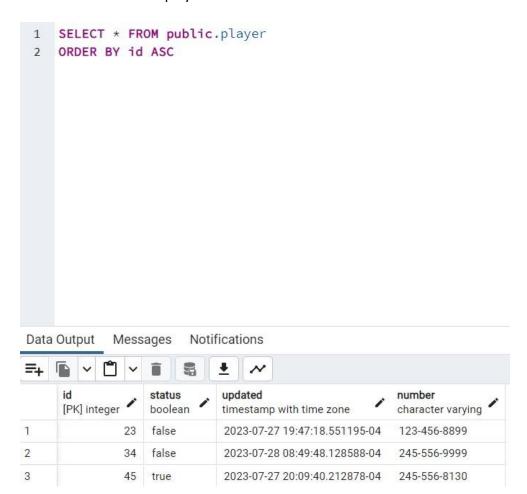# Java Sample Web Service Program

I decided to use Spring Boot Framework for this project. This is developed in Java 11 , with PostgreSQL 15 and pgadmin 4 v7.5.

How to run this project: -

1. Unzip the project.
2. You need to have installed Java 11 , PostgresSQL 15 and pg admin tool.
3. Create a database *playerdb* with user *postgres* and password *1234* in the pgadmin 4 GUI.
4. Import the contents of the *mydatabase_backup.sql* into the database by using the command :- *psql -h localhost -U postgres -d playerdb -f mydatabase_backup.sql* .
5. Once the database is ready, we can start up the spring boot service.
6. Go to the root of the project and run *java -jar target\ticketapp-0.0.1-SNAPSHOT.jar.*
7. The service will start up.
8. Now go to any REST API service test clients like curl, JMeter, wget, postman, etc.

The screenshot of the player table in the database :-

```
1  SELECT * FROM public.player
2  ORDER BY id ASC
```

Data Output   Messages   Notifications

	id [PK] integer	status boolean	updated timestamp with time zone	number character varying
1	23	false	2023-07-27 19:47:18.551195-04	123-456-8899
2	34	false	2023-07-28 08:49:48.128588-04	245-556-9999
3	45	true	2023-07-27 20:09:40.212878-04	245-556-8130

Example REST calls allowed :-

1. GET calls to display the contents of a specific player.

http://127.0.0.1:8080/player/print?id=34

Output JSON :-

```
{
   "id": 34,
   "status": false,
   "number": "245-556-9999",
   "updated": "2023-07-28T12:49:48.128588Z"
}
```

2. POST Calls to update the status and number of a player.

http://127.0.0.1:8080/player/update/34

```
{
  "status" : "false",
  "number" : "245-556-9999"
}
```

Output JSON :-

```
{
   "id": 34,
   "status": false,
   "number": "245-556-9999",
   "updated": "2023-07-28T12:49:48.128588Z"
}
```

Extra Credit:

1. The spring boot application by default is multi-threaded and also can handle multiple clients at once.