Exercise 4: Web Chat

# Cork Board UI

Remember the Cork Board exercise? You have a simple object that manages a list of messages. You are going to add a web front end for this object allowing users to communicate with their browsers instead of RMI.

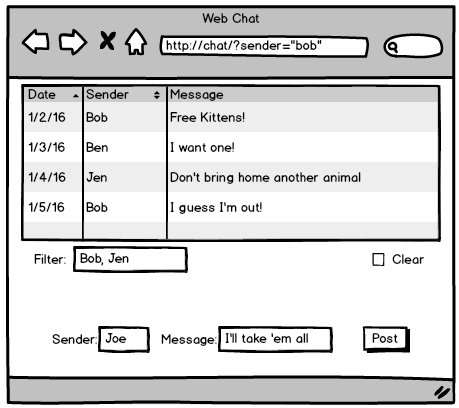
# Exercise

The web user should be able to activate all functions of the CorkBoard:

* See a list of messages
* Post a new message to the board
* Clear the board
* And new – ask the server to filter any return lists. Send a list of names to the server and it will only return messages posted by those names.

You can be as creative as you like. All you really need is ONE servlet that generates ONE page with ONE form that has ONE submit button.

Here is what my super-simple solution looks like:



# Hints

Start with the servlet:

* Create your basic project and get a HelloWorld servlet going.
* Think about the parameters being passed to the servlet in the URL. What are their names?
* Code the servlet to pull these parameters from the request and print them to the console.
* Use the browser to send requests to the servlet. Type your parameters in the URL line.

Add in the CorkBoardImpl:

* Find your old code and copy it into the project.
* You only need the server implementation … no RMI needed in this exercise.
* Flesh out your servlet to talk to the CorkBoard implementation.
* Print outputs to the console and return a simple “OK” to the browser.
* Use the browser to test the servlet.

Experiment with a static HTML file:

* Use a list of text lines to show example messages. Try your hand at making a table instead.
* Add a form and all fields to the HTML file.
* Press the send button and check the parameters in the URL line.
* Use this static form to test your servlet. (Still only an OK response)

Generate the HTML:

* Add print statements to your servlet to return the full HTML.
* Copy lines from the static HTML page you just created.
* Test, test, test! Try chatting with yourself among multiple browser windows.

