**Domain**

To Continue on our chatter application we need to create a simple protocol to send data between server and client. In this lab you will create classes needed for 2 types of packets.

1. Message packet, this packet will hold the username, date and message. This packet will be sent to server whenever client types something. Also it will be sent to client whenever server needs to echo the message it has received.
2. Acknowledgment packet. this packet will hold the username, error message and ack code. This packet will be sent to client when a message packet is received on server side.

**Attachment**

* UML diagrams are attached.
* Skeleton maven project with all the JUnit and Jackson libraries included.
  + Don't forget to copy over your lab 2 code. Lab 2 is not needed in lab 3.

**Import Project to Eclipse**

if you are using other IDE's it is up to you to figure out how to setup your maven, you can copy the pom.xml out of skeleton.

1. download the zip file.
2. unzip it in a desired location.
3. open eclipse and click on file/open project.
4. navigate to unzipped location and only select the ChatterApp eclipse project.

**Run Maven**

1. right click on project/run as/maven clean.
2. right click on project/run as/maven install. This command is like clean and build of Netbeans. It Also runs all your tests however, you still can run your test manually one at a time.

**JSON**

<https://www.w3schools.com/js/js_json_syntax.asp>

This is a format to format to store information like XML. Here is an example of json format:

{ "user" : "cst8288", "error" : null, "ack" : 1}

This can be map to this class:

class AckPacket{  
    private String username;  
    private String error;  
    private int ack;  
}

**New Library Jackson**

In this lab you will use the new library called Jackson, [jenkov.com](http://tutorials.jenkov.com/java-json/jackson-objectmapper.html) has a very good tutorial (you don't need to read everything, lookup what you need). Jackson library can convert a JSON String to an object or an object to a JSON String.

Here are some components:

* [@JsonCreator](http://tutorials.jenkov.com/java-json/jackson-annotations.html#jsoncreator), this annotation is used in conjunction with @JsonProperty. Place @JsonCreator on the constructor of the POJO class and @JsonProperty on every argument of constructor with desired name.

@JsonCreator

AckPacket( @JsonProperty("user") String user,

                   @JsonProperty("error") String error,

                   @JsonProperty("ack") int ack){

This is used to setup a class to be created and filled by Jackson API.

* [ObjectMapper](http://tutorials.jenkov.com/java-json/jackson-objectmapper.html#jackson-objectmapper-example), this class is the center of Jackson API. An object of this class can convert string to an object and vice versa. To create one follow bellow:

ObjectMapper objectMapper = new ObjectMapper();

objectMapper.findAndRegisterModules();

* [Convert JSON String to an Object](http://tutorials.jenkov.com/java-json/jackson-objectmapper.html#read-object-from-json-string). Using ObjectMapper instance call the method readValue(JSON\_String, Target\_Class).  
  AckPacket packet = objectMapper.readValue( jsonString, AckPacket.class);
* [Convert an Object to JSON String](http://tutorials.jenkov.com/java-json/jackson-objectmapper.html#write-json-from-objects). Using ObjectMapper instance call the method writeValueAsString( JSON\_Convertable\_Instance).  
  String jsonString = objectMapper.writeValueAsString( ackPacket);
* You can also manually create a JSON object instead of String. This will be useful for testing.

ObjectNode node = objectMapper.createObjectNode();

node.put( "user", packet.getUsername());

node.put( "error", packet.getError());

node.put( "ack", packet.getAck());

AckPacket packet = objectMapper.treeToValue( node, AckPacket.class);

**Requirement**

You are to create 5 classes in src and 3 junit class in test package of your project. Make sure to make all classes in their correct packages.

1. AckPacket and MessagePacket are both POJO classes with package access constructor and no setters. make sure to add hashCode and equals. Don't forget to use @JsonCreator in the constructors.
2. AckBuilder and MessageBuilder both follow builder design pattern. they have constructors with package level access. When building your packets make sure nothing is null or empty, throw IllegalArgumentException if not. the only exception is for error which can be null.
3. PacketBuilderFactory, follow the sequence diagram attached.
4. Create junit tests for the two builders and the factory. Class diagrams attached have the list of test you should complete.

**Hint**

* All the Jackson code you need is already in the Jackson section.
* Create the Packet classes first then Builders and finally Factory.
* Create Factory test first then the Builders, leave the json tests for last.

**Submission**

You need to submit **one file,** **Zip (archive) of source code**.

1. You need to submit a zip file of your source code.   
   [firstName]-[lastName]-[labSection#].zip  
   ex: shawn-emami-11.zip