Event Listeners

Kevin Jones

@kevinrjones www.rocksolidknowledge.com

Overview

Reminder of Annotations

Introduction to Listeners

Writing Listeners

Using Annotations for Listeners

Annotations - Reminder

Can replace or augment settings

- web.xml

Only these are scanned

- classes in WEB-INF/classes
- JARS in WEB-INF/lib

Can set metadata-complete='true' on web-app

- Stops any scanning for annotations

Annotations - Reminder

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.IO.*;

@WebServlet("/home")
public class OurServlet extends HttpServlet {
  public init() {
    // Perform any instance initialisation
  }

  public void doGet(HttpServletRequest req,
    HttpServletResponse resp)
    throws ServletException, IOException {

    // Process the request and generate the response
  }
  ...
}
```

Annotations - Reminder

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.IO.*;

@WebServlet(urlPatterns = {"/home", "*.do"})
public class OurServlet extends HttpServlet {
  public init() {
    // Perform any instance initialisation
  }

  public void doGet(HttpServletRequest req,
    HttpServletResponse resp)
    throws ServletException, IOException {

    // Process the request and generate the response
  }
  ...
}
```

Annotations - Reminder

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.IO.*;

@WebServlet(urlPatterns = {"/home", "*.do"},
            initParams =
            {@WebInitParam(name="connstr",
                           value="server=...")})
public class OurServlet extends HttpServlet {
  public init() {
    connstr = getInitParameter("connstr");
  }

  ...
}
```

What Annotations Exist?

@WebServlet

@WebFilter

@WebInitParam

@WebListener

@MultipartConfig

Event Listeners

Events are fired at users at appropriate times

- Application Events at app start/end
- Session Events at Session start/end
- Request Events at Request start/end
- Attribute events when attribute added/removed
- Listeners listen for events

Listener Implementation

Listeners Implement Appropriate Interface

- ServletContextListener
- ServletContextAttributeListener
- HttpSessionActivationListener
- HttpSessionAttributeListener
- ServletRequestListener
- ServletRequestAttributeListener
- Passed appropriate "Event" object

Registering Listeners

Listeners are registered in web.xml

- Fire in order of listing for activation
- Fire in reverse order for deactivation

```
<listener>
  <listener-class>
     com.mantiso.WhitePagesListener
  </listener-class>
</listener>
```

Application Listener

Implements `ServletContextListener`

Fired at application Start/End

Passed a ServletContextEvent

- Gives access to **`ServletContext`**

Application Listener

```
public class WhitePagesServletListener
             implements javax.servlet.ServletContextListener
{
    java.util.Hashtable whitePages;

    public void contextDestroyed(ServletContextEvent sce)
    {
    }

    public void contextInitialized(ServletContextEvent sce )
    {
        whitePages = new java.util.Hashtable();
        sce.getServletContext().setAttribute("addrbook", whitePages);
    }
}
```

Session Listener

Fired after Session start

Fired before Session ends

- In servlet 2.3 spec fired *after* session ended

Passed and HttpSessionEvent gives access to

- **`HttpSession`**

Session Listener

```
public class WhitePagesSessionListener
             implements javax.servlet.HttpSessionActivationListener
{
    public void sessionCreated(HttpSessionEvent se)
    {
    }

    public void sessionDestroyed(HttpSessionEvent se)
    {
    }
}
```

Request Listener

Fired before request comes into scope

- before it enters first filter in the chain

Fired before request goes out of scope

- after its left the last filter in the chain

ServletRequestEvent Gives access to

- **`ServletContext`**
- **`ServletRequest`**

Request Listener

```
public class WhitePagesRequestListener
             implements javax.servlet.ServletRequestListener
{
    public void requestInitialized(ServletRequestEvent sre)
    {
    }

    public void requestDestroyed(ServletRequestEvent sre)
    {
    }
}
```

Listener Issues - Exceptions

Application initiated exceptions

- e.g. adding an object to a session causes listener to throw exception
- Exception passed to application
- No more listeners fired

Container initiated exceptions

- e.g. Application listener causes exception at start-up
- Not handled by application
- May cause container to make application unavailable
- Container will return HTTP 500 status code

Listeners Annotation

@WebListener

Summary

Web applications fire 'events'

EventListeners can respond to these

Manage application/session/request start/end

Configure in web.xml or annotations

What’s Next