

Simple Flash container based on AbstractTransformerBehavior

```
package manuelbarzi.wicket.flash;

import org.apache.wicket.Component;
import org.apache.wicket.markup.html.WebMarkupContainer;
import org.apache.wicket.markup.transformer.AbstractTransformerBehavior;

/**
 * simple & rude flash container using transformer :-)
 *
 * example:
 *
 * + html:
 *
 * <object wicket:id="flash" />
 *
 * + java:
 *
 * Flash flash = new Flash("flash");
 * flash.setMovie("http://www.notcelebrity.co.uk/flashcontent/flashHeader.swf?thename=Wicket");
 * flash.setWidth("425");
 * flash.setHeight("280");
 * flash.setQuality("high");
 * flash.setWmode("transparent");
 * add(flash);
 *
 * + output:
 *
 * <object width="425" height="280">
 *   <param name="movie" value="http://www.notcelebrity.co.uk/flashcontent/flashHeader.swf?
thename=Wicket" />
 *   <param name="wmode" value="transparent" />
 *   <param name="quality" value="high" />
 *   <embed src="http://www.notcelebrity.co.uk/flashcontent/flashHeader.swf?thename=Wicket" type="
application/x-shockwave-flash" wmode="transparent" quality="high" width="425" height="280" />
 * </object>
 *
 * @author manuelbarzi
 */
public class Flash extends WebMarkupContainer {

    private static final long serialVersionUID = 1L;

    private String movie;
    private String wmode;
    private String width;
    private String height;
    private String quality;

    public Flash(String id) {
        super(id);
        add(new AbstractTransformerBehavior() {

            private static final long serialVersionUID = 1L;

            @Override
            public CharSequence transform(Component component, CharSequence output) throws
Exception {

                StringBuilder sb = new StringBuilder();
                sb.append("<object");
                if (width != null) {
                    appendParamAttribute(sb, "width", width);
                }
                if (height != null) {
```

```

        appendParamAttribute(sb, "height", height);
    }
    sb.append(">");
    if (movie != null) {
        appendParamTag(sb, "movie", movie);
    }
    if (quality != null) {
        appendParamTag(sb, "quality", quality);
    }
    if (wmode != null) {
        appendParamTag(sb, "wmode", wmode);
    }
    sb.append("<embed");
    if (width != null) {
        appendParamAttribute(sb, "width", width);
    }
    if (height != null) {
        appendParamAttribute(sb, "height", height);
    }
    if (movie != null) {
        appendParamAttribute(sb, "src", movie);
    }
    if (quality != null) {
        appendParamAttribute(sb, "quality", quality);
    }
    if (wmode != null) {
        appendParamAttribute(sb, "wmode", wmode);
    }
    appendParamAttribute(sb, "type", "application/x-shockwave-flash");
    sb.append(" />");
    sb.append("</object>");
    return sb;
}

private void appendParamAttribute(StringBuilder sb, String name, String value) {
    sb.append(" ");
    sb.append(name);
    sb.append("=\");
    sb.append(value);
    sb.append("\");
}

private void appendParamTag(StringBuilder sb, String name, String value) {
    sb.append("<param");
    appendParamAttribute(sb, "name", name);
    appendParamAttribute(sb, "value", value);
    sb.append(" />");
}

});
}

public void setMovie(String value) {
    movie = value;
}

public void setWmode(String value) {
    wmode = value;
}

public void setWidth(String value) {
    width = value;
}

public void setHeight(String value) {
    height = value;
}

public void setQuality(String value) {
    quality = value;
}

```

