

Running Apache and Daisy

Why running Apache in front of Daisy?

Daisy ships with [Jetty](#)¹, which is a nice, compact, and fast servlet container. The Daisy Wiki application can be deployed in Tomcat [as well](#)². However, many people (ourselves included) prefer to run Apache in front of the Daisy servlet container, as this provides the ability to listen to the usual port 80 number, without requiring to run Jetty or Tomcat as root processes. So we need some way to hook up Apache and Daisy, which is described in the document underneath.

Required Apache modules

First of all, we need to install a recent version of Apache (v2), with the [mod_proxy](#)³ and [mod_rewrite](#)⁴ modules activated. Most likely, there will be a binary build of Apache for your platform available - if not, the process of building Apache and both modules is explained on the [Apache httpd website](#)⁵. Apparently, the mod_proxy implementation inside Apache v1 isn't considered very robust, so be sure to install Apache v2.

Configuring Apache

In this example, I'm running my website as a Virtual Host in my Apache config, but these instructions should also apply when using Apache to serve only one website: only the location of the directives will be slightly different. Here's what our server configuration looks like:

```
NameVirtualHost *:80
<VirtualHost *:80>
    ServerName cocoondev.org
    ServerAdmin webmaster.at.cocoondev.org

    DocumentRoot /home/daisy_cd

    RewriteEngine on
    RewriteRule "^/WEB-INF/?(.*)" "$0" [L,F,NC]

    RewriteRule "^/$" http://localhost:8080/main/ [P]
    RewriteRule "^/(.*)" "http://localhost:8080/$1" [P]

    ProxyRequests Off
    ProxyPassReverse / http://localhost:8080/
    ProxyPreserveHost On

    ErrorLog /var/log/apache2/error.log
    LogLevel warn

    CustomLog /var/log/apache2/access.log combined

    ServerSignature On

</VirtualHost>
```

-
1. <http://jetty.mortbay.org/jetty/index.html>
 2. [daisy:208-cd \(Deploying on Tomcat\)](#)
 3. http://httpd.apache.org/docs-2.0/mod/mod_proxy.html
 4. http://httpd.apache.org/docs-2.0/mod/mod_rewrite.html
 5. <http://httpd.apache.org/docs-2.0/>

DocumentRoot

I picked the home directory of the user running Daisy as the Apache DocumentRoot. As all requests will be proxy-forwarded anyhow, this setting isn't particularly relevant.

Securing the /WEB-INF/ directory

Though Jetty doesn't service requests starting with /WEB-INF/ by default, we make sure such requests are blocked on the Apache level as well:

```
RewriteRule "^/WEB-INF/?(.*)" "$0" [L,F,NC]
```

Here, the [F] flag forbids accessing the matched resource. [NC] makes the pattern case-insensitive.

Getting rid of the /daisy/ prefix

Typically, you don't want to see the /daisy/ prefix appear in a Daisy-based website. That requires you to edit Daisy's Cocoon sitemap (we might come up with a different solution for this in the future). Edit \$DAISY_HOME/daisywiki/webapp/sitemap.xmap:

Around line 578, near the <!-- welcome page --> comment, replace:

```
<map:match pattern="">
```

with

```
<map:mount check-reload="yes" src="daisy/" uri-prefix=""/>
```

Then remove the following lines until you reach

```
<map:handle-errors>
```

which should be around line 718.

Proxying Jetty

mod_rewrite can also pass the handling of requests to the mod_proxy module (using the [P] flag), and we make use of this in the statement which makes sure all requests are simply forwarded from Apache to Jetty, where they will be picked up by Cocoon for further processing:

```
RewriteRule "^/(.*)" "http://localhost:8080/$1" [P]

ProxyRequests Off
ProxyPassReverse / http://localhost:8080/
ProxyPreserveHost On
```

By turning ProxyRequests Off, we also make sure our web server can't be used as an open proxy for people who want to surf anonymously. Make sure you don't forget to add the ProxyPreserveHost directive, which is especially needed in Virtual Hosts serving context.

Final touches

Normally, Daisy's main access point is the page listing the different sites available (depending on login and role) from that specific Daisy instance. If we want to skip this page, and forward visitors to one specific site, consider adding this rule to the server config:

```
RewriteRule "^/$" http://localhost:8080/main/ [P]
```

This way, if someone hits the root of the website, he will be automatically forwarded to the homepage of the 'main' Daisy site (<http://host/main/>). The other sites are still available, by simply accessing <http://host/otherdaisysite/>. Also, the administration screens are accessible using <http://host/admin/>.

Acknowledgments

Kudos to Pier for [describing](#)⁶ this setup technique on the Cocoon Wiki. This setup is in use at a very high profile / high volume news website, so it should work for Daisy as well.

6. <http://wiki.apache.org/cocoon/ApacheModProxy>