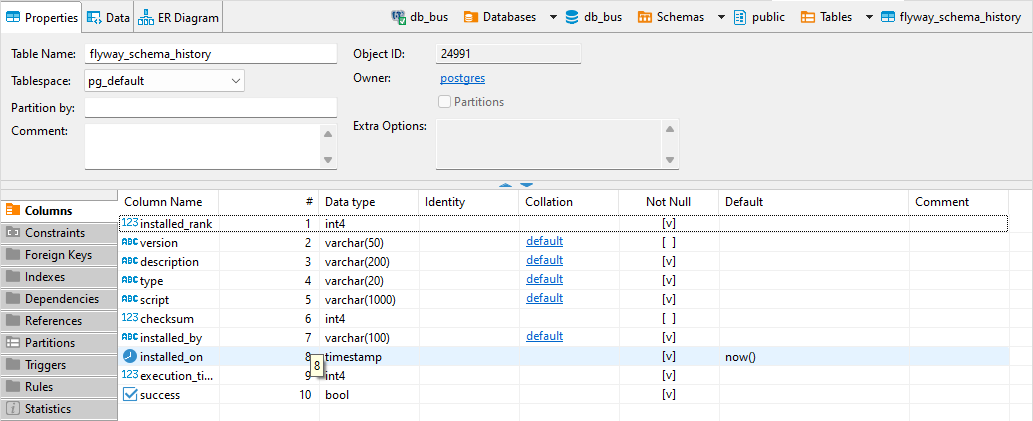
* id serial4 -> Primary Key
* name varchar(20)



Gambar 3. Screenshot DBeaver table roles properties

1. flyway\_schema\_history (ter-generate otomatis oleh flyway)

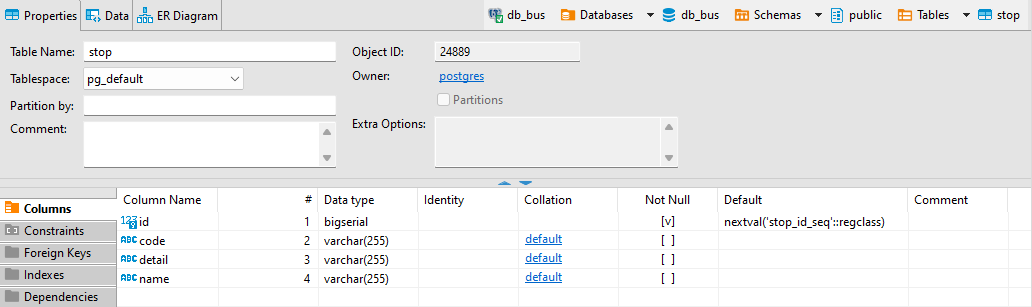
* installed\_rank int4
* version varchar(50)
* description varchar(200)
* type varchar(20)
* script varchar(1000)
* checksum int4
* installed\_by varchar(100)
* installed\_on timestamp
* execution\_time int4
* success bool



Gambar 4. Screenshot DBeaver table flyway\_schema\_history properties

1. stop

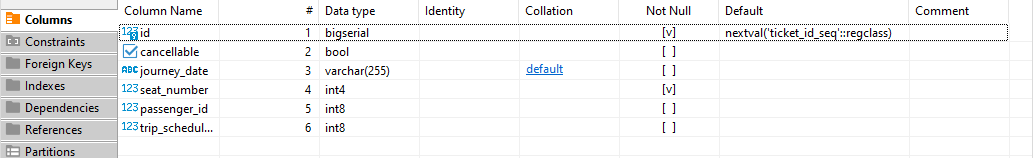
* id bigserial -> Primary Key
* code varchar(255)
* detail varchar(255)
* name varchar(255)



Gambar 5. Screenshot DBeaver table stop properties

1. ticket

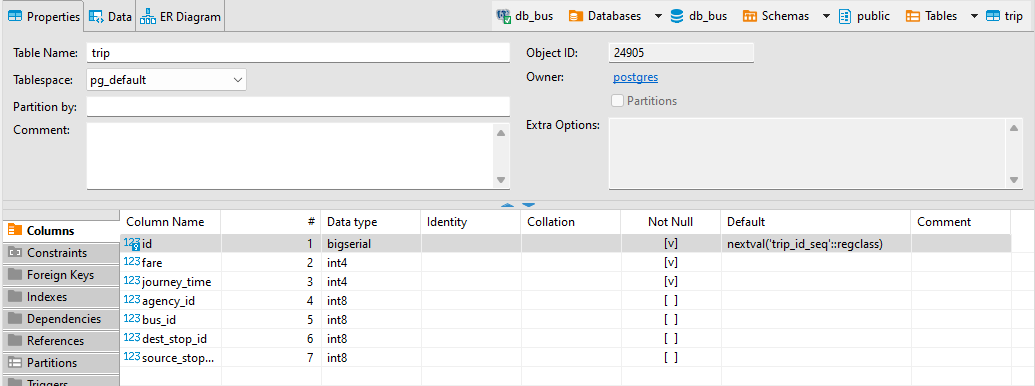
* id bigserial -> Primary Key
* cancellable bool
* journey\_date varchar(255)
* seat\_number int4
* passenger\_id int8 -> Foreign Key ke table user\_info
* trip\_schedule\_id int8 -> Foreign Key ke table trip\_schedule



Gambar 6. Screenshot DBeaver table ticket properties

1. trip

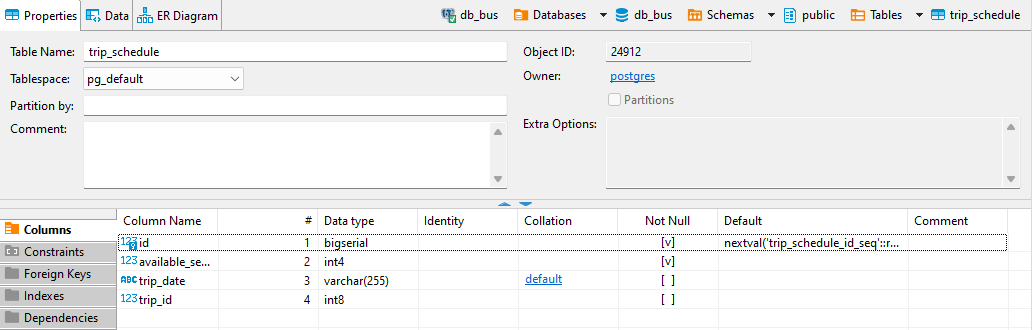
* id bigserial -> Primary Key
* fare int4
* journey\_time int4
* agency\_id int8 -> Foreign Key ke table agency
* bus\_id int8 -> Foreign Key ke table bus
* dest\_stop int8 -> Foreign Key ke table stop
* source\_stop int8 -> Foreign Key ke table stop



Gambar 7. Screenshot DBeaver table trip properties

1. trip\_schedule

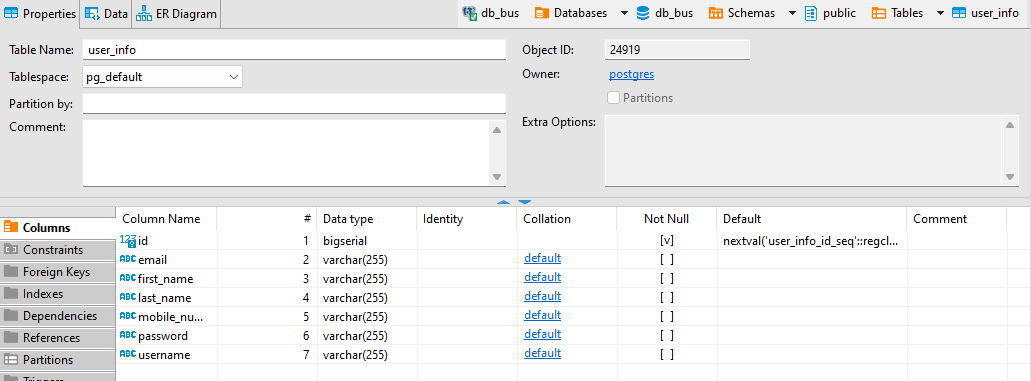
* id bigserial -> Primary Key
* available\_seats int4
* trip\_date varchar(255)
* trip\_id int8 -> Foreign Key ke table trip



Gambar 8. Screenshot DBeaver table trip\_schedule properties

1. user\_info (di PostgreSQL tidak bisa menggunakan nama table “user”)

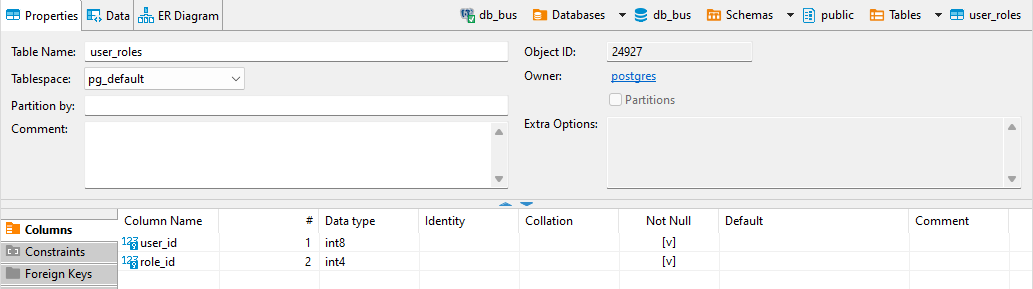
* id bigserial -> Primary Key
* email varchar(255)
* first\_name varchar(255)
* last\_name varchar(255)
* mobile\_number varchar(255)
* password varchar(255)
* username varchar(255)



Gambar 9. Screenshot DBeaver table user\_info properties

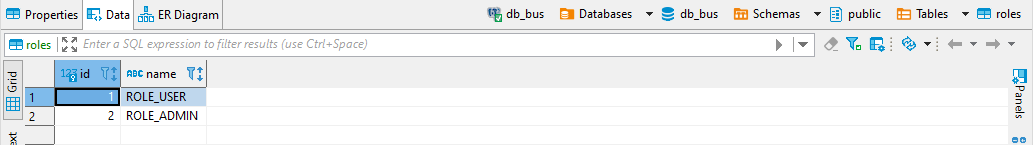
1. user\_roles

* user\_id -> Foreign Key ke table user\_info
* role\_id -> Foreign Key ke table roles

 Gambar 10. Screenshot DBeaver table user\_roles properties

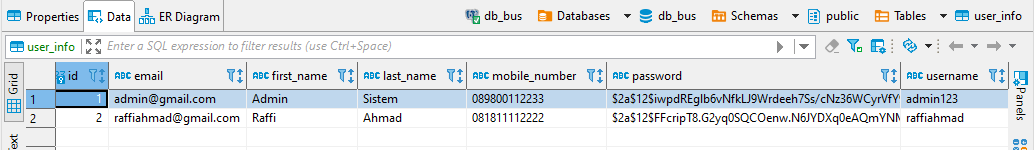
1. **Database Migration**
2. role

|  |  |
| --- | --- |
| **id** | **name** |
| 1 | ROLE\_USER |
| 2 | ROLE\_ADMIN |

 Gambar 11. Screenshot DBeaver table roles data

1. user\_info (passwordnya terenkripsi, tapi di table ini saya tampilkan raw-nya agar bisa dipakai untuk login)

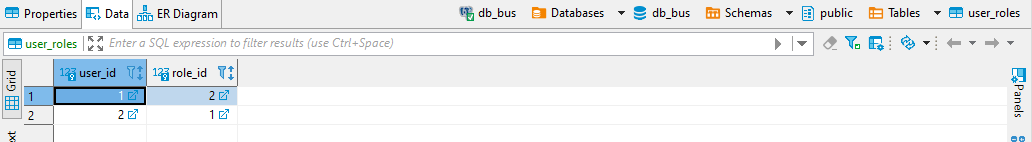
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **id** | **email** | **first\_name** | **last\_name** | **mobile\_number** | **password** | **username** |
| 1 | admin@gmail.com | Admin | Sistem | 089800112233 | admin123 | admin123 |
| 2 | raffiahmad@gmail.com | Raffi | Ahmad | 081811112222 | password | raffiahmad |



Gambar 12. Screenshot DBeaver table user\_info data

1. user\_roles

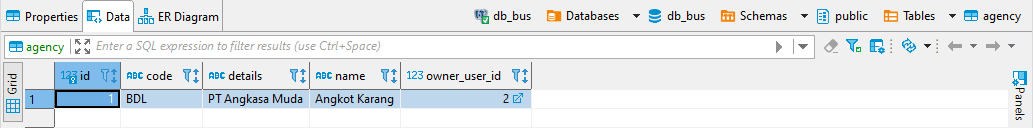
|  |  |
| --- | --- |
| **user\_id** | **role\_id** |
| 1 | 2 |
| 2 | 1 |



Gambar 13. Screenshot DBeaver table user\_roles data

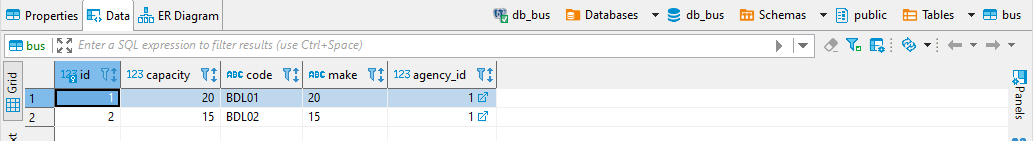
1. agency

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **code** | **details** | **name** | **owner\_user\_id** |
| 1 | BDL | PT Angkasa Muda | Angkot Karang | 2 |

 Gambar 14. Screenshot DBeaver table agency data

1. bus

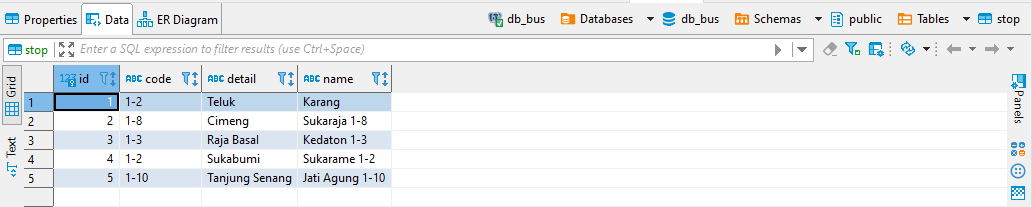
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **capacity** | **code** | **make** | **agency\_id** |
| 1 | 20 | BDL01 | 20 | 1 |
| 2 | 15 | BDL02 | 15 | 1 |



Gambar 15. Screenshot DBeaver table bus data

1. stop

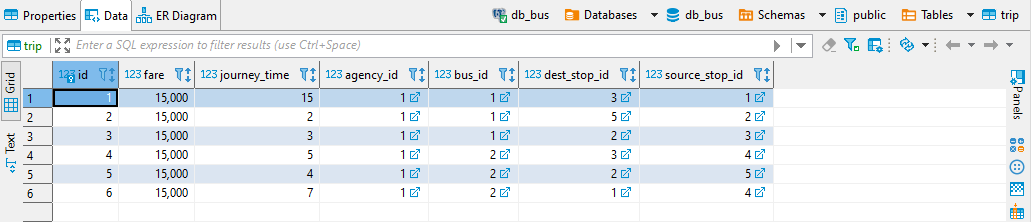
|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **code** | **detail** | **name** |
| 1 | 1-2 | Teluk | Karang |
| 2 | 1-8 | Cimeng | Sukaraja 1-8 |
| 3 | 1-3 | Raja BasaI | Kedaton 1-3 |
| 4 | 1-2 | Sukabumi | Sukarame 1-2 |
| 5 | 1-10 | Tanjung Senang | Jati Agung 1-10 |



Gambar 16. Screenshot DBeaver table stop data

1. trip

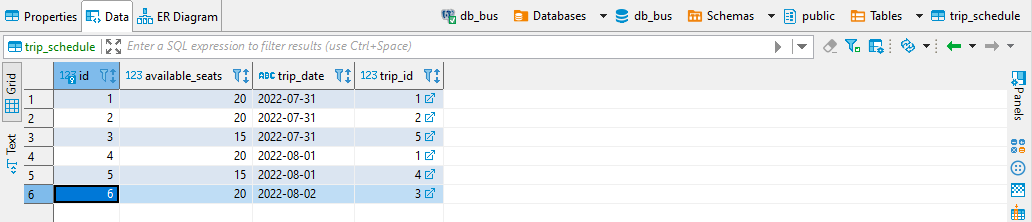
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **id** | **fare** | **journey\_time** | **agency\_id** | **bus\_id** | **dest\_stop\_id** | **source\_stop\_id** |
| 1 | 15000 | 15 | 1 | 1 | 3 | 1 |
| 2 | 15000 | 2 | 1 | 1 | 5 | 2 |
| 3 | 15000 | 3 | 1 | 1 | 2 | 3 |
| 4 | 15000 | 5 | 1 | 2 | 3 | 4 |
| 5 | 15000 | 4 | 1 | 2 | 2 | 5 |
| 6 | 15000 | 7 | 1 | 2 | 1 | 4 |



Gambar 17. Screenshot DBeaver table trip data

1. trip\_schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **available\_seats** | **trip\_date** | **trip\_id** |
| 1 | 20 | 2022-07-31 | 1 |
| 2 | 20 | 2022-07-31 | 2 |
| 3 | 15 | 2022-07-31 | 5 |
| 4 | 20 | 2022-08-01 | 1 |
| 5 | 15 | 2022-08-01 | 4 |
| 6 | 20 | 2022-08-02 | 3 |



Gambar 18. Screenshot DBeaver table trip\_schedule data

1. **Controller List**
2. User Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Endpoint** | **Method** | **Admin** | **User** |
| registerUser | /api/v1/user/signup | POST | No Login | No Login |
| updateUser | /api/v1/user/update | PUT | ✅ | ✅ |
| updatePassword | /api/v1/user/update/password | PUT | ✅ | ✅ |

Catatan: Untuk update user dan password tidak perlu menggunakan user id pada URL param karena otomatis didapatkan berdasarkan user yang login.

1. Auth Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Endpoint** | **Method** | **Admin** | **User** |
| authenticateUser | /api/auth | POST | No Login | No Login |

1. Reservation Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Endpoint** | **Method** | **Admin** | **User** |
| bookTicket | /api/v1/reservation | POST | ❌ | ✅ |
| getAvailableSeatNumberByTripScheduleId | /api/v1/reservation/  available-seat-number/{tripScheduleId} | GET | ❌ | ✅ |
| getAllOwnedTicketByLoggedInUser | /api/v1/reservation/passenger | GET | ❌ | ✅ |
| cancelTicket | /api/v1/reservation/{id} | DELETE | ❌ | ✅ |
| getOwnedTicketById | /api/v1/reservation/{id} | GET | ❌ | ✅ |

Catatan: Controller ini khusus untuk role user, jika suatu akun hanya punya role admin, maka perlu membuat akun user untuk mengakses API yang ada di controller ini.

1. Agency Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Endpoint** | **Method** | **Admin** | **User** |
| getAllAgency | /api/v1/agency | GET | ✅ | ✅ |
| createAgency | /api/v1/agency | POST | ✅ | ❌ |
| getAgencyByOwnerId | /api/v1/agency/owner/{ownerId} | GET | ✅ | ✅ |
| deleteAgency | /api/v1/agency/{id} | DELETE | ✅ | ❌ |
| getAgencyById | /api/v1/agency/{id} | GET | ✅ | ✅ |
| updateAgency | /api/v1/agency/{id} | PUT | ✅ | ❌ |

1. Bus Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Endpoint** | **Method** | **Admin** | **User** |
| getAllBus | /api/v1/bus | GET | ✅ | ✅ |
| createBus | /api/v1/bus | POST | ✅ | ❌ |
| getBusByAgencyId | /api/v1/bus/agency/{agencyId} | GET | ✅ | ✅ |
| deleteBus | /api/v1/bus/{id} | DELETE | ✅ | ❌ |
| getBusById | /api/v1/bus/{id} | GET | ✅ | ✅ |
| updateBus | /api/v1/bus/{id} | PUT | ✅ | ❌ |

1. Stop Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Endpoint** | **Method** | **Admin** | **User** |
| getAllStop | /api/v1/stop | GET | ✅ | ✅ |
| createStop | /api/v1/stop | POST | ✅ | ❌ |
| deleteStop | /api/v1/stop/{id} | DELETE | ✅ | ❌ |
| getStopById | /api/v1/stop/{id} | GET | ✅ | ✅ |
| updateStop | /api/v1/stop/{id} | PUT | ✅ | ❌ |

1. Ticket Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Endpoint** | **Method** | **Admin** | **User** |
| getAllTicket | /api/v1/ticket | GET | ✅ | ❌ |
| getAllTicket | /api/v1/ticket/trip-schedule/{tripScheduleId} | GET | ✅ | ❌ |
| getTicketById | /api/v1/ticket/{id} | GET | ✅ | ❌ |

Catatan: Controller ini hanya untuk admin melihat ticket yang terpesan, namun admin tidak dapat memesan, mengubah, atau membatalkan ticket milik user.

1. Trip Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Endpoint** | **Method** | **Admin** | **User** |
| getAllTrip | /api/v1/trip | GET | ✅ | ✅ |
| createTrip | /api/v1/trip | POST | ✅ | ❌ |
| getTripByAgencyId | /api/v1/trip/agency/{agencyId} | GET | ✅ | ✅ |
| getTripByBusId | /api/v1/trip/bus/{busId} | GET | ✅ | ✅ |
| getTripByDestStopId | /api/v1/trip/dest-stop/{destStopId} | GET | ✅ | ✅ |
| getTripBySourceStopId | /api/v1/trip/source-stop/{sourceStopId} | GET | ✅ | ✅ |
| deleteTrip | /api/v1/trip/{id} | DELETE | ✅ | ❌ |
| getTripById | /api/v1/trip/{id} | GET | ✅ | ✅ |
| updateTrip | /api/v1/trip/{id} | PUT | ✅ | ❌ |

1. Trip Schedule Controller

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Endpoint** | **Method** | **Admin** | **User** |
| getAllTripSchedule | /api/v1/trip-schedule | GET | ✅ | ✅ |
| createTripSchedule | /api/v1/trip-schedule | POST | ✅ | ❌ |
| getTripScheduleByTripId | /api/v1/trip-schedule/trip/{tripId} | GET | ✅ | ✅ |
| deleteTripSchedule | /api/v1/trip-schedule/{id} | DELETE | ✅ | ❌ |
| getTripScheduleById | /api/v1/trip-schedule/{id} | GET | ✅ | ✅ |
| updateTripSchedule | /api/v1/trip-schedule/{id} | PUT | ✅ | ❌ |

1. **Authentication Intsructions**

Berikut ini adalah langkah-langkah untuk autentikasi:

1. Memiliki akun terlebih dahulu, jika belum punya akun, bisa membuat terlebih dahulu melalui endpoint /api/v1/user/signup, namun saya sudah menyediakan masing-masing 1 akun admin dan user yang bisa digunakan, namun bias juga jika ingin membuat akun baru.
2. Melakukan autentikasi melalui endpoint /api/auth dan masukkan username dan password. Jika ingin menggunakan user yang sudah tersedia adalah sebagai berikut:
3. Admin

username: admin123

password: admin123

1. User

username: raffiahmad

password: password

1. Jika berhasil, maka pada response code pada field token berupa “Bearer <kodetoken>” copy token tersebut
2. Klik tombol authorize paste pada isian value, lalu klik Authorize
3. **API Details**
4. User Controller
5. registerUser

Endpoint: /api/v1/user/signup

Method: POST

URL Param: -

Request Body Example:

{

"email": "string",

"firstName": "string",

"lastName": "string",

"mobileNumber": "string",

"password": "string",

"role": [

"string"

],

"username": "string"

}

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | User berhasil didaftarkan |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/”” | 400 | Bad Request | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| Username sudah terpakai | 400 | Bad Request | Username sudah terpakai |
| Email sudah terpakai | 400 | Bad Request | Email sudah terpakai |
| Ada role selain “user”, “admin”, “role\_admin”, “role\_user” (case insensitive) | 400 | Bad Request | Role tidak dikenal |

Screenshot success response:



Gambar 19. Screenshot registerUser

1. updateUser

Endpoint: /api/v1/user/update

Method: PUT

URL Param: -

Request Body Example:

{

"firstName": "string",

"lastName": "string",

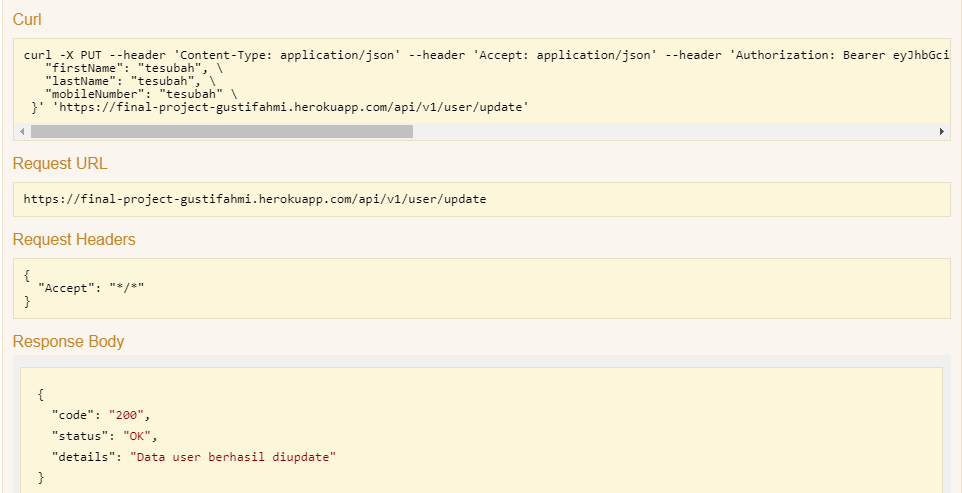
"mobileNumber": "string"

}

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Data user berhasil diupdate |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/”” | 400 | Bad Request | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Data user tidak ditemukan (kemungkinan kalau sudah berhasil authenticate, harusnya ada di database, kecuali diubah manual di database) | 404 | Not Found | Data user tidak ditemukan |

Screenshot success response (authorize terlebih dahulu):



Gambar 20. Screenshot updateUser

1. updatePassword

Endpoint: /api/v1/user/update/password

Method: PUT

URL Param: -

Request Body Example:

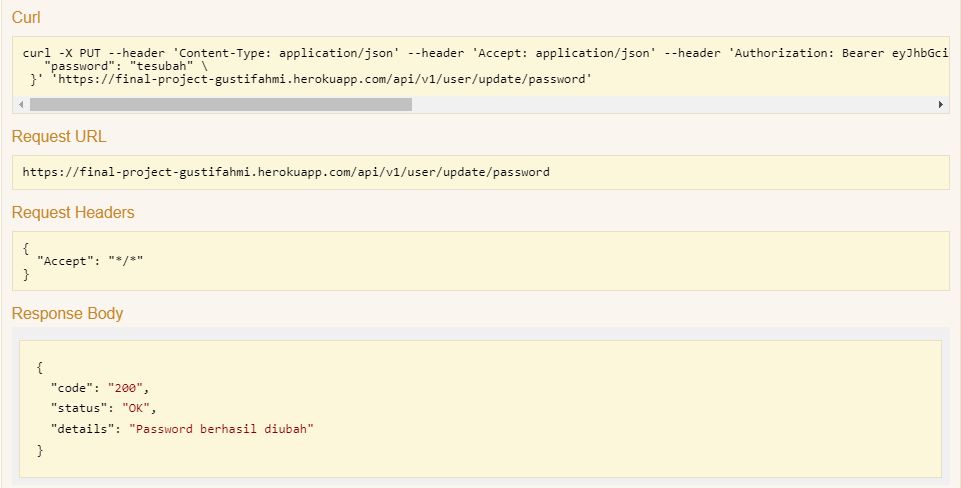
{

"password": "string"

}

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Password berhasil diubah |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/”” | 400 | Bad Request | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Data user tidak ditemukan (kemungkinan kalau sudah berhasil authenticate, seharusnya ada di database, kecuali diubah manual di database) | 404 | Not Found | Data user tidak ditemukan |

Screenshot success response (authorize terlebih dahulu):

Gambar 21. Screenshot updatePassword

1. Auth Controller
2. authenticateUser

Endpoint: /api/auth

Method: POST

URL Param: -

Request Body Example:

{

"password": "string",

"username": "string"

}

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <jwt response > |
| Authentication gagal (username kosong/salah atau password kosong/salah) | 401 | Unauthorized | <unauthorized response> |

Screenshot success response:



Gambar 22. Screenshot authenticateUser

1. Reservation Controller (menggunakan akun user)
2. bookTicket

Endpoint: /api/v1/reservation

Method: POST

URL Param: -

Request Body Example:

{

"cancellable": true,

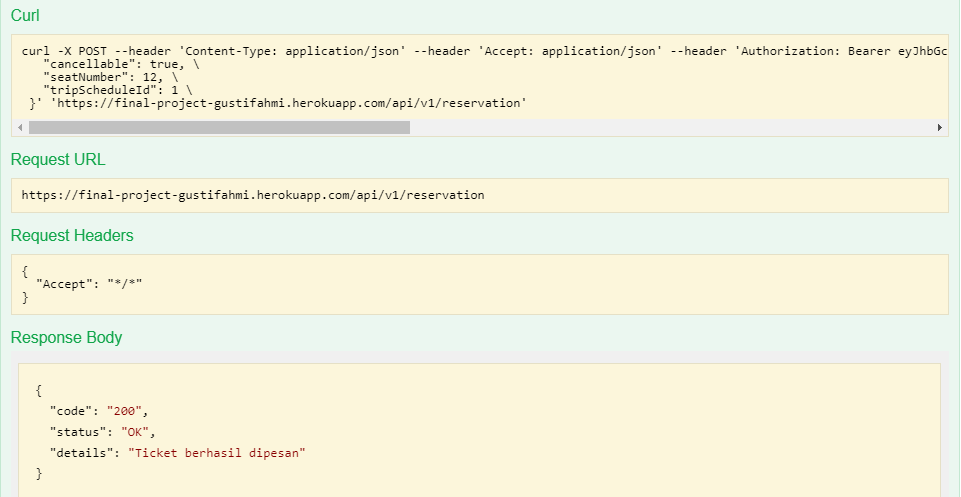
"seatNumber": 0,

"tripScheduleId": 0

}

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Ticket berhasil dipesan |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/”” | 400 | Bad Request | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| Field tripScheduleId tidak valid (tidak ada di database) | 400 | Bad Request | Data trip schedule tidak ditemukan |
| Field tripScheduleId merujuk pada trip schedule yang sudah berlalu | 400 | Bad Request | Trip schedule sudah lewat, silahkan pilih trip schedule yang lain |
| Data passenger tidak ditemukan (kemungkinan kalau sudah berhasil authenticate, seharusnya ada di database, kecuali diubah manual di database) | 400 | Bad Request | Data passenger tidak ditemukan |
| Kolom available\_seats untuk tripScheduleId tersebut 0 (ticket sudah terpesan semua sesuai kapasitas bus) | 400 | Bad Request | Mohon maaf, ticket sudah habis |
| User sudah memesan ticket untuk tripScheduleId tersebut | 400 | Bad Request | Mohon maaf, anda tidak dapat memesan lagi untuk trip schedule ini |
| seatNumber > kapasitas bus, sehingga tidak valid | 400 | Bad Request | Seat number tidak valid. Kapasitas bus hanya <capacity> kursi |
| seatNumber yang dipilih sudah dipesan oleh user lain | 400 | Bad Request | Seat number sudah dipesan, silahkan pilih seat number lain |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun admin | 403 | Forbidden | <forbidden response> |

Screenshot success response:Gambar 23. Screenshot bookTicket

1. getAvailableSeatNumberByTripScheduleId

Endpoint: /api/v1/reservation/available-seat-number/{tripScheduleId}

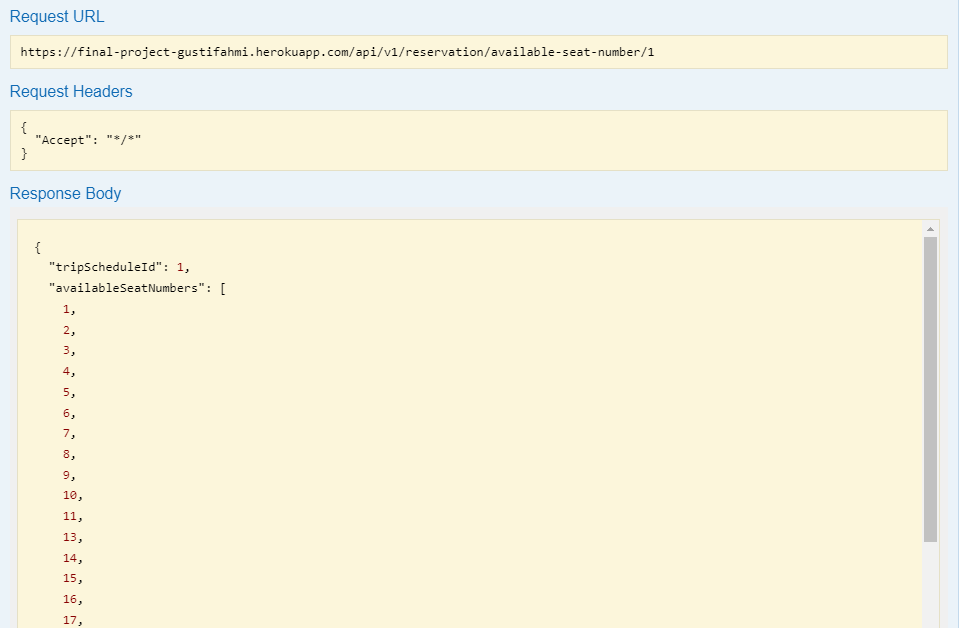
Method: GET

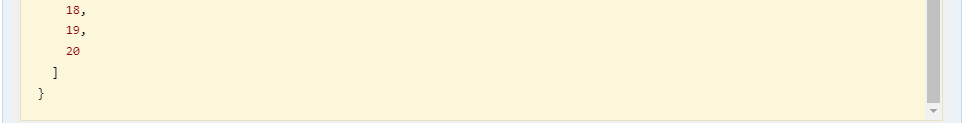
URL Param: tripScheduleId

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <list of available seat number by that trip schedule id> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun admin | 403 | Forbidden | <forbidden response> |
| tripScheduleId tidak valid (tidak ada di database) | 404 | Not Found | Data trip schedule tidak ditemukan |

Screenshot success response:****

****

Gambar 24. Screenshot getAvailableSeatNumberByTripScheduleId

1. getAllOwnedTicketByLoggedInUser

Endpoint: /api/v1/reservation/passenger

Method: GET

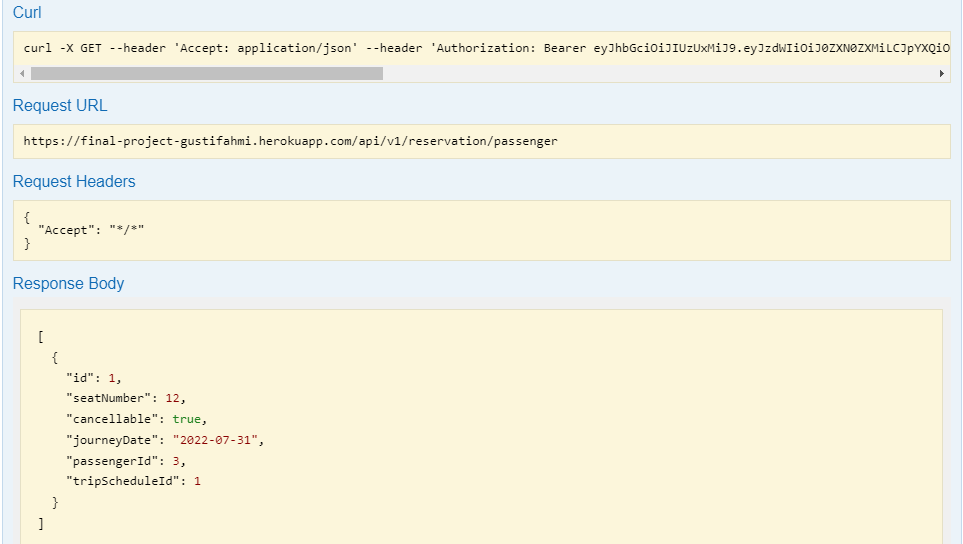
URL Param: -

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <list of owned ticket by logged in user> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun admin | 403 | Forbidden | <forbidden response> |
| User belum memiliki ticket | 404 | Not Found | Anda belum memiliki ticket yang tersimpan |

Screenshot success response:



Gambar 25. Screenshot getAllOwnedTicketByLoggedInUser

1. cancelTicket

Endpoint: /api/v1/reservation/{id}

Method: DELETE

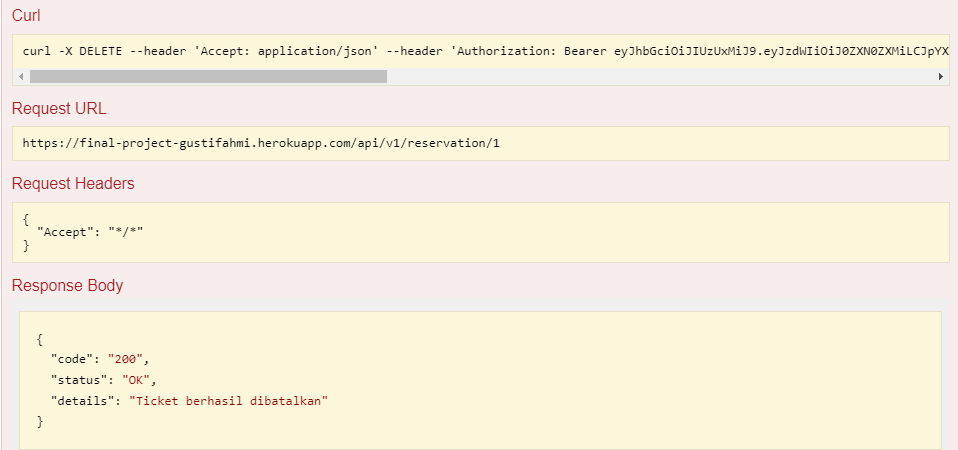
URL Param: id

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Ticket berhasil didaftarkan |
| Id yang dimasukkan valid, namun bukan milik user yang login | 400 | Bad Request | Ticket hanya dapat dibatalkan oleh pemilik ticket |
| Kolom cancellable false, sehingga tidak dapat dibatalkan | 400 | Bad Request | Mohon maaf, ticket tidak dapat dibatalkan |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun admin | 403 | Forbidden | <forbidden response> |
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data ticket tidak ditemukan |

Screenshot success response:



Gambar 26. Screenshot cancelTicket

1. getTicketById

Endpoint: /api/v1/reservation/{id}

Method: GET

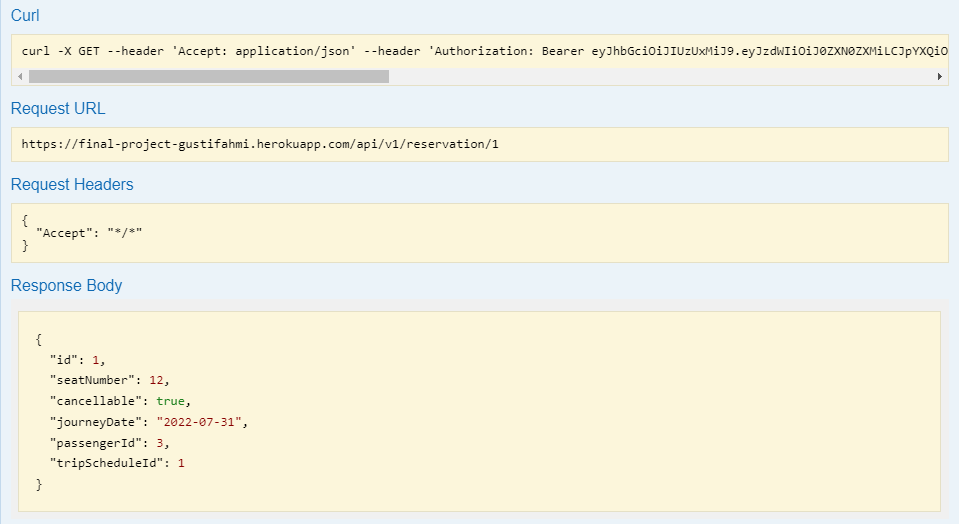
URL Param: -

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <ticket response> |
| Id ticket adalah milik user lain, sehingga tidak dapat melihat ticket tersebut | 400 | Bad Request | Anda tidak dapat melihat ticket milik penumpang lain |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun admin | 403 | Forbidden | <forbidden response> |
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data ticket tidak ditemukan |

Screenshot success response (sebelum delete):



Gambar 27. Screenshot getTicketById

1. Agency Controller (mulai dari sini, hingga akhir, menggunakan akun admin)
2. getAllAgency

Endpoint: /api/v1/agency

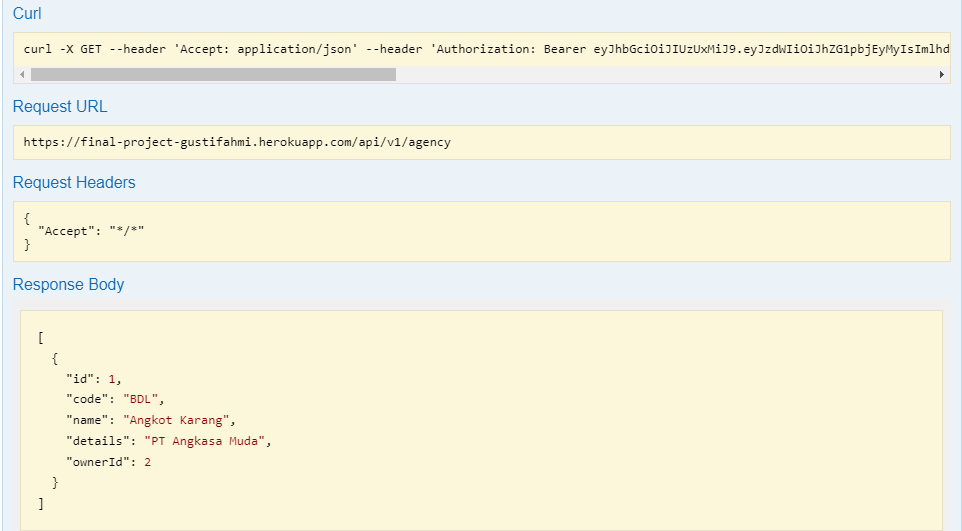
Method: GET

URL Param: -

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <list of agency> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Tidak ada data agency di database | 404 | Not Found | Tidak ada agency yang tersimpan |

Screenshot success response:

Gambar 28. Screenshot getAllAgency

1. createAgency

Endpoint: /api/v1/agency

Method: POST

URL Param: -

Request Body Example:

{

"code": "string",

"details": "string",

"name": "string",

"ownerId": 0

}

Response

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Agency berhasil disimpan |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/”” | 400 | Bad Request | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| ownerId tidak valid (tidak ada di database) | 400 | Bad request | Data owner tidak ditemukan |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |

Screenshot success response:



Gambar 29. Screenshot createAgency

1. getAgencyByOwnerId

Endpoint: /api/v1/agency/owner/{ownerId}

Method: GET

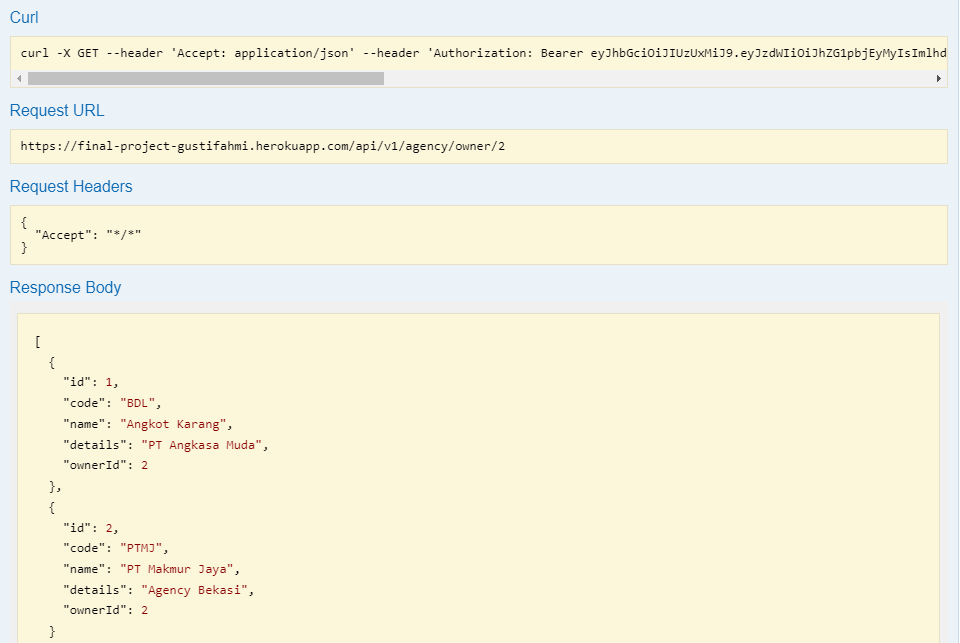
URL Param: ownerId

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <list of agency by ownerId> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| ownerId tidak valid (tidak ada di database) | 404 | Not Found | Data owner tidak ditemukan |
| Owner tidak memiliki agency | 404 | Not Found | User tidak memiliki agency |

Screenshot success response:



Gambar 30. Screenshot getAgencyByOwnerId

1. deleteAgency

Endpoint: /api/v1/agency/{id}

Method: DELETE

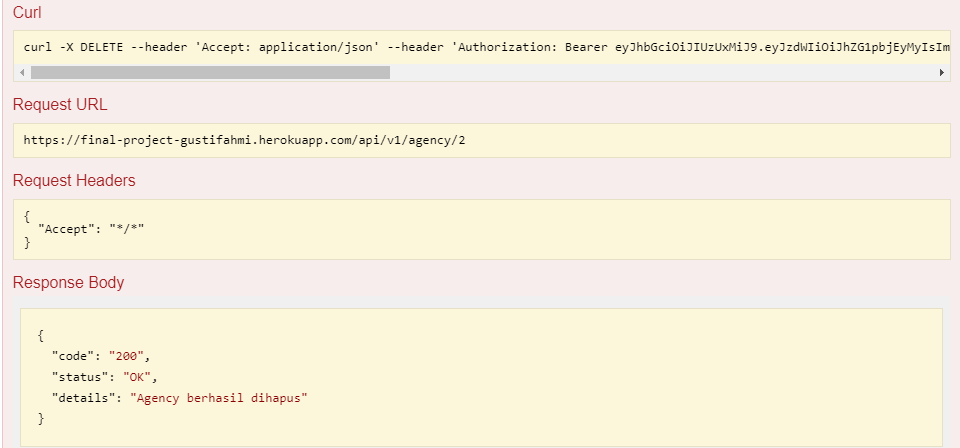
URL Param: id

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Agency berhasil dihapus |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data agency tidak ditemukan |

Screenshot success response:



Gambar 31. Screenshot deleteAgency

1. getAgencyById

Endpoint: /api/v1/agency/{id}

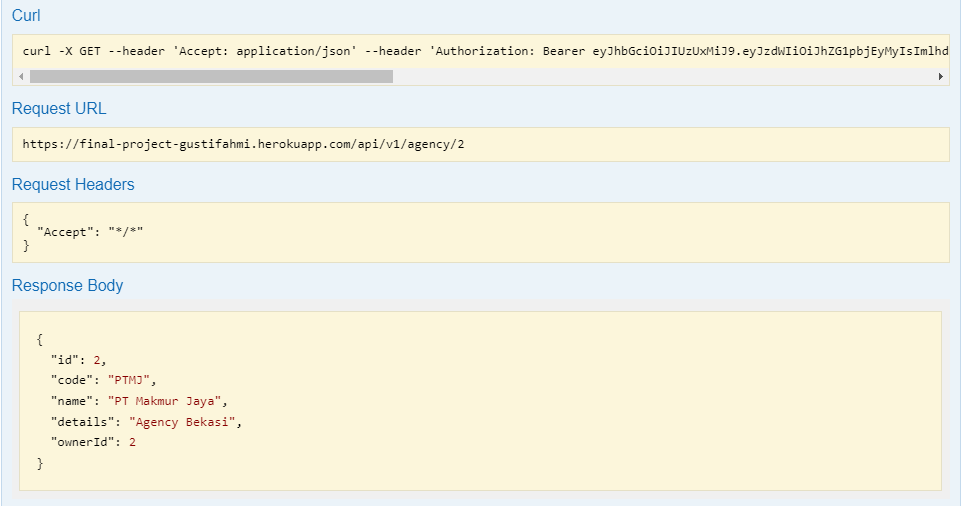
Method: GET

URL Param: id

Request Body Example: -

Response

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <agency response> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data agency tidak ditemukan |

Screenshot success response (sebelun delete): 

Gambar 32. Screenshot getAgencyById

1. updateAgency

Endpoint: /api/v1/agency/{id}

Method: PUT

URL Param: id

Request Body Example:

{

"code": "string",

"details": "string",

"name": "string",

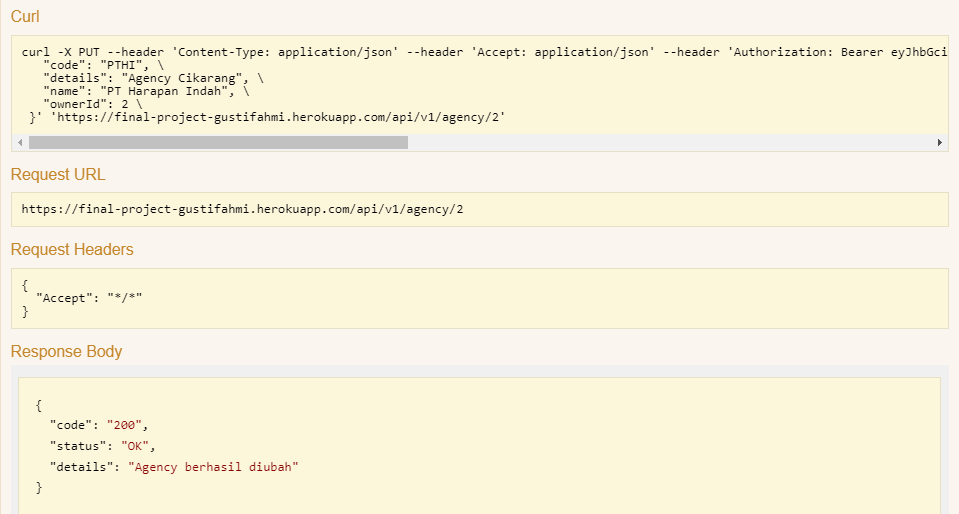
"ownerId": 0

}

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Agency berhasil diubah |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/”” | 400 | Bad request | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| ownerId tidak valid (tidak ada di database) | 400 | Bad request | Data owner tidak ditemukan |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data agency tidak ditemukan |

Screenshot success response (sebelun delete):



Gambar 33. Screenshot updateAgency

1. Bus Controller
2. getAllBus

Endpoint: /api/v1/bus

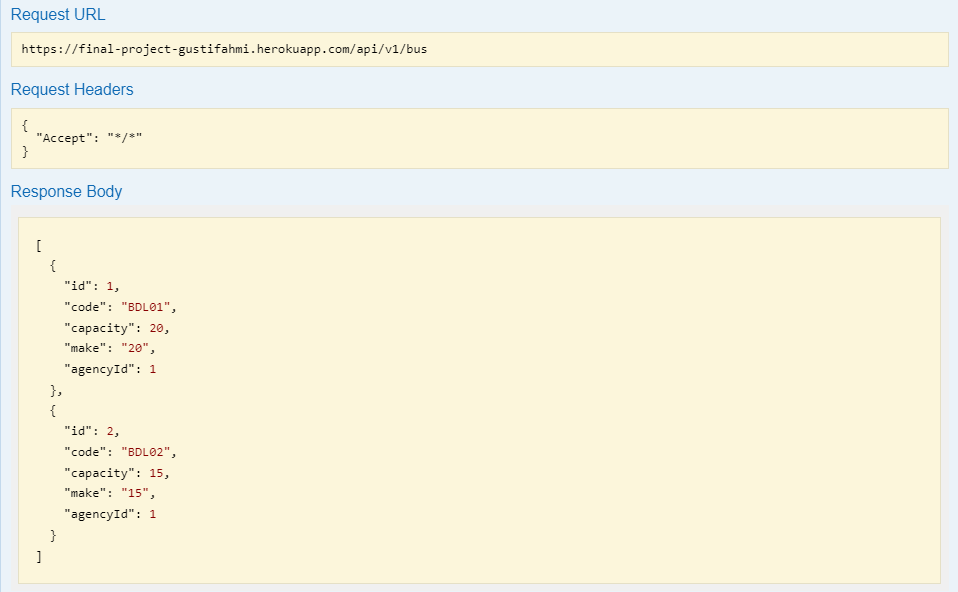
Method: GET

URL Param: -

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <list of bus> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Tidak ada data bus di database | 404 | Not Found | Tidak ada bus yang tersimpan |

Screenshot success response:****

Gambar 34. Screenshot getAllBus

1. createBus

Endpoint: /api/v1/bus

Method: POST

URL Param: -

Request Body Example:

{

"agencyId": 0,

"capacity": 0,

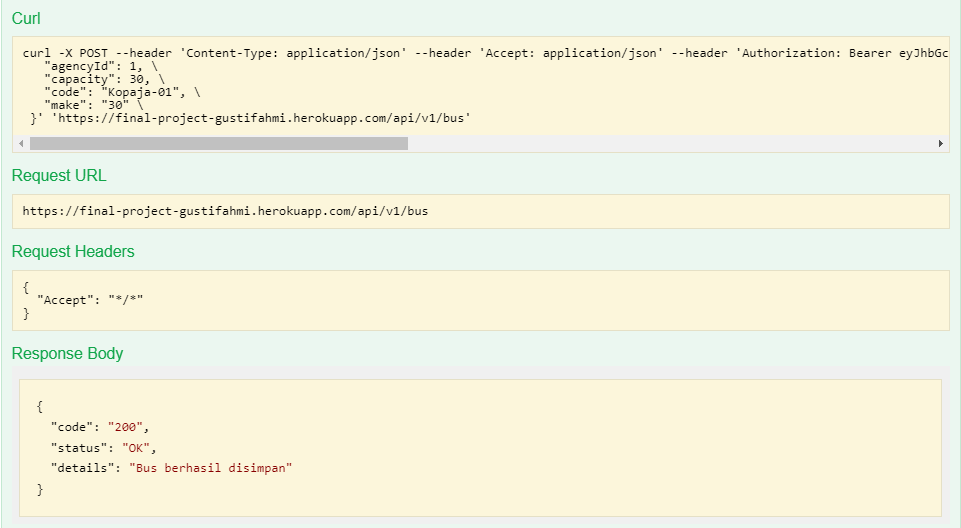
"code": "string",

"make": "string"

}

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Bus berhasil disimpan |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/”” | 400 | Bad Request | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| agencyId tidak valid (tidak ada di database) | 400 | Bad Request | Data agency tidak ditemukan |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |

Screenshot success response:

Gambar 35. Screenshot createBus

1. getBusByAgencyId

Endpoint: /api/v1/bus/agency/{agencyId}

Method: GET

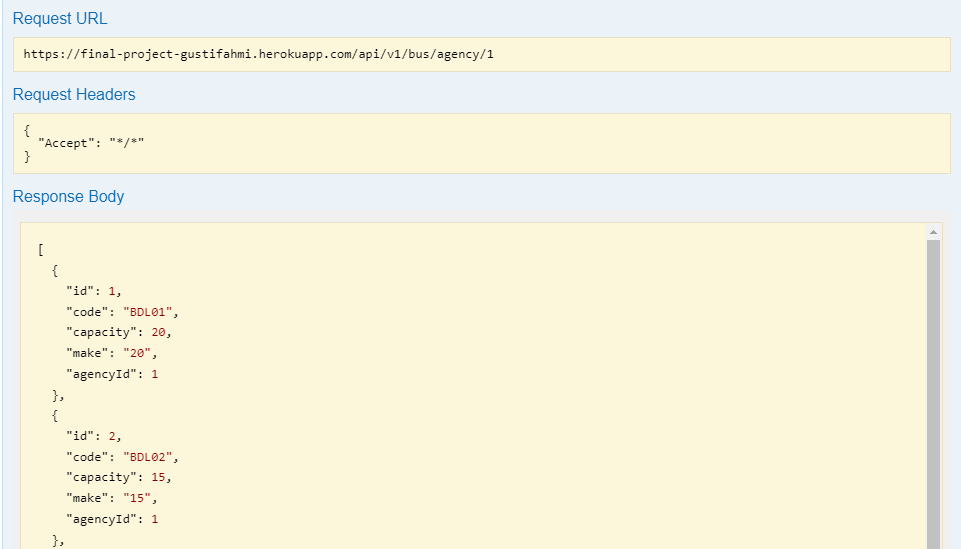
URL Param: agencyId

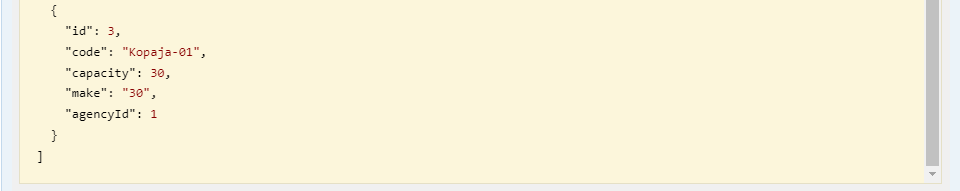
Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <list of bus by agency id> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| agencyId tidak valid (tidak ada di database) | 404 | Not Found | Data agency tidak ditemukan |
| Agency tidak memiliki bus | 404 | Not Found | Agency tidak memiliki bus |

Screenshot success response:





Gambar 36. Screenshot getBusByAgencyId

1. deleteBus

Endpoint: /api/v1/bus/{id}

Method: DELETE

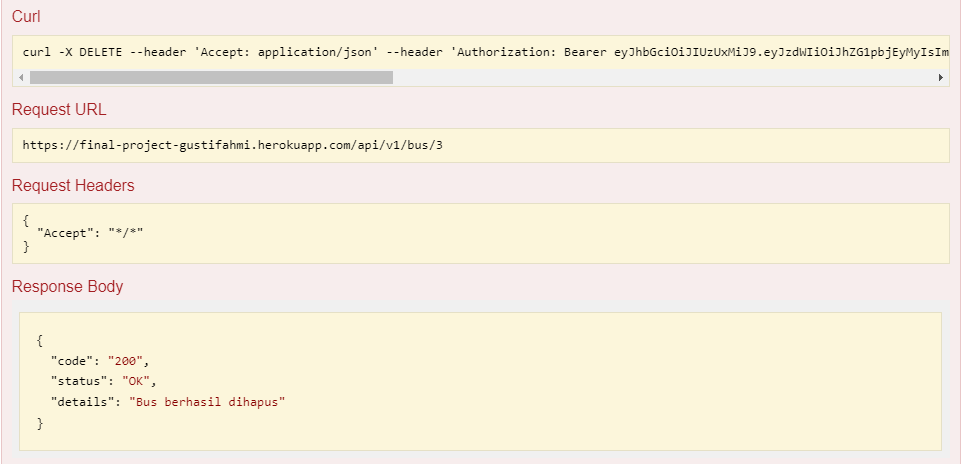
URL Param: id

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Bus berhasil dihapus |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data bus tidak ditemukan |

Screenshot success response:



Gambar 37. Screenshot deleteBus

1. getBusById

Endpoint: /api/v1/bus/{id}

Method: GET

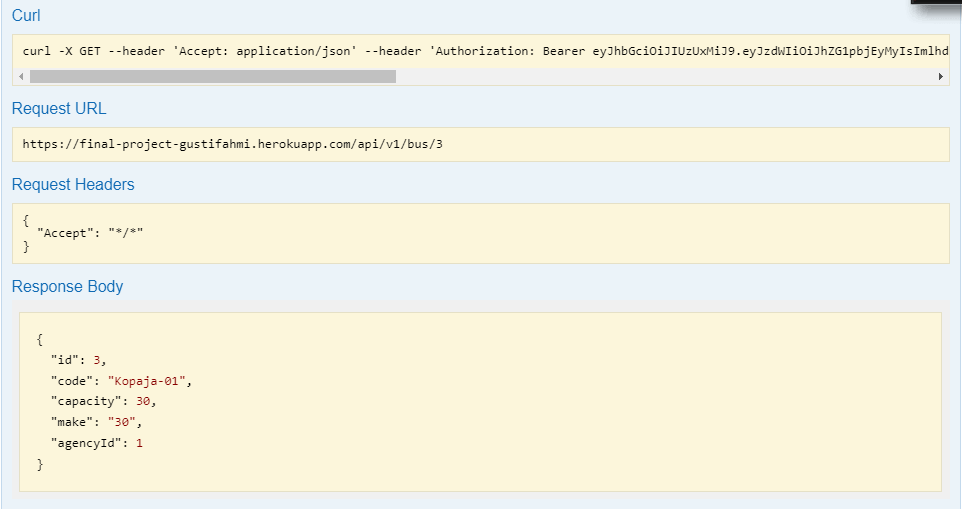
URL Param: id

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <bus response> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Id tidak valid (tidak ada dalam database) | 404 | Not Found | Data bus tidak ditemukan |

Screenshot success response (sebelum delete):



Gambar 38. Screenshot getBusById

1. updateBus

Endpoint: /api/v1/bus/{id}

Method: PUT

URL Param: id

Request Body Example:

{

"agencyId": 0,

"capacity": 0,

"code": "string",

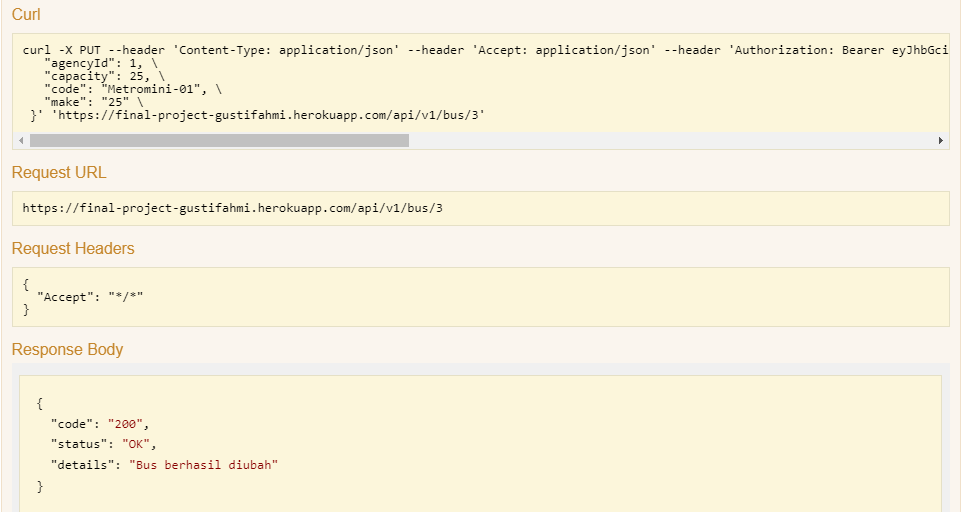
"make": "string"

}

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Bus berhasil diubah |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/”” | 400 | Bad Request | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| agencyId tidak valid (tidak ada di database) | 400 | Bad Request | Data agency tidak ditemukan |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data bus tidak ditemukan |

Screenshot success response (sebelum delete):



Gambar 39. Screenshot updateBus

1. Stop Controller
2. getAllStop

Endpoint: /api/v1/stop

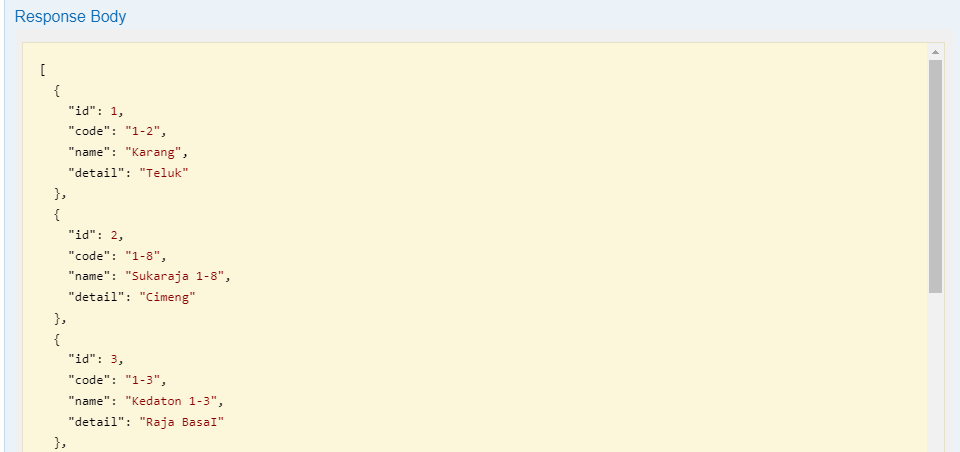
Method: GET

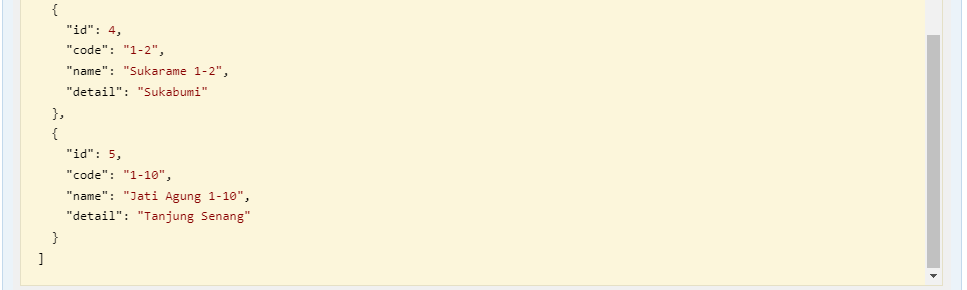
URL Param: -

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <list of all stop> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Tidak ada data stop di database | 404 | Not Found | Tidak ada stop yang tersimpan |

Screenshot success response:



Gambar 40. Screenshot getAllStop

1. createStop

Endpoint: /api/v1/stop

Method: GET

URL Param: -

Request Body Example:

{

"code": "string",

"detail": "string",

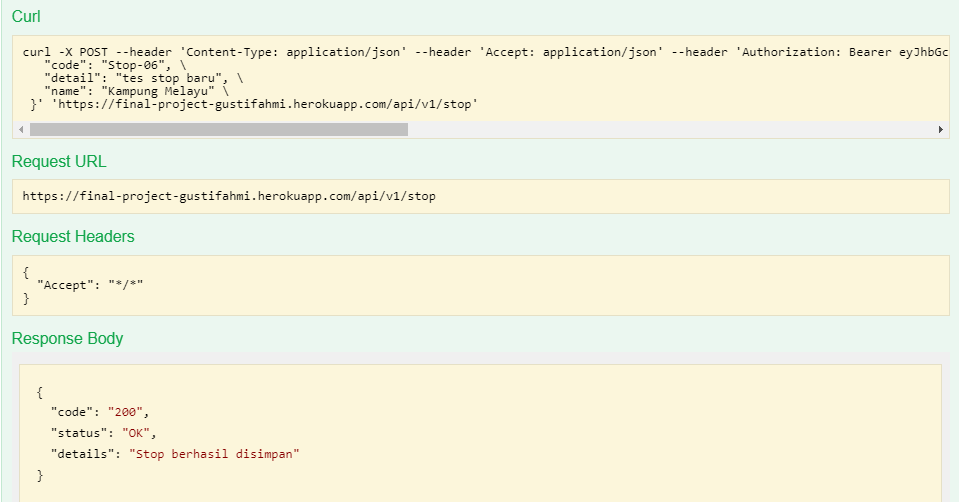
"name": "string"

}

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Stop berhasil disimpan |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/”” | 400 | Bad Request | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |

Screenshot success response:



Gambar 41. Screenshot createStop

1. deleteStop

Endpoint: /api/v1/stop/{id}

Method: DELETE

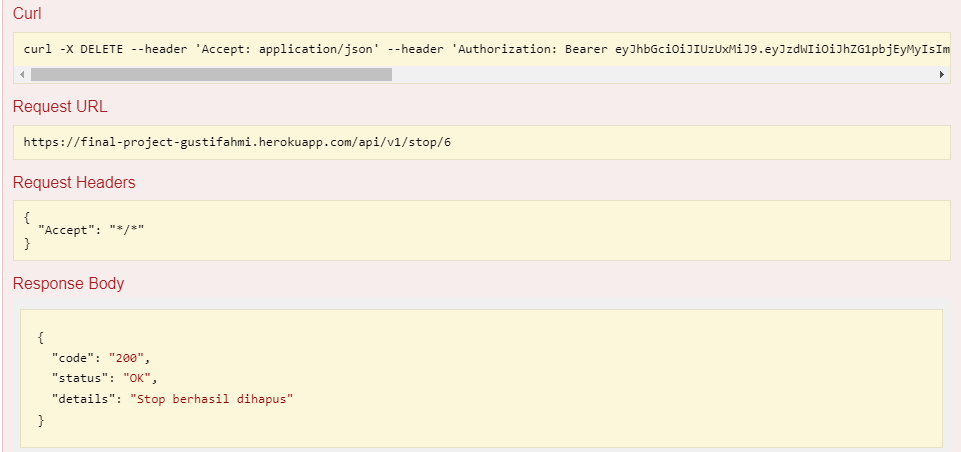
URL Param: id

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Stop berhasil dihapus |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data stop tidak ditemukan |

Screenshot success response:



Gambar 42. Screenshot deleteStop

1. getStopById

Endpoint: /api/v1/stop/{id}

Method: GET

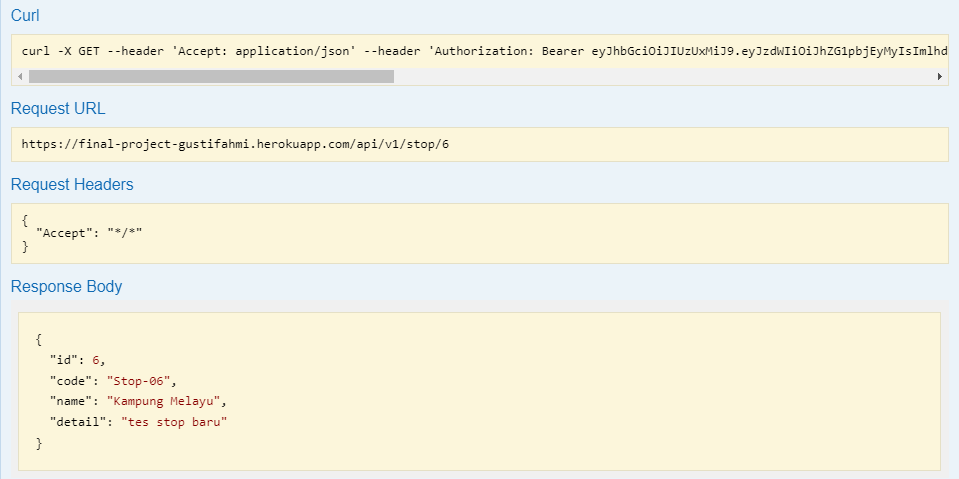
URL Param: id

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <stop response> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data stop tidak ditemukan |

Screenshot response (sebelum delete):



Gambar 43. Screenshot getStopById

1. updateStop

Endpoint: /api/v1/stop/{id}

Method: PUT

URL Param: id

Request Body Example:

{

"code": "string",

"detail": "string",

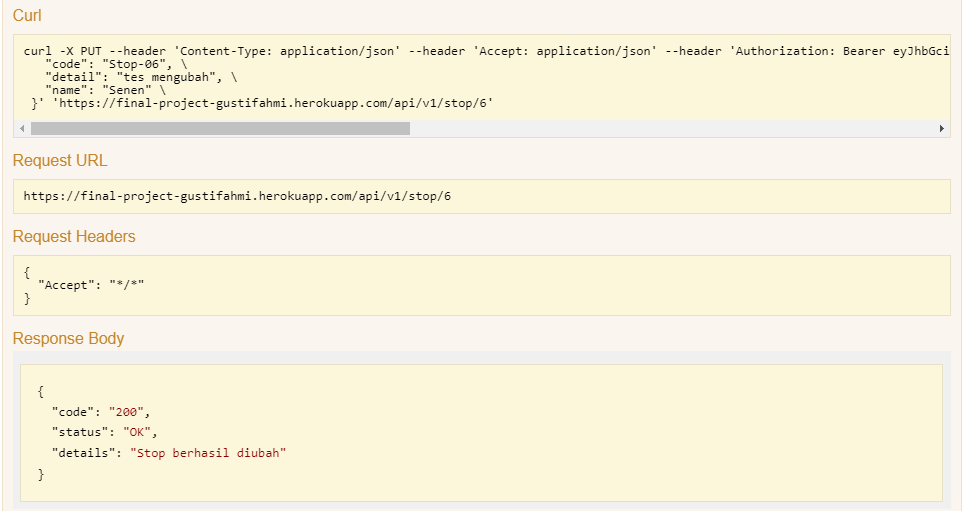
"name": "string"

}

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Stop berhasil diubah |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/”” | 400 | Bad Request | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data stop tidak ditemukan |

Screenshot response (sebelum delete):



Gambar 44. Screenshot updateStop

1. Ticket Controller
2. getAllTicket

Endpoint: /api/v1/ticket

Method: GET

URL Param: -

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <list of all ticket> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |
| Tidak ada data ticket di database | 404 | Not Found | Tidak ada ticket yang tersimpan |

Screenshot success response (pastikan ada tiket yang terpesan):

Gambar 45. Screenshot getAllTicket

1. getAllTicketByTripScheduleId

Endpoint: /api/v1/ticket/trip-schedule/{tripScheduleId}

Method: GET

URL Param: tripScheduleId

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <list of ticket by trip schedule id> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |
| tripScheduleId tidak valid (tidak ada di database) | 404 | Not Found | Data trip schedule tidak ditemukan |
| Tidak ada tiket yang terjual untuk trip schedule tersebut | 404 | Not Found | Tidak ada ticket yang dipesan dari trip schedule tersebut |

Screenshot success response (pastikan ada tiket yang terpesan):



Gambar 46. Screenshot getAllTicketByTripScheduleId

1. getTicketById

Endpoint: /api/v1/ticket/{id}

Method: GET

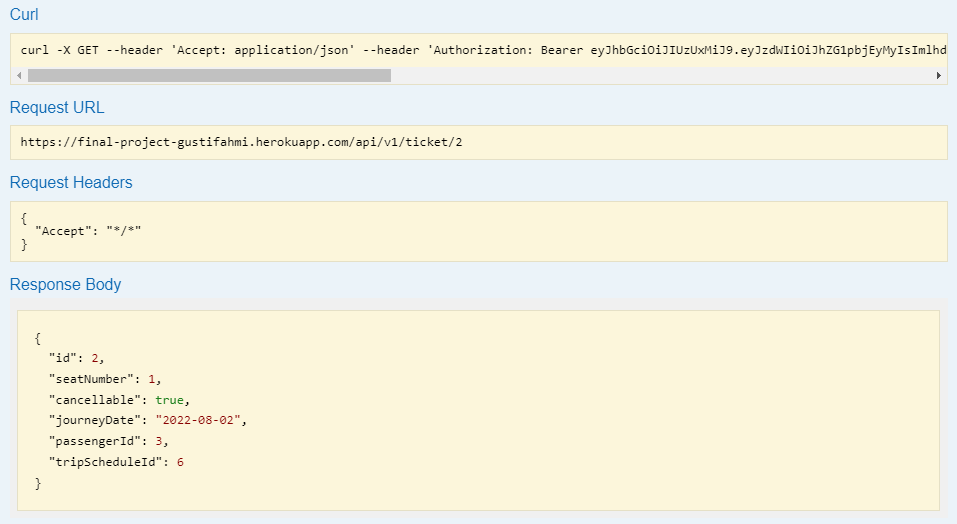
URL Param: id

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <ticket response> |
| Belum authorize | 401 | Unauthorize | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data ticket tidak ditemukan |

Screenshot success response (pastikan ada tiket yang terpesan):



Gambar 47. Screenshot getTicketById

1. Trip Controller
2. getAllTrip

Endpoint: /api/v1/trip

Method: GET

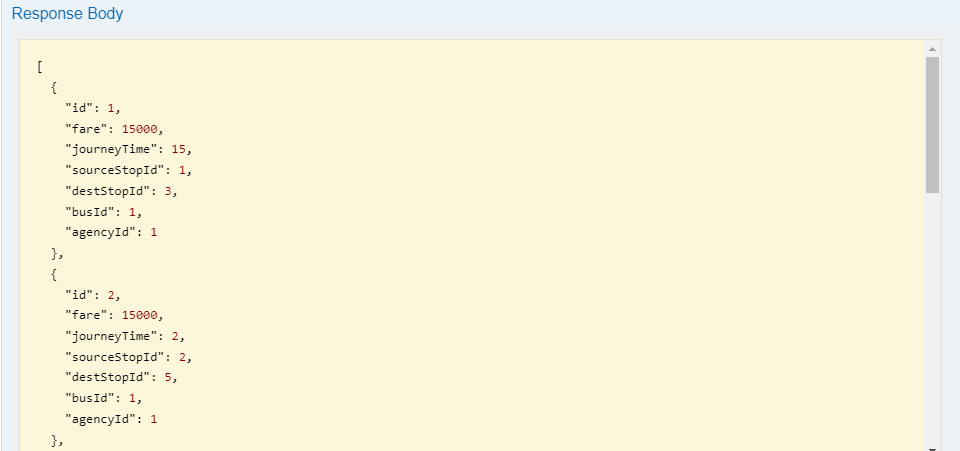
URL Param: -

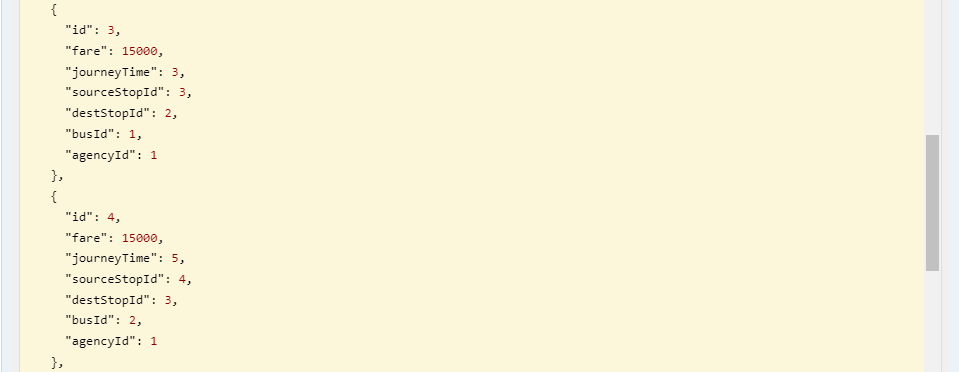
Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <list of all trip> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Tidak ada data trip di database | 404 | Not Found | Tidak ada trip yang tersimpan |

Screenshot response:







Gambar 48. Screenshot getAllTrip

1. createTrip

Endpoint: /api/v1/trip

Method: POST

URL Param: -

Request Body Example:

{

"busId": 0,

"destStopId": 0,

"fare": 0,

"journeyTime": 0,

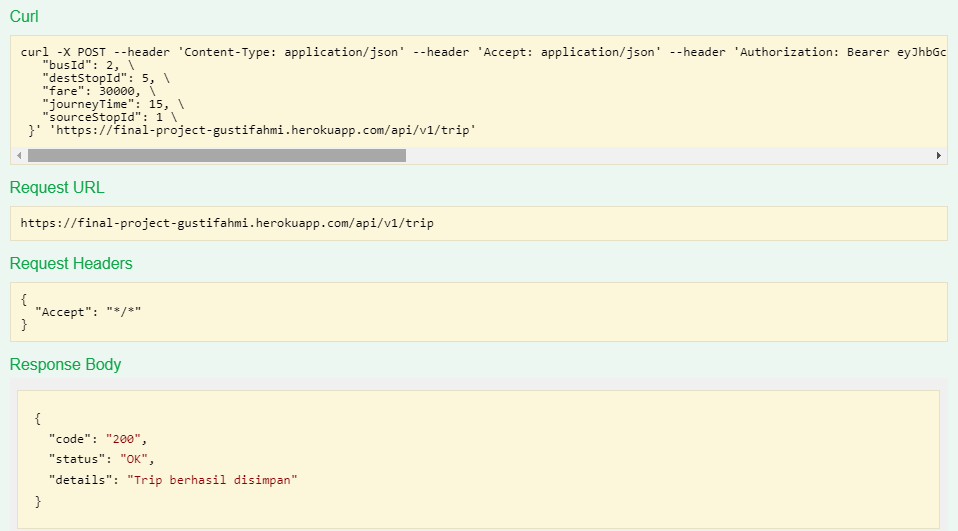
"sourceStopId": 0

}

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Trip berhasil disimpan |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/”” | 400 | Bad Request | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| busId tidak valid (tidak ada di database) | 400 | Bad Request | Data bus tidak ditemukan |
| Bus tidak memiliki kolom agencyId yang valid | 400 | Bad Request | Data agency tidak ditemukan |
| sourceStopId tidak valid (tidak ada di database) | 400 | Bad Request | Data destination stop tidak ditemukan |
| destStopId tidak valid (tidak ada di database) | 400 | Bad Request | Data source stop tidak ditemukan |
| Nilai sourceStopId sama dengan destStopId | 400 | Bad Request | Field sourceStopId dan destStopId tidak boleh sama |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |

Screenshot success response:



Gambar 49. Screenshot createTrip

1. getTripByAgencyId

Endpoint: /api/v1/trip/agency/{agencyId}

Method: GET

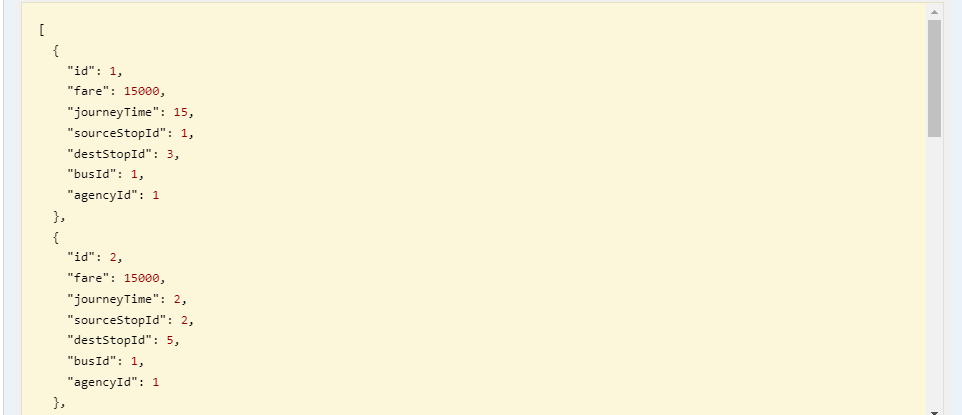
URL Param: agencyId

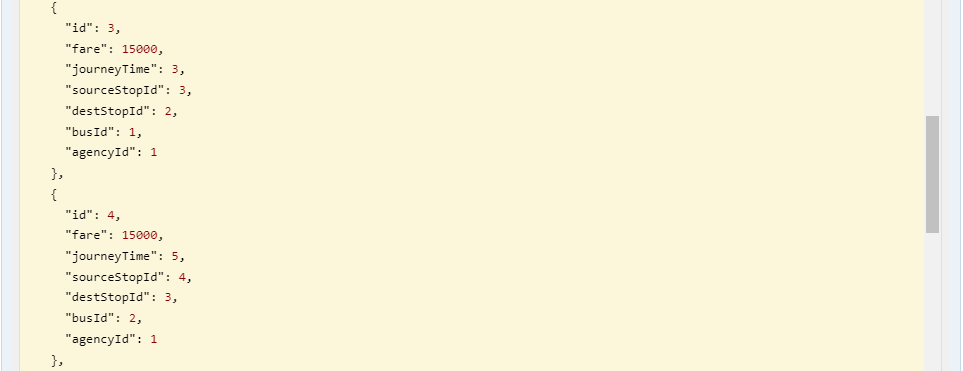
Request Body Example: -

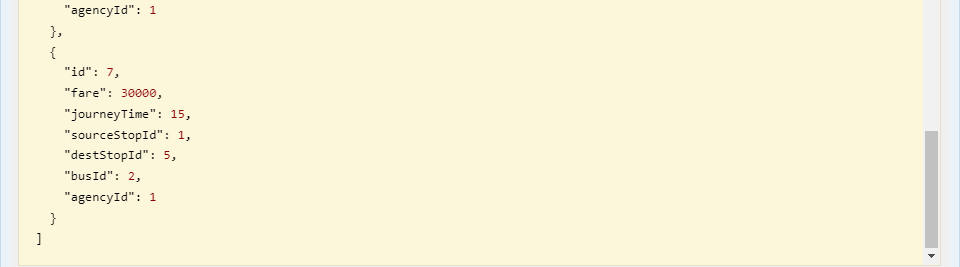
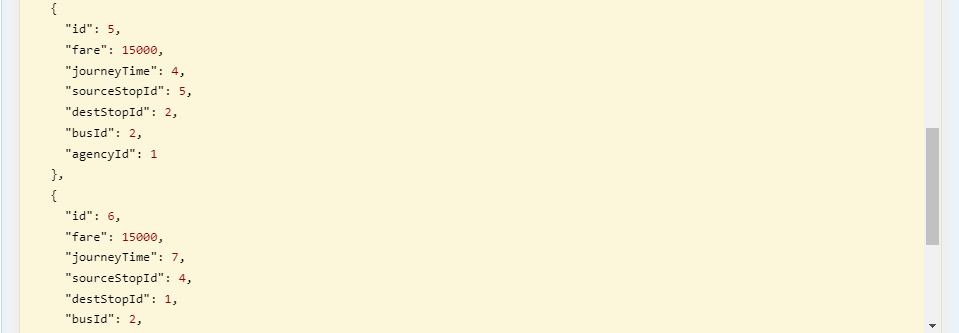
Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <list of trip by agencyId> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Tidak ada trip dari agency tersebut | 404 | Not Found | Agency tidak memliki trip |
| agencyId tidak valid (tidak ada di database) | 404 | Not Found | Data agency tidak ditemukan |

Screenshot success response:







Gambar 50. Screenshot getTripByAgencyId

1. getTripByBusId

Endpoint: /api/v1/trip/bus/{busId}

Method: GET

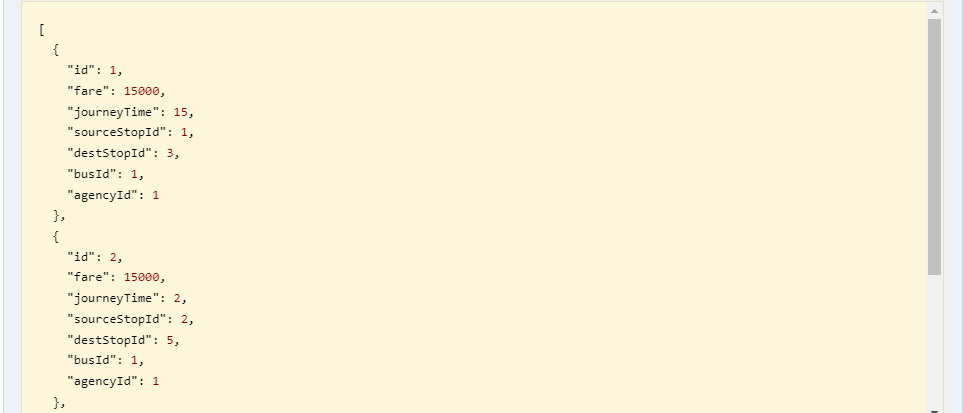
URL Param: busId

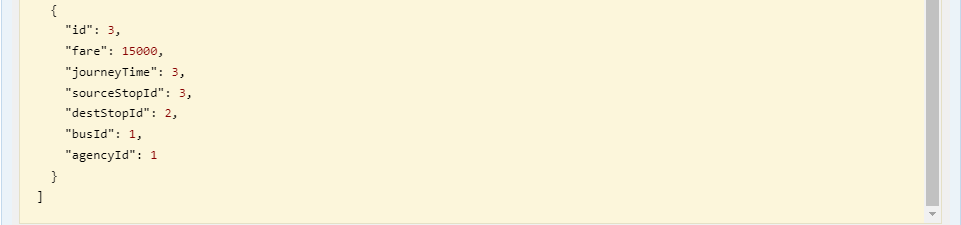
Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <list of trip by busId> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Tidak ada trip dengan menggunakan bus tersebut | 404 | Not Found | Bus tidak memliki trip |
| busId tidak valid (tidak ada di database) | 404 | Not Found | Data bus tidak ditemukan |

Screenshot success response:





Gambar 51. Screenshot getTripByBusId

1. getTripByDestStopId

Endpoint: /api/v1/trip/dest-stop/{destStopId}

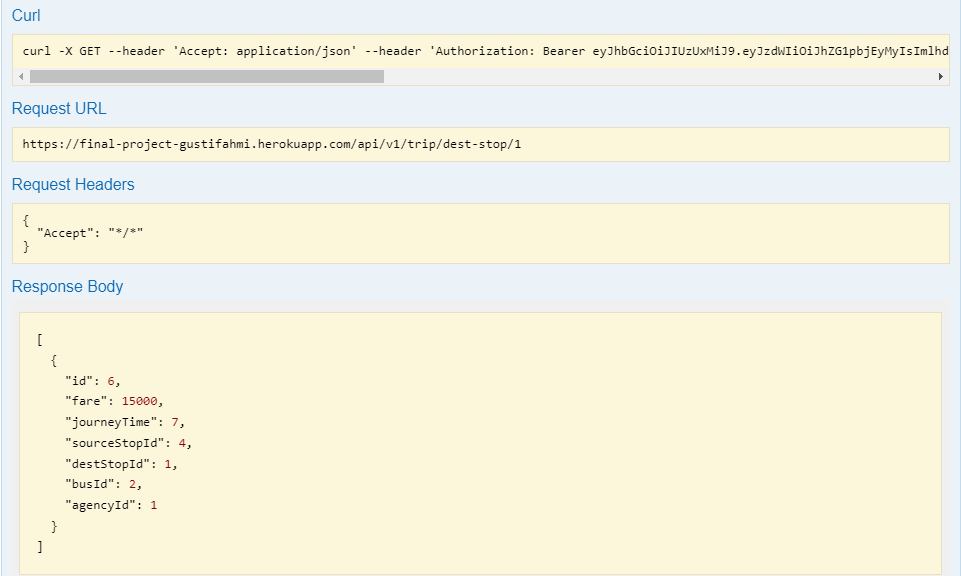
Method: GET

URL Param: destStopId

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <list of trip by destStopId> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Tidak ada trip yang bertujuan ke stop tersebut | 404 | Not Found | Tidak ada trip yang bertujuan ke stop tersebut |
| destStopId tidak valid (tidak ada di database) | 404 | Not Found | Data stop tidak ditemukan |

Screenshot success response:

Gambar 52. Screenshot getTripByDestStopId

1. getTripBySourceStopId

Endpoint: /api/v1/trip/source-stop/{sourceStopId}

Method: GET

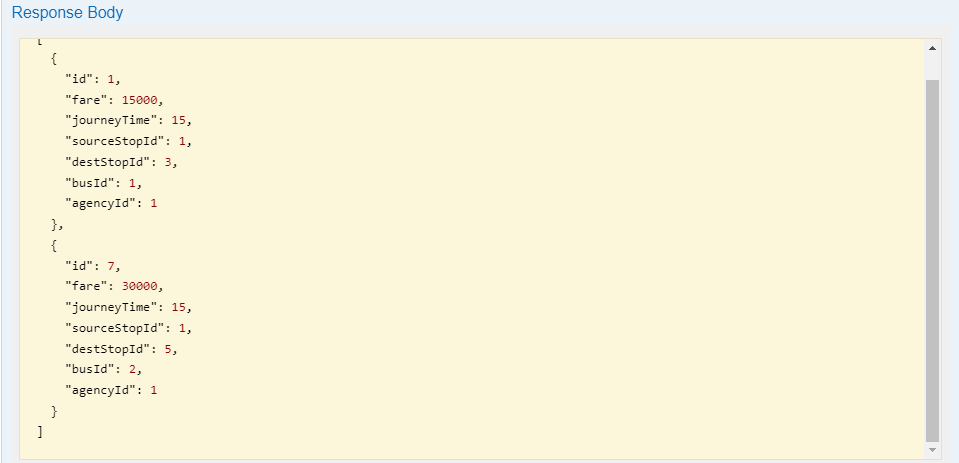
URL Param: sourceStopId

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <list of trip by sourceStopId> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Tidak ada trip yang berangkat dari stop tersebut | 404 | Not Found | Tidak ada trip yang berangkat dari stop tersebut |
| destStopId tidak valid (tidak ada di database) | 404 | Not Found | Data stop tidak ditemukan |

Screenshot success response:



Gambar 53. Screenshot getTripBySourceStopId

1. deleteTrip

Endpoint: /api/v1/trip/{id}

Method: DELETE

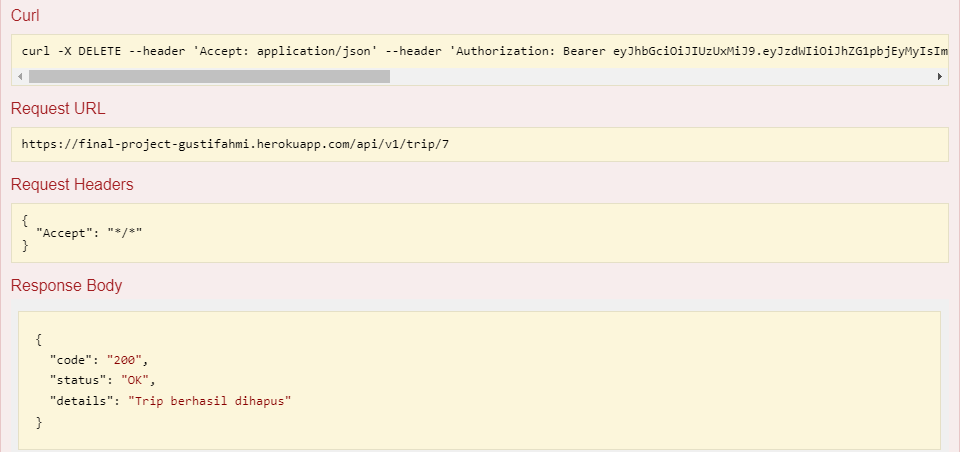
URL Param: id

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Trip berhasil dihapus |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data trip tidak ditemukan |

Screenshot success response:



Gambar 54. Screenshot deleteTrip

1. getTripById

Endpoint: /api/v1/trip/{id}

Method: GET

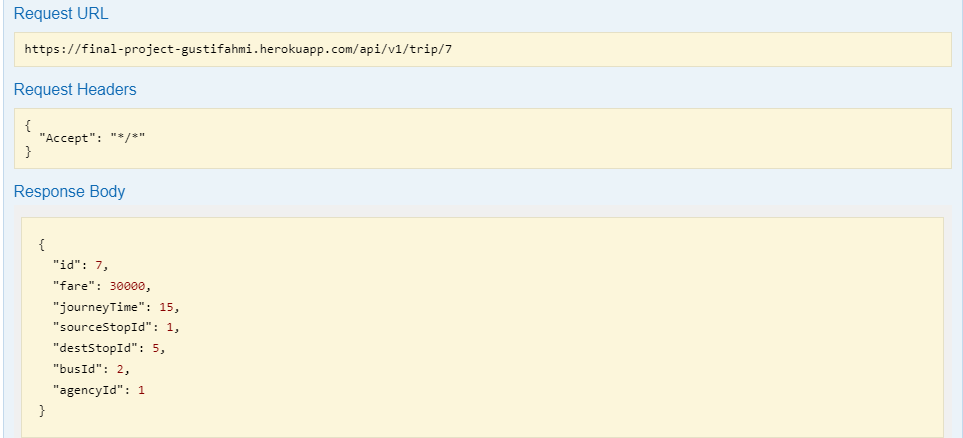
URL Param: id

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <trip response> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data trip tidak ditemukan |

Screenshot success response (sebelum delete):



Gambar 55. Screenshot getTripById

1. updateTrip

Endpoint: /api/v1/trip/{id}

Method: PUT

URL Param: id

Request Body Example:

{

"busId": 0,

"destStopId": 0,

"fare": 0,

"journeyTime": 0,

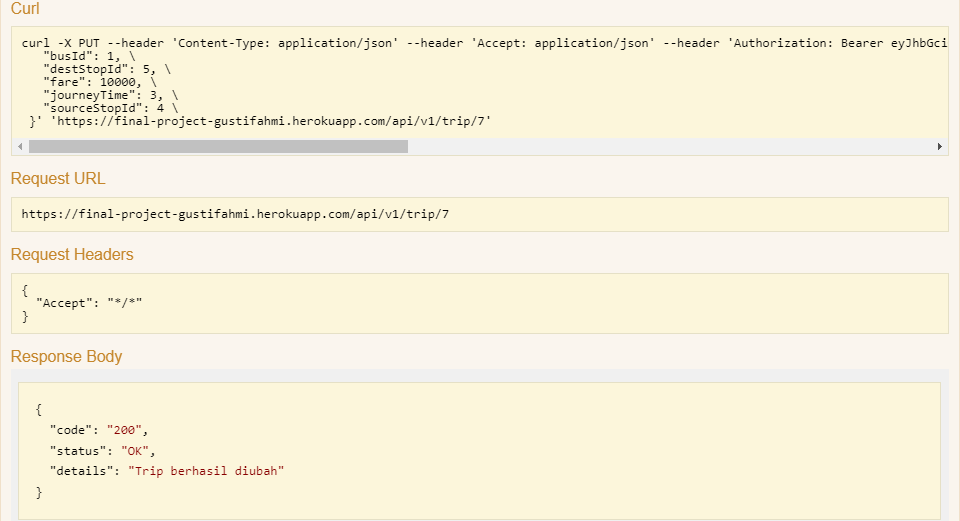
"sourceStopId": 0

}

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Trip berhasil diubah |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/”” | 400 | Bad Request | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| busId tidak valid (tidak ada di database) | 400 | Bad Request | Data bus tidak ditemukan |
| Bus tidak memiliki kolom agencyId yang valid | 400 | Bad Request | Data agency tidak ditemukan |
| sourceStopId tidak valid (tidak ada di database) | 400 | Bad Request | Data destination stop tidak ditemukan |
| destStopId tidak valid (tidak ada di database) | 400 | Bad Request | Data source stop tidak ditemukan |
| Nilai sourceStopId sama dengan destStopId | 400 | Bad Request | Field sourceStopId dan destStopId tidak boleh sama |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data trip tidak ditemukan |

Screenshot success response (sebelum delete):



Gambar 56. Screenshot updateTrip

1. Trip Schedule Controller
2. getAllTripSchedule

Endpoint: /api/v1/trip-schedule

Method: GET

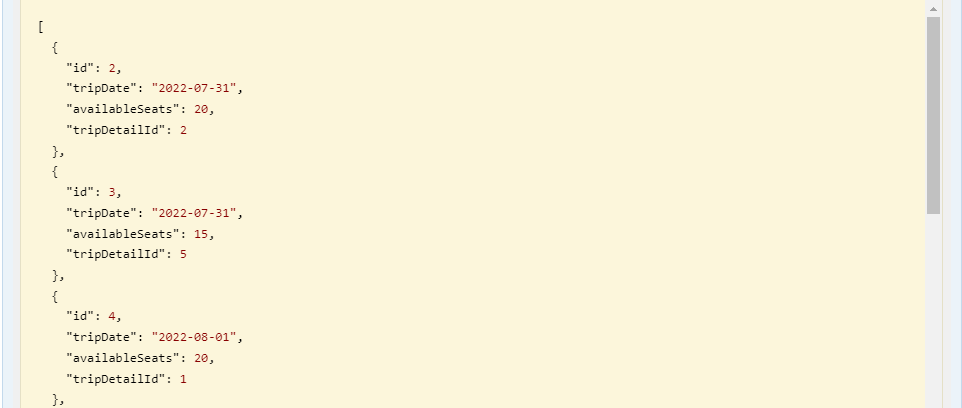
URL Param: -

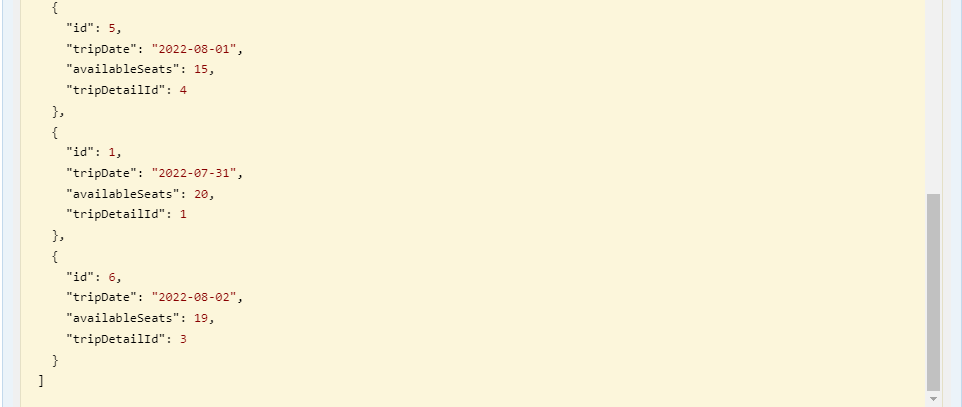
Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <list of all trip schedule> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Tidak ada data trip schedule di database | 404 | Not Found | Tidak ada trip schedule yang tersimpan |

Screenshot success response:





Gambar 57. Screenshot getAllTripSchedule

1. createTripSchedule

Endpoint: /api/v1/trip-schedule

Method: POST

URL Param: -

Request Body Example:

{

"tripDate": "dd-MM-yyyy",

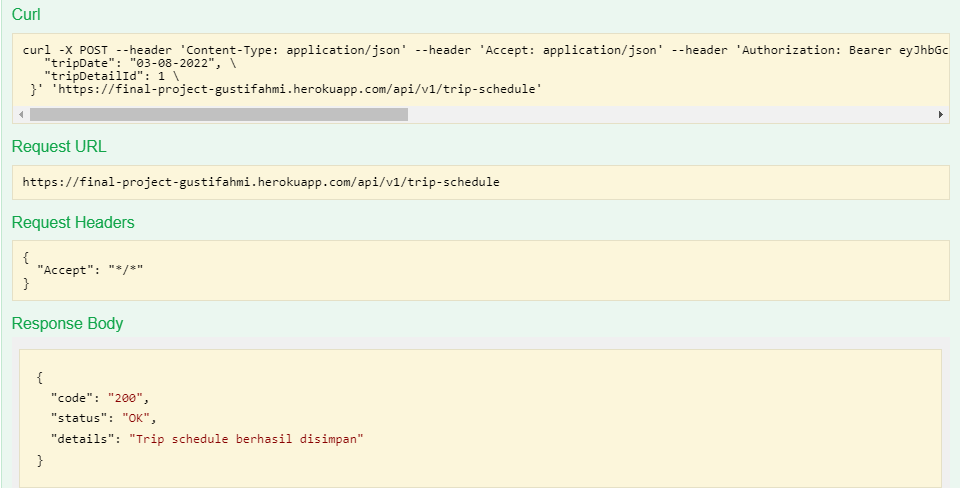
"tripDetailId": 0

}

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Trip schedule berhasil disimpan |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/”” | 400 | Bad Request | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| tripDetailId tidak valid (tidak ada di database) | 400 | Bad Request | Data trip detail tidak ditemukan |
| Trip date sudah terlewat, sehingga tidak bisa membuat trip schedule | 400 | Bad Request | Trip date sudah lewat, silahkan masukkan trip date yang lain |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |

Screenshot success response:



Gambar 58. Screenshot getTripSchedule

1. getTripScheduleByTripId

Endpoint: /api/v1/trip-schedule/trip/{tripId}

Method: GET

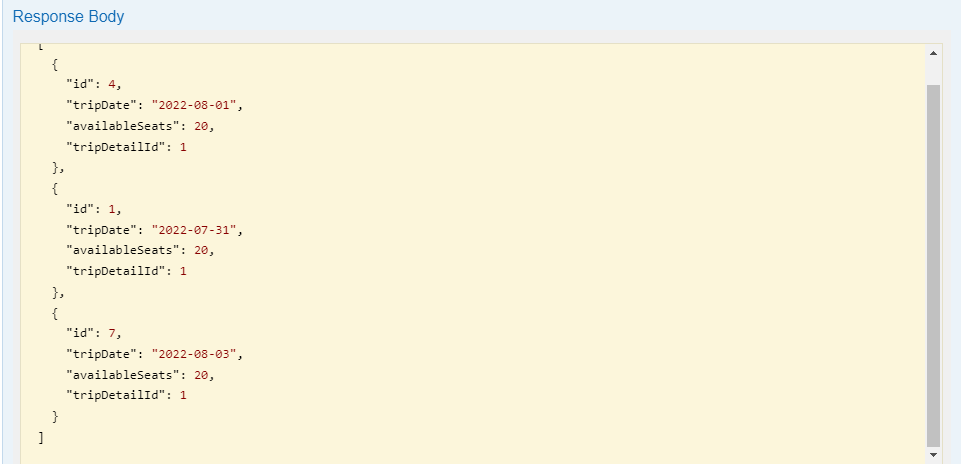
URL Param: tripId

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <list trip schedule by trip id> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| tripId tidak valid (tidak ada di database) | 404 | Not Found | Data trip tidak ditemukan |
| Trip tersebut tidak memiliki trip schedule | 404 | Not Found | Tidak ada trip schedule dari trip tersebut |

Screenshot success response:



Gambar 59. Screenshot getTripScheduleByTripId

1. deleteTripSchedule

Endpoint: /api/v1/trip-schedule/{id}

Method: DELETE

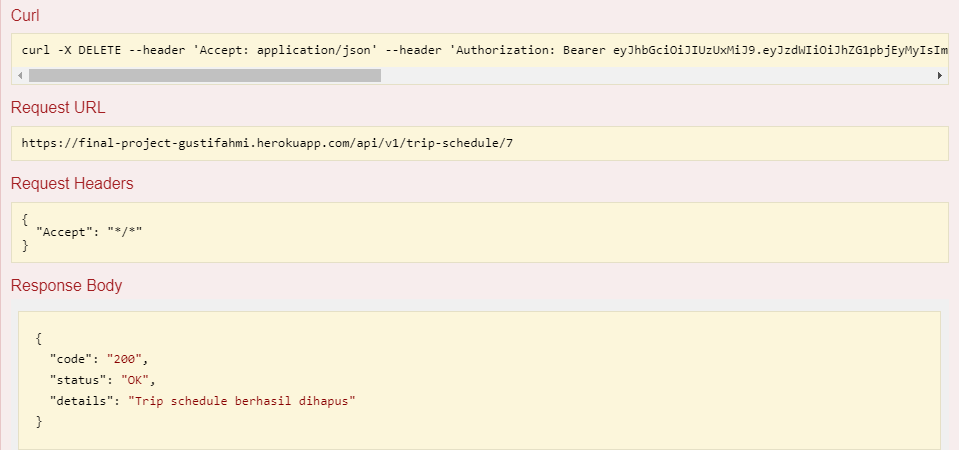
URL Param: id

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Trip schedule berhasil dihapus |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data trip schedule tidak ditemukan |

Screenshot success response:



Gambar 60. Screenshot deleteTripSchedule

1. getTripScheduleById

Endpoint: /api/v1/trip-schedule/{id}

Method: GET

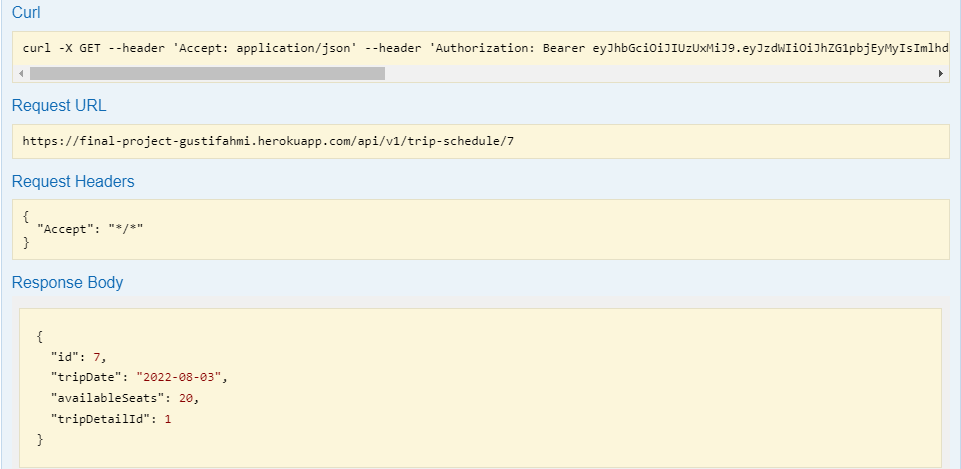
URL Param: id

Request Body Example: -

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | <trip schedule response> |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data trip schedule tidak ditemukan |

Screenshot success response (sebelum delete):



Gambar 61. Screenshot getTripScheduleById

1. updateTripSchedule

Endpoint: /api/v1/trip-schedule/{id}

Method: PUT

URL Param: id

Request Body Example:

{

"tripDate": "dd-MM-yyyy",

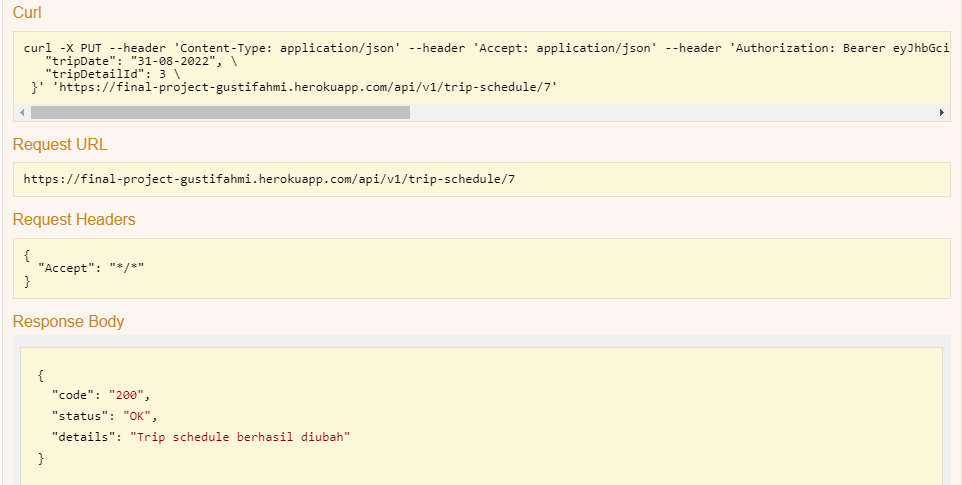
"tripDetailId": 0

}

Response:

|  |  |  |  |
| --- | --- | --- | --- |
| **Condition** | **Code** | **Status** | **Message** |
| Berhasil | 200 | OK | Trip schedule berhasil diubah |
| Ada field yang tidak ada atau ada tapi bernilai 0/null/”” | 400 | Bad Request | Pastikan field tidak kosong dan tidak bernilai 0 atau null atau string kosong |
| tripDetailId tidak valid (tidak ada di database) | 400 | Bad Request | Data trip detail tidak ditemukan |
| Trip date sudah terlewat, sehingga tidak bisa membuat trip schedule | 400 | Bad Request | Trip date sudah lewat, silahkan masukkan trip date yang lain |
| Belum authorize | 401 | Unauthorized | <unauthorized response> |
| Menggunakan akun user | 403 | Forbidden | <forbidden response> |
| Id tidak valid (tidak ada di database) | 404 | Not Found | Data trip schedule tidak ditemukan |

Screenshot success response (sebelum delete):



Gambar 62. Screenshot updateTripSchedule