

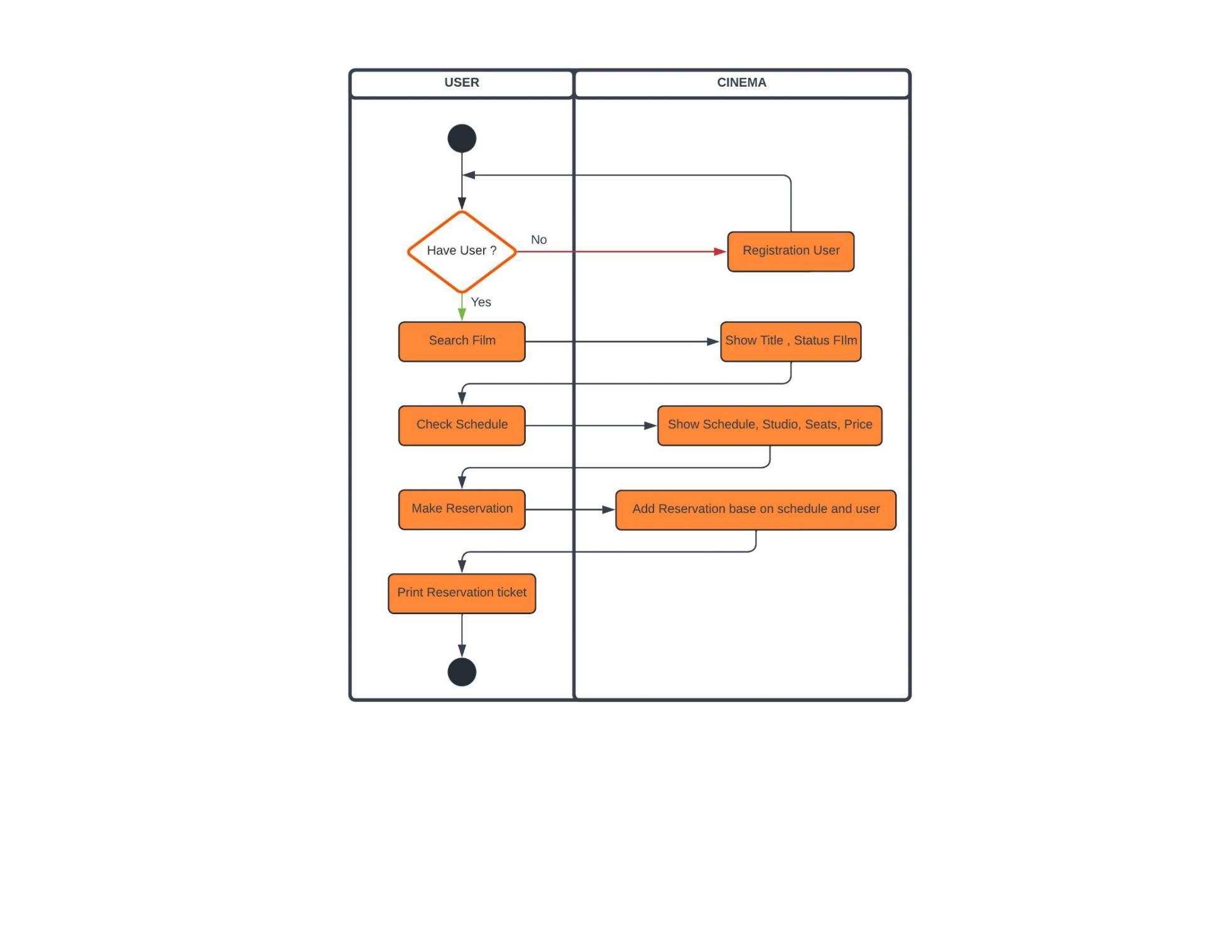
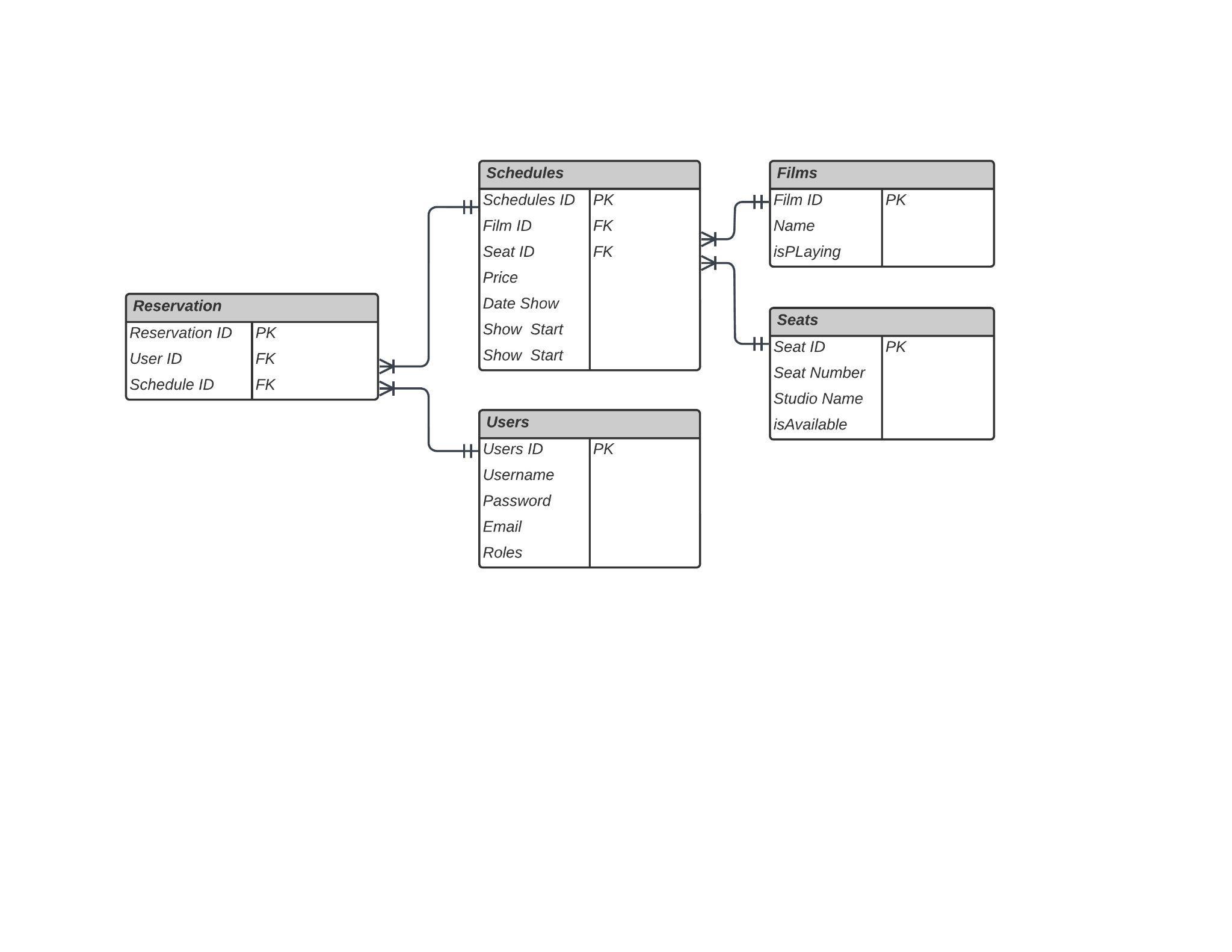
Dokumentasi App Bioskop

Team C – Chapter 6

(Database & Spring Security)

**Catatan Release Dokumen**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tanggal Release** | **Versi** | **Oleh** | **Catatan perubahan** |
| 21 juni 2022 | 1.0.0 | Team C – Chapter 6 | Versi Pertama |

1. **Activity Diagram**
2. **ERD Diagram**

**Query DDL**

1. Pembuatan table “Films”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Column name | Tipe data | Not null | key | default |
| film\_id | int | v | Primary Key |  |
| name | Varchar(100) | v |  |  |
| Is\_playing | bool | v |  | false |
| 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 | bool | v |  | false |
| 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 | Unique Key |  |
| Email\_id | Varchar(225) | v | Unique Key |  |
| password | Varchar(225) | v |  |  |
| roles | Varchar(225) |  |  | ‘ROLE\_USER’::character varying |
| 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',true),  ('The Roundup',true),  ('Avatar 2',true),  ('Sonic the Hedgehog 2',true),  ('The Batman',true),  ('KKN di Desa Penari',true) |  |

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. Security
3. Authentification (Method Post)

**Deksripsi Endpoint**

By Request POST kita masukkan username - password

untuk mendapatkan token

**Request URL**

|  |
| --- |
| localhost:8090/login |

**Sample URL Request**

|  |
| --- |
| localhost:8090/login |

**Sample Request**

|  |  |
| --- | --- |
| Key | Value |
| username | Nur Aida |
| password | 123456aaaa |

**Sample Response**

|  |
| --- |
| Success Response |
| {  "access\_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJOdXIgQWlkYSIsInJvbGVzIjpbIlJPTEVfQURNSU4iXSwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdDo4MDkwL2xvZ2luIiwiZXhwIjoxNjU4NDkwNDE3fQ.6Znj3OkhcWPelFAPpFNNZkbLwgPhNaiEsC9JDd\_Et30",  "refresh\_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJOdXIgQWlkYSIsImlzcyI6Imh0dHA6Ly9sb2NhbGhvc3Q6ODA5MC9sb2dpbiIsImV4cCI6MTY1ODQ5MjIxN30.wr\_3SEiiYdGoXHyN2qVcOehjH13BSGqhFmr-NQTN22U"  } |

|  |
| --- |
| Failed Response |
| Error: Exceeded maxRedirects. Probably stuck in a redirect loop http://localhost:8090/login |

**Request Structure**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Description | Example |
| username | String | username untuk authentication user | Nur Aida |
| password | String | password untuk authentication user | 123456aaaa |

**Response Structure**

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Data Type | Algorithma | Description |
| Access token | JWT | HMAC256 | token jwt untuk anthorization |
| Refreh token | JWT | HMAC256 | token jwt untuk anthorization |

1. Authorization

**Deksripsi Endpoint**

Authorization method http

**Request URL**

|  |
| --- |
| localhost:8090/\*\* |

**Sample URL Request**

|  |
| --- |
| localhost:8090/team3/v2/Seats |

**Authorization**

|  |  |
| --- | --- |
| Type | Token |
| Barrier Token | eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJOdXIgQWlkYSIsInJvbGVzIjpbIlJPTEVfQURNSU4iXSwiaXNzIjoiaHR0cDovL2xvY2FsaG9zdDo4MDkwL2xvZ2luIiwiZXhwIjoxNjU4NDkwNDE3fQ.6Znj3OkhcWPelFAPpFNNZkbLwgPhNaiEsC9JDd\_Et30 |

1. Resources
2. Schedules

* **Get All**

**Deksripsi Endpoint**

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

**Request URL**

|  |
| --- |
| localhost:8090/team3/schedules |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/schedules |

**Sample Response**

|  |
| --- |
| Success Response |
| {      "data": [          {              "scheduleId": 1,              "films": {                  "filmId": 1,                  "filmName": "Jurassic World Dominion",                  "isPlaying": **true**              },              "seats": {                  "seatId": 1,                  "seatNumber": 2,                  "studioName": "A",                  "isAvailable": **true**,                  "createdAt": "2022-07-21T17:21:42.61831",                  "updatedAt": "2022-07-21T17:21:42.61831"              },              "dateShow": "2022-06-19",              "showStart": "18:00:00",              "showEnd": "19:00:00",              "price": 35000,              "updated\_at": "2022-07-21T17:21:42.637057",              "created\_at": "2022-07-21T17:21:42.637057"          }  } |

|  |
| --- |
| 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 |
| studioName | String | M |  | Nama Studio | A |
| 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 |

* + - **GET By ID**

**Deksripsi Endpoint**

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

**Request URL**

|  |
| --- |
| localhost:8090/team3/schedules{id} |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/teamC/schedules/6 |

**Sample Response**

|  |
| --- |
| Success Response |
| {  "data": {  "scheduleId": 9,  "films": {  "filmId": 9,  "filmName": "The Lost City",  "isPlaying": true  },  "seats": {  "seatId": 9,  "seatNumber": 9,  "studioName": "A",  "isAvailable": false,  "createdAt": "2022-07-21T17:21:42.61831",  "updatedAt": "2022-07-21T17:21:42.61831"  },  "dateShow": "2022-06-19",  "showStart": "18:00:00",  "showEnd": "19:00:00",  "price": 35000,  "updated\_at": "2022-07-21T17:21:42.637057",  "created\_at": "2022-07-21T17:21:42.637057"  },  "message": "Successfully retrivied data!",  "status": 200  } |

|  |
| --- |
| Failed Response |
| {  "data": "Data Not Found!",  "message": "Schedules not exist with id : 99",  "status": 404  } |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| scheduleId | Integer | M |  | Nomor Identifikasi schedule dari system. | 9 |
| filmId | Integer | M |  | Nomor Identifikasi film dari system. | 9 |
| seatId | Integer | M |  | Nomor Identifikasi seat dari system. | 9 |
| studioName | String | M |  | Nama Studio | A |
| 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

**Request URL**

|  |
| --- |
| localhost:8090/team3/dashboard/create/schedules |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/dashboard/create/schedules |

**Sample Request**

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

**Sample Response**

|  |
| --- |
| Success Response |
| {      "data": {          "scheduleId": 64,          "filmId": 3,          "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 |
| Films | Entity | M |  | Entity/Model dari Films |  |
| filmId | Integer | M |  | Nomor Identifikasi film dari system. | 3 |
| Seats | Entitiy |  |  | Entity/Model dari seats |  |
| seatId | Integer | M |  | Nomor Identifikasi seat dari system. | 7 |
| 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. | 64 |
| filmId | Integer | M |  | Nomor Identifikasi film dari system. | 3 |
| seatId | Integer | M |  | Nomor Identifikasi seat dari system. | 7 |
| 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

**Request URL**

|  |
| --- |
| localhost:8090/ team3/dashboard/update/schedules{id} |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/dashboard/update/schedules/1 |

**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**

|  |
| --- |
| 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. | 9 |
| filmId | Integer | M |  | Nomor Identifikasi film dari system. | 3 |
| seatId | Integer | M |  | Nomor Identifikasi seat dari system. | 7 |
| dateShow | Date | M |  | Tanggal tayang. | 2022-06-19 |
| showStart | Time | M |  | Tanggal dan waktu film tayang. | 00:00:00 |
| showEnd | Time | M |  | Tanggal dan waktu film selesai. | 00:00:00 |
| price | Integer | M |  | Harga tiket. | 20000 |

* **POST (Search) By Film Name (Like Operator)**

**Deksripsi Endpoint**

By Request POST kita dapat memasukkan nama film yang ingin di cari untuk menampilkan data film tersebut.

**Request URL**

|  |
| --- |
| localhost:8090/team3/search/schedules/byfilmnameLike |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/search/schedules/byfilmnameLike |

**Sample Request**

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

**Sample Response**

|  |
| --- |
| 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 | 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 | 25000 |

* **DELETE**

**Deksripsi Endpoint**

By Request DELETE digunakan untuk menghapus data Schedules menggunakan parameter

**Request URL**

|  |
| --- |
| localhost:8090/ team3/dashboard/delete/schedules{id} |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/dashboard/delete/schedules/15 |

**Sample Request**

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

**Sample 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. | 15 |

**Response Structure**

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

* **Get Schedule Reports**

**Deksripsi Endpoint**

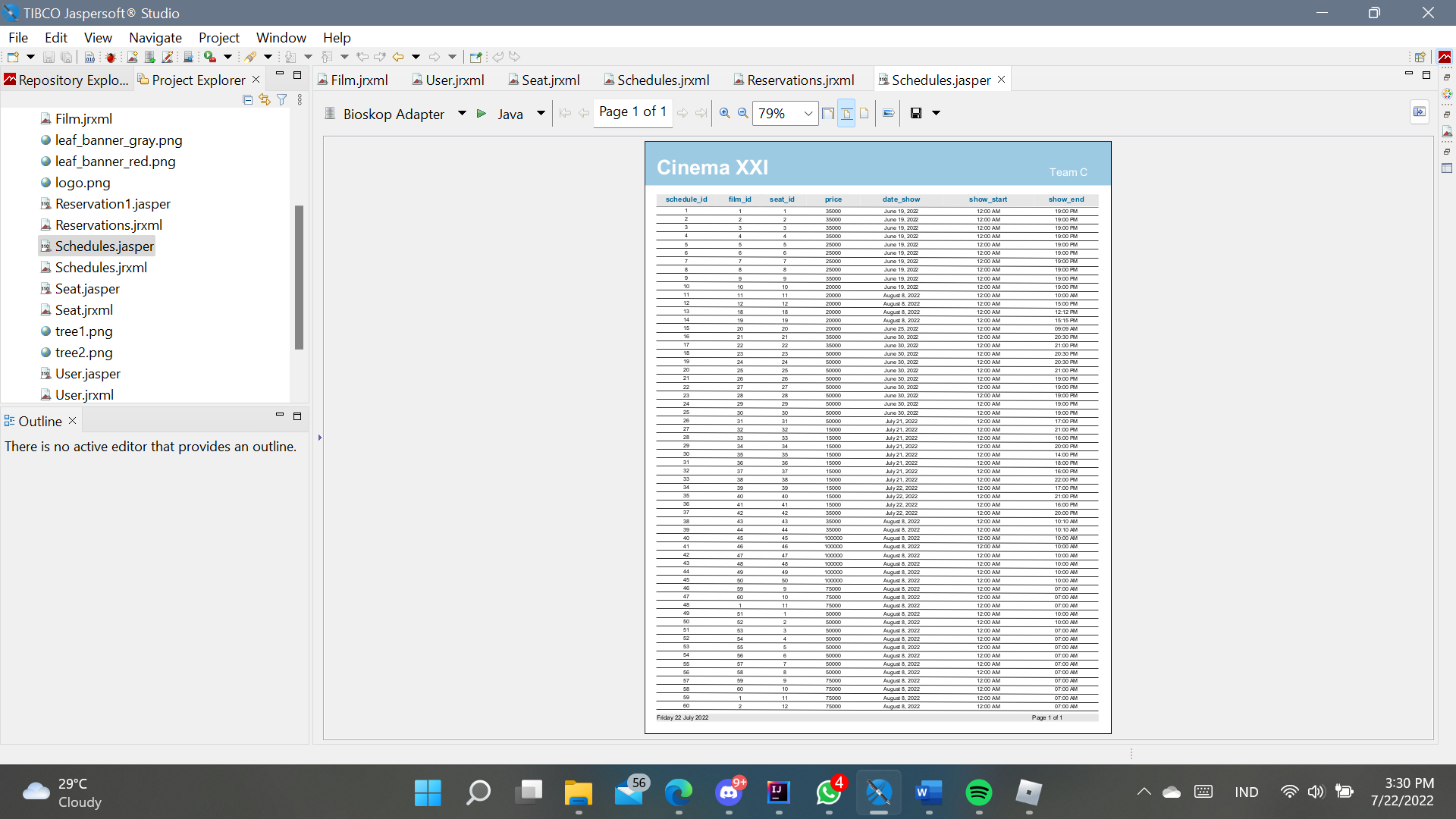
By Request GET kita dapat menggenerate Endpoint ke PDF menggunakan Jasper

**Request URL**

|  |
| --- |
| localhost:8090/team3/print/schedules |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/print/schedules |

**Get Schedules print to PDF with jasper report**

1. Seats

* **Get All**

**Deksripsi Endpoint**

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

**Request URL**

|  |
| --- |
| localhost:8090/team3/v2/Seats |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/seats |

**Sample Response**

|  |
| --- |
| Success Response |
| {  "data": [  "seat\_id": 100,  "number\_seat": 2,  "name\_studio": "E",  "available": **true**  }  ],  "message": "Success Get All",  "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 |
| seat\_Id | Integer | M |  | Nomor Identifikasi seat dari system. | 1 |
| number\_seat | Integer | M |  | Nomor kursi | 1 |
| name\_studio | String | M |  | Nama Studio | A |
| available | Boolean |  |  | Ketersediaan kursi | true |

* + - **GET BY ID**

**Deksripsi Endpoint**

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

**Request URL**

|  |
| --- |
| localhost:8090/team3/seats/{id} |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/seats/3 |

**Sample Response**

|  |
| --- |
| Success Response |
| {  "data": {  "seat\_id": 3,  "number\_seat": 3,  "name\_studio": "A",  "available": true  },  "message": "Succes Get By id",  "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 |
| seat\_Id | Integer | M |  | Nomor Identifikasi seat dari system. | 3 |
| number\_seat | Integer | M |  | Nomor kursi | 3 |
| name\_studio | String | M |  | Nama Studio | A |
| available | Boolean |  |  | Ketersediaan kursi | true |

* **POST/CREATE**

**Deksripsi Endpoint**

By Request POST kita bisa buat data yang baru

**Request URL**

|  |
| --- |
| localhost:8090/team3/dashboard/create/seats |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/dashboard/create/seats |

**Sample Request**

|  |
| --- |
| {  "number\_seat": 3,  "name\_studio": "A",  "available": **true**  } |

**Sample Response**

|  |
| --- |
| Success Response |
| {  "data": {  "seat\_id": 101,  "number\_seat": 3,  "name\_studio": "A",  "available": **true**  },  "message": "Success Create Seat",  "status": 201  } |

|  |
| --- |
| Failed Response |
| {      "data": Failed Create Seat,      "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 |
| number\_seat | Integer | M |  | Nomor kursi | 1 |
| name\_studio | String | M |  | Nama Studio | A |
| available | Boolean |  |  | Ketersediaan kursi | true |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seat\_Id | Integer | M |  | Nomor Identifikasi seat dari system. | 1 |
| number\_seat | Integer | M |  | Nomor kursi | 1 |
| name\_studio | String | M |  | Nama Studio | A |
| available | Boolean |  |  | Ketersediaan kursi | true |

* **PUT**

**Deksripsi Endpoint**

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

**Request URL**

|  |
| --- |
| localhost:8090/ team3/dashboard/update/seats /{id} |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/dashboard/update/seats/101 |

**Sample Request**

|  |
| --- |
| {  "number\_seat": 3,  "name\_studio": "A",  "available": **true**  } |

**Sample Response**

|  |
| --- |
| Success Response |
| {  "data": {  "seat\_id": 101,  "number\_seat": 3,  "name\_studio": "A",  "available": **true**  },  "message": "Success Create Seat",  "status": 201  } |

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

**Resquest Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| number\_seat | Integer | M |  | Nomor kursi | 1 |
| name\_studio | String | M |  | Nama Studio | A |
| available | Boolean |  |  | Ketersediaan kursi | true |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seat\_Id | Integer | M |  | Nomor Identifikasi seat dari system. | 1 |
| number\_seat | Integer | M |  | Nomor kursi | 1 |
| name\_studio | String | M |  | Nama Studio | A |
| available | Boolean |  |  | Ketersediaan kursi | true |

* **POST (Search) By isAvailable**

**Deksripsi Endpoint**

By Request POST kita dapat memasukkan tipe data yang ingin di cari untuk menampilkan tipe data tersebut.

**Request URL**

|  |
| --- |
| localhost:8090/ team3/search/seats/isavailable |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/search/seats/isavailable |

**Sample Request**

|  |
| --- |
| {  "available": **true**  } |

**Sample Response**

|  |
| --- |
| Success Response |
| {  "data": [  {  "seat\_id": 100,  "number\_seat": 2,  "name\_studio": "E",  "available": **true**  },  {  "seat\_id": 3,  "number\_seat": 12,  "name\_studio": "A",  "available": **true**  }  ],  "message": "Success Query",  "status": 200  } |

|  |
| --- |
| 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  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| available | Boolean | M |  | Ketersediaan kursi | true |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| seat\_Id | Integer | M |  | Nomor Identifikasi seat dari system. | 1 |
| number\_seat | Integer | M |  | Nomor kursi | 1 |
| name\_studio | String | M |  | Nama Studio | A |
| available | Boolean |  |  | Ketersediaan kursi | true |

* **DELETE**

**Deksripsi Endpoint**

By Request DELETE digunakan untuk menghapus data Schedules menggunakan parameter

**Request URL**

|  |
| --- |
| localhost:8090/ team3/dashboard/delete/seats /{id} |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/dashboard/delete/seats/51 |

**Sample Response**

|  |
| --- |
| Success Response |
| {  "data": **null**,  "message": "Success Delete",  "status": 200  } |

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

* **Get Seats Reports**

**Deksripsi Endpoint**

By Request GET kita dapat menggenerate Endpoint ke PDF menggunakan Jasper

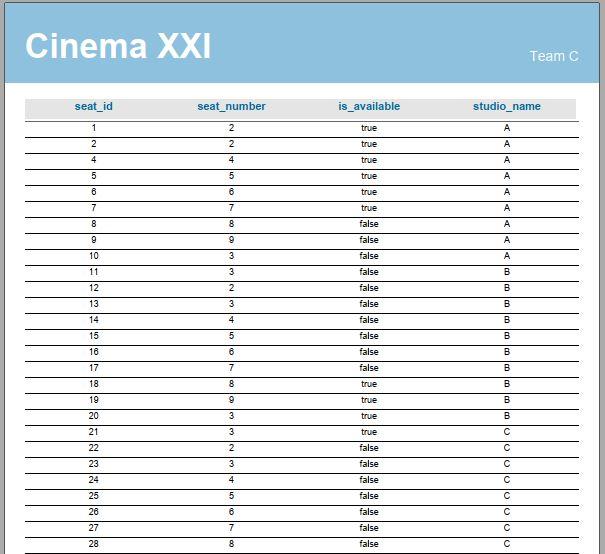
**Request URL**

|  |
| --- |
| localhost:8090/team3/print/seat |

**Sample URL Request**

|  |
| --- |
| http://localhost:8090/team3/print/seat |

**Get Seats print to PDF with jasper report**

****

1. User

* **Get All**

**Deksripsi Endpoint**

By Request GET ALL kita bisa menampilkan semua List Data User yang ada

**Request URL**

|  |
| --- |
| localhost:8090/team3/users |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/users |

**Sample Response**

|  |
| --- |
| 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 |

* + - **GET BY ID**

**Deksripsi Endpoint**

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

**Request URL**

|  |
| --- |
| localhost:8090/team3/users/{id} |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/users/3 |

**Sample Response**

|  |
| --- |
| Success Response |
| {  "data": {  "user\_id": 3,  "user\_name": "Budi Setiawan",  "email\_id": "BudiW@gmail.com"  },  "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 |
| userId | Long | M |  | User Id yang diberikan system untuk konsumen. | 3 |
| username | String | M |  | User name yang dimiliki oleh konsumen. | Budi Setiawan |
| emailId | String | M |  | Email yang digunakan oleh konsumen saat mendaftar. | Budi@gmail.com |

* **POST/CREATE**

**Deksripsi Endpoint**

By Request POST kita bisa buat data user yang baru

**Request URL**

|  |
| --- |
| localhost:8090/ team3/dashboard/sign-up |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/dashboard/sign-up |

**Sample Request**

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

**Sample Response**

|  |
| --- |
| 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  } |

**Resquest 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 |

* **PUT**

**Deksripsi Endpoint**

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

**Request URL**

|  |
| --- |
| localhost:8083/teamC/ team3/update/users |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/update/users |

**Sample Request**

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

**Sample Response**

|  |
| --- |
| 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  } |

**Resquest 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 |

* **POST (Search) By User Name (Like Operator)**

**Deksripsi Endpoint**

**Request URL**

|  |
| --- |
| localhost:8090/team3/dashboard/update/users /{users\_Id} |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/dashboard/update/users/8 |

**Sample Request**

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

**Sample Response**

|  |
| --- |
| 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 |

**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 |

* **DELETE**

**Deksripsi Endpoint**

By Request DELETE digunakan untuk menghapus data User menggunakan parameter

**Request URL**

|  |
| --- |
| localhost:8090/team3/dashboard/delete/users/{users\_Id} |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/dashboard/delete/users/8 |

**Sample Request**

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

**Sample Response**

|  |
| --- |
| 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 |

* **Get User Reports**

**Deksripsi Endpoint**

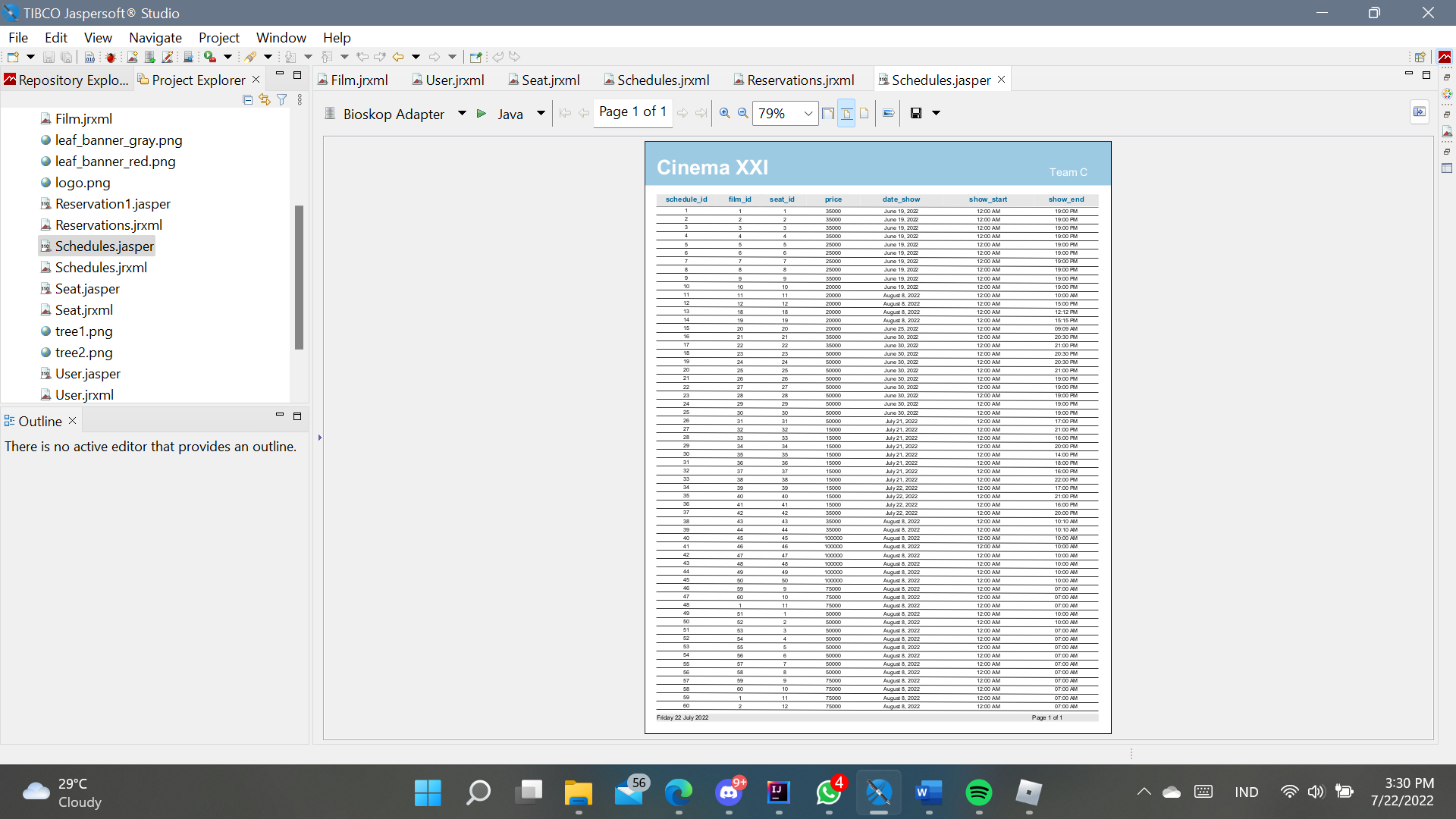
By Request GET kita dapat menggenerate Endpoint ke PDF menggunakan Jasper

**Request URL**

|  |
| --- |
| localhost:8090/team3/print/user |

**Sample URL Request**

|  |
| --- |
| http://localhost:8090/team3/print/user |

**Get Schedules print to PDF with jasper report**

1. Reservations

* **Get All**

**Deksripsi Endpoint**

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

**Request URL**

|  |
| --- |
| localhost:8090/team3/Reservation |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/Reservation |

**Sample Response**

|  |
| --- |
| Success Response |
| {        "reservation\_id": 50,              "user\_id": 13,              "user\_name": "Tasilia",              "email\_user": "Tasilia@hotmail.com",              "film\_name": "Thank you dewi",              "price": 100000,              "schedule\_id": 44,              "studio": "E",              "status\_show": **true**,              "seat\_number": 9,              "status\_seat": **true**,              "date\_film": "2022-08-08",              "start\_film": "07:00:00",              "end\_film": "10:00:00",              "created\_at": "2022-07-21T19:22:01.556854",              "updated\_at": "2022-07-21T19:22:01.556854"          }      ],      "message": "Success Get All",      "status": 200  } |

|  |
| --- |
| Failed Response |
| {    } |

**Request Structure**

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Lengt h | Description | Example |
| reservation\_id | Integer | M |  | Nomor Identifikasi reservation dari system. | 50 |
| user\_id | Integer | M |  | Id pengguna | 13 |
| email\_user | Varchar |  | 50 | Email dari pengguna | Tasilia@hotmail.com |
| Film\_name | Varchar |  | 50 | Nama film yang di pesan user | Thankyou Dewi |
| price | Integer |  |  | Harga dari film yang di pesan si user | 100000 |
| schedule\_id | Integer | M |  | id schedule yang terekam | 44 |
| Status\_show | boo |  |  | Menampilkan status dari film reservation | true |
| Date\_film | timestamp |  |  | Menampilkan waktu tayang film | 2022-08-08 |
| Start\_film | timestamp |  |  | Menampilkan mulai tayang film | 07:00:00 |
| End\_film | timestamp |  |  | Menampilkan waktu selesai tayang film | 10:00:00 |
| created\_at | Timestamp |  |  | Dibuat pada saat | 2022-07-21 19:22:01.556 |
| updated\_at | Timestamp |  |  | Di perbaharui saat | 2022-07-21 19:22:01.556 |

* + - **GET BY ID**

**Deksripsi Endpoint**

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

**Request URL**

|  |
| --- |
| localhost:8090/team3/Rservation/{id} |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/Reservation/4 |

**Sample Response**

|  |
| --- |
| 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  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| userid | Long | M |  | User Id yang ingin di delete. | 4 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Lengt h | Description | Example |
| reservation\_id | Integer | M |  | Nomor Identifikasi reservation dari system. | 4 |
| user\_id | Integer | M |  | Id pengguna | 4 |
| email\_user | Varchar |  | 50 | Email dari pengguna | Aida\_Nur@hotmail.com |
| Film\_name | Varchar |  | 50 | Nama film yang di pesan user | Dibawah Lindungan Kabah |
| price | Integer |  |  | Harga dari film yang di pesan si user | 20000 |
| schedule\_id | Integer | M |  | id schedule yang terekam | 12 |
| Status\_show | boo |  |  | Menampilkan status dari film reservation | false |
| Date\_film | timestamp |  |  | Menampilkan waktu tayang film | 2022-08-08 |
| Start\_film | timestamp |  |  | Menampilkan mulai tayang film | 07:00:00 |
| End\_film | timestamp |  |  | Menampilkan waktu selesai tayang film | 10:00:00 |
| created\_at | Timestamp |  |  | Dibuat pada saat | 2022-07-21 19:22:01.556 |
| updated\_at | Timestamp |  |  | Di perbaharui saat | 2022-07-21 19:22:01.556 |

* + **POST/CREATE**

**Deksripsi Endpoint**

By Request POST kita bisa buat data yang baru

**Request URL**

|  |
| --- |
| localhost:8090/ team3/Reservation |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/Reservation |

**Sample Request**

|  |
| --- |
| {  "user": {  "userId":15  },  "schedules": {  "scheduleId": 1  }  } |

**Sample Response**

|  |
| --- |
| Success Response |
| {  "data": {  "reservation\_id": 51,  "schedule\_id": 1,  "user\_id": 15,  "created\_at": "2022-07-27T21:49:09.9259721",  "updated\_at": "2022-07-27T21:49:09.9259721"  },  "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 |
| User | Entity/Model | M |  | Enitity.User |  |
| userId | Integer | M |  | Nomor Identifikasi user dari system. | 15 |
| Schedules | Entity/Model | M |  | Entity Schedule |  |
| scheduleId | Stirng | M |  | Nomor Identifikasi schedule dari System | 1 |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | lenght | Description | Example |
| reservation\_id | Integer | M |  | Nomor Identifikasi reservation dari system. | 51 |
| user\_id | Integer | M |  | Nomor kursi | 15 |
| schedule\_id | Integer | M |  | Nama Studio | 1 |
| created\_at | Timestamp |  |  | Dibuat pada saat | 2022-07-21 19:22:01.556 |
| updated\_at | Timestamp |  |  | Di perbaharui saat | 2022-07-21 19:22:01.556 |

* + **PUT**

**Deksripsi Endpoint**

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

**Request URL**

|  |
| --- |
| localhost:8090/team3/Reservation/{id} |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/Reservation/15 |

**Sample Request**

|  |
| --- |
| {  "user": {  "userId":16  },  "schedules": {  "scheduleId": 1  }  } |

**Sample Response**

|  |
| --- |
| Success Response |
| {  "data": {  "reservation\_id": 51,  "user\_id": 16,  "user\_name": "Tajungkang",  "email\_user": "Tajungkang@apps.ui.ac.id",  "film\_name": "Avatar 2",  "price": 20000,  "schedule\_id": 1,  "studio": "A",  "status\_show": true,  "seat\_number": 7,  "status\_seat": true,  "date\_film": "2022-06-19",  "start\_film": "00:00:00",  "end\_film": "00:00:00",  "created\_at": null,  "updated\_at": "2022-07-27T21:50:09.5555069"  },  "message": "Success Update Reservation",  "status": 201  } |

|  |
| --- |
| Failed Response |
| {  "data": "Bad Request",  "message": "Reservation not exist with id 99",  "status": 400  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| User | Entity/Model | M |  | Enitity.User |  |
| userId | Integer | M |  | Nomor Identifikasi user dari system. | 16 |
| Schedules | Entity/Model | M |  | Entity Schedule |  |
| scheduleId | Stirng | M |  | Nomor Identifikasi schedule dari System | 1 |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Lengt h | Description | Example |
| reservation\_id | Integer | M |  | Nomor Identifikasi reservation dari system. | 50 |
| user\_id | Integer | M |  | Id pengguna | 13 |
| email\_user | Varchar |  | 50 | Email dari pengguna | Tasilia@hotmail.com |
| Film\_name | Varchar |  | 50 | Nama film yang di pesan user | Thankyou Dewi |
| price | Integer |  |  | Harga dari film yang di pesan si user | 100000 |
| schedule\_id | Integer | M |  | id schedule yang terekam | 44 |
| Status\_show | boo |  |  | Menampilkan status dari film reservation | true |
| Date\_film | timestamp |  |  | Menampilkan waktu tayang film | 2022-08-08 |
| Start\_film | timestamp |  |  | Menampilkan mulai tayang film | 07:00:00 |
| End\_film | timestamp |  |  | Menampilkan waktu selesai tayang film | 10:00:00 |
| created\_at | Timestamp |  |  | Dibuat pada saat | 2022-07-21 19:22:01.556 |
| updated\_at | Timestamp |  |  | Di perbaharui saat | 2022-07-21 19:22:01.556 |

* + **DELETE**

**Deksripsi Endpoint**

By Request DELETE digunakan untuk menghapus data Reservation menggunakan parameter

**Request URL**

|  |
| --- |
| localhost:8090/team3/Reservation/{id} |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/Reservation/16 |

**Sample Response**

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

|  |
| --- |
| Failed Response |
| {    "data": "Data not found",      "message": "Reservation not exist with id 101",  } |

* **Get Reservation Reports**

**Deksripsi Endpoint**

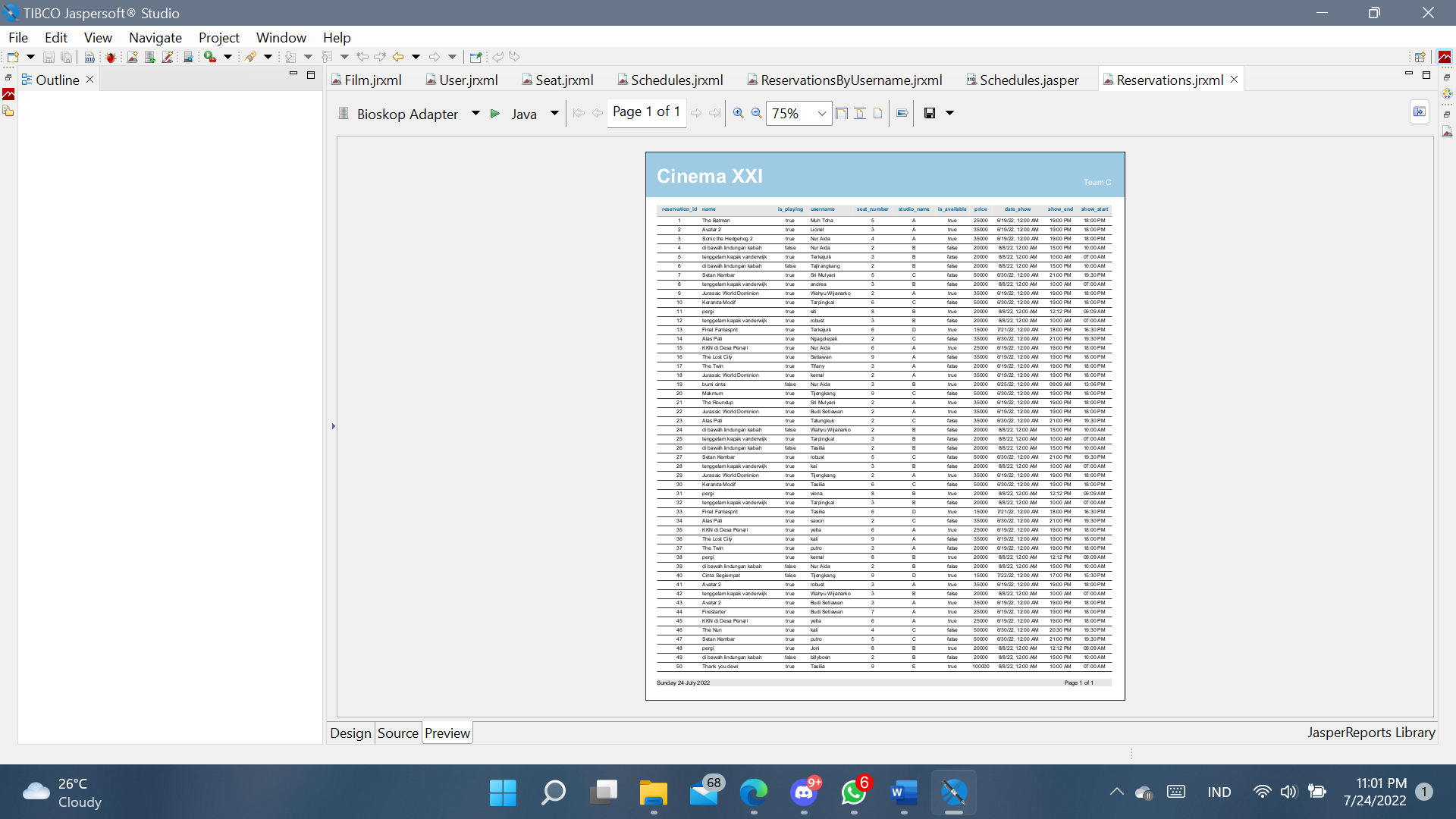
By Request GET kita dapat menggenerate Endpoint ke PDF menggunakan Jasper

**Request URL**

|  |
| --- |
| localhost:8090/team3/print/reservations |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/print/reservations |

**Get Reservations print to PDF with jasper report**

1. Film

* **Get All**

**Deksripsi Endpoint**

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

**Request URL**

|  |
| --- |
| localhost:8090/team3/films |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/films |

**Sample Response**

|  |
| --- |
| Success Response |
| {  "data": [  {  "filmId": 1,  "filmTitle": "Jurassic World Dominion",  "status": **true**  },  {  "filmId": 2,  "filmTitle": "The Roundup",  "status": **true**  }  ],  "message": "Success Get",  "status": 200  } |

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

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Lengt h | Description | Example |
| filmId | Integer | M |  | id film | 1 |
| filmTitle | Varchar |  |  | Judul film | The Roundup |
| status | Boolean | M |  | Film sedang berlangsung | true |

* **GET BY ID**

**Deksripsi Endpoint**

By Request GET BY ID digunakan untuk mengambil data film berdasarkan id film

**Request URL**

|  |
| --- |
| localhost:8090/team3/films/{filmId} |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/films/30 |

**Sample Response**

|  |
| --- |
| Success Response |
| {  "data": {  "filmId": 30,  "filmTitle": "Nenek Gayung",  "status": **true**  },  "message": "Succes Get",  "status": 200  } |

|  |
| --- |
| Failed Response |
| {  "data": "Data Not Found",      "message": "Request processing failed; nested exception is org.springframework.dao.EmptyResultDataAccessException: No class com.TeamC.Chapter6.Model.Film entity with id 30 exists!",      "status": 404  } |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Lengt h | Description | Example |
| filmId | Integer | M |  | id film | 1 |
| filmTitle | Varchar |  |  | Judul film | The Roundup |
| status | Boolean | M |  | Film sedang berlangsung | true |

* **POST (Search) By Film Name (Like Operator)**

**Deksripsi Endpoint**

By Request POST kita bisa mengambil data berdasarkan nama film

**Request URL**

|  |
| --- |
| localhost:8090/team3/search/films/name |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/search/films/name |

**Sample Request**

|  |
| --- |
| {  "filmTitle" : "The"  } |

**Sample Response**

|  |
| --- |
| Success Response |
| {  "data": [  {  "filmId": 5,  "filmTitle": "The Batman",  "status": **true**  },  {  "filmId": 9,  "filmTitle": "The Lost City",  "status": **true**  }  ],  "message": "Success All",  "status": 200  } |

|  |
| --- |
| Failed Response |
| {    "data": "Data Not Found",    "message": "Film with film name ”The” does not exist",    "status": 404  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| filmTitle | String | M |  | Nama film | The |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Lengt h | Description | Example |
| filmId | Integer | M |  | id film | 1 |
| filmTitle | Varchar |  |  | Judul film | The Roundup |
| status | Boolean | M |  | Film sedang berlangsung | true |

* **POST/CREATE**

**Deksripsi Endpoint**

By Request POST kita bisa buat data yang baru

**Request URL**

|  |
| --- |
| localhost:8090/team3/dashboard/create/films |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/dashboard/create/films |

**Sample Request**

|  |
| --- |
| {  "filmTitle": "film merah",  "status": **true**  } |

**Sample Response**

|  |
| --- |
| Success Response |
| {  "data": {  "filmId": 64,  "filmName": "film merah",  "isPlaying": **true**  },  "message": "Success Create",  "status": 201  } |

|  |
| --- |
| 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 |
| filmTitle | varchar | M |  | nama film | “The Magician” |
| status | boolean | M |  | Film sedang berlangsung | true |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Lengt h | Description | Example |
| filmId | Integer | M |  | id film | 1 |
| filmTitle | Varchar |  |  | Judul film | The Roundup |
| isPlaying | Boolean | M |  | Film sedang berlangsung | true |

* **PUT**

**Deksripsi Endpoint**

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

**Request URL**

|  |
| --- |
| localhost:8090/team3/dashboard/update/{filmId} |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/dashboard/update/films/1 |

**Sample Request**

|  |
| --- |
| {  "filmTitle": "The Magician 2",  "status": **true**  } |

**Sample Response**

|  |
| --- |
| Success Response |
| {  "data": {  "filmId": 66,  "filmName": "The Magician 2",  "isPlaying": **true**  },  "message": "Succes Update",  "status": 201  } |

|  |
| --- |
| Failed Response |
| {    "data": "Bad Request",    "message": "Film not exist with id : 62",    "status": 400  } |

**Request Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Length | Description | Example |
| filmTitle | varchar | M |  | nama film | “The Magician 2” |
| status | boolean | M |  | Film sedang berlangsung | true |

**Response Structure**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Data Type | Mandatory | Lengt h | Description | Example |
| filmId | Integer | M |  | id film | 1 |
| filmTitle | Varchar |  |  | Judul film | The Roundup |
| isPlaying | Boolean | M |  | Film sedang berlangsung | true |

* **DELETE**

**Deksripsi Endpoint**

By Request DELETE digunakan untuk menghapus data film menggunakan parameter

**Request URL**

|  |
| --- |
| localhost:8090/team3/dashboard/delete/films/{filmId} |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/dashboard/delete/films/64 |

**Sample Response**

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

|  |
| --- |
| Failed Response |
| {  "data": "Data Not Found",      "message": "Request processing failed; nested exception is org.springframework.dao.EmptyResultDataAccessException: No class com.TeamC.Chapter6.Model.Film entity with id 64 exists!",      "status": 404  } |

**Response Structure**

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

* **Get Film Reports**

**Deksripsi Endpoint**

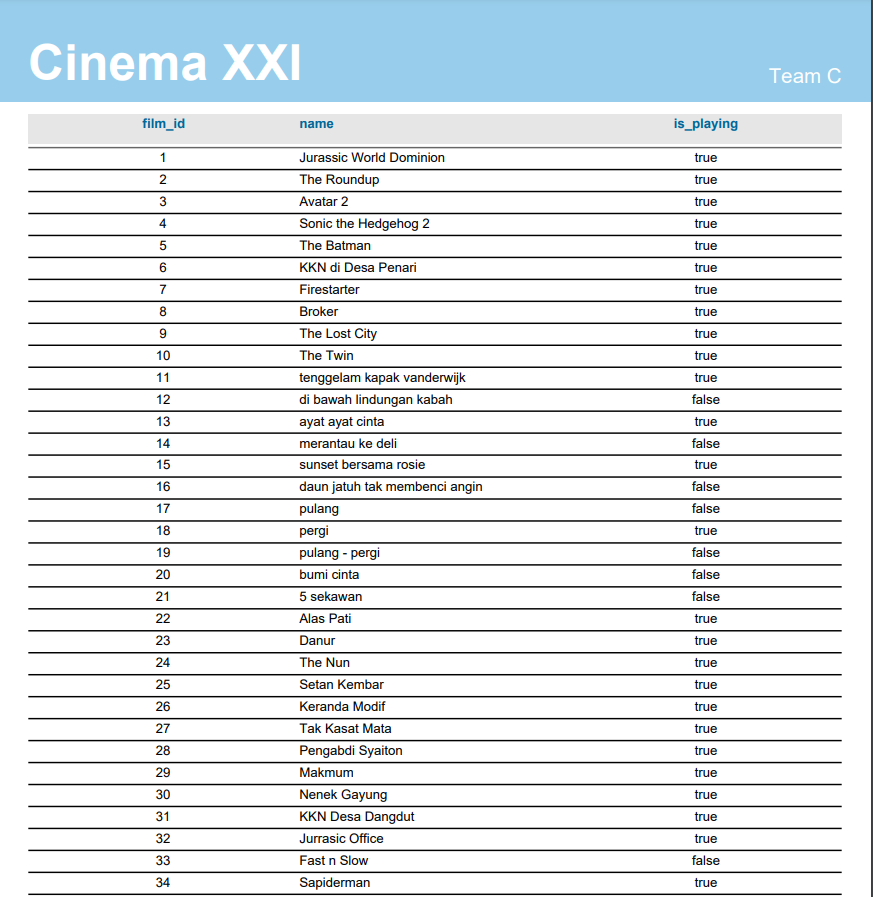
By Request GET kita dapat menggenerate Endpoint ke PDF menggunakan Jasper

**Request URL**

|  |
| --- |
| localhost:8090/team3/print/film |

**Sample Request URL**

|  |
| --- |
| http://localhost:8090/team3/print/film |

**Get Film print to PDF with jasper report**