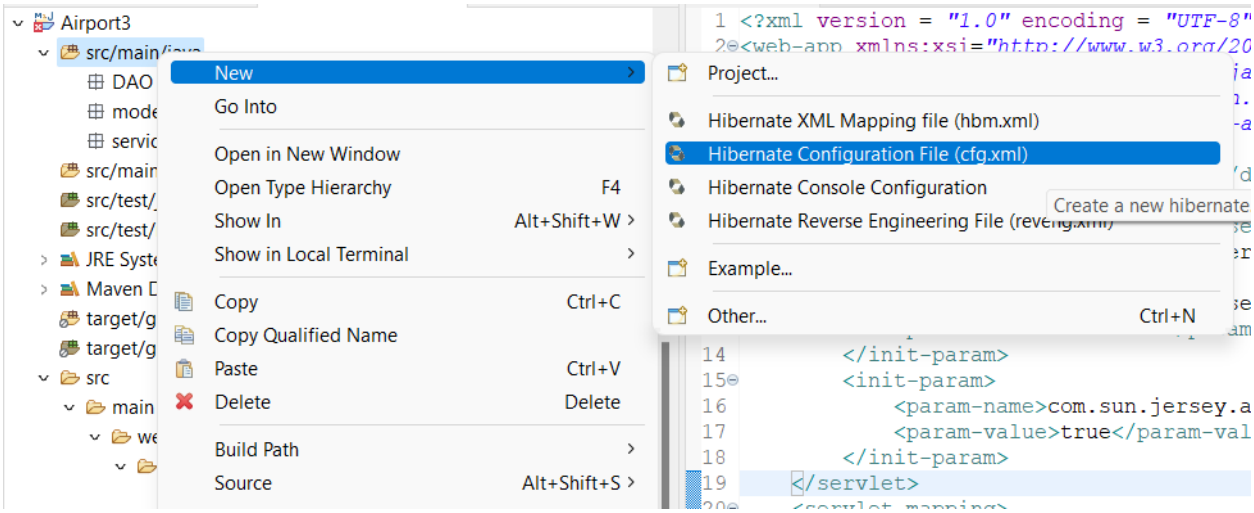
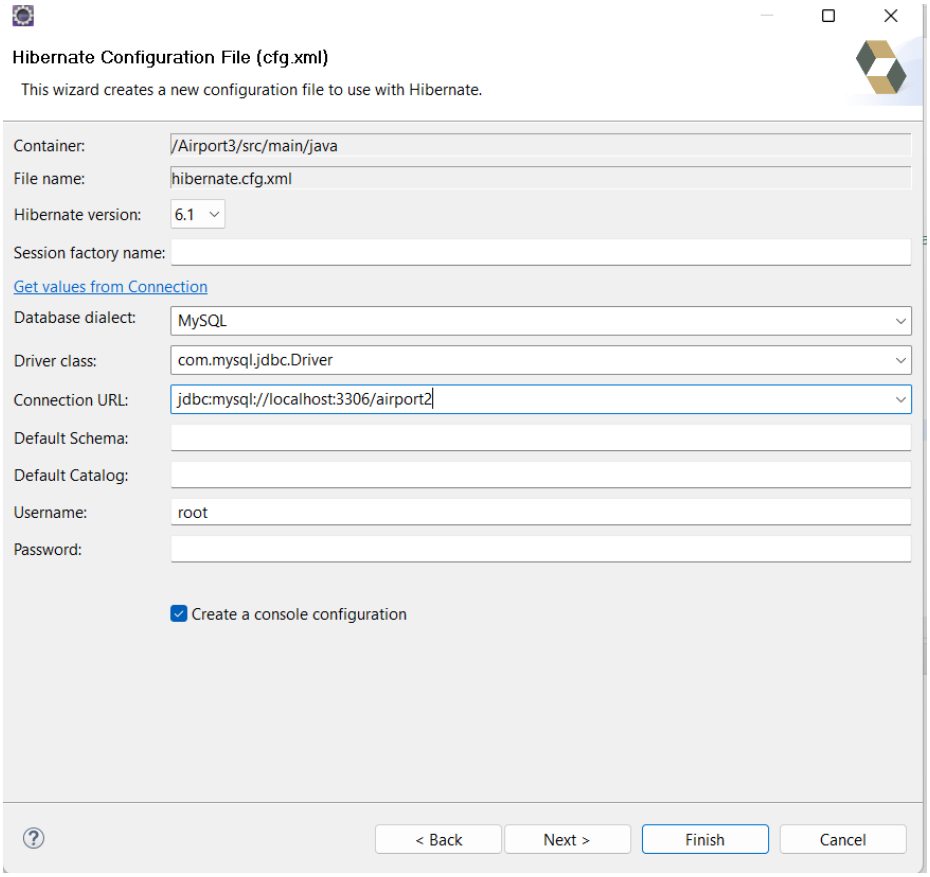
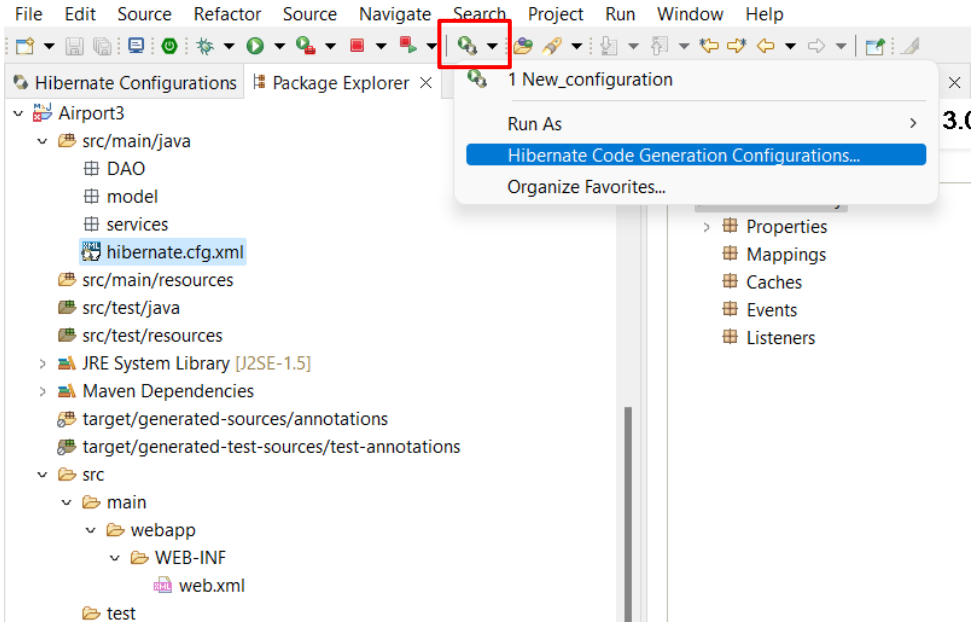
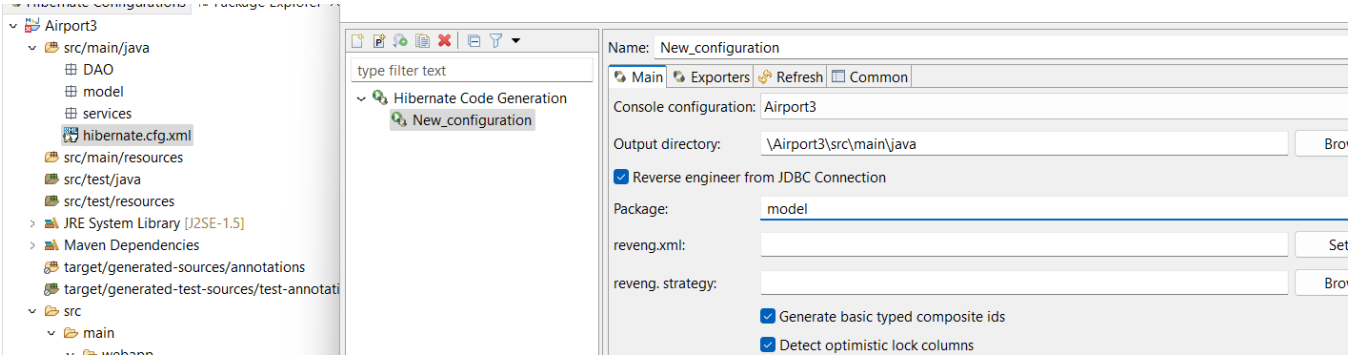
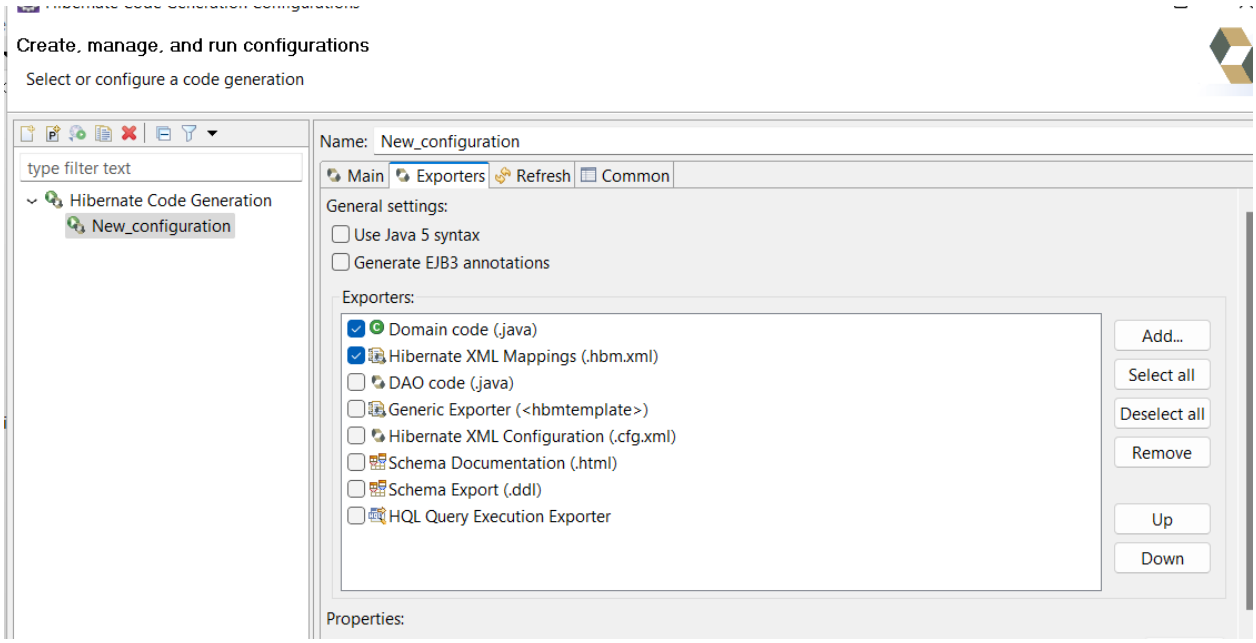

Airport3
src/main/java
DAO
model
services
src/main/resources
src/test/java
src/test/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
src
main
webapp
WEB-INF
web.xml
test
target
pom.xml
demo
<?xml version = "1.0" encoding = "UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd"
id="WebApp_ID" version="3.0">
<display-name>Customer Management</display-name>
<servlet>
<servlet-name>jersey-servlet</servlet-name>
<servlet-class>org.glassfish.jersey.servlet.ServletContainer</servlet-class>
<init-param>
<param-name>jersey.config.server.provider.packages</param-name>
<param-value>services</param-value>
</init-param>
<init-param>
<param-name>com.sun.jersey.api.json.POJOMappingFeature</param-name>
<param-value>true</param-value>
</init-param>
</servlet>
<servlet-mapping>
<servlet-name>jersey-servlet</servlet-name>
<url-pattern>/rest/*</url-pattern>
</servlet-mapping>
</web-app>


Airport3
New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Show in Local Terminal
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Build Path
Source Alt+Shift+S
Project...
Hibernate XML Mapping file (hbm.xml)
Hibernate Configuration File (cfg.xml)
Hibernate Console Configuration
Hibernate Reverse Engineering File
Example...
Other... Ctrl+N
Create a new hibernate

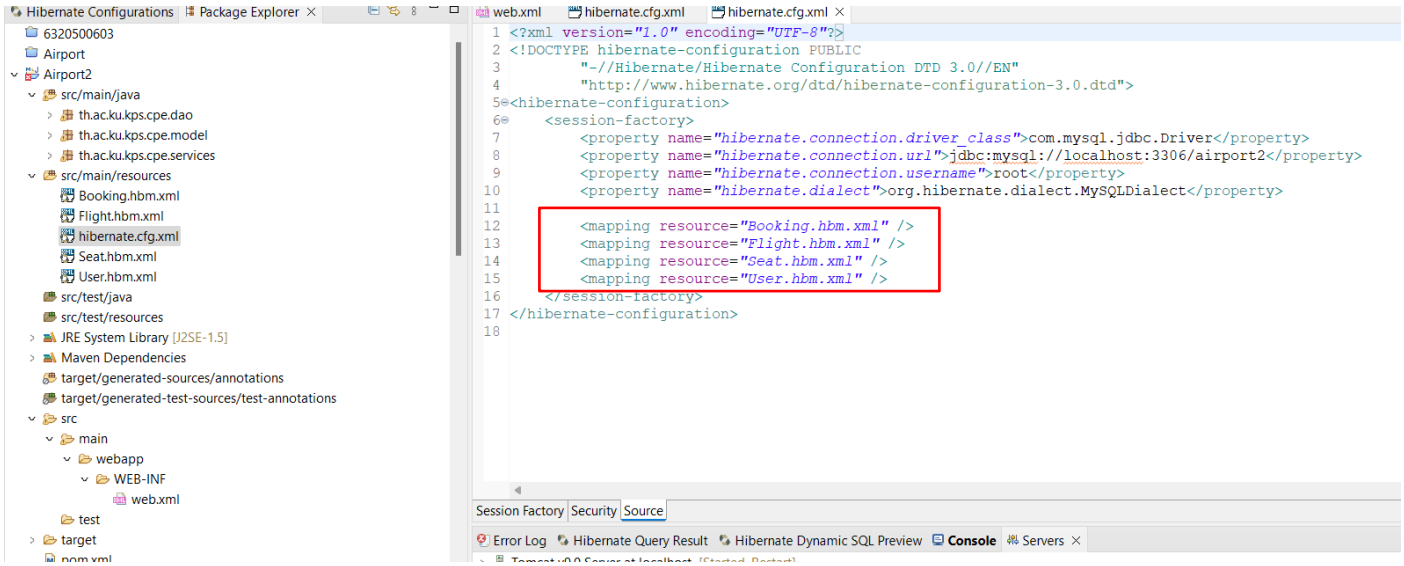

Hibernate Configuration File (cfg.xml)
This wizard creates a new configuration file to use with Hibernate.
Container: /Airport3/src/main/java
File name: hibernate.cfg.xml
Hibernate version: 6.1
Session factory name:
Get values from Connection
Database dialect: MySQL
Driver class: com.mysql.jdbc.Driver
Connection URL: jdbc:mysql://localhost:3306/airport2
Default Schema:
Default Catalog:
Username: root
Password:
Create a console configuration
< Back
Next >
Finish
Cancel


File Edit Source Refactor Source Navigate Search Project Run Window Help
Hibernate Configurations
Package Explorer
1 New_configuration
Run As
Hibernate Code Generation Configurations...
Organize Favorites...
Airport3
src/main/java
DAO
model
services
hibernate.cfg.xml
src/main/resources
src/test/java
src/test/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
target/generated-sources/annotations
target/generated-test-sources/test-annotations
src
main
webapp
WEB-INF
web.xml
test
Properties
Mappings
Caches
Events
Listeners


Airport3
src/main/java
DAO
model
services
hibernate.cfg.xml
src/main/resources
src/test/java
src/test/resources
JRE System Library [J2SE-1.5]
Maven Dependencies
target/generated-sources/annotations
src
main
type filter text
Hibernate Code Generation
New_configuration
Name: New_configuration
Main
Exporters
Refresh
Common
Console configuration: Airport3
Output directory: \Airport3\src\main\java
Reverse engineer from JDBC Connection
Package: model
reveng.xml:
reveng. strategy:
Generate basic typed composite ids
Detect optimistic lock columns


Create, manage, and run configurations
Select or configure a code generation
type filter text
Hibernate Code Generation
New_configuration
Name: New_configuration
Main
Exporters
Refresh
Common
General settings:
Use Java 5 syntax
Generate EJB3 annotations
Exporters:
Domain code (.java)
Hibernate XML Mappings (.hbm.xml)
DAO code (.java)
Generic Exporter (<hbmtemplate>)
Hibernate XML Configuration (.cfg.xml)
Schema Documentation (.html)
Schema Export (.ddl)
HQL Query Execution Exporter
Add...
Select all
Deselect all
Remove
Up
Down
Properties:

ดูที่ hibernate source และเพิ่ม



ไปที่ model เพิ่ม JsonIgnore ที่เป็น foreign key

```java
package th.ac.ku.kps.cpe.model;
// Generated Mar 28, 2023, 10:13:40 AM by Hibernate Tools 6.1.3.Final

import java.sql.Date;

/**
 * Booking generated by hbm2java
 */
public class Booking implements java.io.Serializable {

    private Integer bookingId;
    @JsonIgnore
    private Seat seat;
    @JsonIgnore
    private User user;
    @JsonIgnore
    private Flight flight;
    private Date date;

    public Booking() {
    }

    public Booking(Seat seat, User user, Flight flight, Date date) {
        this.seat = seat;
        this.user = user;
        this.flight = flight;
        this.date = date;
```

ตัวอย่าง query Booking

```
public List<Booking> getAllUser(){
    Session session = SessionUtil.getSession();
    Query query = session.createQuery("from Booking");
    ArrayList<Booking> booking = (ArrayList<Booking>) query.list();
    session.close();
    return booking;
}
```

```
public Booking getBookingById(int bookingId) {
    Session session = SessionUtil.getSession();
    String hql = "from Booking where bookingId = :bookingId";
    Query query = session.createQuery(hql).setParameter("bookingId", bookingId);
    List<Booking> booking = (List<Booking>) query.list();
    session.close();
    return booking.get(0);
}
```

```
public List<Booking> getAllBookingByUserId(int userId) {
    Session session = SessionUtil.getSession();
    String hql = "from Booking where userId = :userId";
    Query query = session.createQuery(hql).setParameter("userId", userId);
    List<Booking> bookings = (List<Booking>) query.list();
    session.close();
    return bookings;
}
```

```
public void addBooking(Booking booking) {
    Session session = SessionUtil.getSession();
    Transaction tx = session.beginTransaction();
    //tx.begin();
    session.save(booking);
    tx.commit();
    session.close();
}

public void update(Booking booking) {
    Session session = SessionUtil.getSession();
    Transaction tx = session.beginTransaction();
    session.update(booking);
    tx.commit();
    session.close();
}

public void delete(Booking booking) {
    Session session = SessionUtil.getSession();
    Transaction tx = session.beginTransaction();
    session.delete(booking);
    tx.commit();
    session.close();
}
```

BookingService

```
@Path("services")
public class BookingServices {

    BookingDAO bookingDAO = new BookingDAO();
    UserDAO userDAO = new UserDAO();
    FlightDAO flightDAO = new FlightDAO();
    SeatDAO seatDAO = new SeatDAO();
```

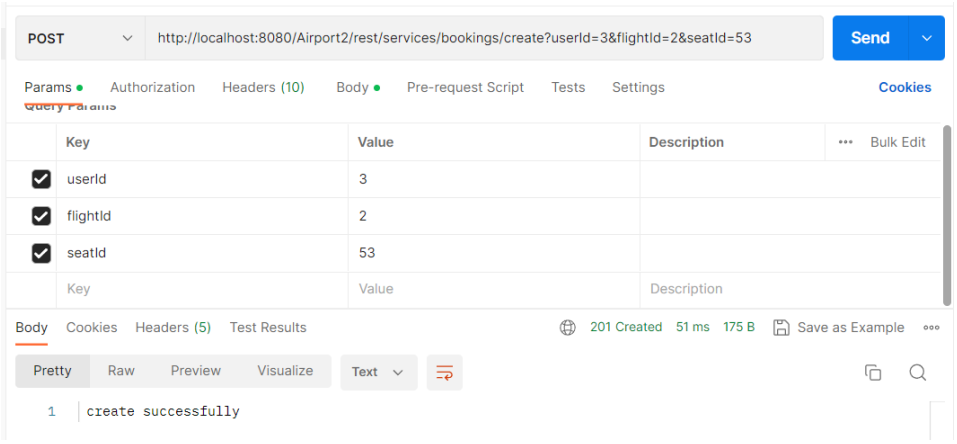
GET

```
@GET
@Path("/bookings")
@Produces(MediaType.APPLICATION_JSON)
public String getAllBooking() throws JsonProcessingException{
    ObjectMapper obj = new ObjectMapper();
    ArrayList<Booking> booking = (ArrayList<Booking>) bookingDAO.getAllUser();

    return obj.writeValueAsString(booking);
}

@GET
@Path("/bookings/{bookingId}")
@Produces(MediaType.APPLICATION_JSON)
public String getUserById(@PathParam("bookingId") int bookingId) throws JsonProcessingException{
    ObjectMapper obj = new ObjectMapper();
    Booking booking = bookingDAO.getBookingById(bookingId);

    return obj.writeValueAsString(booking);
}
```

```
@GET
@Path("/users/{userId}/bookings")
@Produces(MediaType.APPLICATION_JSON)
public String getAllBookingByUserId(@PathParam("userId")int userId) throws JsonProcessingException {
    ArrayList<Booking> bookings = (ArrayList<Booking>) bookingDAO.getAllBookingByUserId(userId);
    ObjectMapper obj = new ObjectMapper();
    return obj.writeValueAsString(bookings);

}
```

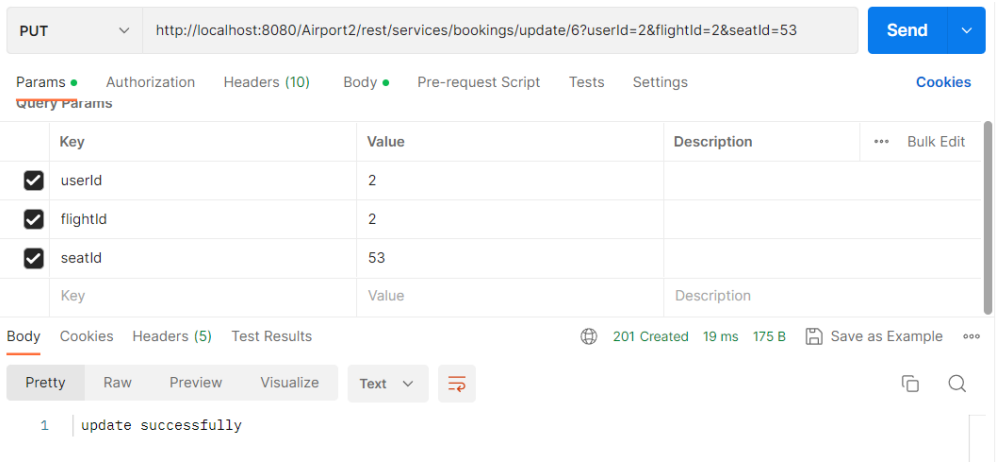
POST

```
@POST
@Path("/bookings/create")
@Consumes(MediaType.APPLICATION_JSON)
public Response addBooking(@QueryParam("userId")int userId,@QueryParam("flightId")int flightId,@QueryParam("seatId")int seatId)
        throws JsonMappingException, JsonProcessingException {

    User user = userDAO.getUserById(userId);
    Flight flight = flightDAO.getFlightById(flightId);
    Seat seat = seatDAO.getSeatById(seatId);
    long millis = System.currentTimeMillis();

    bookingDAO.addBooking(new Booking(seat,user,flight,new Date(millis)));

    return Response.status(201).entity(" create successfully").build();
}
```



PUT

```
@PUT
@Path("/bookings/update/{bookingId}")
@Consumes(MediaType.APPLICATION_JSON)
public Response updateBooking(@QueryParam("userId") int userId, @QueryParam("flightId") int flightId,
        @QueryParam("seatId") int seatId,@PathParam("bookingId")int bookingId) throws JsonMappingException, JsonProcessingEx
    Booking booking = bookingDAO.getBookingById(bookingId);
    User user = userDAO.getUserById(userId);
    Flight flight = flightDAO.getFlightById(flightId);
    Seat seat = seatDAO.getSeatById(seatId);
    booking.setFlight(flight);
    booking.setSeat(seat);
    booking.setUser(user);
    bookingDAO.update(booking);

    return Response.status(201).entity(" update successfully").build();
}
```



DELETE

```
@DELETE
@Path("/bookings/delete/{bookingId}")
@Consumes(MediaType.APPLICATION_JSON)
public Response deleteUser(@PathParam("bookingId")int bookingId) {

    Booking booking = bookingDAO.getBookingById(bookingId);
    bookingDAO.delete(booking);

    return Response.status(201).entity(" delete successfully").build();
}
```

ตัวอย่าง query User

```
public List<User> getAllUser(){
    Session session = SessionUtil.getSession();
    Query query = session.createQuery("from User");
    ArrayList<User> users = (ArrayList<User>) query.list();
    session.close();
    return users;
}
```

```
public User getUserById(int userId) {
    Session session = SessionUtil.getSession();
    String hql = "from User where userId = :userId";
    Query query = session.createQuery(hql).setParameter("userId", userId);
    List<User> users = (List<User>) query.list();
    session.close();
    return users.get(0);
}
```

```
public User getUserByUsername(String username) {
    Session session = SessionUtil.getSession();
    String hql = "from User where username = :username";
    Query query = session.createQuery(hql).setParameter("username", username);
    List<User> users = (List<User>) query.list();
    session.close();
    return users.get(0);
}
```

```
public void addUser(User user) {
    Session session = SessionUtil.getSession();
    Transaction tx = session.beginTransaction();
    //tx.begin();
    session.save(user);
    tx.commit();
    session.close();
}

public void update(User user) {
    Session session = SessionUtil.getSession();
    Transaction tx = session.beginTransaction();
    session.update(user);
    tx.commit();
    session.close();
}

public void delete(User user) {
    Session session = SessionUtil.getSession();
    Transaction tx = session.beginTransaction();
    session.delete(user);
    tx.commit();
    session.close();
}
```

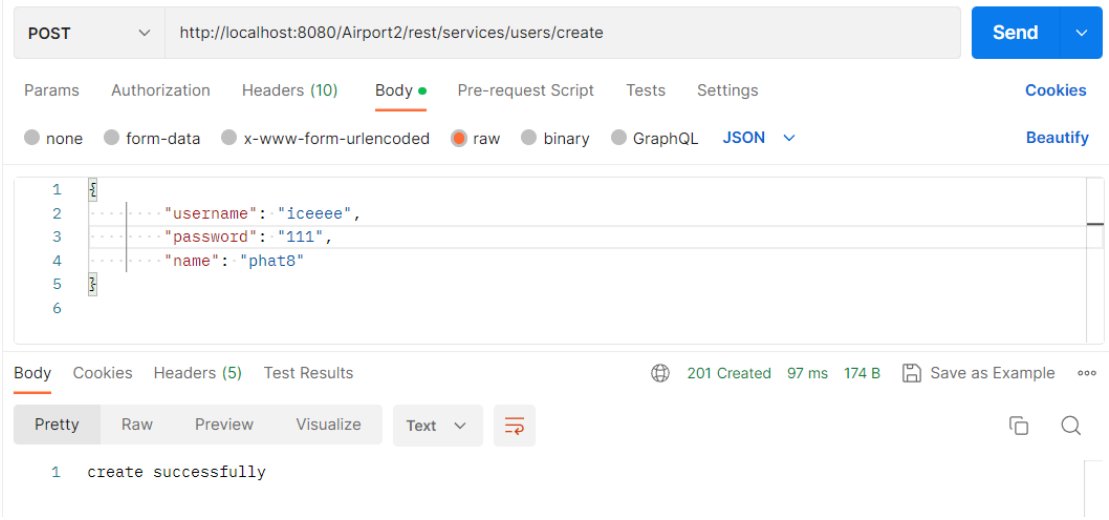
```java
package th.ac.ku.kps.cpe.services;

import java.io.IOException;
import java.util.ArrayList;

import javax.ws.rs.Consumes;
import javax.ws.rs.DELETE;
import javax.ws.rs.GET;
import javax.ws.rs.POST;
import javax.ws.rs.PUT;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;
import javax.ws.rs.core.Response;

import com.fasterxml.jackson.core.JsonProcessingException;
import com.fasterxml.jackson.databind.JsonMappingException;
import com.fasterxml.jackson.databind.ObjectMapper;


import th.ac.ku.kps.cpe.dao.UserDAO;
import th.ac.ku.kps.cpe.model.User;
```

```java
@Path("services")
public class UserService {

    UserDAO userDAO = new UserDAO();

    @GET
    @Path("/users")
    @Produces(MediaType.APPLICATION_JSON)
    public String getAllUsers() throws JsonProcessingException{
        ObjectMapper obj = new ObjectMapper();
        ArrayList<User> users = (ArrayList<User>) userDAO.getAllUser();
        for(User user : users) {
            System.out.println(user.getName());
        }
        return obj.writeValueAsString(users);
    }
```

```java
//findById
@GET
@Path("/users/{userId}")
@Produces(MediaType.APPLICATION_JSON)
public String getUserById(@PathParam("userId") int userId) throws JsonProcessingException{
    ObjectMapper obj = new ObjectMapper();
    User user = userDAO.getUserById(userId);

    return obj.writeValueAsString(user);
}

@GET
@Path("/users/username/{username}")
@Produces(MediaType.APPLICATION_JSON)
public String getUserByUsername(@PathParam("username") String username) throws JsonProcessingException{
    ObjectMapper obj = new ObjectMapper();
    User user = userDAO.getUserByUsername(username);

    return obj.writeValueAsString(user);
}
```

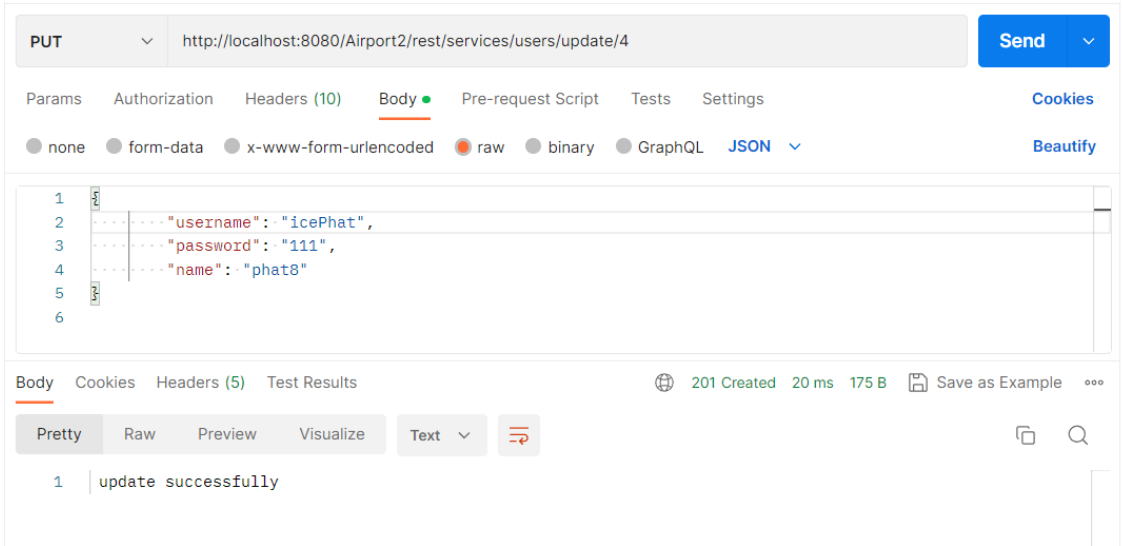
POST

```
@POST
@Path("/users/create")
@Consumes(MediaType.APPLICATION_JSON)
public Response addUser(String value) throws IOException {

    ObjectMapper obj = new ObjectMapper();
    User user = obj.readValue(value, User.class);
    userDAO.addUser(user);
    return Response.status(201).entity("create successfully").build();
}
```



PUT

```
@PUT
@Path("/users/update/{userId}")
@Consumes(MediaType.APPLICATION_JSON)
public Response updateUser(String value,@PathParam("userId")int userId) throws JsonMappingException, JsonProcessingException {
    ObjectMapper obj = new ObjectMapper();
    User user = obj.readValue(value, User.class);
    User u = userDAO.getUserById(userId);

    u.setName(user.getName());
    u.setPassword(user.getPassword());
    u.setUsername(user.getUsername());
    userDAO.update(u);

    return Response.status(201).entity(" update successfully").build();
}
```



```java
@DELETE
@Path("/users/delete/{userId}")
@Consumes(MediaType.APPLICATION_JSON)
public Response deleteUser(@PathParam("userId")int userId) {

    User user = userDAO.getUserById(userId);
    userDAO.delete(user);

    return Response.status(201).entity(" delete successfully").build();
}
```