Exercises: MVP Framework

Make a copy of the GwtMVP2 Eclipse project and name it “ExercisesMVP2” (or some such – my solution set is called GwtMVP2Excercises).

1. I explained during lecture time that the view should ideally not save any state. However, the Google-provided example violates this practice. The HelloViewImpl.java stores the ‘name’ as a local instance variable. Fix this. Remember it’s the Presenter’s responsibility to cache data (i.e., to contain state). Remember that in GWT MVP framework, the Presenter interface is defined by the view and implemented by the Activity associated with that view.

2. Create a button inside the HelloView. On click of that button, an alert box should pop up showing a random number within some hardcoded range. Once that random number is shown, any subsequent clicks on the button should show the *same* number as it showed the first time, i.e., no new random number should be generated.

3. Create a new Place called SameRandomNumberPlace. It should know how to interpret a single token string that will be a range within which to generate a random number. Create SameRandomView and its UiBinder-based implementation. The view should be very simple: just a place to display a number, nothing more.
   Place an Anchor widget into the HelloView. When that anchor is clicked, it should go to the SameRandomNumberPlace, passing it the range that’s currently stored in HelloActivity. The end result should be such that the SameRandomView is displayed with the random number within the passed-in range.

   Where did you save the random number for the SameRandomNumberActivity? What can you do such that every time you navigate to the SameRandomNumberAvtivity, the random number generated once stays the same?

   Even if you copy/paste the same logic from #2 and make the number stay the same (by doing whatever), when you refresh the page, a new random number will show up. Why is that?

4. Create a button in the SameRandomView with the caption “Show Range”. Upon clicking on that button, an alert should pop up showing the user the range within which the random number was created.

5. Test the project in production mode and deployed mode.