

Lab03	Servlet based Inventory .v0
IT618 Enterprise Computing, Autumn'2020; pm_jat @ dalict	

This week you create a Servlet that implements few use cases of web based inventory system.

[This is working copy of lab description; may have some added elaborations; please download fresh when lab begins]

Usage Scenario of the Application:

1. You request getting list of all items in inventory (Let inventory be URL for the servlet)

<http://localhost/lab03/inventory?action=listitems>

You get list of all items. Need not to be formatted simple list as following is fine.

All items in the Inventory:

101,USB Mouse,5,10,450.0

102,DVD Drive,3,2,2100.0

103,500 GB Hard Disk,3,1,3500.0

2. You request getting details of a given item

<http://localhost/lab03/inventory?action=getitem&code=103>

You get item details as following

Details of requested item:

Item-code: 101

item-description: 500 GB Hard Disk

Item stock: 3

Item required min stock: 1

item-cost: 3500

3. You add stock to an item

<http://localhost/lab03/inventory?action=addstock&code=101&qty=5>

Responds as “Successful” or “Appropriate Error Message”.

There will be exception when there is no item in inventory with code=101

4. You withdraw stock of an item

<http://localhost/lab03/inventory?action=withdraw&code=101&qty=2>

Responds as “Successful” or “Appropriate Error Message”.

Again there will be error when there is no item in inventory with code=101

5. You add a new item to inventory.

To display a blank html form, let us say you send an http request as following:
<http://localhost/lab03/additem.html> that returns html form as shown below

Add New Inventory Item

Item Code	<input type="text"/>
Item Description	<input type="text"/>
Item Stock	<input type="text"/>
Item Min Stock	<input type="text"/>
Item Cost	<input type="text"/>
<input type="button" value="Add Item"/>	

On clicking [Add Item], a post request is sent and item is added to the inventory.

Response: Appropriate acknowledgement is returned “**Successful**” or “**Error Message**”

There will be exception raised when there is already an item in inventory with entered item code here.

6. Let you also add functionality in use case 5 above. Since here we are adding new item, and item code entered here should be nonexistent in inventory database. Recall that **addItem** method of inventory raises exception “Item Already Exists”.

To avoid disappointment later, we would like to check if entered item code is valid (that is it not already there in inventory database). For performing this check let you use ajax. Let there be a button against item code field, called [**check it**]; on click of this button you make an ajax call to inventory servlet. Let the URL for ajax call here be as following:

<http://localhost/lab03/inventory?action=checkitem&code=104>

Inventory servlet responds **true** for this request if item with given code exists, otherwise returns **false**.

Implementation recommendations:

1. Create new Dynamic Web Project; give name **lab03** to it.
2. Before proceeding, you are advised to study example “[DataSet servlet](#)” placed at moodle.
3. For business logic, let us use Inventory classes of Lab01 without any modification. Fresh download the Inventory classes from [here](#). There are some updates to it.
4. Place them in package “**lab03.inventory.model**” of your web project; you may have to change package name appropriately in these classes when you place them here.
5. Create a single servlet named as “**InventoryServlet**” to handle all use cases here. Put the servlet in package “**lab03.inventory.servlets**”.
6. Specify URL pattern “**/inventory**” for this servlet.
7. You require importing inventory classes in servlet class as these classes are in different package.
8. Store an **Inventory object as servlet field**.

9. Add some items to the inventory object in Constructor of the Servlet; your item adding code may look something as following-

```
try {
    inventory.addItem( new Item(101, "USB Mouse", 10, 5, 450 ) );
    inventory.addItem( new Item(102, "DVD Drive", 2, 3, 2100 ) );
    inventory.addItem( new Item(103, "500 GB Hard Disk", 0, 3, 3500 ) );
}
catch(ItemAlreadyExists e) { //We ensure that it does not land here }
```

10. Have a private method “processRequest” in servlet class with following signature

```
processRequest(HttpServletRequest request, HttpServletResponse response)
                throws ServletException, IOException
```

All types of actions are handled in the method - may use appropriate switch or so.

And call this “processRequest” method in doGet and doPost so that you have common functionality handling Get and Post requests.

11. For implementing Use Case 5, do following-

- a. Create an html page “additem.html” to add a new item. Have a HTML form in this with action specified as “inventory”. (Note: you do not begin with / here). typically as following:
`<form method = "post" action="inventory"> </form>`
- b. Have a hidden field called “action” with value of “additem”.
- c. In In HTML form, let the field names be as following for capturing attributes of inventory item: code, description, stock, minstock, and cost respectively