



SERVLET SCOPE & FILTER

LẬP TRÌNH JAVA #4 (P5.1)

- Giới thiệu Scope
- Chia sẻ dữ liệu qua HttpServletRequest
- Chia sẻ dữ liệu qua HttpSession
- Chia sẻ dữ liệu qua ServletContext
- Chia sẻ dữ liệu qua PageContext
- WebListener



- ❑ Chia sẻ dữ liệu là việc dữ liệu được tạo ra bởi một thành phần (Servlet, JSP, filter, listener) này và được sử dụng bởi một thành phần khác. Nơi duy trì dữ liệu tạm thời đó được gọi là scope (phạm vi chia sẻ).
- ❑ Trong servlet/jsp tồn tại 4 scope với tên gọi là **page, request, session** và **application**



- ❑ Request (**HttpServletRequest**) sẵn có dưới dạng đối số của phương thức dịch vụ (doGet(), doPost(), service())
- ❑ Tham chiếu đến Session và Application
 - ❖ **HttpSession** session = request.get**Session**()
 - ❖ **ServletContext** application = this.getServletContext()
 - ❖ ServletContext application = request.get**ServletContext**()
 - ❖ ServletContext application = session.getServletContext()

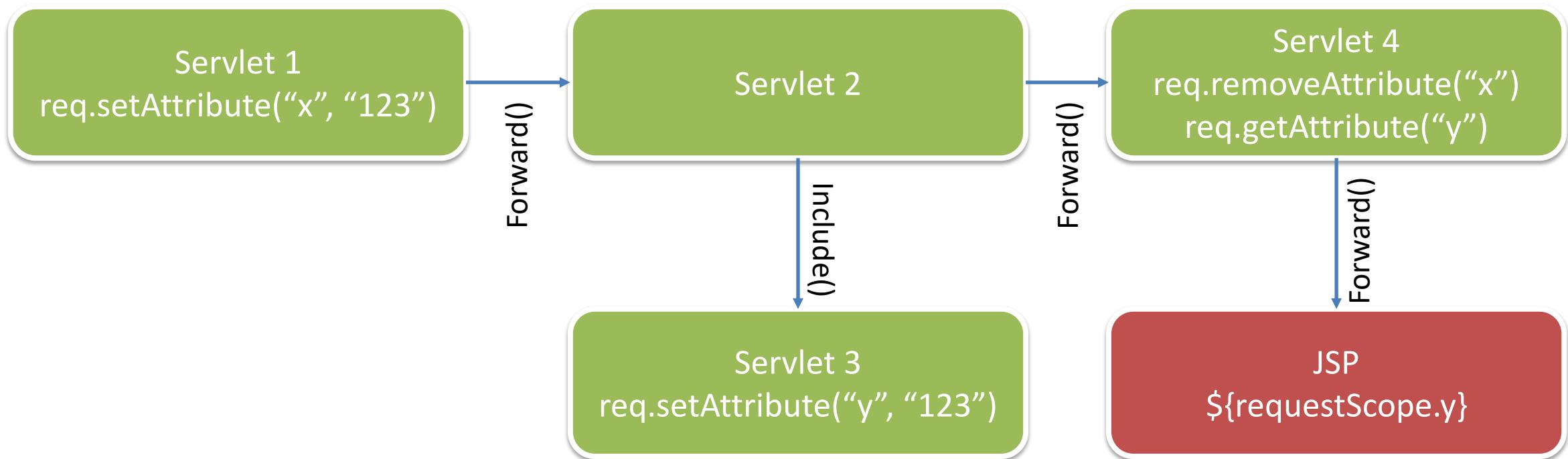
❑ Trong Servlet

- ❖ **<scope>. setAttribute(String, Object)**
 - Thêm biến vào scope
- ❖ **<scope>.getAttribute(String): Object**
 - Lấy giá trị của biến scope
- ❖ **<scope>. removeAttribute(String)**
 - Xóa biến khỏi scope

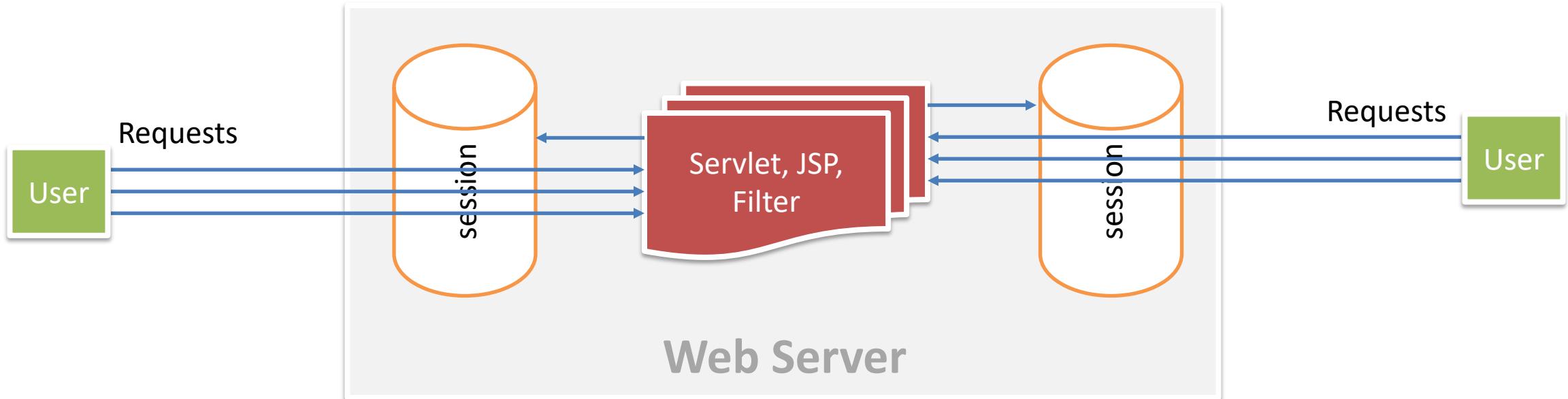
❑ Trong JSP (xyz=page|request|session|application)

- ❖ **<c:set var="name" value="" scope="xyz"/>**
 - Thêm biến vào scope
- ❖ **\${xyzScope.attribute} hoặc \${xyzScope['attribute']}**
 - Lấy giá trị của biến scope
- ❖ **<c:remove var="name" scope="xyz"/>**
 - Xóa biến khỏi scope

- ❑ **Request** (`HttpServletRequest`) được sử dụng để chia sẻ dữ liệu giữa các thành phần hoạt động trên cùng một dây chuyền `include()` và `forward()`
- ❑ Tất cả attribute trong request sẽ bị giải phóng khi request kết thúc

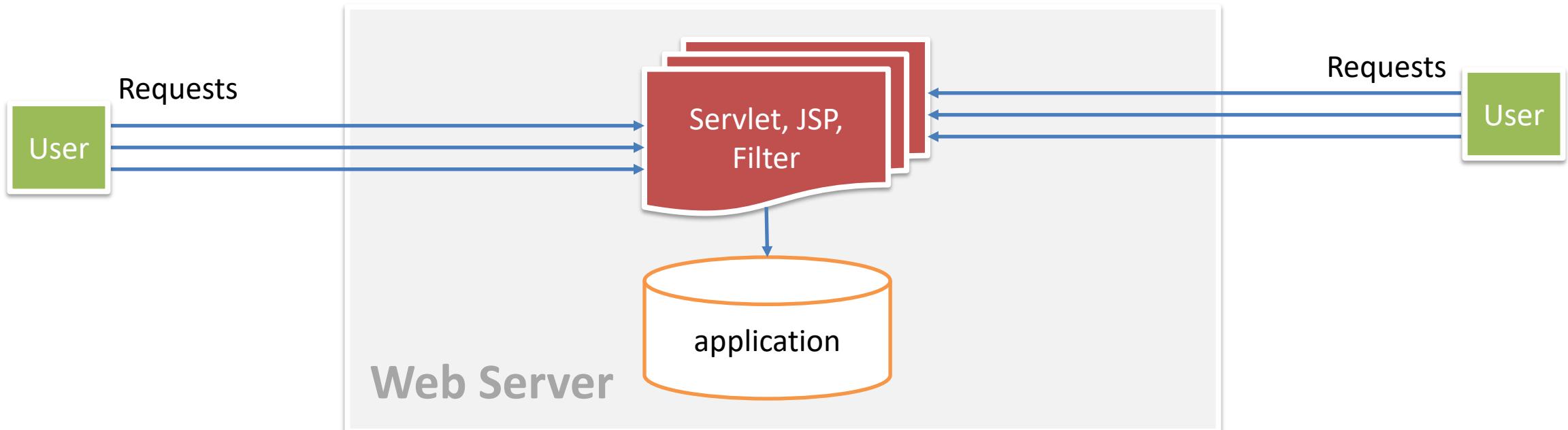


- ❑ Session (HttpSession) được sử dụng để chia sẻ dữ liệu giữa các request hoạt động trong cùng một phiên làm việc.
- ❑ Khi session timeout (hết hạn) thì các attribute sẽ bị giải phóng



- ❑ Có thể sử dụng session để duy trì giỏ hàng điện tử, thông tin đăng nhập, thông tin cá nhân hóa...

- ❑ Application (ServletContext) được sử dụng để chia sẻ dữ liệu cho tất cả mọi request của tất cả mọi user với mọi thời điểm.
- ❑ Attribute chia sẻ qua application chỉ bị giải phóng khi ứng dụng web bị shutdown



- ❑ Có thể sử dụng application để duy trì bộ đếm khách thăm web, Cache dữ liệu dùng chung để giảm truy vấn CSDL,...



DEMOSTATION

- ❑ @WebListener được tạo ra để **lắng nghe và xử lý những sự kiện** xảy ra bên trong ứng dụng web.
 - ❖ Khi nào ứng dụng bắt đầu, kết thúc
 - ❖ Khi nào một phiên làm việc bắt đầu, hết hạn
- ❑ **ServletContextListener** được sử dụng để xử lý 2 sự kiện: bắt đầu và kết thúc ứng dụng
- ❑ **HttpSessionListener** được sử dụng để xử lý 2 sự kiện: session được tạo ra và session hết hạn
- ❑ ...

```
@WebListener
public class AppListener implements ServletContextListener{
    @Override
    public void contextDestroyed(ServletContextEvent sce) {
        // TODO Chạy trước khi ứng dụng web bị shutdown
        ServletContext app = sce.getServletContext();
    }
    @Override
    public void contextInitialized(ServletContextEvent sce) {
        // TODO Chạy ngay sau khi ứng dụng khởi động
        ServletContext app = sce.getServletContext();
    }
}
```

```
@WebListener
public class UserListener implements HttpSessionListener{
    @Override
    public void sessionCreated(HttpSessionEvent e) {
        // TODO Chạy ngay sau khi một phiên làm việc được tạo
        HttpSession session = e.getSession();
    }
    @Override
    public void sessionDestroyed(HttpSessionEvent e) {
        // TODO Chạy ngay trước khi phiên làm việc hết hạn
        HttpSession session = e.getSession();
    }
}
```

```
@WebListener
```

```
public class AppListener implements ServletContextListener, HttpSessionListener {  
    @Override  
    public void sessionCreated(HttpSessionEvent e) {  
    }  
    @Override  
    public void sessionDestroyed(HttpSessionEvent e) {  
    }  
    @Override  
    public void contextDestroyed(ServletContextEvent e) {  
    }  
    @Override  
    public void contextInitialized(ServletContextEvent e) {  
    }  
}
```



DEMOSTATION

- ✓ Giới thiệu Scope
- ✓ Chia sẻ dữ liệu qua HttpServletRequest
- ✓ Chia sẻ dữ liệu qua HttpSession
- ✓ Chia sẻ dữ liệu qua ServletContext
- ✓ Chia sẻ dữ liệu qua PageContext
- ✓ WebListener





Cảm ơn