com.booking

Class AirportController

java.lang.Object
     com.booking.AirportController

```
public class AirportController
extends java.lang.Object
```

This class provides methods pertaining to any required functionality and processing that is related to airports.

Field Summary

Fields

Modifier and Type	Field and Description
private **AirportEntity**	**ae** AirportController requires the AirportEntity class to write/read data to the database.
private java.util.Scanner	**in** Scanner allows class to use the basic input from the console.

Constructor Summary

Constructors

Constructor and Description
AirportController() Default constructor.

Method Summary

Methods

Modifier and Type	Method and Description
private void	**addAirport**() Method called by the manageAirportsOption() method when the Flight Manager chooses to add an airport.
private void	**deleteAirport**() Method called by the manageAirportsOption() method when the Flight Manager chooses to delete an airport.
boolean	**doesAirportExist**(java.lang.String airport_name) This method checks whether this airport name already exists in the database or not.

boolean	**doesIATAExist**(java.lang.String IATA) This method checks whether this IATA already exists in the database or not.
private void	**editAirport**() Method called by the manageAirportsOption() method when the Flight Manager chooses to edit an airport.
private java.lang.String	**enterAirportName**(boolean check) A UI method that allows the Flight Manager to enter an airport name.
private int	**enterAltitude**() A UI method that allows the Flight Manager to enter the altitude for an airport.
private java.lang.String	**enterCity**() A UI method that allows the Flight Manager to enter an city name.
private java.lang.String	**enterCountry**() A UI method that allows the Flight Manager to enter an country name.
private char	**enterDST**() A UI method that allows the Flight Manager to enter the daylight savings time for an airport.
private java.lang.String	**enterIATA**() A UI method that allows the Flight Manager to enter an IATA for an airport.
private double	**enterLatitude**() A UI method that allows the Flight Manager to enter the latitude for an airport.
private double	**enterLongitude**() A UI method that allows the Flight Manager to enter the longitude for an airport.
private int	**enterTimezone**() A UI method that allows the Flight Manager to enter the timezone for an airport.
private java.lang.String	**enterTzDatabaseTimezone**() A UI method that allows the Flight Manager to enter the TZ database timezone for an airport.
java.lang.String	**getAirportCity**(java.lang.String IATA) Gets the airport city associated with the airport IATA given.
java.lang.String	**getAirportCode**(java.lang.String airport_name) Gets the airport code associated with the airport name given.
java.lang.String	**getAirportCountry**(java.lang.String airport_name) Gets the airport country associated with the airport name given.
void	**manageAirportsOption**() Provides an interface for the Flight Manager to choose whether to add, edit or delete airports.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

ae

private AirportEntity ae

AirportController requires the AirportEntity class to write/read data to the database.