Scope and State Handling in Java Server Pages

Jeff Offutt

http://www.cs.gmu.edu/~offutt/

SWE 432

Design and Implementation of Software for the Web

Session Tracking (2)

review

***Session*: A series of related interactions between a client and a web server (similar to a use case)**

- Session tracking refers to keeping data between multiple HTTP requests
- This problem is essential to maintaining state, which we understand quite well in the context of traditional procedural programming and object-oriented programming
- The Web brings in unique constraints

HTTP is stateless

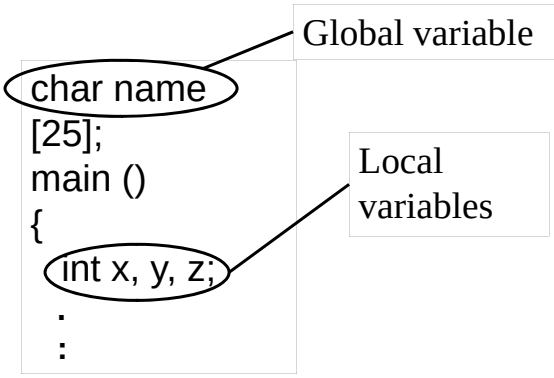
Distributed

© Offutt, 2011

review

Handling State in Procedural Languages

- The C programming language has simple ways to handle state

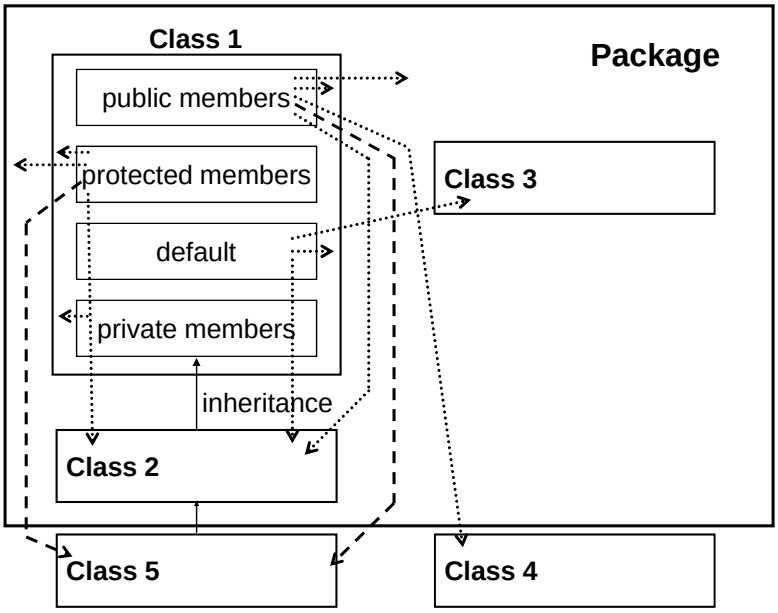


- We added several layers of scope in OO languages

© Offutt, 2011

review

Handling State in Java



© Offutt, 2011

State on the Web

review

- These schemes have two simple, subtle, assumptions :

1. The software components share physical memory

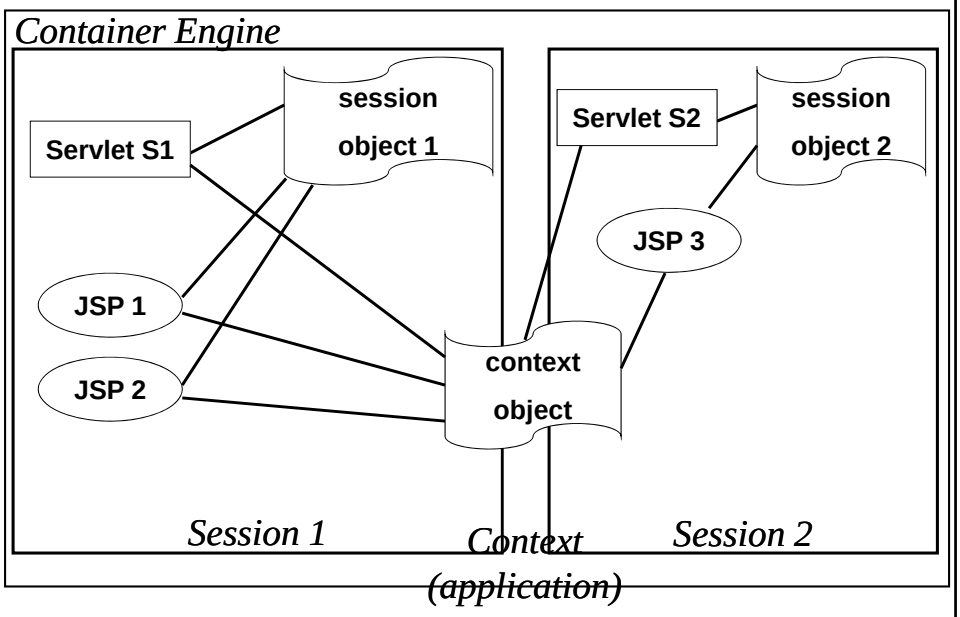
2. The program runs to completion with active memory

- But these assumptions are violated in web applications !
  1. Distributed software components
  2. Stateless nature of HTTP
- To keep state in web applications, we need different ways to store and access variables and objects

Public access and parameter passing are <u>not</u> <u>enough</u> for Web applications!

Context Scope

review



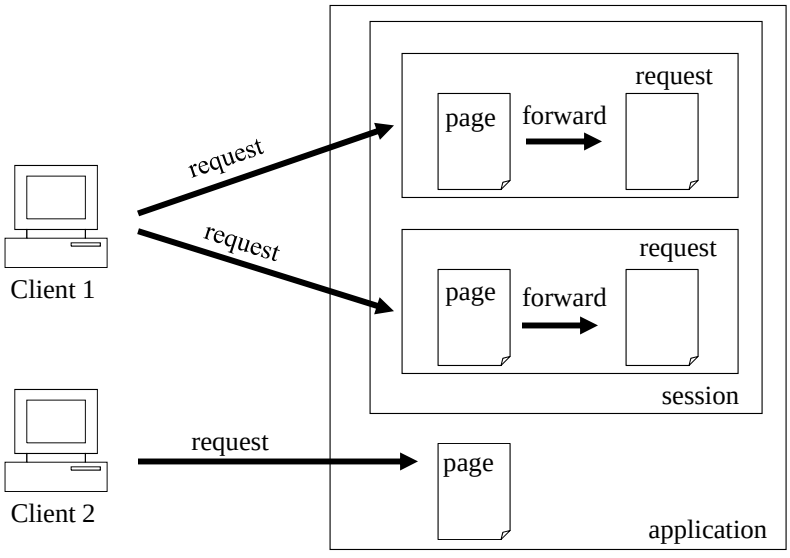
JSP Scope & State Management

- JSPs formalize this with four separate scopes new ...
  1. Page : Within the same program component (web page)
  2. Request : Within the same request
  3. Session : Within all requests from the same session
  4. Application : Within all sessions for one servlet context

- Each can be accessed by different sets of program components
- Some exist for different periods of time

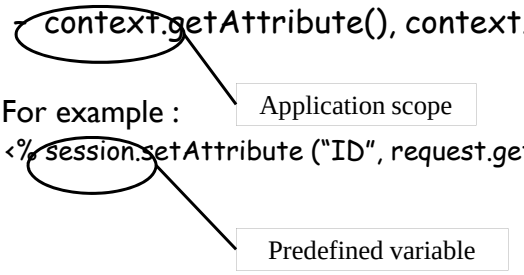
http://cs.gmu.edu:8080/offutt/jsp/432/counterScope.jsp

Sharing Data with Scope



Sharing Data with Scope (2)

- Using JSP Scriptlets
  - getParameter(); // retrieves client form data
  - request.getAttribute(), request.setAttribute();
  - session.getAttribute(), session.setAttribute();
  - context.getAttribute(), context.setAttribute();
- For example :
  <% session.setAttribute ("ID", request.getParameter ("ID")); %>



Sharing Data with Scope (3)

- The previous approach makes the code kind of clumsy
- Alternative approach – expanded use of JavaBean
- Use the scope attribute in the <jsp:useBean> action

```
<jsp:useBean id="languageBean" class="lang.LanguageBean"
             scope="session">
<jsp:getProperty name="languageBean" property="name">
```

- Scoping keywords:
  - scope="request" for the request object
  - scope="session" for the session object
  - scope="application" for the context object
- The page scope is default – local variables

JSP State Management Summary

- Programmers often get state management wrong
  - They learned "how" without learning "why" (the *theory*)
  - They don't understand the differences in the various scopes
  - They forget to consider which scope to use as part of design
- State management is very different from traditional programming
- These scopes are quite powerful
- New frameworks beyond J2EE often add different scopes and different semantics on the same scopes

http://cs.gmu.edu/~offutt/classes/432/examples/jsp/