

# Exercises: RPC

Make a new Google Web Application Project called “ExercisesRpc” (or some such). Note that my solution set is called GwtRpcExercises. If you can do the first two problems, you have the idea of RPC. The other problems are moderately important in that they let you practice with serializable custom classes, but the first two are the *really* critical ones.

- 1.** Make a server-side method `flipCoin()` that randomly returns either "Heads" or "Tails". Set up RPC so that you can press a button that invokes the server-side method and puts the result into the page. This super-simple example is the most important one of the bunch, since it requires you to get all the structure in place.
- 2.** Make a server-side method `flipCoins(int n)` that returns an n-length array of randomly selected "Heads" or "Tails" values. Make a button that calls the method with `n=5` and builds a bulleted (<ul>) list out of the results.
- 3.** Create a Customer class for sending to the client. Give your class bean properties (i.e., getters/setters) `id (int)`, `first name (String)`, `last name (String)`, and `balance (double)`. Remember to have it implement `Serializable` and to have a zero-arg constructor. In your server code, create a very small array or List of customers.
- 4.** Make a server-side method `getRichestCustomer()` that returns the Customer with the highest balance. Make a button that calls the method and builds some output from the result.
- 5.** Make a similar method `getPoorestCustomer()`. Make a button that calls the method and builds some output from the result.
- 6.** Update the client-side code (`onSuccess`) that handled the richest and poorest customers. Now, store the richest and poorest customers in instance variables for later use.
- 7.** Make a server-side method `redistribute(int id1, int id2)` that finds the customers with the given ids, sets each of their balances to the average of the two balances, then returns both the customers in a List. Make a button that, when pressed, tries to call `redistribute` on the already-stored richest and poorest customers. If either one of the instance variables (from #6 above) is null, just give an error message.
- 8.** Test in production mode. Test in deployed mode on Tomcat.