

# ServingStaticFiles

## How to serve static files such as images, CSS stylesheets, etc.?

By default a web server will serve up any file that can be found under its document root. Therefore if it serves a web page that references additional images, applets or other supporting media then, as long as those files can be found under the document root, it'll serve them when the browser requests them.

However Cocoon's default mode of operation is different: unless there is an explicit [Pipeline](#) that instructs Cocoon on how to respond to a particular request and serve the required file, then there will be a 'resource not found' or a 'no pipeline matched request' error.

So, in short, a web server 'knows' how to serve files automatically. Cocoon doesn't, so you have to map the request to the file system using a [Pipeline](#).

The simplest way to achieve this is to use a [Reader](#) (usually the `ResourceReader`, which is the default). E.g:

```
<map:match pattern="**.jar">
  <map:read src="jars/{1}.jar" mime-type="application/java-archive"/>
</map:match>
<map:match pattern="**.css">
  <map:read src="style/{1}.css" mime-type="text/css"/>
</map:match>
<map:match pattern="**.gif">
  <map:read src="images/{1}.gif" mime-type="image/gif"/>
</map:match>
<map:match pattern="static-**.html">
  <map:read src="html/{1}.html" mime-type="text/html"/>
</map:match>
```

---

*IMHO this is overengineered, because in order to serve static content directly from Apache (see below) the URI-to-filesystem mapping in the example above must also be implemented in the Apache configuration. Cocoon already needs a separate rule for each (in order to specify the MIME type), but Apache doesn't, so it's a pain to have to add it just so that URIs don't have to start with "images/", etc. See the simpler version of these sitemap rules given in [SimpleModProxy](#) under "Serving Static Content". --ML*

---

It's more efficient to serve static files from Apache (see [SimpleModProxy](#)), but we still want our Cocoon app to have the ability to serve static content all by itself without Apache, for when we're developing in a non-server (desktop/laptop) environment.

TODO: show how the expires parameter can help

## Mime types

As you can see the reader needs the mime type (Multipurpose Internet Mail Extensions):

- [Content Types and Subtypes as a list](#)
- [General information](#)