

## [Faking a Postback with JSF + Facelets](#)

By [Roger Keays](#), 27 February 2007

Have you ever needed to post a form to a different action URL with JSF? How about posting a form from an email or when a session has expired? The JSF spec doesn't allow for these situations because it requires a view to be present in the user's session before the complete lifecycle will be invoked. Here's a handy trick using an extended Facelets ViewHandler that allows you to make a fake postback to a view that doesn't yet exist in the user's session.

The motivation for this exercise was to implement a "quick subscribe" form for a mailing list in the sidebar of a site. I had already implemented a thorough form, validations and actions which I wanted to reuse as much as I could. In fact, all I wanted was a small version of the form that would operate in exactly the same way as the main form, except the results of posting the small form would land the user on the page showing the main form.

JSF doesn't allow you to specify the action attribute of a form, but you can always create your own `<form>` tag by hand and set the action attribute manually. By reproducing the request parameters as they appear in the real form, you can trick JSF into thinking the original form was submitted. Here is an example of the quick subscribe form I've implemented:

```
<!--
  A miniature version of the subscription form. Variable names
  must match the actual subscription form for the fake postback
  to work properly. The $list variable should be a number which
  is the index of the list to subscribe to from the real form.
-->
<form id="fb-quickSubscribe" action="modules/lists/user/subscriptions.jsf">
  <input type="hidden" name="javax.faces.ViewState" value="0:0"/>
  <input type="hidden" name="subscriptions" value="subscriptions"/>
  <input type="hidden" name="submit" value="Submit"/>
  <input type="hidden" name="lists:${list}:subscribe" value="true"/>
  <input type="text" id="name" name="name" value="name" max="255"/><br/>
  <input type="text" id="email" name="email" value="email" max="319"/>
  <input type="image" id="submit" src="modules/core/icons/send.png" width="160" height="16"/>
</form>
```

At first, once you've magically crafted your parameters to imitate the form you might think you've finished and the problem is solved. Unfortunately though, it may only be working because you've

visited the real form's page and this has created a view for the form in your session. If there is no such view already created, this method will fail because the `restoreView()` method will return null and the complete lifecycle will not be invoked.

Enter Facelets. Facelets (unlike the JSP ViewHandler) was designed sensibly to separate page rendering into two phases - *build* and *render*. We can, in fact, build the view at any time in the lifecycle. This means we can create an extended ViewHandler which overrides `restoreView()` to build a view for us if one doesn't already exist:

```
public class SeamlessViewHandler extends FaceletViewHandler {

    /** constructor */
    public SeamlessViewHandler(ViewHandler parent) {
        super(parent);
    }

    /**
     * To allow the user to postback to a view which was not rendered in the
     * first place, or has expired, we modify the default restoreView
     * method to build a new view if there was not one to restore. This
     * allows us to invoke the full JSF lifecycle on an initial page view,
     * which is useful for email forms and doing 'fake' postbacks.
     */
    public UIViewRoot restoreView(FacesContext context, String viewId) {
        UIViewRoot viewRoot = super.restoreView(context, viewId);
        if (viewRoot == null) {
            viewRoot = createView(context, viewId);
            context.setViewRoot(viewRoot);
            try {
                this.buildView(context, viewRoot);
                context.getExternalContext().getRequestMap().put(
                    "seamless.viewBuilt", getRenderedViewId(context, viewId)
                );
            } catch (IOException ioe) {
                log.log(Level.SEVERE, "Error Building View", ioe);
            }
        }
        return viewRoot;
    }

    /** only build the view if it wasn't built by restoreView */
    protected void buildView(FacesContext context, UIViewRoot viewToRender)
```

```
throws IOException, FacesException {

String viewId = getRenderedViewId(context, viewToRender.getViewId());
String viewBuilt = (String) context.getExternalContext()
    .getRequestMap().get("seamless.viewBuilt");

if (viewBuilt == null || !viewId.equals(viewBuilt)) {
    super.buildView(context, viewToRender);
}
}
}
```

Voila! The key ingredient to making our fake postbacks work. Have a go yourself by [submitting the subscription form using this link](#).

This view handler is a part of the [Furnace Webapp Framework](#). Please feel free to use it in your own applications!

## About Roger Keays



Roger Keays is an artist, an engineer, and a student of life. He has no fixed address and has left footprints on 40-something different countries around the world.

Roger is addicted to surfing. His other interests are music, psychology, languages, the proper use of semicolons, and finding good food.