Welcome to sillyview

1. Audience

This site is of potential use to Java programmers who want to replace hardcoded Swing user interfaces with a model-view-controller architecture based on HTML models.

This site does not claim to have anything useful or interesting to non-programmers.

2. What is sillyview?

sillyview is a Swing frontend for HTML specified user interfaces. In particular, templated user interfaces, such as those provided by the Apache Jakarta Project's Velocity Template Engine.

In brief: Write your interface once, in easy-to-maintain HTML and re-use it in your Java applets, applications, and servlets with no change to the program logic. Make multiple HTML interfaces available to the end-user and you have a "skinnable" program.

sillyview solves two problems:

- my Swing code is hard to write and maintain, and
- I want to use the same program logic for applications, applets, and servlets

Using sillyview adds a level of abstraction to your code which separates your program logic and data (the model), processing user events (the controller) and your user interface (the view). Furthermore, many of sillyview's features are geared towards the use of HTML to draw your user interface, whether inside an application, applet, or servlet.

sillyview uses tokens which are variables and their values (often called name/value pairs) and string processing to automatically change the view when the model is updated.

3. Real World Example

sillyview was created during the upgrade of Pavlov 1.0 to 1.1 in order to help along the process of making a Servlet version of Pavlov. At one point in development, Pavlov had 165 classes and 22,400 lines of code. After integrating sillyview, and removing nasty hardcoded Swing user interface pieces, there were 100 classes and 16,000 lines of code. The result: more functionality and a code base much easier to maintain.

Copyright © 2004 T.J. Willis All rights reserved.

4. I want to keep my legacy Swing UI's

As long as a class implements the WidgetView interface, you can use it as a view. This gives you the option of doing a small rewrite and having your cake and eating it too. I.e., your legacy interface will be hot-swappable with other sillyview interfaces at runtime.

Copyright © 2004 T.J. Willis All rights reserved.