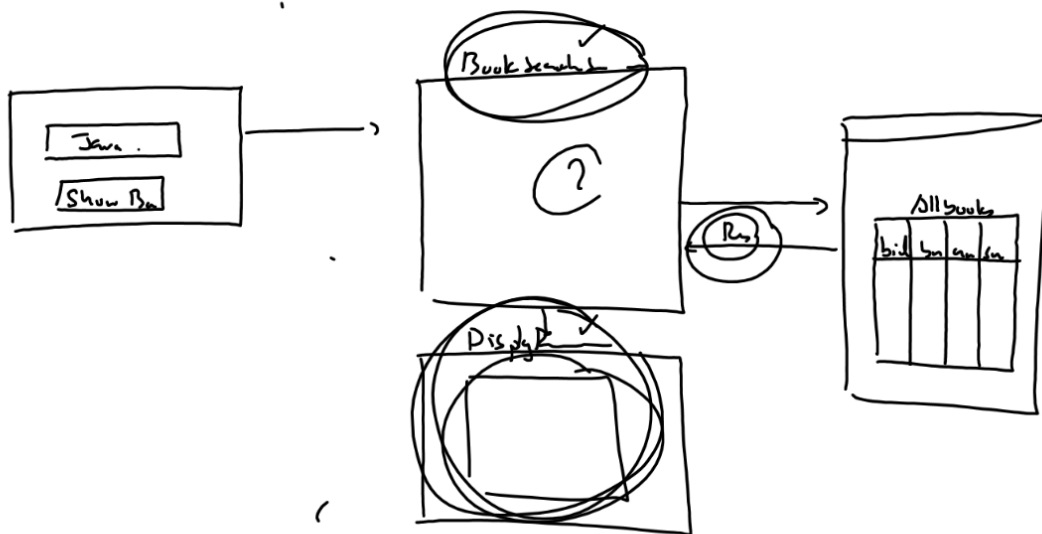


Request Dispatching ✓



Request Dispatching

1. Many times in a Java web application it might happen that our Servlet can perform the initial processing of the request but for further processing of that request it may want to transfer the control to some other web resource.

2. This is achieved using Request Dispatching mechanism in Java.

3. For example: suppose we are developing an E-commerce application and we have two Servlets called **ProductSearchServlet** and **ProductDisplayServlet**.

The user fills up the name of the product in an HTML page, clicks on the search button which generates a request to **ProductSearchServlet** and the **ProductSearchServlet** fetches the details of the searched products from the DB but these details have to be displayed by **ProductDisplayServlet**. This in Java is done with the help of **Request Dispatcher** i.e. the **ProductSearchServlet** will dispatch the request to **ProductDisplayServlet** along with the product data to be displayed.

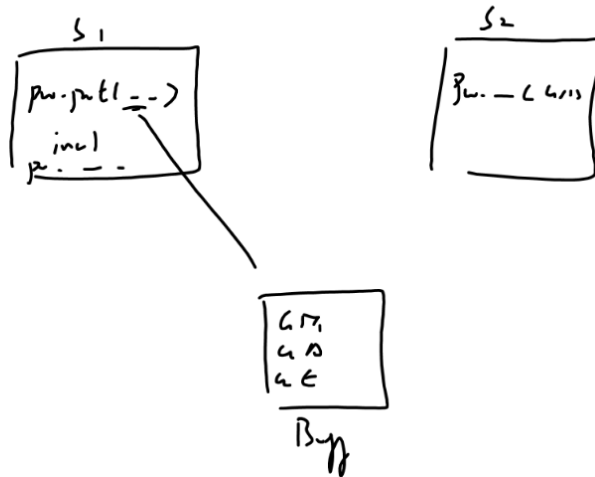
Redirection

1. Used for transferring the request from one resource to another.
2. Current request is finished and a new request is generated by the browser to a new URL.
3. All request and response data of the original request is lost.
4. The original URL is changed and a new URL is displayed in the browser which generates the output.
5. Redirection can happen within the application or to any website on the internet.

v/s

Dispatching

1. Used for transferring the request from one resource to another.
2. Current request remains alive but the output is generated by multiple resources.
3. All request and response data of the original remains alive b/w resources.
4. In the browser still the original URL shown but the output comes from multiple resources.
5. Dispatching is only allowed within the application not to any external resource.



How to perform request dispatching

=====

1. To perform request dispatching we need an object of **RequestDispatcher** interface.

2. To get a **RequestDispatcher** object we call the method **getRequestDispatcher()** belonging to **HttpServletRequest** object.

3. This method has the following prototype:

public RequestDispatcher getRequestDispatcher(String);

4. The argument passed to this method is the name of the resource to which request has to be dispatched.

5. Then we have to either of the following two methods of **RequestDispatcher** object which are :

a) **include()**

b) **forward()**