

CSS-enabled feedback panel

This is for wicket 1.2, wicket 1.3 has appropriate classes set on all markup elements which you can use to style the FeedbackPanel with your own CSS.

The problem I had with FeedbackPanel is that it doesn't allow me to change the CSS style based on the kinds of messages it contains. So, with some help from wicket-user, I was able to put this together:

```
public class CSSFeedbackPanel extends FeedbackPanel {
    public CSSFeedbackPanel(final String id) {
        super(id);
        WebMarkupContainer feedbackul = (WebMarkupContainer) get("feedbackul");
        if (feedbackul != null) {
            feedbackul.add(new AttributeModifier("class", true, new Model() {
                public Object getObject(Component comp) {
                    if (anyMessage()) {
                        if (anyMessage(FeedbackMessage.INFO)) {
                            return "feedbackulINFO";
                        }
                        else if (anyMessage(FeedbackMessage.ERROR)) {
                            return "feedbackulERROR";
                        }
                        else {
                            return "feedbackulNONE";
                        }
                    }
                    else {
                        return "feedbackulNONE";
                    }
                }
            }));
        }
    }
}
```

It's just a drop-in replacement for FeedbackPanel that changes the CSS style based on the type of messages contained. The problem is that it doesn't work well if there are multiple message types in the panel.

Alternative Example (Wicket 1.4+)

An alternative example using the built-in method for getting the message rendering component. This method simply appends a class attribute to the component that Wicket is about to render for the particular message. The resulting classes are: feedbackUNDEFINED feedbackDEBUG feedbackINFO feedbackWARNING feedbackERROR feedbackFATAL.

```

public class CSSFeedbackPanel extends FeedbackPanel {

    public CSSFeedbackPanel(final String id, final IFeedbackMessageFilter filter) {
        super(id, filter);
    }

    public CSSFeedbackPanel(final String id) {
        super(id);
    }

    @Override
    protected Component newMessageDisplayComponent(final String id,
        final FeedbackMessage message) {
        final Component newMessageDisplayComponent = super
            .newMessageDisplayComponent(id, message);

        /*
         * CSS class resulting: feedbackUNDEFINED feedbackDEBUG feedbackINFO
         * feedbackWARNING feedbackERROR feedbackFATAL
         */
        newMessageDisplayComponent
            .add(new AttributeAppender("class", new Model<String>(
                "feedback" + message.getLevelAsString()), " "));

        return newMessageDisplayComponent;
    }
}

```