

# Including JSP files in HTML templates

To include JSP files in your Wicket templates with much of the same flexibility as `jsp:include`, we will create a custom Wicket component that will take care of this.

## Custom Wicket Component

```
public class WicketJspResolver implements IComponentResolver {
    private static final long serialVersionUID = 1L;
    private static final Logger log = LoggerFactory.getLogger(WicketJspResolver.class);

    static {
        // Register the jsp tag within the wicket tag namespace
        WicketTagIdentifier.registerWellKnownTagName("jsp");
    }

    public boolean resolve(final MarkupContainer container, final MarkupStream markupStream, final
ComponentTag tag) {
        if (tag instanceof WicketTag) {

            WicketTag wtag = (WicketTag)tag;

            if ("jsp".equalsIgnoreCase(wtag.getName())) {
                String file = wtag.getAttributes().getString("file");
                if (file == null || file.trim().length() == 0) {
                    throw new MarkupException("Wrong format of <wicket:jsp file='foo.jsp'>:
attribute 'file' is missing");
                }

                // Add the jsp component to the component hierarchy
                container.autoAdd(new JspFileContainer(file), markupStream);

                // We did process the tag
                return true;
            }

            // We did not process the tag
            return false;
        }

        private static class JspFileContainer extends MarkupContainer {
            private static final long serialVersionUID = 1L;

            private String _file;

            public JspFileContainer(String file) {
                super(file);

                _file = file;
            }

            protected void onRender(MarkupStream markupStream) {
                markupStream.next();

                WebRequestCycle cycle = (WebRequestCycle)RequestCycle.get();
                ServletRequest request = cycle.getWebRequest().getHttpServletRequest();
                ServletResponse response = cycle.getWebResponse().getHttpServletResponse();
                ServletContext context = ((WebApplication)Application.get()).getServletContext();

                try {
                    // Make an attempt at locating the resource.
                    // If it does not exist, the requestDispatcher.include will simply ignore it.
                    if (context.getResource(_file) == null) {
                        if (shouldThrowExceptionForMissingFile()) {
                            throw new WicketRuntimeException(String.format("Cannot locate
resource %s within current context: %s", _file, context.getContextPath()));
                        } else {
                            log.warn("File will not be processed. Cannot locate resource {}")
                        }
                    }
                }
            }
        }
    }
}
```

```
within current context: {"", _file, context.getContextPath());
    }
} catch (MalformedURLException e) {
    throw new WicketRuntimeException(e);
}

try {
    context.getRequestDispatcher(_file).include(request, response);
} catch (ServletException e) {
    throw new WicketRuntimeException(e);
} catch (IOException e) {
    throw new WicketRuntimeException(e);
}

private boolean shouldThrowExceptionForMissingFile() {
    return Application.get().getResourceSettings().getThrowExceptionOnMissingResource();
}
}
```

#### Register the custom component in your application class

```
protected void init() {
    getPageSettings().addComponentResolver(new WicketJspResolver());
}
```

#### Include the JSP file in your HTML template

```
<wicket:jsp file="/foo.jsp"/>
```