{:title “Creating JFrames using Swing APIs in Clojure (Part 2)” :layout :post :tags [“Clojure Guide Code”]}

Ok so continuing on with what we have so far. If you havent read the first part I suggest reading it to follow through. I want to show you a quick way to add more flavor to your frame since it looks rather dull. One way to do this is by adding substance to your dependencies which is basically like a bunch of themes for your frames. Add the following to your dependencies in your project.clj

```
[com.github.insubstantial/substance "7.1"]
```

Now you can import substance by changing your namespace to the following.

```
(ns hello-swing.core
  (:import org.pushingpixels.substance.api.SubstanceLookAndFeel) ;; add this
  (:gen-class))
```

Now that you can use the substance themes you can set a default theme by adding to your -main.

```
(defn -main []
  (native!)
  (invoke-later
   (-> jframe show!)
   (SubstanceLookAndFeel/setSkin "org.pushingpixels.substance.api.skin.GraphiteAquaSkin")))
```

Now that you have set the theme automatically you can run what you have so far.

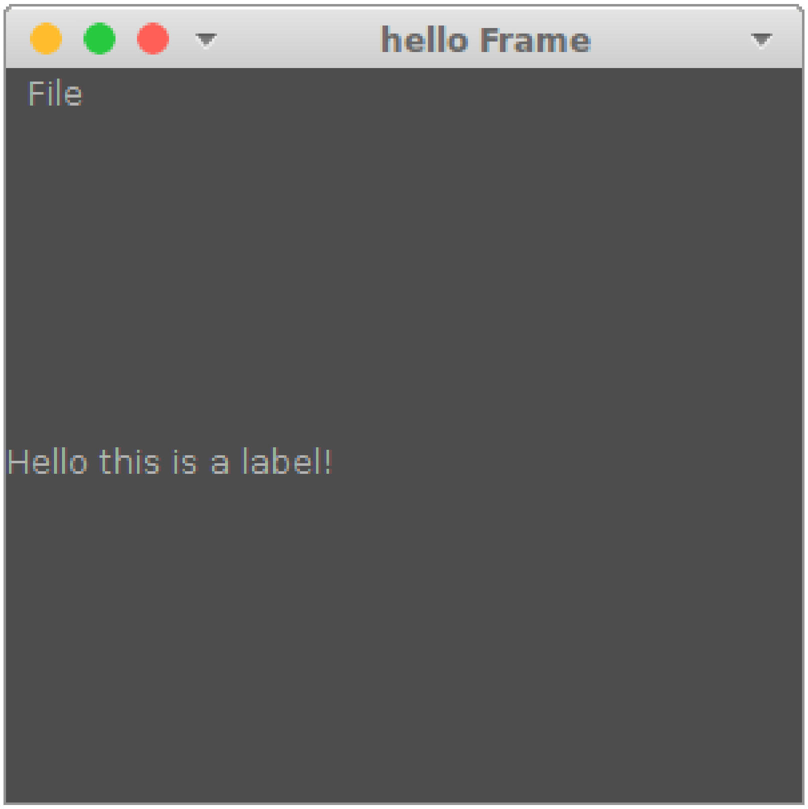
```
$ lein run
```

A frame like this should appear now and as you notice it has been decorated.

However you may now notice that there is a problem when opening the menubar and it just disappears. To fix this add pack! to your -main.

```
(defn -main []
  (native!)
  (invoke-later
   (-> jframe pack! show!) ;; add pack! Which should fix the menubar problem.
   (SubstanceLookAndFeel/setSkin "org.pushingpixels.substance.api.skin.GraphiteAquaSkin")))
```

Now what if we want to select what theme we want? Now we can use the menubar to select what theme we want. In order to do this make a menubar item called theme-selector.



Hello this is a label!

Figure 1: jframe 2

```
(def jframe (frame :title "hello Frame"
                   :menubar (menubar :items [(menu :text "File" :items [close-frame theme-se
                   :height 300
                   :width 300
                   :on-close :exit ;; Exits on close
                   :content (label :text "Hello this is a label!")))
```

Now make the menu-item for the theme-selector. Also don't forget to add theme-select into :items in your jframe.

```
(def theme-select (menu-item :text "Theme Select"
                               :tip "This will allow you to select a theme."
                               :listen [:action handler]))
```

Now that you have the theme-select you can change your handler to do something when that is clicked on.

```
(defn handler [event]
  (let [e (.getActionCommand event)]
    (if (= e "Close Frame")
      (System/exit 0))
    (if (= e "Theme Select")  ;; add this to incorporate the theme selector.
      (-> (frame :title "Themes" :id 3 :content (theme-selector) :on-close :hide :height 600
```

Now we create the actual combobox that will select the theme.

```
(defn theme-selector []
  (horizontal-panel
   :items [
           (combobox
            :model    (vals (SubstanceLookAndFeel/getAllSkins)) ;; list out all of the skins
            :renderer (fn [this {:keys [value]}]  ;; takes the values and displays them from
                        (text! this (.getClassName value)))
            :listen   [:selection (fn [e]
                                    (invoke-later
                                     (-> e
                                         selection
                                         .getClassName
                                         SubstanceLookAndFeel/setSkin)))])])) ;; changes the
```

Now you can open it up and go to file and select theme-selector which will allow you to change the theme!