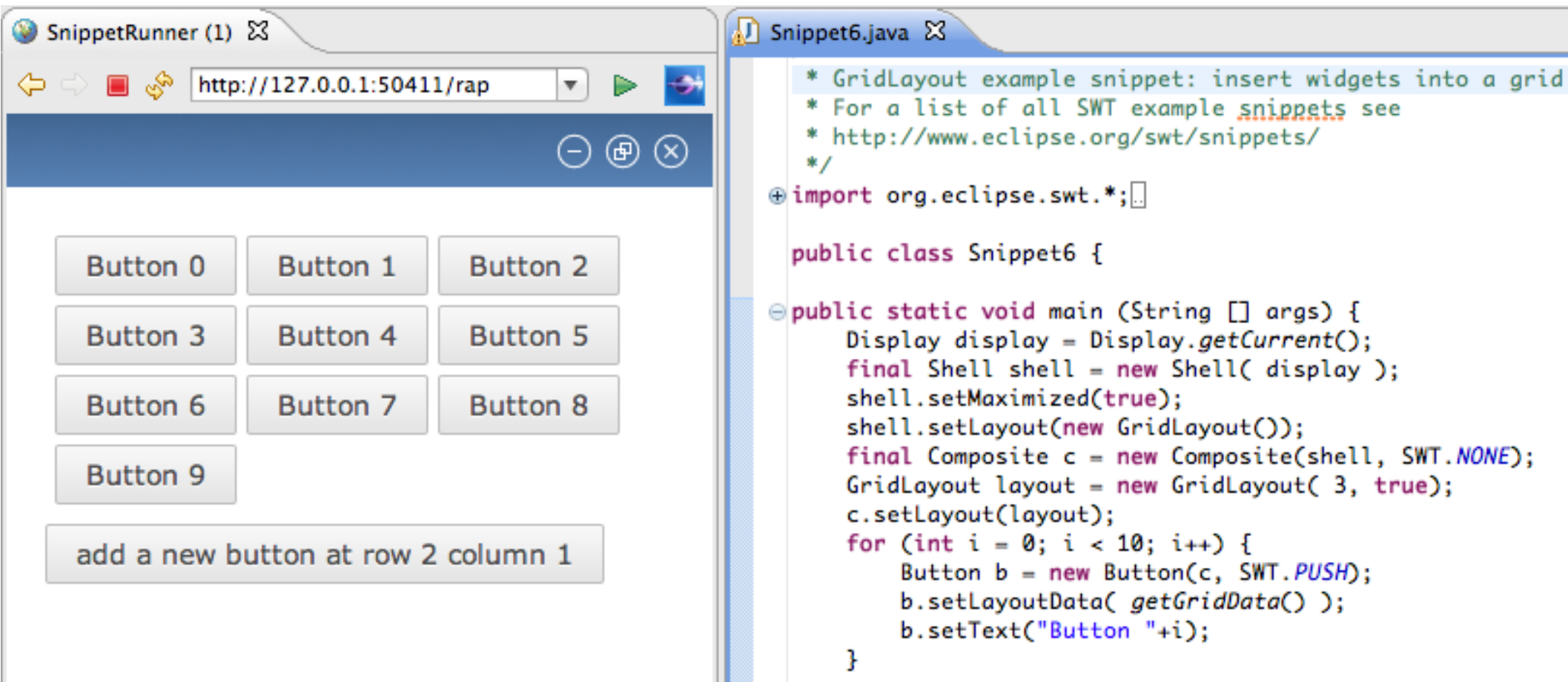


# What has happened?

# Style Footprint Integration Interactivity



The image shows a web browser window on the left and a code editor window on the right. The browser window, titled "SnippetRunner (1)", displays a web application with a grid of buttons. The buttons are arranged in three rows: the first row has "Button 0", "Button 1", and "Button 2"; the second row has "Button 3", "Button 4", and "Button 5"; the third row has "Button 6", "Button 7", and "Button 8". Below the grid is a single button labeled "Button 9". At the bottom of the browser window is a text input field containing the text "add a new button at row 2 column 1". The code editor window, titled "Snippet6.java", shows the source code for the application. The code is a Java class named "Snippet6" with a "main" method. The "main" method creates a shell window, sets its layout to a grid, and then creates a composite widget with a grid layout. It then creates a loop of buttons, each with a unique text label.

```
* GridLayout example snippet: insert widgets into a grid
* For a list of all SWT example snippets see
* http://www.eclipse.org/swt/snippets/
*/
import org.eclipse.swt.*;

public class Snippet6 {

    public static void main (String [] args) {
        Display display = Display.getCurrent();
        final Shell shell = new Shell( display );
        shell.setMaximized(true);
        shell.setLayout(new GridLayout());
        final Composite c = new Composite(shell, SWT.NONE);
        GridLayout layout = new GridLayout( 3, true);
        c.setLayout(layout);
        for (int i = 0; i < 10; i++) {
            Button b = new Button(c, SWT.PUSH);
            b.setLayoutData( getGridData() );
            b.setText("Button "+i);
        }
    }
}
```

Nice out of the box style

## Runtime choices

Standard Java Classpath (war deployment)

- 1 jar, 3 MB

Minimal OSGi

- 8 bundles, 5 MB

**RECOMMENDED**

minimal configuration with Eclipse Workbench

- 33 bundles, 14 MB

**Ready for cloud and data center**

**No more UI thread in default execution mode**

**Threadlocals work as expected (no more hassle to integrate with Spring, Security Context, ...)**

**Session Clustering with failover works out of the box**

## Highly interactive

ClientScripting - Validation / local processing of events in JavaScript

KeyEvents - Fine grain control of keys being sent to the server for processing

Note: ClientScripting is not part of RAP 1.5, it will be in 2.0