CYBAGE
Delivering Value. Scientifically.

Servlet Features

Presented by : Asfiya Khan
(Sr. Technical Trainer)

www.cybage.com

Agenda

- **Servlet Config**
- **Servlet Context**
- **Request Dispatcher**

Servlet Config

- The web container carries out following activities :
- It loads & instantiates a Servlet
- It initializes the servlet
- Delivers requests from clients

- Certain Servlet parameters can be initialized by reading from configuration files.
- To read this configuration environment during initialization a Servlet -Configuration object is used by the servlet
- Servlet API provides various means of accessing Servlet Config object

Methods of ServletConfig

- **String getInitParameter(String name)**: returns a String value initialized parameter, or NULL if the parameter does not exist.
- **Enumeration getInitParameterNames()**: returns the names of the servlet's initialization parameters as an Enumeration of String objects, or an empty Enumeration if the servlet has no initialization parameters.
- **ServletContext getServletContext()**: returns a reference to the ServletContext
- **String getServletName()**: returns the name of the servlet instance

Web-Init Parameters

- The @WebInitParam annotation is used to specify an initialization parameter for a servlet or a filter. It is used in conjunction with the @WebServlet and @WebFilter annotations.

Syntax of @WebInitParam Annotation:

```
@WebInitParam (
    name = <name>,
    value = <value>,
    description = <value>
)
```

```
@WebServlet(
        urlPatterns = "/uploadFiles",
        initParams = @WebInitParam(name = "location", value = "D:/Uploads")
)
```

Servlet Context

- Interface **ServletContext** is used by servlet to communicate with web container
- Servlet Context allows servlets in an application to share data
- A servlet can log events, store attributes that are accessible by other servlets

@WebListener

- The **@WebListener** annotation is used to register a class as a listener of a web application.
- The annotated class must implement one or more of the following interfaces: javax.servlet.ServletContextListener
- javax.servlet.ServletContextAttributeListener
- javax.servlet.ServletRequestListener
- javax.servlet.ServletRequestAttributeListener
- javax.servlet.http.HttpSessionListener
- javax.servlet.http.HttpSessionAttributeListener

Request Dispatcher

- Defines an object that receives requests from the client

  and

- Sends them to any resource (such as a servlet, HTML file, or JSP file) on the server
- The servlet container creates this RequestDispatcher object
- It is used as a wrapper around a server resource located at a particular path
- Methods : include(…,….)   & forward(….,…..)

Response Redirection

- If you want the browser to initiate a new request for a resource which is not available on same web application ,then use **response.sendRedirect()**

- public void sendRedirect (String path) : This method is available for **HttpServeltResponse** object

CYBAGE

Difference between sendRedirect() & forward()

Forward	Redirect
In forwarding, the destination resource must be java enabled resource only.	In redirection, the destination resource can be either java or non-java resource also.
In forwarding, both source and destination resource must run within the same server. It is not possible to communicate across the server.	In redirection, it is possible to communicate, either within the serve or even across the server.
In forwarding, both the data and control are forwarded to destination (by default).	In redirection, only control is redirected, but not the data (by default).

Any questions?

CYBAGE
Delivering Value. Scientifically.

Thank You!

www.cybage.com