

Exercises: web.xml

1. Make a new Eclipse project. Copy any servlet you wrote for a previous exercise, give it a custom address in web.xml, and access it using that address.

For the remaining problems, you are the CTO of Enroff, a company that is so popular that other big companies keep buying you out for higher-and-higher prices. That's the good news. The bad news: the company name keeps changing, making you keep changing your home page.

2. Make a servlet that reads the company name from an initialization parameter and then uses it in the same page.

3. Repeat problem 2 (using an init parameter), but use a JSP page instead of a servlet.

4. Make sure that this JSP page is displayed if the user requests *http://host/yourCompany/*, with no filename specified.

5. Turn off the invoker servlet and verify that it is disabled.

6. Have a servlet read the company name from an init parameter, store it in the servlet context, and use it in the servlet's output. Have two JSP pages that use the same company name, but don't have the JSP pages re-read the init parameter.

7. Be sure nothing bad happens if someone tries to access the above JSP pages before the above servlet that sets the company name.

8. Make a servlet or JSP page that reads a request parameter and prints a message saying if that value is a substring of the current company name. If the request parameter is missing, display a designated page that is not accessible directly (i.e., there is no URL that the client can supply that directly yields the error page). Don't use any try/catch blocks or explicit checks for null in the JSP page.