

SCOPES, ATTRIBUTES AND PARAMETERS

4 Scopes:

- **Request context**
- **Servlet context**
- **Session context**
- **Application context**

Scope = Опсег на важење т.е. опсег на пристапност, т.е. кој може да пристапува или не до атрибутите или параметрите.

Параметри

Не се променуваат, кога ги читаш **return String** враќаат.

Атрибути

Се променуваат, кога ги читаш **return Object** враќаат.

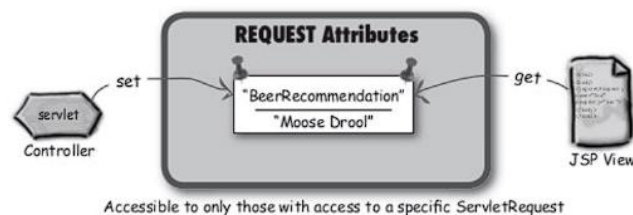
Request Context – Барање

Пристап има: И параметрите и атрибутите се достапни од `HttpServletRequest`, можеш да пристапиш до нив само во едно request барање. Со други зборови сите барања си имаат свои уникатни контексти.

Параметри:

```
public void doGet(HttpServletRequest request, HttpServletResponse response){  
    request.getParameter("page");  
}
```

Атрибути:



```
HttpServletRequest.setAttribute("student", new Student());  
Student student = (Student) HttpServletRequest.getAttribute("student");
```

Note: При земање на атрибут, мораш да го кастираш бидејќи го добиваш како Object.

Note: Request-от е различен при редирекција, но е ист при dispatching.

Note: Attributes are thread safe!

Servlet Context – Сервлет aka ServletConfig

Пристап има: Сите клиенти во рамките на еден сервлет. Со други зборови, сите сервлети си имаат свои сервлет контексти.

Параметри:

Два начина на декларирање на сервлет параметри:

- pom.xml (Deployment Descriptor - DD)
- Во анотацијата над сервлетот

```
<servlet>
  <servlet-name>scopeServlet</servlet-name>
  <servlet-class>finki.wp_servlets.ScopeServlet</servlet-class>
  <init-param>
    <param-name>max-users</param-name>
    <param-value>5</param-value>
  </init-param>
  <init-param>
    <param-name>min-users</param-name>
    <param-value>2</param-value>
  </init-param>
</servlet>
```

или

```
@WebServlet(name = "scopeServlet", value = "/ss",
  initParams = {
    @WebInitParam(name = "max-users", value = "5"),
    @WebInitParam(name = "min-users", value = "2")
  })
```

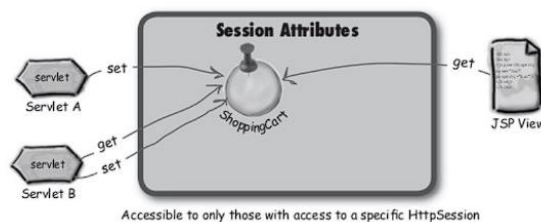
```
ServletConfig config = getServletConfig();
String servletMaxUsers = config.getInitParameter("max-users");
```

Note: Сервлетите немаат атрибути.

Session Context - Сесија

Пристап има: Било кои сервлети, но само од еден клиент. Со други зборови, сите клиенти имаат свои сесии.

Атрибути:



```
HttpSession session = request.getSession();
Integer counter = (Integer) session.getAttribute("user-counter");
session.setAttribute("counter", ++counter);
```

или

```
request.getSession().getAttribute("user-counter");
```

Note: Attributes are not thread safe! Два таба истовремено да повикаме - не е thread safe.

Application Context – Апликација aka ServletContext

Пристап има: Сите сервлети, сите филтери, сите клиенти, целосно јавно достапни. Use-case на овие би бил конекција до база, таа треба да биде достапна до сите сервлети.

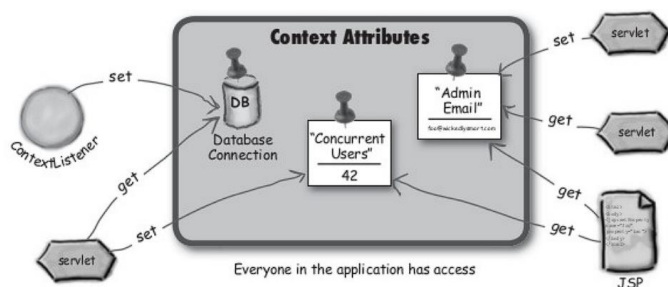
Параметри:

Начин на декларирање на ServletContext параметри е во pom.xml (Deployment Descriptor - DD).

```
<context-param>
  <param-name>max-users</param-name>
  <param-value>15</param-value>
</context-param>
<context-param>
  <param-name>min-users</param-name>
  <param-value>10</param-value>
</context-param>
```

```
ServletContext context = getServletContext();
String contextMaxUsers = context.getInitParameter("max-users");
```

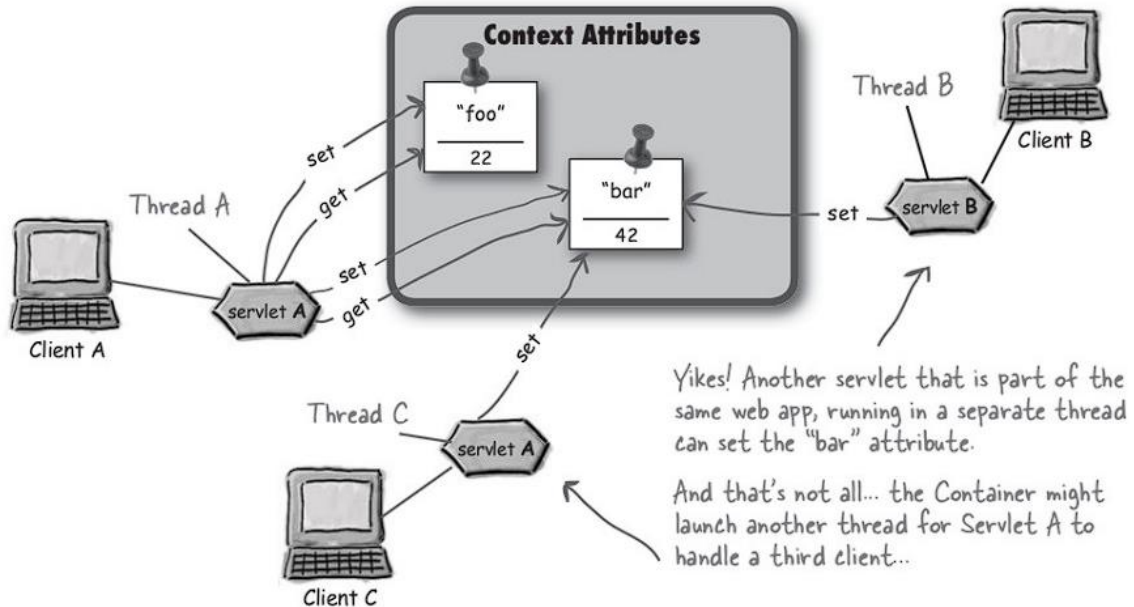
Атрибути:



```
Servlet context = getServletContext();
Integer counter = (Integer) context.getAttribute("counter");
context.setAttribute("counter", ++counter);
```

Note: Attributes are not thread safe!

Context scope isn't thread-safe!



EXTRA CONTENT

Attributes are not parameters!

	Attributes	Parameters
Types	Application/context Request Session <i>There is no servlet-specific attribute (just use an instance variable).</i>	Application/context init parameters Request parameters <u>Servlet init parameters</u> <i>No such thing as session parameters!</i>
Method to set	setAttribute(String name, Object value)	You CANNOT set Application and Servlet init parameters—they're set in the DD, remember? (With Request parameters, you can adjust the query String, but that's different.)
Return type	Object	String ← Big difference!
Method to get	getAttribute(String name) <i>Don't forget that attributes must be cast, since the return type is Object.</i>	getInitParameter(String name)

Attribute Scope

	Accessibility (who can see it)	Scope (how long does it live)	What it's good for
Context (NOT thread-safe!)	Any part of the web app including servlets, JSPs, ServletContextListeners, ServletContextAttributeListeners.	Lifetime of the ServletContext, which means life of the deployed app. If server or app goes down, the context is destroyed (along with its attributes).	Resources you want the entire application to share, including database connections, JNDI lookup names, email addresses, etc.
HttpSession (NOT thread-safe!)	Any servlet or JSP with access to this particular session. Remember, a session extends beyond a single client request to span multiple requests by the same client, which could go to different servlets.	The life of the session. A session can be destroyed programmatically or can simply time-out. (We'll go into the details in the Session Management chapter.)	Data and resources related to this client's session, not just a single request. Something that requires an ongoing conversation with the client. A shopping cart is a typical example.
Request (Thread-safe)	Any part of the application that has direct access to the Request object. That mostly means only the Servlets and JSPs to which the request is forwarded using a RequestDispatcher. Also Request-related listeners.	The life of the Request, which means until the Servlet's service() method completes. In other words, for the life of the thread (stack) handling this request.	Passing model info from the controller to the view... or any other data specific to a single client request.