

CHALLENGE

TIM D - CHAPTER 6

BIOSKOP API

(SECURITY)

Contents

# Daftar Isi

[Daftar Isi 2](#_Toc109849273)

[CATATAN RELEASE DOKUMEN 4](#_Toc109849274)

[Version History 5](#_Toc109849275)

[FLOWCHART AND DIAGRAM ENTITY 6](#_Toc109849276)

[Project Flow 6](#_Toc109849277)

[A. Entity Relationship Diagram 7](#_Toc109849278)

[TABEL DAN DATA INSERT 8](#_Toc109849279)

[a) Users 8](#_Toc109849280)

[b) Films 8](#_Toc109849281)

[c) Schedules 8](#_Toc109849282)

[d) Seats 9](#_Toc109849283)

[e) Booking 9](#_Toc109849284)

[f) Role 9](#_Toc109849285)

[g) User\_roles 10](#_Toc109849286)

[1. INSERT DATA 10](#_Toc109849287)

[a) Insert data users table 10](#_Toc109849288)

[b) Insert data films table 10](#_Toc109849289)

[c) Insert data schedules table 11](#_Toc109849290)

[d) Insert data seats table 11](#_Toc109849291)

[e) Insert data booking table 11](#_Toc109849292)

[f) Insert data role 11](#_Toc109849293)

[g) Insert data users\_roles 11](#_Toc109849294)

[RUNNING API SECURITY 12](#_Toc109849295)

[1. Login 12](#_Toc109849296)

[2. USER 12](#_Toc109849297)

[a) Get By ID 13](#_Toc109849298)

[b) PUT 14](#_Toc109849299)

[c) Delete 15](#_Toc109849300)

[d) Post (sign up) 16](#_Toc109849301)

[e) Get All 17](#_Toc109849302)

[3. Seats 19](#_Toc109849303)

[a) Get By ID 19](#_Toc109849304)

[b) PUT 20](#_Toc109849305)

[c) Delete 21](#_Toc109849306)

[d) Get All 22](#_Toc109849307)

[e) Post 23](#_Toc109849308)

[f) Post is Available 24](#_Toc109849309)

[4. Schedule 25](#_Toc109849310)

[a) Get All 25](#_Toc109849311)

[b) Put Schedule 26](#_Toc109849312)

[c) Post/ Create Schedule 28](#_Toc109849313)

[d) Post/ Find Schedule By Film Name Like 30](#_Toc109849314)

[e) Get By Id 31](#_Toc109849315)

[g) Delete 32](#_Toc109849316)

[5. Film 33](#_Toc109849317)

[a) Get By Id 33](#_Toc109849318)

[b) Put 34](#_Toc109849319)

[c) Delete 35](#_Toc109849320)

[d) Get All 35](#_Toc109849321)

[e) Post 36](#_Toc109849322)

[f) Post Film is Playing 37](#_Toc109849323)

[6. Booking 39](#_Toc109849324)

[a) Get All 39](#_Toc109849325)

[b) Get By Id 40](#_Toc109849326)

[c) Create/ Post 41](#_Toc109849327)

[d) Update / Put 43](#_Toc109849328)

[e) Delete 44](#_Toc109849329)

[f) Post/ Query By FilmName 45](#_Toc109849330)

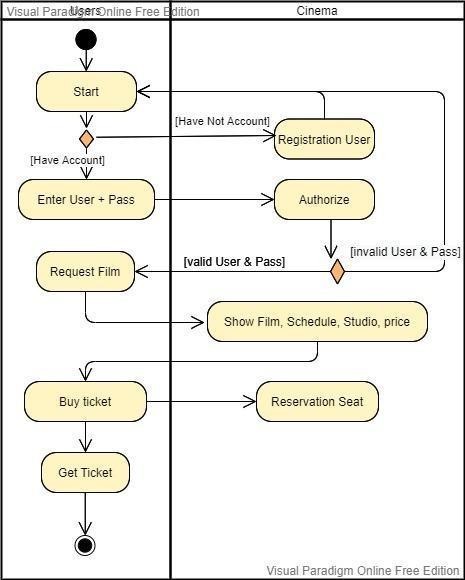
# CATATAN RELEASE DOKUMEN

|  |  |  |  |
| --- | --- | --- | --- |
| Tanggal Release | Versi | Oleh | Catatan Perubahan |
| 18 Juli 20221 | 1.0.0 | Tim D CH 6 | Versi Pertama |

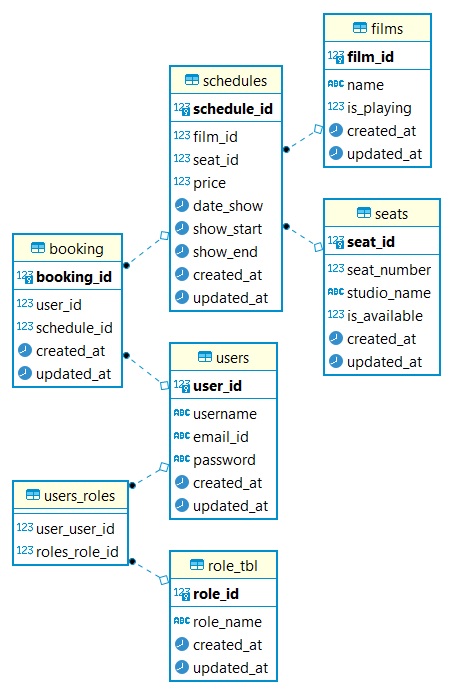
# Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Author | Description |
| 18/07/2022 | 1.0.0 | TIM D CH 6 | Upload first at Github, Setup New Trello, Add security Dependency |
| 19/07/2022 | 1.0.1 | TIM D CH 6 | Documentascion to doing, Build new project on bioskop Api,security add security dependency on bioskop Api |
| 27/07/2022 | 1.0.2 | TIM D CH 6 | Build new project to done, documentasi to done, deployment, jasperreports, check security acces, check endpoint, all final check, fitur login |

# FLOWCHART AND DIAGRAM ENTITY

Project Flow

1. Entity Relationship Diagram



# TABEL DAN DATA INSERT

1. Users

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Data Type | Length | Not Null | Auto Increment | Key |
| user\_id | int | - | v | - | Primary Key |
| email | varchar | 255 | - | - |  |
| name | varchar | 255 | - |  |  |
| password | varchar | 255 | - | - |  |
| username | varchar | 255 | - | - |  |

* Query

CREATE TABLE public.users(

user\_id int8 NOT NULL,

email varchar (255) NULL,

“name varchar” (255) NULL,

“password” varchar (255) NULL,

username varchar (255) NULL,

constrant users pkey PRIMARY KEY (user\_id)

);

1. Films

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Data Type | Length | Not Null | Auto Incremenet | Key |
| film\_id | int |  | v | - | Primary Key |
| is\_playing | tinyint | - | v | - |  |
| name | timestamp | - | v | - |  |

* Query

CREATE TABLE public.films(

film\_id bigserial NOT NULL,

is\_playing int4 not NULL,

name varrchar (255) NULL,,

CONSTRAINT films\_pkey PRIMARY KEY (film\_id)

);

1. Schedules

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Data Type | Length | Not Null | Auto Incremenet | Key |
| schedule\_id | int | - | v | - | Primary Key |
| date\_show | date | - | - | - |  |
| price | int | 4 | - | - |  |
| show\_end | time | - | - | - |  |
| show\_start | time | - | - | - |  |
| film\_id | int | 8 | - | - |  |
| Seat\_id | int | 8 | - | - |  |

* Query

CREATE TABLE public.schedules(

schedule\_id serial not null,

date\_show date NULL,

price int4 NULL,

show\_end time NULL,

show\_start time NULL,

film\_id int8 NULL,

seat\_id int8 NULL,

CONSTRAINT schedules\_pkey PRIMARY KEY (schedule\_id)

);

1. Seats

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Data Type | Length | Not Null | Auto Incremenet | Key |
| seat\_id | int | - | v | - | Primary Key |
| Is\_available | int | 4 | - | - | - |
| seat\_number | varchar | 8 | - | - | - |
| studio\_name | tinyint | 255 | - | - | - |

* Query

CREATE TABLE public.seats(

seat id bigserial NOT NULL,

is\_available int4 NULL,

seat\_number int8 NULL,

studio\_name varchar(255) NULL,

CONSTRAINT seats\_pkey PRIMARY KEY (seat\_id)

);

1. Booking

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Data Type | Length | Not Null | Auto Incremenet | Key |
| booking\_id | int | - | v | - | Primary Key |
| schedule\_id | int | 4 | - | - | - |
| user\_id | timestamp | 8 | - | - | - |

* Query

CREATE TABLE public.booking(

booking\_id bigserial NOT NULL,

schedule\_id int4 NULL

user\_id int8 NULL,

CONTRAINT booking\_pkey PRIMARY KEY (booking\_id)

);

1. Role

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Data Type | Length | Not Null | Auto Incremenet | Key |
| id | int | 8 | v | - | Primary Key |
| name | varchar | 255 | - | - | - |

* Query

CREATE TABLE public.”role” (

id int8 NOT NULL,

“name” varchar(255) NULL,

CONSTRAINT role\_pkey PRIMARY KEY (id),

);

1. User\_roles

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Column Name | Data Type | Length | Not Null | Auto Incremenet | Key |
| User\_user\_id | int | 8 | v | - | - |
| Roles\_id | int | 8 | v | - | - |

* Query

CREATE TABLE public.users\_roles (

user\_user\_id int8 NOT NULL,

roles\_id int8 NOT NULL

);

1. INSERT DATA
2. Insert data users table

INSERT INTO public.users (user\_id, email, “name”, “password”, username)

VALUES

(‘value’,’value’,’value,’value’,’value);

Example:

INSERT INTO public.users (user\_id, email, “name”,”password”, username)

VALUES

(1, 'bofihillie0@addtoany.com', 'Brice O''Fihillie', '$2a$06$qBO.idYe6n/AIqYgvOcrdOoFbKCeykRJnusfi31axiTEkfiUYJ3rC', 'bofihillie0');

1. Insert data films table

INSERT INTO public.films

(film\_id, is\_playing, “name”)

VALUES

(‘value’,’value’, ‘value’);

Example:

INSERT INTO public.films (film\_id, is\_playing, “name”)

VALUES

(1, 1, 'Jurassic World Dominion');

1. Insert data schedules table

INSERT INTO public.schedules (schedules\_id, date\_show, price, show\_end, show\_start, film\_id, seat\_id)

VALUES

(‘value’,‘value’,‘value’,’value,’value’,’value’,’value’);

Example:

INSERT INTO schedules (schedules\_id, date\_show, price, show\_end, show\_start, film\_id, seat\_id)

VALUES

(1, '2022-06-19', 35000, '19:00:00', '18:00:00', 1, 1);

1. Insert data seats table

INSERT INTO public.seats(seat\_id,is\_available, seat\_number, studio\_name)

VALUES

(‘value’,’value’,’value’,’value’);

Example:

INSERT INTO seats(seat\_id,is\_available, seat\_number, studio\_name)

VALUES

(1, 1, 1, 'A');

### Insert data booking table

INSERT INTO public.booking (booking\_id, schedule\_id, user\_id)

VALUES

(‘value’,’value’,’value’);

Example:

INSERT INTO booking (booking\_id, schedule\_id, user\_id)

VALUES

(1, 5, 6);

### Insert data role

INSERT INTO public.”role” (id, “name”)

VALUES

(‘value’,’value’);

Example:

INSERT INTO role (id, “name”)

VALUES

(1, 'ROLE\_USER');

### Insert data users\_roles

INSERT INTO public.users\_roles (user\_user\_id, roles)

VALUES

(‘value’,’value’);

Example:

INSERT INTO public.users\_roles(user\_user\_id, roles\_id)

VALUES

(1, 2);

RUNNING API SECURITY

Deskripsi : pada aplikasi bioskop ini previlage admin bisa untuk mengakses semua endpoint.

Privilege untuk user bisa untuk mengakses, mengedit dan menghapus akunnya sendiri dan paling utama untuk booking.

Privilege untuk user(customer) bisa untuk mengupdate user(put), menghapus user delete(delete) dan post.

Untuk endpoint film, schedule dan login bisa dilihat oleh semua orang baik oleh user(customer), admin, guest user (user yang belum terdaftar) dan endpoint user post juga bisa dilihat oleh semua orang untuk kebutuhan regristrasi.

1. Login

* Request url

http://localhost:8083/teamD/v1/login

Sample Request

|  |
| --- |
| {  “username” : “bofihillie0”,  “password” : “K1o4roL7eKI”  } |

Sample Response Request

|  |
| --- |
| Succes Response |
| {      "access\_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJib2ZpaGlsbGllMCIsInJvbGVzIjpbIlJPTEVfQURNSU4iXSwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdDo4MDgzL3RlYW1EL3YxL2xvZ2luIiwiZXhwIjoxNjU4NzYwNTgyfQ.GvCcvX2e5NbnVErrteeOe3XUhOjN0evju4odZHEJl6g",      "refresh\_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJib2ZpaGlsbGllMCIsImlzcyI6Imh0dHA6Ly9sb2NhbGhvc3Q6ODA4My90ZWFtRC92MS9sb2dpbiIsImV4cCI6MTY1ODc2MjM4Mn0.leM4Yz1bxfI\_AC7-7KgFwU-5Orl91Ip70x9ykt0IoRY"  } |

1. USER
2. Get By ID

* Request url

http://localhost:8083/teamD/v1/users/51

* Sample Request

|  |
| --- |
| {  “userId”:51,  } |

* Sample Response Request

|  |
| --- |
| Succes Response |
| {      "data": [          {              "ID             ": 51,              "Username       ": "anikja",              "Email          ": "anik@gmail.com",              "Password       ": "12345"          }      ],      "message": "Successfully Get User By ID!",      "status": 200  } |

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found!",      "message": "User not exist with id :51",      "status": 404  } |

* Response Strukture

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data type | Mandatory | Length | Description | Example |
| ID | String | M |  | Nomor ID | 51 |
| username | String | M |  | Username yang dimiliki oleh konsumen | anikja |
| emailId | String | M |  | Email yang digunakan oleh konsumen saat mendaftar | anik@gmail.com |
| password | String | M |  | Password yang diinput oleh konsumen saat mendaftar | 12345 |
| roles | ID | M |  | Id role | 2 |
| string | M |  | Role name | ROLE\_ADMIN |

1. PUT

* Request URL

http://localhost:8083/teamD/v1/users/51

* Sample Request

|  |
| --- |
| {  "userId": 51,  "name": "bella nita",  "username": "anikja",  "email": "anik@gmail.com",  "password": "12345"  } |

* Sample Response Request

|  |
| --- |
| Succes Response |
| {      "data": {          "userId": 51,          "name": "bella nita",          "username": "anikja",          "email": "anik@gmail.com",          "password": "12345",          "roles": []      },      "message": "Successfully Updated User!",      "status": 200  } |

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found!",      "message": "User not exist with id :51",      "status": 404  } |

* Request Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| userId | string | M |  | User Id | 51 |
| name | string | M |  | Nama usernama | Bella nita |
| username | String | M |  | User name baru konsumen yang ingin diupdate | anikja |
| email | String | M |  | Email baru konsumen yang ingin diupdate | anik@gmail.com |
| password | String | M |  | Password baru konsumen yang ingin diupdate | 12345 |

* Response Strukture

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| userId | string | M |  | User Id | 51 |
| name | string | M |  | Nama yang telah di update | bellanita |
| username | String | M |  | User name baru yang telah diupdate | anikja |
| emailId | String | M |  | Email baru yang telah diupdate | anik@gmail.com |
| password | String | M |  | Password baru konsumen yang telah diupdate | 12345 |
|  |  |  |  |  |  |

1. Delete

* Endpoint URL

http://localhost:8083/teamD/v1/users/51

* Sample Request

|  |
| --- |
| {  userId: 51  } |

* Sample Response Request

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

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found!",      "message": "User not exist with id :51",      "status": 404  } |

* Request Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| userid | Long | M |  | User Id yang ingin didelete | 51 |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| deleted | Boolean | M |  | Status perintah delete | 51 |

1. Post (sign up)

* Request URL:

http://localhost:8083/teamD/v1/user/save

* Sample Request

|  |
| --- |
| Sample Request |
| {  "userId": 51,  "name": "anik",  "username": "anikj",  "email": "anik@gmail.com",  "password": "1234"  } |

* Sample Response Request

|  |
| --- |
| Succes Response |
| {      "userId": 51,      "name": "anik",      "username": "anikj",      "email": "anik@gmail.com",      "password": "$2a$10$y9J658ytldPSoncb2wieGu05XlE8FyCPjdA6gNPGBZUtY83/Npcdi",      "roles": []  } |

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found!",      "message": "User not exist with id :51",      "status": 404  } |

* Request Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| userId | string | M |  | User Id konsumen baru yang ingin di insert | 51 |
| name | string | M |  | Nama yang ingin di insert | anik |
| username | String | M |  | User name konsumen yang ingin di insert | anikj |
| email | String | M |  | Email konsumen baru yang ingin diinsert | anik@gmail.com |
| password | String | M |  | Password konsumen baru yang ingin di insert | 1234 |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| userId | String | M |  | User Id konsumen baru yang berhasil di insert | 51 |
| name | String | M |  | Nama yang sudah di insert | anik |
| username | String | M |  | User name konsumen yang berhasil di insert | anikj |
| emailId | String | M |  | Email konsumen baru yang berhasil diinsert | anik@gmail.com |
| password | String | M |  | Password konsumen baru yang berhasil di insert | $2a$10$y9J658ytldPSoncb2wieGu05XlE8FyCPjdA6gNPGBZUtY83/Npcdi |

1. Get All

* Request URL:

http://localhost:8083/teamD/v1/users

* Sample Request

|  |
| --- |
|  |

* Sample Response Request

|  |
| --- |
| Succes Response |
| {              "userId": 49,              "name": "Zack Hunnisett",              "username": "zhunnisett1c",              "email": "zhunnisett1c@prnewswire.com",              "password": "$2a$06$1tIX0DKg2F2g6hKbQfN8OOytofUNFBwAI2TQyOccUkhF8t1aWWRDS",              "roles": []          },          {              "userId": 50,              "name": "Albertine Pavelka",              "username": "apavelka1d",              "email": "apavelka1d@miitbeian.gov.cn",              "password": "$2a$06$qEbDiG2vZIGRFSJ8ShVB0.0ERCUSZly8L2mupsrbaS5d2UB4URxxa",              "roles": []          }      ],      "message": "Successfully Get All User!",      "status": 200  } |
|  |

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found!",      "message": "User not exist with id :51",      "status": 404  } |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| userId | string | M |  | User Id yang diberikan system untuk konsumen | Albertina Pavelka |
| name | string | M |  | Nama user |  |
| username | String | M |  | User name yang di miliki oleh konsumen | apavelkaid |
| email | String | M |  | Email yang digunakan konsumen saat mendaftar | apavelkaid@mitbeian.gov.cn |
| password | String | M |  | Password yang di input oleh konsumen saat mendaftar | $2a$06$qEbDiG2vZIGRFSJ8ShVB0.0ERCUSZly8L2mupsrbaS5d2UB4URxxa |
| roles | ID | M |  | Id role | 2 |
| string | M |  | Role name | ROLE\_ADMIN |

* Endpoint Report PDF:

<http://localhost:8080/teamD/v1/reports/schedule>

1. Seats
2. Get By ID

* Request URL:

http://localhost:8083/teamD/v1/seats/1

* Sample Request

|  |
| --- |
| {  seatId:1  } |

* Sample Response Request

|  |
| --- |
| Succes Response Request |
| {      "data": {          "seatId": 1,          "seatNumber": 1,          "studioName": "A",          "isAvailable": 1      },      "message": "Successfully retrieved data!",      "status": 200  } |

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Seats not Exist with id :100",      "status": 404  } |

* Response Structure

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

1. PUT

* Request URL :

<http://localhost:8083/teamD/v1/seats/53>

* Sample Request

|  |
| --- |
| {      "seatId": 53,        "seatNumber": 20,          "studioName": "B",          "isAvailable": 1  } |

* Sample Response Request

|  |
| --- |
| Succes Response |
| {      "data": {          "seatId": 53,          "seatNumber": 20,          "studioName": "A",          "isAvailable": 1      },      "message": "Success Update Seats",      "status": 201  } |

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Seats not Exist with id :100",      "status": 404  } |

* Request Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seatId | Long | M |  | Seats Id yang ingin diupdate | 53 |
| seatNumber | Long | M |  | Nomor seats konsumen yang ingin diupdate | 1 |
| studioName | String | M |  | Nama Studio konsumen yang ingin di update. | B |
| isAvailable | int | M |  | Status seats konsumen yang ingin diupdate | 0 |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seatId | Long | M |  | Seats Id yang telah diupdate | 53 |
| seatNumber | Long | M |  | Nomor seats yang telah diupdate | 20 |
| studioName | String | M |  | Nama Studio yang telah di update. | A |
| isAvailable | int | M |  | Status seats konsumen yang telah diupdate | 1 |

1. Delete

* Endpoint URL:

http://localhost:8083/teamD/v1/seats/51

* Sample Request:

|  |
| --- |
| {  seatId:51  } |

* Sample Response Request

|  |
| --- |
| Succes Response |
| {      "data": {          "headers": {},          "body": {              "deleted": **true**          },          "statusCode": "OK",          "statusCodeValue": 200      },      "message": "Deleted! ",      "status": 200  } |

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Seats not Exist with id :51",      "status": 404  } |

* Request Strukture

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seatId | Long | M |  | Seat Id yang ingin di delete | 51 |

* Response Structure

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

1. Get All

* Request URL:

http://localhost:8083/teamD/v1/seats

* Sample Request

|  |
| --- |
|  |

* Sample Response Request

|  |
| --- |
| Succes Response |
| {              "seatId": 49,              "seatNumber": 9,              "studioName": "E",              "isAvailable": 1          },          {              "seatId": 50,              "seatNumber": 10,              "studioName": "E",              "isAvailable": 1          }      ],      "message": "Succesfully Read All Data Seats !",      "status": 200 |

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Seats not Exist with id :100",      "status": 404  } |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seatId | Long | M |  | Nomor Idenstifikasi seat dari system | 50 |
| seatNumber | Long | M |  | Nomor seats di studio | 10 |
| studioName | String | M |  | Nama Studio. | E |
| isAvailable | int | M |  | Status tersedia atau tidak | 1 |

1. Post

* Request URL:

http://localhost:8083/teamD/v1/seats

* Sample Request:

|  |
| --- |
| {      "seatId": 51,          "seatNumber": 20,          "studioName": "A",          "isAvailable": 1  } |

* Sample Response Request

|  |
| --- |
| Succes Response |
| {      "data": {          "seatId": 51,          "seatNumber": 20,          "studioName": "A",          "isAvailable": 1      },      "message": "Succesfully Add Data Seats !",      "status": 201  } |

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Seats not Exist with id :100",      "status": 404  } |

* Request Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seatId | Long | M |  | Seats Id User name konsumen baru yang ingin di insert | 51 |
| seatNumber | Long | M |  | Nomor seats konsumen baru yang ingin di insert | 20 |
| studioName | String | M |  | Nama Studio konsumen baru yang ingin di insert | A |
| isAvailable | int | M |  | Status Seats konsumen baru yang ingin di insert | 1 |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seatId | Long | M |  | Seats Id User name konsumen baru yang berhasil di insert | 51 |
| seatNumber | Long | M |  | Nomor Seats konsumen baru yang berhasil di insert | 20 |
| studioName | String | M |  | Nama Studio konsumen baru yang berhasil di insert | A |
| isAvailable | int | M |  | Status Seats konsumen baru yang berhasil di insert | 1 |

1. Post is Available

* Request URL

<http://localhost:8083/teamD/v1/seats/isAvailable>

* Sample Request

|  |
| --- |
| {      "seatId": 53,          "seatNumber": 20,          "studioName": "A",          "isAvailable": 1  } |

* Sample Response Request

|  |
| --- |
| Succes Response |
| {              "seatId": 52,              "seatNumber": 20,              "studioName": "A",              "isAvailable": 1          },          {              "seatId": 53,              "seatNumber": 20,              "studioName": "A",              "isAvailable": 1          }      ],      "message": "Successfully seats is Available ! ",      "status": 200  } |

* Failed Response

|  |
| --- |
| {      "data": "Data Not Found",      "message": " Seats not Exist with id :100",      "status": 404  } |

* Request Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seatId | Long | M |  | Nomor Identifikasi seat dari system | 53 |
| seatNumber | Long | M |  | Nomor Seats studio | 20 |
| studioName | String | M |  | Nama Studio | A |
| isAvailable | int | M |  | Status tersedia atau tidak | 1 |

* Response Strukture

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seatId | Long | M |  | Nomor Identifikasi seat dari system | 53 |
| seatNumber | Long | M |  | Nomor Seats studio | 20 |
| studioName | String | M |  | Nama Studio | A |
| isAvailable | int | M |  | Status tersedia atau tidak | 1 |

1. Schedule
2. Get All

* Request URL

http://localhost:8083/teamD/v1/schedule

* Sample Request

|  |
| --- |
|  |

* Sample Response Request

|  |
| --- |
| Succes Response |
| {              "scheduleId": 62,              "films": {                  "filmId": 2,                  "name": "The Roundup",                  "isPlaying": 1              },              "seats": {                  "seatId": 12,                  "seatNumber": 2,                  "studioName": "B",                  "isAvailable": 1              },              "dateShow": "2022-08-08",              "showStart": "00:05:00",              "showEnd": "00:07:00",              "price": 35000          }      ],      "message": "Successfully  getAll data!",      "status": 200  } |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| scheduleId | integer | M |  | Nomor Identifikasi schedule dari system | 62 |
| filmId | integer | M |  | Id film | 2 |
| name | integer | M |  | Nomor Identifikasi seat dari system | The Roundup” |
| isPlaying | integer | M |  | Yang sedang tayang atau tidak | 1 |
| seatsId | integer | M |  | id kursi | 12 |
| seatNumber | integer | M |  | Nomor kursi | 2 |
| studioName | time | M |  | Tanggal dan waktu film tayang | B |
| isAvailable | integer | M |  | Sedang tayang | 1 |
| dateShow | date | M |  | Tanggal tayang | 2022-08-08 |
| showStar | time | M |  | Tanggal dan waktu film selesai | 00:05:00 |
| ShowEnd | time | M |  | Tanggal dan waktu film selesai | 00:07:00 |
| price | integer | M |  | Harga Tiket | 35000 |

1. Put Schedule

* Request URL:

<http://localhost:8083/teamD/v1/schedule/3>

* Sample Request

|  |
| --- |
| {              "scheduleId": 3,              "films": {                  "filmId": 2                },              "seats": {                  "seatId": 7              },              "dateShow": "2022-06-19",              "showStart": "18:00:00",              "showEnd": "19:00:00",              "price": 35000          } |

* Sample Response Request

|  |
| --- |
| Succes Response |
| {      "data": {          "scheduleId": 3,          "filmId": 2,          "seatId": 7,          "dateShow": "2022-06-19",          "showStart": "18:00:00",          "showEnd": "19:00:00",          "price": 35000      },      "message": "Data Updated!",      "status": 201  } |

* Failed Response

|  |
| --- |
| Failed Structure |
| {      "data": "Data Not Found",      "message": " Schedule not Exist with id :100",      "status": 404  } |

* Request Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| scheduleId | Integer | M |  | Nomor Identifikasi schedule dari system | 3 |
| filmId | Integer | M |  | Nomor Identifikasi seat dari system | 2 |
| 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 | 3 |
| filmId | Integer | M |  | Nomor Identifikasi seat dari system | 2 |
| seatsId | Entity | M |  | Entity / Model dari seats | 7 |
| dateshow | Date | M |  | Tanggal Tayang | 2022-06-19 |
| showstart | Time | M |  | Tanggal dan waktu film tayang | 18:00:00 |
| showEnd | Time | M |  | Tanggal dan waktu film selesai | 19:00:00 |
| price | Integer | M |  | Harga Tiket | 35000 |

1. Post/ Create Schedule

* Request URL

http://localhost:8083/teamD/v1/schedule

* Sample Request

|  |
| --- |
| {            "scheduleId": 64,          "films": {              "filmId": 3          },          "seats": {              "seatId": 2          },          "dateShow": "2022-06-19",          "showStart": "18:00:00",          "showEnd": "19:00:00",         "price": 35000      } |

* Sample Response Request

|  |
| --- |
| Success Response |
| {      "data": {         "scheduleId": 64,         "filmId": 3,         "seatId": 2,         "dateShow": "2022-06-19",         "showStart": "18:00:00",          "showEnd": "19:00:00",          "price": 35000      },      "message": "Successfully added data!",      "status": 201  } |

* Failed Response

|  |
| --- |
| {      "data": "Data Not Found",      "message": " Schedule not Exist with id :100",      "status": 404  } |

* Request Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| scheduleId | Integer | M |  | Nomor Identifikasi schedule dari system | 64 |
| filmId | Integer | M |  | Nomor Identifikasi films dari system | 3 |
| seatId | Integer | M |  | Nomor Identifikasi seat dari system | 2 |
| dateshow | Date | M |  | Tanggal Tayang | 2022-06-19 |
| showstart | Time | M |  | Tanggal dan waktu film tayang | 18:00:00 |
| showEnd | Time | M |  | Tanggal dan waktu film selesai | 19:00:00 |
| price | Integer | M |  | Harga Tiket | 35000 |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| scheduleId | Integer | M |  | Nomor Identifikasi schedule dari system | 64 |
| filmId | Integer | M |  | Nomor Identifikasi seat dari system | 3 |
| seatId | Integer | M |  | Nomor Identifikasi seat dari system | 2 |
| 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 |

1. Post/ Find Schedule By Film Name Like

* Request URL

<http://localhost:8083/teamD/v1/schedule/byfilmnameLike>

* Sample Request

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

* Sample Response Request

|  |
| --- |
| Succes Response |
| {      "data": [          {              "filmName": "Avatar 2",              "studioName": "A",              "price": 35000          },          {              "filmName": "Avatar 2",              "studioName": "A",              "price": 35000          }      ],      "message": "Success",      "status": 201  } |

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Schedule not Exist with id :100",      "status": 404  } |

* Request Structure

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

* 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 | 355000 |

1. Get By Id

* Request URL

<http://localhost:8083/teamD/v1/schedule/1>

Sample Request

|  |
| --- |
| {  “scheduleId”: 1  } |

* Sample Response

|  |
| --- |
| Succes Response |
| {      "data": {          "scheduleId": 1,          "films": {              "filmId": 1,              "name": "Jurassic World Dominion",              "isPlaying": 1          },          "seats": {              "seatId": 1,              "seatNumber": 1,              "studioName": "A",              "isAvailable": 1          },          "dateShow": "2022-06-19",          "showStart": "18:00:00",          "showEnd": "19:00:00",          "price": 35000      },      "message": "Successfully retrivied data!",      "status": 200  } |

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Schedule not Exist with id :100",      "status": 404  } |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| scheduleId | int | M |  | Nomor Identifikasi schedule dari system | 1 |
| filmId | int | M |  | Nomor Identifikasi film dari system | 1 |
| name | string | M |  | Nama Film | “Jurassic World Dominion” |
| isPlaying | integer | M |  | Status sedang tayang atau tidak | 1 |
| seatId | long | M |  | Nomor Identifikasi seat dari system | 1 |
| seatNumber | long | M |  | Nomor Kursi | 1 |
| studioName | string | M |  | Nama Studio | A |
| isAvailable | int | M |  | Status Kursi tersedia atau tidak | 1 |
| dateShow | date | M |  | Tanggal Tayang | 2022-06-19 |
| showStart | date | M |  | waktu film tayang | 18:00:00 |
| showEnd | date | M |  | waktu film selesai | 19:00:00 |
| price | integer | M |  | Harga Tiket | 35000 |

1. Delete

* Request URL

<http://localhost:8083/teamD/v1/schedule/63>

* Sample Request

|  |
| --- |
| {  “schedule”:63  } |

* Sample Response

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

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Schedule not Exist with id :100",      "status": 404  } |

* Request Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Lenght | Description | Example |
| schedule | long | M |  | Seat Id yang ingin di delete | 63 |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Lenght | Description | Example |
| deleted | Boolean | M |  |  | true |

## Film

1. Get By Id

* Request URL:

http://localhost:8083/teamD/v1/films/1

* Sample Request

|  |
| --- |
| {  “filmId”:1  } |

* Sample Response

|  |
| --- |
| Succes Response |
| {      "data": {          "code": 1,          "title": "Jurassic World Dominion",          "status": 1      },      "message": "Succes Get",      "status": 200  } |

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Films not Exist with id :100",      "status": 404  } |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| code | long | M |  | Nomor kode film dari system | 1 |
| title | string | M |  | Judul film | Jurassic World Dominion |
| status | integer | M |  | Film tersedia atau tidak | 1 |

1. Put

* Request URL

<http://localhost:8083/swagger-ui/index.html#/films-controller-dto/filmupdate>

* Sample Request

|  |
| --- |
| {          "code": 62,          "title": "Jurassic",          "status": 1      } |

* Sample Response

|  |
| --- |
| Succes Response |
| {      "data": {          "code": 62,          "title": "Jurassic",          "status": 1      },      "message": "Succes Update",      "status": 201  } |

* Failed Strukture

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Films not Exist with id :100",      "status": 404  } |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| code | long | M |  | kode Id konsumen yang berhasil di update | 62 |
| title | string | M |  | Nama Film konsumen yang berhasil di update | Jurassic |
| status | integer | M |  | Status tayang film yang berhasil di update | 1 |

1. Delete

* Request URL

http://localhost:8083/teamD/v1/films/61

* Sample Request

|  |
| --- |
| {  “filmId”:61  } |

* Sample Response

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

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Films not Exist with id :100",      "status": 404  } |

* Request Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| filmId | Long | M |  | Film Id yang ingin di delete | 61 |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| deleted | Long | M |  | Status Perintah Delete | true |

1. Get All

* Request URL

<http://localhost:8083/teamD/v1/films>

* Sample Request

|  |
| --- |
|  |

* Sample Response

|  |
| --- |
| Succes Response |
| {              "code": 61,              "title": "Siap 86",              "status": 1          }      ],      "message": "Succes All",      "status": 200  } |

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Films not Exist with id :100",      "status": 404  } |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Typre | Mandatory | Length | Description | Example |
| code | long | M |  | Nomor code film dari system | 61 |
| film | string | M |  | Nama film | Siap 86 |
| status | int | M |  | Film tersedia atau tidak | 1 |

1. Post

* Request URL

http://localhost:8083/teamD/v1/films

* Sample Request

|  |
| --- |
| Sanple Request |
| {          "code": 62,          "title": "sunrise",          "status": 1      } |

* Sampel Response

|  |
| --- |
| Succes Response |
| {      "data": {          "code": 62,          "title": "sunrise",          "status": 1      },      "message": "Succes Create",      "status": 201  } |

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Films not Exist with id :100",      "status": 404  } |

* Request Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Typre | Mandatory | Length | Description | Example |
| code | long | M |  | Code Id konsumen yang ingin di insert | 62 |
| film | string | M |  | Nama film konsumen yang ingin di insert | sunrise |
| status | int | M |  | Ststus tayang film tersedia atau tidak yang ingin di insert | 1 |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Typre | 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 | sunrise |
| isPlaying | int | M |  | Status tayang film tersedia atau tidak yang berhasil di insert. | 1 |

1. Post Film is Playing

* Request URL:

http://localhost:8083/teamD/v1/films/1

* Sample Request

|  |
| --- |
|  |

* Sample Response

|  |
| --- |
| Succes Response |
| {              "code": 62,              "title": "Jurassic",              "status": 1          }      ],      "message": "Find Film Succes",      "status": 200  } |

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Films not Exist with id :100",      "status": 404  } |

* Request Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Typre | Mandatory | Length | Description | Example |
| code | long | M |  | kode Id konsumen yang ingin diinginkan | 0 |
| title | string | M |  | Nama judul film konsumen yang ingin dimainkan | 0 |
| status | int | M |  | Status tayang film tersedia atau tidak yang berhasil di insert. | 0 |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Typre | Mandatory | Length | Description | Example |
| code | long | M |  | kode Id konsumen yang ingin diinginkan | 62 |
| title | string | M |  | Nama judul film konsumen yang ingin dimainkan | Jurassic |
| status | int | M |  | Status tayang film tersedia atau tidak tayang | 1 |

1. Booking
2. Get All

* Request URL:

http://localhost:8083/teamD/v1/booking

* Sample Response Request

|  |
| --- |
| Succes Response |
| {              "book\_id": 9,              "usr\_id": 1,              "usr\_name": "bofihillie0",              "email\_usr": "bofihillie0@addtoany.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

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Booking not Exist with id :100",      "status": 404  } |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| book\_id | integer | M |  | Nomor Identifikasi booking dari system | 1 |
| usr\_id | integer | M |  | Nomor Identifikasi user dari system | 1 |
| usr\_name | string | M |  | Nama User | bofihillie0 |
| email\_usr | string | M |  | Email User | bofihillie0@addtoany.com |
| film\_name | string | M |  | Nama Film | Jurassic World Dominition |
| price | integer | M |  | Harga Tiket Film | 35000 |
| studio | string | M |  | Nama Studio | A |
| status\_show | integer | M |  | Film tayang (1) atau tidak(0) | 1 |
| seat\_num | integer | M |  | Nomor Kursi | 1 |
| status\_seat | integer | M |  | Ketersediaan 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 |

1. Get By Id

* Request URL

http://localhost:8083/teamD/v1/booking/1

* Sample Response Request

|  |
| --- |
| Succes Response |
| {      "data": {          "book\_id": 1,          "usr\_id": 6,          "usr\_name": "lknapman5",          "email\_usr": "lknapman5@creativecommons.org",          "film\_name": "The Batman",          "price": 35000,          "studio": "A",          "status\_show": 1,          "seat\_num": 5,          "status\_seat": 1,          "date\_film": "2022-06-19",          "start\_film": "18:00:00",          "end\_film": "19:00:00"      },      "message": "Success Get By Id",      "status": 200  } |

* Failed Response

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

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| book\_id | integer | M |  | Nomor Identifikasi booking dari system | 1 |
| usr\_id | integer | M |  | Nomor Identifikasi user dari system | 6 |
| usr\_name | string | M |  | Nama User | lknapman5 |
| email\_usr | string | M |  | Email User | lknapman5@creativecommons.org |
| film\_name | string | M |  | Nama Film | The Batman |
| price | integer | M |  | Harga Tiket Film | 35000 |
| studio | string | M |  | Nama Studio | A |
| status\_show | integer | M |  | Film tayang (1) atau tidak(0) | 1 |
| seat\_num | integer | M |  | Nomor Kursi | 5 |
| status\_seat | integer | M |  | Ketersediaan 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 |

1. Create/ Post

* Request URL

http://localhost:8083/teamD/v1/booking

* Sample Request for admin

|  |
| --- |
| {     "usr": {         "userId": 9     },     "sch": {         "scheduleId":1     }  } |

* Sample Request for user(customer).

|  |
| --- |
| {     "sch": {         "scheduleId":1     }  } |

* Sample Response Request

|  |
| --- |
| Succes Response |
| {      "data": {          "book\_id": 10,          "sch\_id": 1,          "usr\_id": 9      },      "message": "Success Create Booking",      "status": 201  } |

* Failed Response

|  |
| --- |
| Failed Response |
| {  “data”: “Data not found”,  “message”: “Booking not exist with id 15”,  “status”: 404  } |

* Request Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| usr | Entity/model | M |  | Entity user |  |
| userId | integer | M |  | Nomor identifikasi user dari system | 1 |
| sch | Entity/model | M |  | Entity schedule |  |
| scheduleId | string | M |  | Nomor identifikasi schedule | 1 |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| Book\_id |  | M |  | Nomor Identifikasi booking dari system | 11 |
| Sch\_id | integer | M |  | Nomor identifikasi schedule dari system | 12 |
| Usr\_id | integer | M |  | Nomor identifikasi user dari system | 13 |

1. Update / Put

* Request URL:

http://localhost:8083/teamD/v1/booking/8

* Sample Request

|  |
| --- |
| {     "usr": {         "userId": 9     },     "sch": {         "scheduleId":2     }  } |

* Sample Response Request

|  |
| --- |
| Succes Response |
| {      "data": {          "book\_id": 8,          "usr\_id": 9,          "usr\_name": "sarunowicz8",          "email\_usr": "sarunowicz8@engadget.com",          "film\_name": "The Roundup",          "price": 35000,          "studio": "A",          "status\_show": 1,          "seat\_num": 2,          "status\_seat": 1,          "date\_film": "2022-06-19",          "start\_film": "18:00:00",          "end\_film": "19:00:00"      },      "message": "Success Update Booking",      "status": 201  } |

* Failed Response

|  |
| --- |
| Failed Response |
| {  “data” Bad Request”  “message” Booking not exist with id 200”  “status” 400  } |

* Request Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| usr | Entity/model | M |  | Entity user |  |
| userId | integer | M |  | Nomor identifikasi user dari system | 1 |
| sch | Entity/model | M |  | Entity schedule |  |
| scheduleId | string | M |  | Nomor identifikasi schedule | 1 |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| book\_id | integer | M |  | Nomor Identifikasi booking dari system | 8 |
| usr\_id | integer | M |  | Nomor Identifikasi user dari system | 9 |
| usr\_name | string | M |  | Nama User | sarunowicz8 |
| email\_usr | string | M |  | Email User | sarunowicz8@engadget.com |
| film\_name | string | M |  | Nama Film | The Roundup |
| price | integer | M |  | Harga Tiket Film | 35000 |
| studio | string | M |  | Nama Studio | A |
| status\_show | integer | M |  | Film tayang (1) atau tidak(0) | 1 |
| seat\_num | integer | M |  | Nomor Kursi | 2 |
| status\_seat | integer | M |  | Ketersediaan 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 |

1. Delete

* Request URL:

<http://localhost:8083/teamD/v1/booking/9>

* Sample Request Response

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

* Failed Response

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Booking not Exist with id :9",      "status": 404  } |

* Request Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| bookingId | Long | M |  | Booking Id yang ingin di delete | 9 |

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| deleted | Long | M |  | Status Perintah Delete | true |

1. Post/ Query By FilmName

* Request URL:

<http://localhost:8083/teamD/v1/booking/Filmname>

* Sample Request

|  |
| --- |
| {  “name”: Avatar”  } |

* Sample Response Request

|  |
| --- |
| Succes Response |
| {      "data": [          {              "book\_id": 1,              "usr\_id": 6,              "usr\_name": "lknapman5",              "email\_usr": "lknapman5@creativecommons.org",              "film\_name": "The Batman",              "price": 35000,              "studio": "A",              "status\_show": 1,              "seat\_num": 5,              "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

|  |
| --- |
| Failed Response |
| {      "data": "Data Not Found",      "message": " Booking not Exist with id :100",      "status": 404  } |

* Request Strukture

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

* Response Structure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| book\_id | integer | M |  | Nomor Identifikasi booking dari system | 1 |
| usr\_id | integer | M |  | Nomor Identifikasi user dari system | 1 |
| usr\_name | string | M |  | Nama User | lknapman5 |
| email\_usr | string | M |  | Email User | lknapman5@creativecommons.org |
| film\_name | string | M |  | Nama Film | The Batman |
| price | integer | M |  | Harga Tiket Film | 35000 |
| studio | string | M |  | Nama Studio | A |
| status\_show | integer | M |  | Film tayang (1) atau tidak(0) | 1 |
| seat\_num | integer | M |  | Nomor Kursi | 5 |
| status\_seat | integer | M |  | Ketersediaan 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 |