First Steps With Clojure & Swing

Swing is the GUI standard for Java. Clojure is the awesomeness standard for Java. Let's make an awesome GUI.

Make a Window

Interactive development is fun. Fire up a Clojure REPL, and type this:

```
(import 'javax.swing.JFrame)
(def frame (JFrame. "Hello Frame"))
(.setSize frame 200 200)
(.setVisible frame true)
```

Hey Presto, there's a window! (It might not pop to the front, so check your task bar.)

JFrame is Swing's all-purpose window class. This example created a frame, set its dimensions to 200×200 pixels, and "turned it on" by calling setVisible.

Containers and Frames

Our new frame doesn't look like much; let's give it some content. More specifically, let's give it a *container*. Swing GUIs are laid out as a nested hierarchy of containers. All containers — **except** the top-level window — are sub-classes of JComponent. This is called the containment hierarchy.

Although it's technically possible to add GUI elements directly to a top-level container like JFrame, it's more correct to use a content pane. We'll use JPanel, a general-purpose container:

```
(import 'javax.swing.JPanel)
(def panel (JPanel.))
(.setContentPane frame panel)
```

A panel by itself doesn't show anything, so lets make a button and add it to the panel.

```
(import 'javax.swing.JButton)
(def button (JButton. "Click Me!"))
(.add panel button)
```

Hey, it's not there! That's because Swing wasn't designed with interactive development in mind. To make your button visible, call:

```
(.revalidate button)
```

The revalidate method is not something you'll read about in most Swing tutorials, because in pre-compiled Java it's rarely necessary. Basically, it tells Swing, "I just changed the layout, you need to redraw stuff." Starting at our JButton, Swing searches up the containment hierarchy to the top-level container, and redraws it.

Next step: Action! My next post will talk about enabling GUI events with ActionListener.

This entry was posted in Programming and tagged Clojure, Swing on January 2, 2010.

← Objects Are Not Abstract Data Types

Swing Into Actions with Clojure →

4 thoughts on "First Steps With Clojure & Swing"

Benoît Huron
January 3, 2010 at 6:56 pm

Since Clojure 1.1, import is a macro form. So this should work : (import javax.swing.JFrame)

Pingback: Clojure – Destillat #2 | duetsch.info - Open Source, Wet-, Web-, Software

Pingback: Funny Java Standards | Wisdom and Wonder

Anders
February 19, 2010 at 4:39 am

Thanks, i have been looking around for som easy introduction to Clojure and Swing and this was great.

Comments are closed.