Session Variables

1

Last time: request variables

We saw how to attach request variables to the request object in the servlet or Spring Controller handler method.

request.setAttribute("user", user); // in servlet code

model.setAttribute("allSizes", sizes); // in Spring handler, causing request.setAttribute("allSizes", sizes) in the DispatcherServlet.

The request variables ride from the servlet/handler to the forwarded-to JSP along with the request itself.

This is great for providing data from the servlet/handler to its JSP helper, but these variables evaporate along with the request object at the end of the request-response cycle.

2

Request variables vs. request parameters

Request **variables** are "attributes" of request, manipulated with request.get/setAttribute(…)

Request **parameters** are "parameters" of a request that come in with the original request, and are not changeable. In a GET, they show up after the path, in the query string that starts with ?.

We use request.getParameter(String name) to access a query string like productCode=pf01 and request.getParameterValues(String name) for query strings like x=1&x=2.

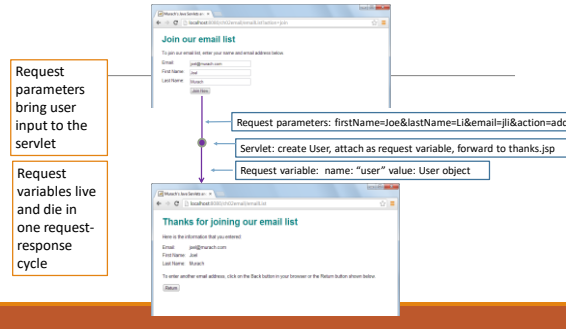
Parameters bring in the form data from a form, and can come in from decorated links like this:

pf01 info

And also from radio buttons, check boxes, and dropdown lists.

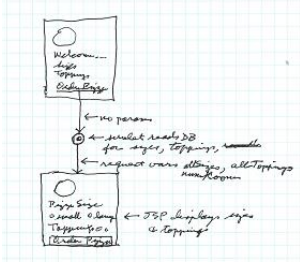
You see that request parameters bring in input from the user to the servlet or Spring Controller handler, which in turn directs what the app should do. Request variables are created in the servlet/handler for communication to the forwarded-to JSP (or servlet/handler)

3



4

Spring controller that sets up the pizza order page in pizza3: uses request variables allSizes, allToppings, and numRooms



5

Examples of longer-lived variables

In our client-server programs, we could keep longer-lived variables going in our presentation code.

In pizza1, TakeOrder asks for the room number for each action, but it could ask for it once and then hold it in a field, and use it for orders and status requests.

In music1, UserApp has a field cart that holds the user's cart, which changes over actions of the user. Also a field user that remembers info on the current user. Finally, there is a local variable that holds the Product that the user is interested in, across some actions.

In a web app, we want to use session variables for these variables.

6

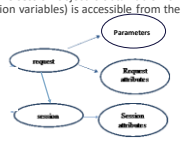
Next: MVC webapps with session variables.

- Start reading Chap 7 on Session Variables.
- So far, we have been using "request variables" like the "user" attribute in ch05email and "allSizes" for pizza3's show-order-form page.
- In many cases, we need to remember information in the server for longer, from one request cycle to another. That's when we use **session variables**, or of course the database.
- The session variables belong to a certain user, like the variables in presentation in client-server apps.
- We'll see how tomcat sets up and maintains them across multiple requests

7

Accessing session variables

- The session variables hang off the "session" object just like the request variables hang off the request object. The session object is obtainable from the request object, so everything (among request and session variables) is accessible from the request object.



Parameters: request.getParameter(...)

Request variables: request.getAttribute(...)

Session variables
request.getSession().getAttribute(...)

(also setAttribute in the lower two cases)

8

Chapter 7: How to work with sessions

Look at Chapter7 slides (6pp)

9