

Exercises: History

Make a new Google Web Application Project called "ExercisesGwtHistory" (or some such -- my solution set is called GwtHistoryExercises). Even though you are going to do some server-side coding, you can still delete the classes out of the .server package and delete the shared package altogether. Don't forget to edit/erase the web.xml entries that refer to the GWT-RPC classes that you deleted. If you can do the first two problems, you have the basic idea of history handling. The others combine the idea of history handling with some of the ideas in earlier sections.

1. Designate an area on the host HTML page called "addition". If the history token starts with "add=", assume it is followed by an integer and display "2 + whatever number was in the history token = sum". If there is no history token or it doesn't start with "add=", it should display a default "2 + 2 = 4". Test this by changing the number directly on the URL. Hint: to get a feel for the expected functionality, run GwtHistoryExercises (but don't look at the code), and try both these URLs:

`http://127.0.0.1:8888/GwtHistoryExercises.html?gwt.codesvr=127.0.0.1:9997`

`http://127.0.0.1:8888/GwtHistoryExercises.html?gwt.codesvr=127.0.0.1:9997#add=5`

2. Designate an area on the host HTML page called "greeting". If the history token starts with "greet=" (i.e., the URL ends in something like #greet=Joe), assume it is followed by some name for someone to greet and display "We greet you dear XXX", where XXX is the name from the token. If there is no history token or it doesn't start with "greet=", it should display a default "We greet you dear No one!".
3. Create a GWT-RPC service called ServerDateService with one method: public Date getServerDate(). Return the current date of the server using that method.
4. Designate an area on the host HTML page called "server-date". If the history token starts with "days=", assume it is followed by an integer and display "Server date was xxx, plus *numberOfDaysInTheToken* is *newDate*". To see what I have in mind, run GwtHistoryExercises with #days=5 on the end of the URL.
5. Designate an area on the host HTML page called "buttons". Place 3 buttons in it: "Show 2+30 Page State", "Greet GWT class", and "Add 20 days to Server Date". Implement those buttons using the History API to put the page in an appropriate navigational state corresponding to the names of the buttons.
6. Test your code in production mode and then deploy it to Tomcat.