

Dokumentasi Reservasi Tiket Bioskop

Team C – Chapter 5

(Database & SpringBoot)

Catatan Release Dokumen

|  |  |  |  |
| --- | --- | --- | --- |
| **Tanggal Release** | **Versi** | **Oleh** | **Catatan perubahan** |
| 30 juni 2022 | 1.0.0 | Team C – Chapter 5 | Versi Pertama |

1. Project Flow
2. **Query DDL**
3. Pembuatan table “films”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column name | Tipe data | Not null | key | default |
| film\_id | int | v | Primary Key |  |
| name | Varchar(100) | v |  |  |
| Is\_playing | int | v |  |  |
| created\_at | Timestamp | v |  | now() |
| updated\_at | Timestamp | v |  | now() |

1. Pembuatan table “schedule”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column name | Tipe data | Not null | key | default |
| schedule\_id | int | v | Primary Key |  |
| Film\_id | int | v | Foreign Key |  |
| Seat\_id | int | v | Foreign Key |  |
| Price | int | v |  |  |
| Date\_show | date | v |  |  |
| Show\_start | time | v |  |  |
| Show\_end | time | v |  |  |
| Created\_at | Timestamp | v |  | now() |
| Updated\_at | Timestamp | v |  | now() |

1. Pembuatan table “seats”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column name | Tipe data | Not null | key | default |
| Seat\_id | Int | v | Primary Key |  |
| Seat\_number | Int | v |  |  |
| Studio\_name | Varchar(50) | v |  |  |
| Is\_available | int | v |  | 0 |
| Created\_at | Timestamp | v |  | now() |
| Updated\_at | Timestamp | v |  | now() |

1. Pembuatan table “Reservation”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column name | Tipe data | Not null | key | default |
| Reservation\_id | int | v | Primary key |  |
| User\_id | int | v | Foreign Key |  |
| Schedule\_id | int | v | Foreign Key |  |
| Created\_at | Timestamp | v |  | now() |
| Updated\_at | Timestamp | v |  | now() |

1. Pembuatan table “user”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column name | Tipe data | Not null | key | default |
| User\_id | int | v | Primary Key |  |
| username | Varchar(50) | v |  |  |
| Email\_id | Varchar(225) | v |  |  |
| password | Varchar(225) | v |  |  |
| Created\_at | Timestamp | v |  | now() |
| Updated\_at | Timestamp | v |  | now() |

1. **Query DML Insert**

|  |  |
| --- | --- |
| Query | Tampilan |
| **INSERT** **INTO** "project".public.Films(**name**,is\_playing)  **VALUES**  ('Jurassic World Dominion',1),  ('The Roundup',1),  ('Avatar 2',1),  ('Sonic the Hedgehog 2',1),  ('The Batman',1),  ('KKN di Desa Penari',1) |  |

1. **Query DML Select**

|  |  |
| --- | --- |
| Query | Tampilan |
| **SELECT** film\_id, **name**, is\_playing  **FROM** cinema.public.films  **WHERE** film\_id ='60'; |  |

1. **Query DML Join**

|  |  |
| --- | --- |
| Query | Tampilan |
| **select** \* **from** cinema.public.reservation r  **join** cinema.public.films **on** (r.reservation\_id = films.film\_id); |  |

1. **Running API**
2. Ringkasan

Semua permintaan BIOSKOP API dibuat ke titik akhir mulai: [http://localhost:8089/teamC/v1/](http://localhost:8089/teamC/v1/%20) dan <http://localhost:8089/teamC/v1/DTO>

1. Resources
   1. User
      * **GET By ID**

**Deksripsi Endpoint**

Request GET BY ID digunakan untuk menampilkan data User sesuai dengan (id) yang diinputkan.

**End Point :** http://localhost:8089/teamC/v1/user-controller/getUserById

**Request URL**

|  |
| --- |
| http://localhost:8089/swagger-ui/index.html#/user-controller/getUserById |

**Sample Request**

|  |
| --- |
| {  “userId” : 6  } |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {    "data": [      { "ID             ": 6,        "Username       ": "Lionel",        "Email          ": "Lionel@gmail.com",        "Password       ": "Rahasiaaaa4x1"      }    ],    "message": "Successfully Get User By ID!",    "status": 200  } |

|  |
| --- |
| Failed Response |
| {    "data": "Data Not Found!",    "message": "User not exist with id :66666",    "status": 404  } |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| username | String | M |  | User name yang dimiliki oleh konsumen. | Lily |
| emailId | String | M |  | Email yang digunakan oleh konsumen saat mendaftar. | Lily@gmail.com |
| password | String | M |  | Password yang diinput oleh konsumen saat mendaftar. | 123456aaaa1 |
| userId | Long | M |  | User Id yang diberikan system untuk konsumen. | 9 |

* **PUT**

**Deksripsi Endpoint**

Request PUT digunakan untuk mengubah/memodifikasi data User yang ada menjadi data yang baru

**End Point** : http://localhost:8089/teamC/v1/user-controller/updateUser

**Request URL**

|  |
| --- |
| http://localhost:8089/swagger-ui/index.html#/user-controller/updateUser |

**Sample Request**

|  |
| --- |
| {  "userId": 8,  "username": "string",  "emailId": "string",  "password": "string"  } |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {    "data": {      "userId": 8,      "username": "string",      "emailId": "string",      "password": "string"    },    "message": "Successfully Updated User!",    "status": 200  } |

|  |
| --- |
| Failed Response |
| {    "data": "Data Not Found!",    "message": "User not exist with id :66666",    "status": 404  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| username | String | M |  | User name baru konsumen yang ingin diupdate. | Lily |
| emailId | String | M |  | Email baru konsumen yang ingin diupdate. | Lily@gmail.com |
| password | String | M |  | Password baru konsumen yang ingin diupdate. | 123456aaaa1 |
| userId | Long | M |  | User Id konsumen yang ingin diupdate.. | 9 |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| username | String | M |  | User name baru yang telah diupdate. | Lily |
| emailId | String | M |  | Email baru baru yang telah diupdate. | Lily@gmail.com |
| password | String | M |  | Password baru baru yang telah diupdate. | 123456aaaa1 |
| userId | Long | M |  | User Id konsumen yang ingin diupdate.. | 9 |

* **DELETE**

**Deksripsi Endpoint**

By Request DELETE digunakan untuk menghapus data User menggunakan parameter

**End Point :** <http://localhost:8089/teamC/v1/user-controller/deleteUser>

**Request URL**

|  |
| --- |
| http://localhost:8089/swagger-ui/index.html#/user-controller/deleteUser |

**Sample Request**

|  |
| --- |
| {  userId : 8  } |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {    "data": {      "deleted": **true**    },    "message": "Successfully Delete User! ",    "status": 200  } |

|  |
| --- |
| Failed Response |
| {    "data": "Data Not Found!",    "message": "User not exist with id :8999999",    "status": 404  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| userid | Long | M |  | User Id yang ingin di delete. | 5 |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| deleted | Boolean | M |  | Status perintah Delete. | true |

* **POST**

**Deksripsi Endpoint**

By Request POST digunakan untuk menginput/memasukan data User baru ke database

**End Point :** <http://localhost:8089/teamC/v1/user-controller/createUser>

**Request URL**

|  |
| --- |
| http://localhost:8083/swagger-ui/index.html#/user-controller/createUser |

**Sample Request**

|  |
| --- |
| {  "username": "Luffy",  "emailId": "luffybajaklaut@onepiece.com",  "password": "1231231",  "userId": 101,  "id": 0  } |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {    "data": [      {        "ID             ": 62,        "Username       ": "Luffy",        "Email          ": "luffybajaklaut@onepiece.com",        "Password       ": "1231231"      }    ],    "message": "Successfully Created User!",    "status": 201  } |

|  |
| --- |
| Failed Response |
| {      "data": "User Already Exist!",      "message": "could not execute statement; SQL [n/a]; constraint [null]; nested exception is org.hibernate.exception.ConstraintViolationException: could not execute statement",      "status": 400  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| username | String | M |  | User name konsumen baru yang ingin diinsert. | Luffy |
| emailId | String | M |  | Email konsumen baru yang ingin diinsert. | luffybajaklaut@onepiece.com |
| password | String | M |  | Password konsumen baru yang ingin diinsert. | 1231231 |
| userId | Long | M |  | User Id konsumen baru yang ingin diinsert. | 101 |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| username | String | M |  | User name konsumen yang berhasil diinsert. | Luffy |
| emailId | String | M |  | Email konsumen yang berhasil diinsert. | luffybajaklaut@onepiece.com |
| password | String | M |  | Password konsumen yang berhasil diinsert. | 1231231 |
| userId | Long | M |  | User Id konsumen yang berhasil diinsert. | 101 |

* **GET ALL**

**Deksripsi Endpoint**

Request GET digunakan untuk menampilkan semua list data entry di table User

**End Point :** http://localhost:8083/teamC/v1/user-controller/getAllUser

**Request URL**

|  |
| --- |
| http://localhost:8083/swagger-ui/index.html#/user-controller/getAllUser |

**Sample Request**

|  |
| --- |
|  |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {    "data": [      {        "userId": 1,        "username": "Wahyu Wijanarko",        "emailId": "WahyuW@gmail.com",        "password": "Rahasiaaaa4x"      },      ......      {        "userId": 62,        "username": "Luffy",        "emailId": "luffybajaklaut@onepiece.com",        "password": "1231231"      }    ],    "message": "Successfully Get All User!",    "status": 200  } |

|  |
| --- |
| Failed Response |
| {      "data": [],      "message": "Tabel has no value!",      "status": 404  } |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| username | String | M |  | User name yang dimiliki oleh konsumen. | Lily |
| emailId | String | M |  | Email yang digunakan oleh konsumen saat mendaftar. | Lily@gmail.com |
| password | String | M |  | Password yang diinput oleh konsumen saat mendaftar. | 123456aaaa1 |
| userId | Long | M |  | User Id yang diberikan system untuk konsumen. | 9 |

* 1. Seats
* **GET By Id**

**Dekripsi Endpoint**

Request GET BY ID digunakan untuk menampilkan data Seats sesuai dengan (id) yang diinputkan.

**End Point :** <http://localhost:8089/seats-controller/getseatsById>

**Request URL**

|  |
| --- |
| http://localhost:8089/swagger-ui/index.html#/seats-controller/getseatsById |

**Sample Request**

|  |
| --- |
| {  seatId : 3  } |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {    "data": 3,    "message": "Successfully retrieved data!",    "status": 200  } |

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Seats not Exist with id :500",      "status": 404  } |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seatId | Long | M |  | Nomor Identifikasi seat dari system. | 1 |
| seatNumber | Long | M |  | Nomor Seat di Studio. | 1 |
| studioName | String | M |  | Nama Studio. | A |
| isAvailable | int | M |  | Status tersedia atau tidak, | 1 |

* **PUT**

**Deksripsi Endpoint**

Request PUT digunakan untuk mengubah/memodifikasi data yang ada menjadi data yang baru

**End Point :** <http://localhost:8089/seats-controller/updateSeats>

**Request URL**

|  |
| --- |
| http://localhost:8089/swagger-ui/index.html#/seats-controller/updateSeats |

**Sample Request**

|  |
| --- |
| {  "seatId": 10,  "seatNumber": 3,  "studioName": "string",  "isAvailable": 0  } |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {    "data": {      "seatId": 10,      "seatNumber": 3,      "studioName": "string",      "isAvailable": 0    },    "message": "Success Update Seats",    "status": 200  } |

|  |
| --- |
| Failed Response |
| {      "data": "data not found",      "message": " Seats not Exist with id :500",      "status": 404  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seatId | Long | M |  | Seats Id kosumen yang ingin di update. | 1 |
| seatNumber | Long | M |  | Nomor Seats konsumen yang ingin di update. | 2 |
| studioName | String | M |  | Nama Studio konsumen yang ingin di update. | C |
| isAvailable | int | M |  | Status Seats konsumen yang ingin di update. | 0 |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seatId | Long | M |  | Seats Id baru yang telah diupdate. | 1 |
| seatNumber | Long | M |  | Nomor Seats yang telah diupdate. | 2 |
| studioName | String | M |  | Nama Studio yang telah diupdate. | C |
| isAvailable | int | M |  | Status Seats konsumen yang telah diupdate | 0 |

* **DELETE**

**Deksripsi Endpoint**

By Request DELETE digunakan untuk menghapus data Seats menggunakan parameter

**End Point :** <http://localhost:8083/seats-controller/deleteseats>

**Request URL**

|  |
| --- |
| http://localhost:8083/swagger-ui/index.html#/seats-controller/deleteseats |

**Sample Request**

|  |
| --- |
| {  seatId : 3  } |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {    "data": 3,    "message": "Deleted! ",    "status": 200  } |

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Seats not Exist with id :500",      "status": 404  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seatId | Long | M |  | Seat Id yang ingin di delete. | 52 |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| deleted | Boolean | M |  | Status perintah Delete. | true |

* **GET ALL**

**Deksripsi Endpoint**

Request GET digunakan untuk menampilkan semua list data entry di table seats

**End Point :** http://localhost:8083/seats-controller/getSeats

**Request URL**

|  |
| --- |
| http://localhost:8083/swagger-ui/index.html#/seats-controller/getSeats |

**Sample Request**

|  |
| --- |
|  |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {    "data": [      {        "seatId": 1,        "seatNumber": 1,        "studioName": "A",        "isAvailable": 1      },      ......      {        "seatId": 50,        "seatNumber": 10,        "studioName": "E",        "isAvailable": 1      }    ],    "message": "Succesfully Read All Data Seats !",    "status": 200  } |

|  |
| --- |
| Failed Response |
| {      "data": "Table has no value",      "message": "table seats have not value",      "status": 404  } |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seatId | Long | M |  | Nomor Identifikasi seat dari system. | 2 |
| seatNumber | Long | M |  | Nomor Seat di Studio. | 2 |
| studioName | String | M |  | Nama Studio. | A |
| isAvailable | int | M |  | Status tersedia atau tidak, | 1 |

* **POST**

**Deksripsi Endpoint**

By Request DELETE digunakan untuk menghapus data seats menggunakan parameter

**End Point :** <http://localhost:8083/seats-controller/createseats>

**Request URL**

|  |
| --- |
| http://localhost:8083/swagger-ui/index.html#/seats-controller/createseats |

**Sample Request**

|  |
| --- |
| {      "seatId": 2,      "seatNumber": 2,      "studioName": "A",      "isAvailable": 1  } |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {    "data": {      "seatId": 2,      "seatNumber": 2,      "studioName": "A",      "isAvailable": 1    },    "message": "Succesfully Add Data Seats !",    "status": 200  } |

|  |
| --- |
| Failed Response |
| {      "data": [],      "message": "Tabel has no value!",      "status": 404  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seatId | Long | M |  | Seats Id User name konsumen  baru yang ingin diinsert. | 52 |
| seatNumber | Long | M |  | Nomor Seats konsumen  baru yang ingin diinsert | 1 |
| studioName | String | M |  | Nama Studio konsumen  baru yang ingin diinsert | B |
| isAvailable | int | M |  | Status Seats konsumen  baru yang ingin diinsert | 1 |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seatId | Long | M |  | Seats Id User name konsumen  baru yang berhasil diinsert. | 52 |
| seatNumber | Long | M |  | Nomor Seats konsumen  baru yang berhasil diinsert | 1 |
| studioName | String | M |  | Nama Studio konsumen  baru yang berhasil diinsert | B |
| isAvailable | int | M |  | Status Seats konsumen  baru yang berhasil diinsert | 1 |

* **POST isAvailable**

**Deksripsi Endpoint**

Request POST isAvailable digunakan untuk membuat data seat yang baru dan masih tersedia (is\_available = 1)

**End Point :** <http://localhost:8083/swaggerui/index.html#/seatscontroller/findSeats>

**Request URL**

|  |
| --- |
| localhost:8083/teamC/v1/schedule |

**Sample Request**

|  |
| --- |
| {  "seatId": 51 ,  "seatNumber": 5,  "studioName": "B",  "isAvailable": 1  } |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {    "data": [      {        "seatId": 1,        "seatNumber": 1,        "studioName": "A",        "isAvailable": 1,        ....        {        "seatId": 50,        "seatNumber": 10,        "studioName": "E",        "isAvailable": 1      },    ],    "message": "Successfully seats is Available ! ",    "status": 200  } |

|  |
| --- |
| Failed Response |
|  |

**Request struktur**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| isAvailable | int | M |  | Menampilkan semua Status Seats konsumen | 1 |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seatId | Long | M |  | Nomor Identifikasi seat dari system. | 2 |
| seatNumber | Long | M |  | Nomor Seat di Studio. | 2 |
| studioName | String | M |  | Nama Studio. | A |
| isAvailable | int | M |  | Status tersedia atau tidak, | 1 |

* 1. Schedule
* **GET ALL**

**Deksripsi Endpoint**

By Request GET kita bisa menampilkan semua List Data yang ada

**End Point :** <http://localhost:8083/teamC/v1/schedule-controller/> ScheduleList

**Request URL**

|  |
| --- |
| localhost:8083/teamC/v1/schedule |

**Sample Request**

|  |
| --- |
| http://localhost:8083/swagger-ui/index.html#/schedule-controller-dto/ScheduleList\_1 |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {      "data": {          "scheduleId": 1,          "filmId": 1,          "seatId": 1,          "dateShow": "2022-06-19",          "showStart": "18:00:00",          "showEnd": "19:00:00",          "price": 35000      },      "message": "Successfully retrivied data!",      "status": 200  } |

|  |
| --- |
| Failed Response |
| {      "data": "Tabel has no Value",      "message": "Schedule not exist",      "status": 404  } |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| scheduleId | Integer | M |  | Nomor Identifikasi schedule dari system. | 1 |
| filmId | Integer | M |  | Nomor Identifikasi film dari system. | 1 |
| seatId | Integer | M |  | Nomor Identifikasi seat dari system. | 1 |
| dateShow | Date | M |  | Tanggal tayang. | 2022-06-19 |
| showStart | Time | M |  | Tanggal dan waktu film tayang. | 3:00:00 |
| showEnd | Time | M |  | Tanggal dan waktu film selesai. | 15:00:00 |
| price | Integer | M |  | Harga tiket. | 35000 |

* **PUT**

**Deksripsi Endpoint**

By Request PUT kita bisa mengubah/mengedit data yang ada menjadi data yang baru

**End Point :** <http://localhost:8083/swagger-ui/index.html#/schedule-> controller/updateSchedule

**Request URL**

|  |
| --- |
| localhost:8083/teamC/v1/schedule |

**Sample Request**

|  |
| --- |
| {  "scheduleId": 9,  "films": {  "filmId": 3  },  "seats": {  "seatId": 7  },  "dateShow": "2022-06-19",  "showStart": "00:00:00",  "showEnd": "00:00:00",  "price": 20000  } |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {      "data": {          "scheduleId": 9,          "filmId": 3,          "seatId": 7,          "dateShow": "2022-06-19",          "showStart": "00:00:00",          "showEnd": "00:00:00",          "price": 20000      },      "message": "Data Updated!",      "status": 201  } |

|  |
| --- |
| Failed Response |
| {    "data": "Bad Request",    "message": "Schedule not exist with id : 900",    "status": 400  } |

**Resquest Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| scheduleId | Integer | M |  | Nomor Identifikasi schedule dari system. | 1 |
| Films | Entity | M |  | Entity/Model dari Films |  |
| filmId | Integer | M |  | Nomor Identifikasi film dari system. | 1 |
| Seats | Entitiy |  |  | Entity/Model dari seats |  |
| seatId | Integer | M |  | Nomor Identifikasi seat dari system. | 1 |
| dateShow | Date | M |  | Tanggal tayang. | 2022-06-19 |
| showStart | Time | M |  | Tanggal dan waktu film tayang. | 3:00:00 |
| showEnd | Time | M |  | Tanggal dan waktu film selesai. | 15:00:00 |
| price | Integer | M |  | Harga tiket. | 35000 |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| scheduleId | Integer | M |  | Nomor Identifikasi schedule dari system. | 1 |
| filmId | Integer | M |  | Nomor Identifikasi film dari system. | 1 |
| seatId | Integer | M |  | Nomor Identifikasi seat dari system. | 1 |
| dateShow | Date | M |  | Tanggal tayang. | 2022-06-19 |
| showStart | Time | M |  | Tanggal dan waktu film tayang. | 3:00:00 |
| showEnd | Time | M |  | Tanggal dan waktu film selesai. | 15:00:00 |
| price | Integer | M |  | Harga tiket. | 35000 |

* **POST/CREATE**

**Deksripsi Endpoint**

By Request POST kita bisa buat data yang baru

**End Point :** <http://localhost:8083/swagger-ui/index.html#/schedule-> controller/createSchedule

**Request URL**

|  |
| --- |
| localhost:8083/teamC/v1/schedule |

**Sample Request**

|  |
| --- |
| {    "films": {      "filmId": 90    },    "seats": {      "seatId": 7    },    "dateShow": "2022-06-19",    "showStart": "00:00:00",    "showEnd": "00:00:00",    "price": 20000  } |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {      "data": {          "scheduleId": 64,          "filmId": 1,          "seatId": 7,          "dateShow": "2022-06-19",          "showStart": "00:00:00",          "showEnd": "00:00:00",          "price": 20000      },      "message": "Successfully added data!",      "status": 201  } |

|  |
| --- |
| Failed Response |
| {      "data": **null**,      "message": "could not execute statement; SQL [n/a]; constraint [null]; nested exception is org.hibernate.exception.ConstraintViolationException: could not execute statement",      "status": 400  } |

**Resquest Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| scheduleId | Integer | M |  | Nomor Identifikasi schedule dari system. | 1 |
| Films | Entity | M |  | Entity/Model dari Films |  |
| filmId | Integer | M |  | Nomor Identifikasi film dari system. | 1 |
| Seats | Entitiy |  |  | Entity/Model dari seats |  |
| seatId | Integer | M |  | Nomor Identifikasi seat dari system. | 1 |
| dateShow | Date | M |  | Tanggal tayang. | 2022-06-19 |
| showStart | Time | M |  | Tanggal dan waktu film tayang. | 3:00:00 |
| showEnd | Time | M |  | Tanggal dan waktu film selesai. | 15:00:00 |
| price | Integer | M |  | Harga tiket. | 35000 |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| scheduleId | Integer | M |  | Nomor Identifikasi schedule dari system. | 1 |
| filmId | Integer | M |  | Nomor Identifikasi film dari system. | 1 |
| seatId | Integer | M |  | Nomor Identifikasi seat dari system. | 1 |
| dateShow | Date | M |  | Tanggal tayang. | 2022-06-19 |
| showStart | Time | M |  | Tanggal dan waktu film tayang. | 3:00:00 |
| showEnd | Time | M |  | Tanggal dan waktu film selesai. | 15:00:00 |
| price | Integer | M |  | Harga tiket. | 35000 |

* **POST By Film Name**

**Deksripsi Endpoint**

By Request POST kita bisa buat data yang baru dengan hanya memasukkan nama film

**End Point :** <http://localhost:8083/swagger-ui/index.html#/schedule-> controller/findScheduleByFilmName

**Request URL**

|  |
| --- |
| localhost:8083/schedule/byfilmname |

**Sample Request**

|  |
| --- |
| {    "name": "Avatar 2"  } |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {    "data": [      {              "filmName": "Avatar 2",              "studioName": "A",              "price": 20000          }      ],      "message": "Success",      "status": 200  } |

|  |
| --- |
| Failed Response |
| {    "data": "Bad Request",    "message": "Schedule with film name Comic 8888 is not exist!!!!!",    "status": 400  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| name | String | M |  | Nama film. | 1 |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| filmName | String | M |  | Nama Film | Avatar 2 |
| studioName | String | M |  | Nama Studio | A |
| price | Integer | M |  | harga Film | 25000 |

* **POST By Film Name Like**

**Deksripsi Endpoint**

**End Point :** <http://localhost:8083/swagger-ui/index.html#/schedule-> controller/findScheduleByFilmNameLike

**Request URL**

|  |
| --- |
| localhost:8083/schedule/ByfilmnameLike |

**Sample Request**

|  |
| --- |
| {    "name": "Avatar"  } |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {    "data": [      {              "filmName": "Avatar 2",              "studioName": "A",              "price": 20000          }      ],      "message": "Success",      "status": 200  }: 200  } |

|  |
| --- |
| Failed Response |
| {    "data": "Data Not Found",    "message": "Schedule with film name Comichghb jb is not exist!!!!!",    "status": 404  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| name | String | M |  | Nama film. | 1 |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| filmName | String | M |  | Nama Film | Avatar 2 |
| studioName | String | M |  | Nama Studio | A |
| price | Integer | M |  | harga Film | 25000 |

* **GET By ID**

**Deksripsi Endpoint**

Request GET BY ID digunakan untuk menampilkan data Schedules sesuai dengan (id) yang diinputkan

**End Point :** <http://localhost:8083/swagger-ui/index.html#/schedule-> controller/getscheduleById

**Request URL**

|  |
| --- |
| localhost:8083/teamC/v1/schedule/33 |

**Sampel Request**

|  |
| --- |
| {  "scheduleId": 33  } |

**Sampel Response**

|  |
| --- |
| Success Response |
| {    "data": {      "scheduleId": 3,      "filmId": 3,      "seatId": 3,      "dateShow": "2022-06-19",      "showStart": "18:00:00",      "showEnd": "19:00:00",      "price": 35000    },    "message": "Successfully retrivied data!",    "status": 200  } |

|  |
| --- |
| Failed Response |
| {    "data": "Data Not Found!",    "message": "Schedule not exist with id : 1000",    "status": 404  } |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| scheduleId | int | M |  | Nomor Identifikasi schedule dari system. | 3 |
| filmId | int | M |  | Nomor Identifikasi film dari system. | 3 |
| name | String | M |  | Nama film. | Avatar 2 |
| isPlaying | Integer | M |  | Status sedang tayang atau tidak. | 0 |
| dateShow | Date | M |  | Tanggal tayang. | 2022-06-19T00:00:00.000+00:00 |
| seatId | Long | M |  | Nomor Identifikasi seat dari system. | 5 |
| seatNumber | Long | M |  | Nomor kursi. | 4 |
| studioName | String | M |  | Nama Studio. | E |
| isAvailable | int | M |  | Status kursi tersedia atau tidak. | 0 |
| showStart | Date | M |  | Tanggal dan waktu film tayang. | 2022-06-19T13:00:00.000+00:00 |
| showEnd | Date | M |  | Tanggal dan waktu film selesai. | 2022-06-19T15:00:00.000+00:00 |
| price | Integer | M |  | Harga tiket. | 35000 |

* **DELETE**

**Deksripsi Endpoint**

By Request DELETE digunakan untuk menghapus data Schedules menggunakan parameter

**End Point :** <http://localhost:8083/swagger-ui/index.html#/schedule-> controller/delete

**Request URL**

|  |
| --- |
| localhost:8083/schedule/15 |

**Sampel Request**

|  |
| --- |
| {  "scheduleId": 15  } |

**Sampel Response**

|  |
| --- |
| Success Response |
| {      "data": {          "deleted": **true**      },      "message": "Deleted! ",      "status": 200  } |

|  |
| --- |
| Failed Response |
| {  "data": "Data Not Found",      "message": "Schedule not exist with id :15",      "status": 404  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| scheduleId | Long | M |  | Seat Id yang ingin di delete. | 13 |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| deleted | Boolean | M |  | Status perintah Delete. | true |

* 1. Film
* **GET By Id**

**Deksripsi Endpoint**

Request GET BY ID digunakan untuk menampilkan data Film sesuai dengan Film (id) yang diinputkan

**End Point :** http://localhost:8083/teamC/v1/films-controller/getUserById\_1

**Request URL**

|  |
| --- |
| http://localhost:8083/swagger-ui/index.html#/schedule-controller-dto/ScheduleList\_1 |

**Sampel Request**

|  |
| --- |
| {  "filmId": 1  } |

**Sampel Response**

|  |
| --- |
| Success Response |
| {    "data": {      "filmId": 1,      "name": "Jurassic World Dominion",      "isPlaying": 1    },    "message": "Succes Get",    "status": 200  } |

|  |
| --- |
| Failed Response |
| {    "data": "Data not found",    "message": "Films with id 100 is not exist",    "status": 404  } |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| filmId | Long | M |  | Nomor Identifikasi film dari system. | 1 |
| name | String | M |  | Nama film. | Jurassic World Dominion |
| isPlaying | Integer | M |  | Film tersedia atau tidak. | 1 |

* **PUT**

**Deksripsi Endpoint**

Request PUT digunakan untuk mengubah/memodifikasi data yang ada menjadi data yang baru

**End Point :** http://localhost:8083/teamC/v1/films-controller/updateFilms

**Request URL**

|  |
| --- |
| http://localhost:8083/teamC/v1/DTO/films/65 |

**Sampel Request**

|  |
| --- |
| {  "title": "kamu",  "status": 0  } |

**Sampel Response**

|  |
| --- |
| Success Response |
| {      "data": {          "filmId": 65,          "name": "kamu",          "isPlaying": 0      },      "message": "Succes Update",      "status": 201  } |

|  |
| --- |
| Failed Response |
| {      "data": "Data not found",      "message": "Films with id 100 is can't to update",      "status": 404  } |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| filmId | Long | M |  | Film Id konsumen yang  Berhasi diupdate.. | 1 |
| name | String | M |  | Nama film konsumen yang  berhasil diupdate.. | kamu |
| isPlaying | Integer | M |  | Status tayang film yang  berhasil diupdate. | 0 |

* **DELETE**

**Deksripsi Endpoint**

By Request DELETE digunakan untuk menghapus data Film menggunakan parameter

**End Point :** http://localhost:8083/ films-controller/deleteFilms

**Request URL**

|  |
| --- |
| http://localhost:8083/swagger-ui/index.html#/films-controller-dto/deleteFilm |

**Sampel Request**

|  |
| --- |
| {  "filmId": 1  } |

**Sampel Response**

|  |
| --- |
| Success Response |
| {    "data": {      "deleted": **true**    },    "message": "Succes Delete",    "status": 200  } |

|  |
| --- |
| Failed Response |
| {      "data": "Failed delete data",      "message": "Films with id 100 is failed to delete",      "status": 404  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| filmId | Long | M |  | Film Id yang ingin di delete. | 20 |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| deleted | Boolean | M |  | Status perintah Delete. | true |

* **GET ALL**

**Deksripsi Endpoint**

By Request Get ALL kita bisa menampilkan semua List Data yang ada

**End Point :** http://localhost:8083/teamC/v1/films-controller/findAllFilms

**Request URL**

|  |
| --- |
| http://localhost:8083/swagger-ui/index.html#/films-controller-dto/findAllFilms\_1 |

**Sampel Request**

|  |
| --- |
|  |

**Sampel Response**

|  |
| --- |
| Success Response |
| {    "data": [      {        "filmId": 1,        "name": "Jurassic World Dominion",        "isPlaying": 1      },  {        "filmId": 66,        "name": "GERAKAN UJI NYALI",        "isPlaying": 1      }    ],    "message": "Succes All",    "status": 200 } |

|  |
| --- |
| Failed Response |
| {      "data": "Table has no value",      "message": "Table films has not value",      "status": 404  } |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| filmId | Long | M |  | Nomor Identifikasi film dari system. | 1 |
| name | String | M |  | Nama film. | kamu |
| isPlaying | Integer | M |  | Film tersedia atau tidak. | 0 |

* **POST**

**Desksripsi Endpoint**

By Request POST kita bisa buat data yang baru

**End Point :** http://localhost:8083/teamC/v1/films-controller/createFilm

**Request URL**

|  |
| --- |
| http://localhost:8083/swagger-ui/index.html#/films-controller-dto/createFilm\_1 |

**Sampel Request**

|  |
| --- |
| {    "name": "dzikir 24 hours",    "isPlaying": 1  } |

**Sampel Response**

|  |
| --- |
| Success Response |
| {    "data": {      "filmId": 69,      "name": "dzikir 24 hours",      "isPlaying": 0    },    "message": "Succes Create",    "status": 200  } |

|  |
| --- |
| Failed Response |
| {      "data": "Failed create data",      "message": "could not execute statement; SQL [n/a]; constraint [null]; nested exception is org.hibernate.exception.ConstraintViolationException: could not execute statement",      "status": 400  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| filmId | Long | M |  | Film Id konsumen yang  ingin di insert.. | 62 |
| name | String | M |  | Nama film konsumen yang  ingin di insert. | dzikir 24 hours |
| isPlaying | Integer | M |  | Status tayang film tersedia tau tidak yang ingin di insert. | 1 |

|  |
| --- |
| Response Body |
| {  "filmId": 62,  "name": "dzikir 24 hours",  "isPlaying": 1  } |

Code Description

200 OK

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| filmId | Long | M |  | Film Id konsumen yang  berhasil di insert. | 62 |
| name | String | M |  | Nama film konsumen yang  berhasil di insert. | dzikir 24 hours |
| isPlaying | Integer | M |  | Status tayang film tersedia  atau tidak yang berhasil di insert. | 1 |

* 1. Reservation
* **GET ALL**

**Deksripsi endpoint**

By Request GET kita bisa menampilkan semua list data entry di table Reservation

**End Point :** http://localhost:8083/teamC/v1/DTO/Reservation

**Request URL**

|  |
| --- |
| localhost:8083/teamC/v1/Reservation |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {      "data": [              "reservation\_id": 9,              "usr\_id": 1,              "usr\_name": "Wahyu Wijanarko",              "email\_usr": "WahyuW@gmail.com",              "film\_name": "Jurassic World Dominion",              "price": 35000,              "studio": "A",              "status\_show": 1,              "seat\_num": 1,              "status\_seat": 1,              "date\_film": "2022-06-19",              "start\_film": "18:00:00",              "end\_film": "19:00:00"          }      ],      "message": "Success Get All",      "status": 200  } |

|  |
| --- |
| Failed Response |
| {      "data": "Table has no value",      "message": "Data Reservation not exist",      "status": 404  } |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| reservation\_id | Integer | M |  | Nomor Identifikasi Reservation dari system. | 1 |
| usr\_id | Integer | M |  | Nomor Identifikasi user dari system. | 1 |
| usr\_name | String | M |  | Nama user | Budiman Piatu |
| email\_usr | Stirng | M |  | Email User | BudimanPiatu@gmail.com |
| film\_name | String | M |  | Nama film | Jurassic World Domination |
| price | Integer | M |  | Harga tiket film | 20000 |
| studio | String | M |  | Nama Studio | A |
| status\_show | Integer | M |  | Film tayang (1) atau tidak (0) | 1 |
| seat\_num | Integer | M |  | Nomor kursi | 69 |
| status\_seat | Integer | M |  | Ketersedian kursi (1) atau tidak (0) | 1 |
| date\_film | date | M |  | Tanggal Tayang Film | 2022-06-19 |
| start\_film | time | M |  | Waktu Mulai Film | 18:00:00 |
| end\_film | time | M |  | Waktu Selesai Film | 19:00:00 |

* **GET BY ID**

**Deksripsi Endpoint**

Request GET BY ID digunakan untuk menampilkan data Reservation sesuai dengan (id) yang diinputkan

**End Point :** http://localhost:8083/teamC/v1/DTO/Reservation/{id}

**Request URL**

|  |
| --- |
| localhost:8083/teamC/v1/Reservation/{id} |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {      "data": {          "reservation\_id": 4,          "usr\_id": 4,          "usr\_name": "Nur Aida",          "email\_usr": "Aida\_Nur@gmail.com",          "film\_name": "di bawah lindungan kabah",          "price": 35000,          "studio": "B",          "status\_show": 0,          "seat\_num": 2,          "status\_seat": 1,          "date\_film": "2022-08-08",          "start\_film": "00:10:00",          "end\_film": "00:15:00"      },      "message": "Success Get By Id",      "status": 200  } |

|  |
| --- |
| Failed Response |
| {      "data": "Data not found",      "message": "Reservation not exist with id 15",      "status": 404  } |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| reservation\_id | Integer | M |  | Nomor Identifikasi Reservation dari system. | 1 |
| usr\_id | Integer | M |  | Nomor Identifikasi user dari system. | 1 |
| usr\_name | String | M |  | Nama user | Budiman Piatu |
| email\_usr | Stirng | M |  | Email User | BudimanPiatu@gmail.com |
| film\_name | String | M |  | Nama film | Jurassic World Domination |
| price | Integer | M |  | Harga tiket film | 20000 |
| studio | String | M |  | Nama Studio | A |
| status\_show | Integer | M |  | Film tayang (1) atau tidak (0) | 1 |
| seat\_num | Integer | M |  | Nomor kursi | 69 |
| status\_seat | Integer | M |  | Ketersedian kursi (1) atau tidak (0) | 1 |
| date\_film | date | M |  | Tanggal Tayang Film | 2022-06-19 |
| start\_film | time | M |  | Waktu Mulai Film | 18:00:00 |
| end\_film | time | M |  | Waktu Selesai Film | 19:00:00 |

* **CREATE/POST**

**Dekskripsi endpoint**

By Request POST kita bisa buat data yang baru

**End Point :** http://localhost:8083/teamC/v1/DTO/Reservation

**Request URL**

|  |
| --- |
| localhost:8083/teamC/v1/DTO/Reservation |

**Sample Request**

|  |
| --- |
| {         "usr": {              "userId":15          },          "sch": {              "scheduleId": 1          }  } |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {      "data": {          "reservation\_id": 15,          "sch\_id": 1,          "usr\_id": 15      },      "message": "Success Create Reservation",      "status": 201  } |

|  |
| --- |
| Failed Response |
| {      "data": "Data not found",      "message": "Reservation not exist with id 15",      "status": 404  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| Usr | Entity/Model | M |  | Enitity.User |  |
| userId | Integer | M |  | Nomor Identifikasi user dari system. | 1 |
| Sch | Entity/Model | M |  | Entity Schedule |  |
| scheduleId | Stirng | M |  | Nomor Identifikasi schedule dari System | 1 |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| reservation\_id | Integer | M |  | Nomor Identifikasi Reservation dari system. | 11 |
| sch\_id | Integer | M |  | Nomor Identifikasi schedule dari system. | 12 |
| usr\_id | Integer | M |  | Nomor Identifikasi user dari system. | 13 |

* **UPDATE/PUT**

**Deksripsi Endpoint**

By Request PUT kita bisa mengubah/mengedit data yang ada menjadi data yang baru

**End Point :** http://localhost:8083/teamC/v1/DTO/booking/{id}

**Request URL**

|  |
| --- |
| localhost:8083/teamC/v1/DTO/Reservation/{id} |

**Sample Request**

|  |
| --- |
| {         "usr": {              "userId":15          },          "sch": {              "scheduleId": 1          }  } |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {      "data": {          "reservation\_id": 15,          "usr\_id": 4,          "usr\_name": "Nur Aida",          "email\_usr": "Aida\_Nur@gmail.com",          "film\_name": "Sonic the Hedgehog 2",          "price": 35000,          "studio": "A",          "status\_show": 1,          "seat\_num": 4,          "status\_seat": 1,          "date\_film": "2022-06-19",          "start\_film": "18:00:00",          "end\_film": "19:00:00"      },      "message": "Success Update Reservation",      "status": 201  } |

|  |
| --- |
| Failed Response |
| {      "data": "Bad Request",      "message": "Reservation not exist with id 200",      "status": 400  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| Usr | Entity/Model | M |  | Enitity.User |  |
| userId | Integer | M |  | Nomor Identifikasi user dari system. | 1 |
| Sch | Entity/Model | M |  | Entity Schedule |  |
| scheduleId | Stirng | M |  | Nomor Identifikasi schedule dari System | 1 |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| reservation\_id | Integer | M |  | Nomor Identifikasi Reservation dari system. | 1 |
| usr\_id | Integer | M |  | Nomor Identifikasi user dari system. | 1 |
| usr\_name | String | M |  | Nama user | Budiman Piatu |
| email\_usr | Stirng | M |  | Email User | BudimanPiatu@gmail.com |
| film\_name | String | M |  | Nama film | Jurassic World Domination |
| price | Integer | M |  | Harga tiket film | 20000 |
| studio | String | M |  | Nama Studio | A |
| status\_show | Integer | M |  | Film tayang (1) atau tidak (0) | 1 |
| seat\_num | Integer | M |  | Nomor kursi | 69 |
| status\_seat | Integer | M |  | Ketersedian kursi (1) atau tidak (0) | 1 |
| date\_film | date | M |  | Tanggal Tayang Film | 2022-06-19 |
| start\_film | time | M |  | Waktu Mulai Film | 18:00:00 |
| end\_film | time | M |  | Waktu Selesai Film | 19:00:00 |

* **DELETE**

**Deksripsi Endpoint**

By Request DELETE kita bisa delete data sesuai request

**End Point :** http://localhost:8083/teamC/v1/DTO/booking/{id}

**Request URL**

|  |
| --- |
| localhost:8083/teamC/v1/DTO/Reservation/{id} |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {      "data": **true**,      "message": "Success Delete Reservation by ID",      "status": 200  } |

|  |
| --- |
| Failed Response |
| {      "data": "Data not found",      "message": "Reservation not exist with id 100",      "status": 404  } |

* **POST/QUERY BY FILMNAME**

**Dekskripsi Endpoint**

By Request POST kita bisa buat data yang baru hanya dengan memasukkan nama film

**End Point :** http://localhost:8083/teamC/v1/DTO/booking/Filmname

**Request URL**

|  |
| --- |
| localhost:8083/teamC/v1/DTO/Reservation/Filmname |

**Sample Request**

|  |
| --- |
| {    "name": "avatar"  } |

**Sample Response Request**

|  |
| --- |
| Success Response |
| {      "data": [          {              "reservation\_id": 2,              "usr\_id": 6,              "usr\_name": "Lionel",              "email\_usr": "Lionel@gmail.com",              "film\_name": "Avatar 2",              "price": 35000,              "studio": "A",              "status\_show": 1,              "seat\_num": 3,              "status\_seat": 1,              "date\_film": "2022-06-19",              "start\_film": "18:00:00",              "end\_film": "19:00:00"          }      ],      "message": "Succes Query By Filmname",      "status": 200  } |

|  |
| --- |
| Failed Response |
| {      "data": "Failed Query By Filename",      "message": "Reservation not exist with Filmname cinta itu nyata",      "status": 400  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| name | String | M |  | Filmname | Avatar |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| reservation\_id | Integer | M |  | Nomor Identifikasi Reservation dari system. | 1 |
| usr\_id | Integer | M |  | Nomor Identifikasi user dari system. | 1 |
| usr\_name | String | M |  | Nama user | Budiman Piatu |
| email\_usr | Stirng | M |  | Email User | BudimanPiatu@gmail.com |
| film\_name | String | M |  | Nama film | Jurassic World Domination |
| price | Integer | M |  | Harga tiket film | 20000 |
| studio | String | M |  | Nama Studio | A |
| status\_show | Integer | M |  | Film tayang (1) atau tidak (0) | 1 |
| seat\_num | Integer | M |  | Nomor kursi | 69 |
| status\_seat | Integer | M |  | Ketersedian kursi (1) atau tidak (0) | 1 |
| date\_film | date | M |  | Tanggal Tayang Film | 2022-06-19 |
| start\_film | time | M |  | Waktu Mulai Film | 18:00:00 |
| end\_film | time | M |  | Waktu Selesai Film | 19:00:00 |