Listener, Exception Handling

Sandeep, SunBeam

PowerPoint

Event Listeners

- Event is an instance which is used to send as a notification if make changes in state of another instance.
- The servlet specification includes the capability to track key events in your Web applications through *event listeners*.
- This functionality allows more efficient resource management and automated processing based on event status.
- There are three levels of servlet events
    1. **Servlet context-level (application-level) event**
    2. **Session-level event**
    3. **Request-level event**
- Each of these three levels has two event categories:
    1. **Lifecycle changes**
    2. **Attribute changes**

Event Category And Java Interface

Sr. No.	Event Category	Java Interface
1	Servlet context lifecycle changes	javax.servlet.ServletContextListener
2	Servlet context attribute changes	javax.servlet.ServletContextAttributeListener
3	Session lifecycle changes	javax.servlet.http.HttpSessionListener
4	Session attribute changes	javax.servlet.http.HttpSessionAttributeListener
5	Request lifecycle changes	javax.servlet.http.ServletRequestListener
6	Request attribute chanages	javax.servlet.ServletRequestAttributeListener

Typical Event Listener Scenario

- Consider a Web application comprising servlets that access a database. A typical use of the event listener mechanism would be to create a servlet context lifecycle event listener to manage the database connection.
- This listener may function as follows:
    1. The listener is notified of application start-up.
    2. The application logs in to the database and stores the connection object in the servlet context.
    3. Servlets use the database connection to perform SQL operations.
    4. The listener is notified of imminent application shutdown (shutdown of the Web server or removal of the application from the Web server).
    5. Prior to application shutdown, the listener closes the database connection.

Event Listener Declaration and Invocation

- Assume that **MyConnectionManager** implement the **ServletContextListener** interface.

- Event listeners are declared in the application web.xml deployment descriptor through **<listener>** elements under the top-level **<web-app>** element. Each listener has its own **<listener>** element, with a **<listener-class>** sub element specifying the class name.

```
<web-app>

        <display-name>MyListeningApplication</display-name>

        <listener>

                <listener-class>com.sunbeam.MyConnectionManager</listener-class>

        </listener>

</web-app>
```

Exception Handling

- An HTTP error code or an exception thrown by a serlvet can be mapped to a resource bundled with the application to customize the appearance of content when a servlet generates an error.
- This is done using error pages. These pages should be configured in web.xml.
- For HTTP error code, we can do mapping as follows:

```
<error-page>
        <error-code>404</error-code>
        <location>/Error-404.jsp</location>
</error-page>
```

- For exception, we can do mapping as follows:

```
<error-page>
        <exception-type> javax.servlet.ServletException </exception-type>
<error-page>
```

Exception Handling

- Before servlet container invokes the servlet to handle the exception, it sets some attributes in the request to get useful information about the exception, some of them are :

    1. **javax.servlet.error.exception**
    2. **javax.servlet.error.status_code**
    3. **javax.servlet.error.servlet_name**
    4. **javax.servlet.error.request_uri**

Auto-Refresh/Wait Pages

- Another response header technique that is uncommon but helpful is to send a wait page or a page that will auto-refresh to a new page after a given period of time.
- This tactic is helpful in any case where a response might take an uncontrollable time to generate response.
- The entire mechanism revolves around setting the Refresh response header.
- The header can be set using the following:
  - response.setHeader("Refresh", "time; URL=url");
  - time” should be replaced with the amount of seconds
- For example:

```
response.setHeader("Refresh", "10; URL=http://127.0.0.1/foo.html");
```

Thank you