

Notes on Lab 6

General suggestions:

- Place your servers and accompanying classes in a different package (for instance, you can call it **server**). They should *not* be in the hotelapp package.
- When you copy your *runnable jar file* to the departmental file system, copy the */input* folder to the same directory as well.
- When you load data into ThreadSafeHotelData, load reviews from the **/reviews** subfolder, not from /reviewsLargerSet.

Jetty server:

1) How to pass ThreadSafeHotelData to servlets?

One way to do it is to pass it to the constructor of a servlet; it requires using ServletHolder :

```
handler.addServletWithMapping(new ServletHolder(new  
HotelServlet(data));
```

2) Important to "clean" parameters of the request (using Apache Commons Lang library)

3) Which Content type should be used in the response? application/json
response.setContentType("application/json");

4) Ok to use JSONObject for this lab.

5) A returned json file will **not** look nicely formatted in the browser - it is ok.
You can always view it using a json viewer I mentioned earlier.

RawSocketsServer:

1) Needs to be general! Not specific to lab 6.

Suggested design:

- Have an **HttpHandler** interface that contains a method processRequest().
- Write classes HotelHandler, AttractionsHandler and ReviewsHandler that implement this interface and override processRequest method.
- Maintain a map called **handlers** in the server class that maps each URL path to the appropriate "handler". You can fill it with values in the Driver - you would call server.addMapping ("reviews", ReviewHandler.class) etc.
- When you process the request, extract the path from the GET request, then lookup the corresponding value in the handlers map, create an object of that class and invoke processRequest method.
- You need to read **all the lines** in the header of the request. The following website describes the syntax of the request:
<https://code.tutsplus.com/tutorials/http-headers-for-dummies--net-8039>

If you only are reading the first line, you will lose 5% of the grade for the lab.

Having said it, I recommend working on this last , once you have other functionality working and your server is as general as possible (since it is only 5%)

- Need to have a class to represent an HttpRequest - it will contain parameters in a map, as long as data you parsed from the request.
- How do you know if the request is of GET type or POST type etc?
You can use startsWith in the String class.
- Your server needs to be multithreaded. For each client request, create a new GetRequestWorker.