Zack Chambers

CST 235

Charbel Elkhoury

February 17, 2019

GitHub Link: <https://github.com/zchambers3/CST235>

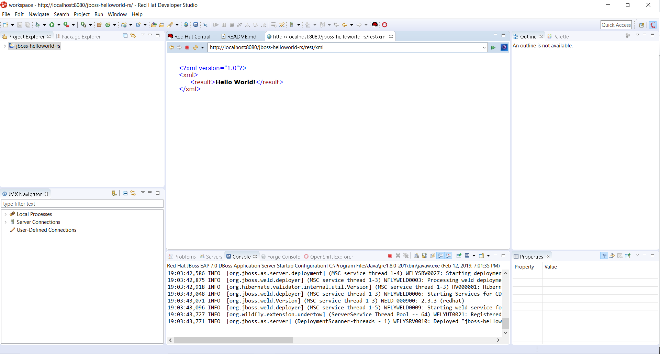
**Assignment 1**

*Part 1: HelloWorld*

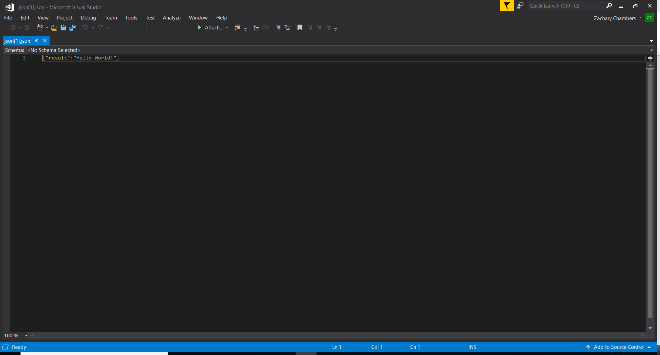
The first screenshot is validation of Servlet application in XML. Second screenshot is validation of Servlet application in JSON. Final screenshot is validation of the change in String.

Note: Click photo to redirect to Git Repository.

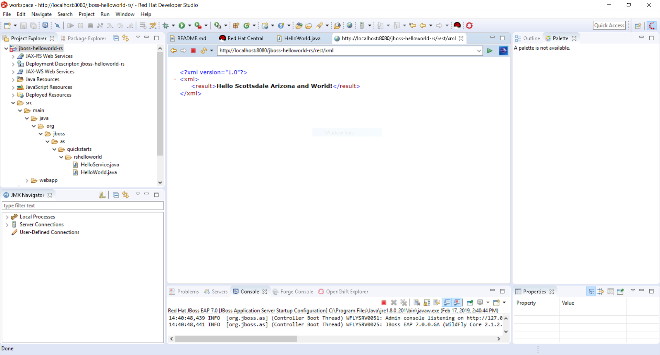
#1

[](https://github.com/zchambers3/CST235/blob/master/Week1/Assignments/Hello%20World%20XML.PNG)

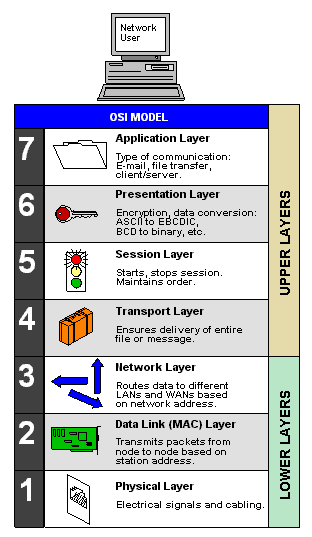
#2

[](https://github.com/zchambers3/CST235/blob/master/Week1/Assignments/Hello%20World%20JSN.PNG)

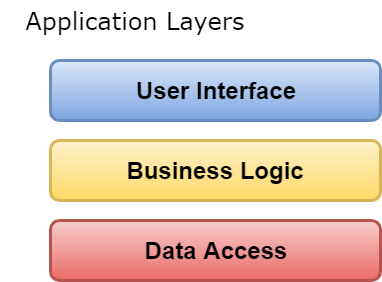
#3

[](https://github.com/zchambers3/CST235/blob/master/Week1/Assignments/Hello%20World%20XML.PNG)

*Part 2: 200 to 300-word write-up that provides the advantages of properly supporting designed N-Layer architecture.*

 N-tier and n-layer are entirely different concepts. N refers to # of tiers or # of layers in layered architecture, an architecture in which data moves from one defined level of processing to another. A good example of this would be communication protocols. The Open Systems Interconnection model below serves as a standard template for describing a network protocol stack.

Typical layers for smaller applications might include Presentation, Business and Data, the same as the traditional 3-tier model. But when we are talking about layers, we are only talking about logical organization of code. In no way is it implied that these layers might run on different computers or in different processes on a single computer or even in a single process on a single computer. All we are doing is discussing a way of organizing a code into a set of layers defined by specific function. Physical tiers however, are only about where the code runs. Specifically, tiers are places where layers are deployed and where layers run. In other words, tiers are the physical deployment of layers.



The most simple architecture pattern is the layered architecture pattern. This pattern is the de facto basic for most Java EE applications and therefore is generally known by most architects, designers and planners. The layered architecture pattern hardly matches the traditional IT communication and organizational structures found in most companies, making it a common choice for most business application development efforts.

References

Cole, K., McChesney, R., & Raszka, R. (2011). Advanced java ee development for rational application developer 7.5 : developers' guidebook. Retrieved from https://ebookcentral-proquest-com.lopes.idm.oclc.org

Pilgrim, P. A. (2013). Java EE 7 handbook. Retrieved from https://ebookcentral proquest-com.lopes.idm.oclc.org