

Exercises: Request Headers

Do whichever exercises fit your background and interests. They are ordered in approximate order of difficulty. As always, feel free to experiment on your own with other related exercises.

Make a new Eclipse project called `exercises-request-headers` (or some such).

1. Go to the `request-headers` project and copy the servlet that shows all request headers (`ShowRequestHeaders`). Paste it in a package of your new project. Access it from both Firefox and Internet Explorer. Use the URL `http://localhost/project-name/show-request-headers`.
2. Make a tiny Web page that contains a hypertext link to the `ShowRequestHeaders` servlet. To make the HTML page, you can copy `hello.html` from my test-app project, or copy one of the forms you had earlier. When you click to get to the `ShowRequestHeaders` servlet, what new header shows up that wasn't there previously?
 - Note: if you are bit rusty with HTML, hypertext links look like this:
`some text to display`
The browser shows the text, but takes you to the address when you click on it.
3. Some informational sites want users to access pages in a certain order. They want to prohibit users from jumping directly to a bookmarked page later in the sequence because the data on the pages may have changed since the user bookmarked the page.

Create two pages. The first page should be a normal HTML page with a link to the second page (which should be a servlet). If a user accesses the first page and then follows the link to the second page, it works normally. But, if the user directly types in the address of the second page (or follows a link from a page with a different name than the first page), they should get sent back to the first page automatically. Hints:

- Use `response.sendRedirect` to send users back to page1. See the description of `sendRedirect` on the bottom of the previous set of exercises.
 - It is not necessary to make page1 be a servlet, but page2 must be.
4. Write a servlet that just says "Hello." Use a red background and a yellow foreground for Internet Explorer users; use a yellow background and a red foreground for Firefox and other non-IE users. If you are a bit rusty with HTML, you set colors as follows:
`<body bgcolor="colorName" text="colorName">` or
`<body bgcolor="#rrggbb" text="#rrggbb">`
(where r, g, and b are hex values for the red, green and blue components. I.e., `#ff00ff` is magenta -- 255 for red, 0 for green, and 255 for blue).