```java
package com.nxkundu.server.bo;

import java.io.Serializable;
import java.util.Base64;
import java.util.Date;
import java.util.UUID;

import com.google.gson.Gson;

/**
 *
 * @author nxkundu
 *
 * @email nxk161830@utdallas.edu
 * @name Nirmallya Kundu
 *
 * DataPacket
 * This DataPacket object is sent and received
 * between the server and the client
 *
 * The ACTION_TYPE defines the the type of Action that needs to taken
 * when the server or the client receives
 *
 * Type of ACTION_TYPE are:
 * ACTION_TYPE_MESSAGE, ACTION_TYPE_SIGNUP, ACTION_TYPE_LOGIN, ACTION_TYPE_LOGIN_SUCCESS,
 * ACTION_TYPE_LOGIN_FAILED, ACTION_TYPE_SIGNUP_FAILED, ACTION_TYPE_LOGOUT,
 * ACTION_TYPE_ONLINE, ACTION_TYPE_ACK
 *
 * The MESSAGE_TYPE defines the the type of Message received
 * and what the specific action the server or the client
 * will take upon receiving
 *
 * Type of MESSAGE_TYPE are:
 * MESSAGE_TYPE_MESSAGE, MESSAGE_TYPE_BROADCAST_MESSAGE, MESSAGE_TYPE_IMAGE_MESSAGE
 *
 * Each DataPacket has a unique Id UUID id
 *
 * Methods:
 *
 * 1> clone() - Creates a clone of the DataPacket object
 * This is used in case of Broadcast Message
 * as we need to make a clone of the same DataPacket object
 * and send it to all the clients
 *
 * 2> toJSON() - Converts the DataPacket object to the
 * JSON object making it suitable to send and receive
 * over the network
 *
 * 3> getByteImage() - Decodes the Image
 *
 * 4> setByteImage() - Encodes the Image
 *
 *
 */
public class DataPacket implements Serializable, Cloneable{

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    public static final String ACTION_TYPE_MESSAGE = "MESSAGE";
    public static final String ACTION_TYPE_SIGNUP = "SIGNUP";
    public static final String ACTION_TYPE_LOGIN = "LOGIN";
    public static final String ACTION_TYPE_LOGIN_SUCCESS = "LOGIN_SUCCESS";
    public static final String ACTION_TYPE_LOGIN_FAILED = "LOGIN_FAILED";
    public static final String ACTION_TYPE_SIGNUP_FAILED = "SIGNUP_FAILED";
    public static final String ACTION_TYPE_LOGOUT = "LOGOUT";
    public static final String ACTION_TYPE_ONLINE = "ONLINE";
    public static final String ACTION_TYPE_ACK = "ACK";

    public static final String MESSAGE_TYPE_MESSAGE = "SINGLE_MESSAGE";
    public static final String MESSAGE_TYPE_BROADCAST_MESSAGE = "BROADCAST_MESSAGE";
    public static final String MESSAGE_TYPE_IMAGE_MESSAGE = "IMAGE_MESSAGE";
```

```java

    private UUID id;

    private String action;

    private String messageType;

    private Client fromClient;
    private Client toClient;

    private String message;

    private String stringImage;

    private long timestamp;

    private boolean isACK;

    private int timesResentDataPacket;

    /****************************** Constructors ************************************/

    public DataPacket(Client fromClient, String action) {
        super();
        this.action = action;
        this.setId(UUID.randomUUID());
        this.setACK(false);
        this.setTimestamp(new Date().getTime());
        this.fromClient = fromClient;
        this.setTimesResentDataPacket(0);
    }

    /****************************** Object Methods ************************************/

    /**
     * Creates a clone of the DataPacket object
     * This is used in case of Broadcast Message
     * as we need to make a clone of the same DataPacket object
     * and send it to all the clients
     */
    @Override
    public Object clone() throws CloneNotSupportedException {
        return super.clone();
    }

    /**
     * Converts the DataPacket object to the
     * JSON object making it suitable to send and receive
     * over the network
     * @return
     */
    public String toJSON() {

        Gson gson = new Gson();
        String strJSON = gson.toJson(this);
        return strJSON;
    }

    /**
     * Decodes the Image
     * @return
     */
    public byte[] getByteImage() {

        return Base64.getDecoder().decode(stringImage);
    }

    /**
     * Encodes the Image
     * @param byteImage
     */
    public void setByteImage(byte[] byteImage) {

        this.stringImage = Base64.getEncoder().encodeToString(byteImage);
    }


    /****************************** toString ****************************************************/

    @Override
    public String toString() {
        return "DataPacket [id=" + id + ", action=" + action + ", messageType=" + messageType + ", fromClient="
                + fromClient + ", toClient=" + toClient + ", message=" + message + ", stringImage=" + stringImage
                + ", timestamp=" + timestamp + ", isACK=" + isACK + "]";
    }


    /****************************** Getters and Setters ***************************************/

    public String getStringImage() {
        return stringImage;
    }

    public void setStringImage(String stringImage) {
        this.stringImage = stringImage;
    }

    public boolean isACK() {
        return isACK;
    }

    public void setACK(boolean isACK) {
        this.isACK = isACK;
    }

    public String getAction() {
        return action;
    }

    public void setAction(String action) {
        this.action = action;
    }

    public String getMessageType() {
        return messageType;
    }

    public void setMessageType(String messageType) {
        this.messageType = messageType;
    }

    public Client getFromClient() {
        return fromClient;
    }

    public void setFromClient(Client fromClient) {
        this.fromClient = fromClient;
    }

    public Client getToClient() {
        return toClient;
    }

    public void setToClient(Client toClient) {
        this.toClient = toClient;
    }

    public String getMessage() {
        return message;
    }

    public void setMessage(String message) {
        this.message = message;
    }

    public UUID getId() {
        return id;
    }
```

```java

    public void setId(UUID id) {
        this.id = id;
    }

    public long getTimestamp() {
        return timestamp;
    }

    public void setTimestamp(long timestamp) {
        this.timestamp = timestamp;
    }

    public int getTimesResentDataPacket() {
        return timesResentDataPacket;
    }

    public void setTimesResentDataPacket(int timesResentDataPacket) {
        this.timesResentDataPacket = timesResentDataPacket;
    }

    public void incrementTimesResentDataPacket() {
        this.timesResentDataPacket += 1;
    }

}

```