

# Apache httpd in front of Tomcat on a local dev machine

This page explains how to set up an [Apache HTTP Server \(httpd\)](#) in front of Tomcat on a local machine for development purposes. This page is not intended as a guide for a system administrator to set up a production environment.

At the end you can access the Magnolia contexts with different domains which both serve from root. This kind of setup may help testing multisite scenarios during local development to emulate a production environment scenario.



This page assumes that you are using a Mac or something similar. This is especially true concerning the paths to Apache configuration files. However, the information here should also apply to other OS.



Start with a Tomcat bundle of your choice, for a selection of bundles see [list of preconfigured Magnolia bundles](#). If you want to test multisite scenarios, choose an EE pro bundle (such as `magnolia-enterprise-pro-demo-bundle`).

If want to get rid of the demo modules to start with a vanilla multisite module - remove the appropriate modules before you start the bundle for the first time. (On `magnolia-enterprise-pro-demo-bundle` delete the modules starting with `magnolia-travel*` from the folder `magnolia-enterprise-x.y.z/apache-tomcat-8.5.5/webapps/magnoliaAuthor/WEB-INF/lib`).

Before following the steps explained below, start the bundle as usual, to make sure both webapps `magnoliaAuthor` and `magnoliaPublic` are installed and work properly.

- [Domains and /etc/hosts](#)
- [Apache setup](#)
  - [Helpful Apache commands](#)
    - [Restart the process](#)
    - [Check the configuration](#)
  - [Enable mod\\_proxy, proxy\\_ajp\\_module](#)
  - [Enable vhosts](#)
  - [Define virtual hosts](#)
    - [Using ajp](#)
    - [Redirecting to Tomcat root](#)
- [Tomcat adjustments to serve one context from root](#)
  - [Usage in a production environment](#)
- [Serve both contexts from root from localhost](#)
  - [Virtual hosts configuration](#)
  - [Tomcat adjustments](#)
    - [webapps directory](#)
    - [server.xml](#)
  - [Final adjustments on the Magnolia instance](#)
    - [Adjust the subscriber for content activation](#)
  - [URLs of the final setup](#)

## Domains and /etc/hosts

This is an overview of the the multisite setup:

	author instance	public instance
<i>comics</i>	author.best-comics.net	www.best-comics.net
<i>vinyl</i>	author.best-comics.net	www.best-vinyl.net

Note that we use only one domain - a subdomain of the "main domain" - to access the author instance (which is recommended practice).

To enable this without DNS, add the following line to your `/etc/hosts` file:

```
127.0.0.1          author.best-comics.net www.best-comics.net www.best-vinyl.net
```

## Apache setup

Make sure you have a working Apache on your local machine.

### Helpful Apache commands

Whenever you change Apache configuration files, you have to restart the `httpd` process.

#### Restart the process

```
sudo apachectl restart
```

#### Check the configuration

```
apachectl configtest
```

### Enable `mod_proxy`, `proxy_ajp_module`

Make sure your Apache has the modules `mod_proxy` and `proxy_ajp_module` enabled.

Check the `/etc/apache2/httpd.conf` file and make sure the following lines are *not* commented:

```
LoadModule proxy_module libexec/apache2/mod_proxy.so
# more line in between ...
LoadModule proxy_ajp_module libexec/apache2/mod_proxy_ajp.so
```

On a different OS, the path and suffix of the modules may be different. If you edit the file, you must restart Apache.

### Enable vhosts

We use virtual hosts configuration in this setup. Make sure `vhosts` are enabled. Check the `/etc/apache2/httpd.conf` file and ensure the line with the `include` directive is *not* commented:

```
# Virtual hosts
Include /private/etc/apache2/extra/httpd-vhosts.conf
```

If you edit the file, you must restart Apache.

### Define virtual hosts

If `vhosts` is properly enabled, edit the `/etc/apache2/extra/httpd-vhosts.conf` file and add something similar to this:

```
<VirtualHost *:80>
    ServerName author.best-comics.net
    # ProxyPass / http://localhost:8080/
    # ProxyPassReverse / http://localhost:8080/
      ProxyPass / ajp://localhost:8009/
      ProxyPassReverse / ajp://localhost:8009/

    ErrorLog "/private/var/log/apache2/projekt.local-error_log"
    CustomLog "/private/var/log/apache2/projekt.local-access_log" common
</VirtualHost>
```

### Using ajp

See lines 3, 4 above which are commented: At first sight one may think that these `ProxyPass` rules should be sufficient. However, they're not. Some things will work, but especially the multisite setup won't do what we expect 🤔.



Use the `ajp` protocol to redirect from apache to Tomcat.

`mod_jk` is a known variant which works too, see [apache documentation](#).

Since we use `ajp` - the `ajp` connector must be configured in the Tomcat `server.xml` - which is the case in the default `server.xml` which is provided by a Magnolia bundle. In the `server.xml` of your Tomcat you should find an entry as this:

```
<Connector port="8009" enableLookups="false" redirectPort="8443" protocol="AJP/1.3" />
```

This is the connector to which apache will redirect to.

### Redirecting to Tomcat root

The apache redirect points to the Tomcat root context (see lines 5, 6 in the excerpt from `httpd-vhosts.conf` shown above).

Redirecting to <ajp://localhost:8009/magnoliaAuthor> - would work, but again, only partially. (Try it out, if you want to have fun.) Web resources would not load and redirections done by Magnolia would be incorrect.

So, since the apache redirect points to the Tomcat root context, we must make sure that the webapp `magnoliaAuthor` is served from the root context. This needs a few modifications on Tomcat (see below).

### Tomcat adjustments to serve one context from root

I guess there are different possible solutions to serve a Magnolia instance from Tomcat root context. Here comes one:

Use the `<Context>` directive which allows to define the path of a webapp.

```
<Context path="" docBase="magnoliaAuthor" />
```

The empty string (`" "`) as value of the `path` attribute ensures to serve the webapp `magnoliaAuthor` from the root context.

## server.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Server port="8005" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />

  <GlobalNamingResources>
    <Resource name="UserDatabase" auth="Container"
      type="org.apache.catalina.UserDatabase"
      description="User database that can be updated and saved"
      factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
      pathname="conf/tomcat-users.xml" />
  </GlobalNamingResources>

  <Service name="Catalina">

    <Connector port="8080" protocol="HTTP/1.1"
      connectionTimeout="20000"
      redirectPort="8443" />

    <Connector port="8009" enableLookups="false" redirectPort="8443" protocol="AJP/1.3" />

    <Engine name="Catalina" defaultHost="localhost">

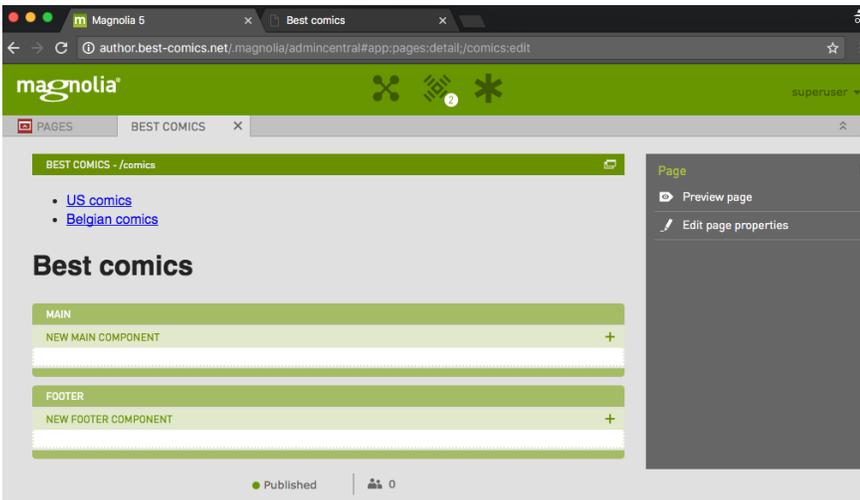
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
          resourceName="UserDatabase"/>
      </Realm>

      <Host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true">
        <!-- serve magnoliaAuthor from root -->
        <Context path="" docBase="magnoliaAuthor" />

        <Valve className="org.apache.catalina.valves.AccessLogValve"
          directory="logs"
          prefix="localhost_access_log"
          suffix=".txt" pattern="%h %l %u %t &quot;%r&quot; %s %b" />
      </Host>
    </Engine>
  </Service>
</Server>
```

**See line 37:** Added <Context> directive.

Tomcat can serve only one webapp from root within the same host and within the same service.



It works.

However, since now we have only one context served from root.

## Usage in a production environment

In a productive environment, you probably have 2 hosts / servers in different networks, in which one server contains one Tomcat serving a single instance. If this is true, the setup described above is feasible.



Use in a production environment at your own risk. This page is *not* intended as a guide for a System administrator.

## Serve both contexts from root from localhost

The following, extended setup allows to run and access both Magnolia instances from one Tomcat instance.

**⚠** Once again, this is a not required setup for productive environments, but it works fine on a local development machine.

These is the demanded setup:

author instance	public instance
author.best-comics.net	www.best-comics.net
	www.best-vinyl.net

Keep in mind that we already have defined redirects to localhost for these three domains (in the `/etc/hosts` file).

## Virtual hosts configuration

In summary we need three virtual hosts on Apache - here is the modified `httpd-vhosts.conf` with one virtual host for the author instance and two for the public instances:

```

<VirtualHost *:80>
  ServerName author.best-comics.net
  ProxyPass / ajp://localhost:8009/
  ProxyPassReverse / ajp://localhost:8009/
</VirtualHost>

<VirtualHost *:80>
  ServerName www.best-comics.net
  ProxyPass / ajp://localhost:8010/
  ProxyPassReverse / ajp://localhost:8010/
</VirtualHost>

<VirtualHost *:80>
  ServerName www.best-vinyl.net
  ProxyPass / ajp://localhost:8010/
  ProxyPassReverse / ajp://localhost:8010/
</VirtualHost>

```

- **Lines 3-4:** `author.best-comics.net` redirects to localhost on port 8080  
=> author instance
- **Lines 9-10, 15-16:** `www.best-comics.net` and `www.best-vinyl.net` are redirected to localhost on port 8081  
=> public instance
- The lines configuring the log files of the virtual hosts have been removed to keep this configuration excerpt small.

## Tomcat adjustments

As mentioned above, one Tomcat host cannot serve two contexts from root.

We achieve this by the the following:

- Define two `<Service>` sections in the `server.xml`.
- Each `<Service>`:
  - Has `<Connector>`s with distinct ports.
    - AJP connector
    - HTTP connector \*
  - Serves from its own `webapps` directory.
  - Defines one host which wraps exactly one context to be served from root.

\*) Actually the AJP connector would be sufficient to use tomcat via apache. However, if you also keep the HTTP connector, you can access tomcat also directly. And we also will use the HTTP connector for JCR activations from author to public context.

## webapps directory

We assume that the bundle has already been started once and that both webapps have been properly installed.

This is the adapted Tomcat directory structure:

```

bin
conf
lib
webapps
  ROOT
  magnoliaAuthor
webapps2
  magnoliaPublic

```

- Create the folder `webapps2`.
- Move the webapp `magnoliaPublic` to `webapps2`.
- You may want delete the webapp `ROOT`, you won't be able to access it anymore.

## server.xml

This is the server.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<Server port="8005" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />

  <GlobalNamingResources>
    <Resource name="UserDatabase" auth="Container"
      type="org.apache.catalina.UserDatabase"
      description="User database that can be updated and saved"
      factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
      pathname="conf/tomcat-users.xml" />
  </GlobalNamingResources>

  <Service name="Catalina">
    <Connector port="8080" protocol="HTTP/1.1"
      connectionTimeout="20000"
      redirectPort="8443" />
    <Connector port="8009" enableLookups="false" redirectPort="8443" protocol="AJP/1.3" />
    <Engine name="Catalina" defaultHost="localhost">
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm" resourceName="UserDatabase"/>
      </Realm>
      <Host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true">
        <Context path="" docBase="magnoliaAuthor" />
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" prefix="
localhost_access_log" suffix=".txt" pattern="%h %l %u %t &quot;%r&quot; %s %b" />
      </Host>
    </Engine>
  </Service>

  <Service name="Catalina2">
    <Connector port="8081" protocol="HTTP/1.1"
      connectionTimeout="20000"
      redirectPort="8444" />
    <Connector port="8010" enableLookups="false" redirectPort="8444" protocol="AJP/1.3" />
    <Engine name="Catalina2" defaultHost="localhost">
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm" resourceName="UserDatabase"/>
      </Realm>
      <Host name="localhost" appBase="webapps2" unpackWARs="true" autoDeploy="true">
        <Context path="" docBase="magnoliaPublic" />
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" prefix="
localhost_access_log" suffix=".txt" pattern="%h %l %u %t &quot;%r&quot; %s %b" />
      </Host>
    </Engine>
  </Service>

</Server>
```

Note the following:

- There are two <Service> sections, their name attributes have different values.
- Every <Service> has its own <Connector>s with distinct ports
- Each connector must have its own distinct ports on the attributes port and redirectPort.
- The port of the AJP-<Connector> is the same port to which we redirect on the Apache virtual host configuration.

## Final adjustments on the Magnolia instance

## Adjust the subscriber for content activation

To ensure content activation still works, you have to adjust the property `/server/activation/subscribers/magnoliaPublic8080@URL:`

Node name	Value
server	
activation	
subscribers	
magnoliaPublic8080	
subscriptions	
URL	http://localhost:8081

Note that the activation is using the HTTP connector.

## URLs of the final setup

If you have all done correctly - you can access the Magnolia instance by the following URLs

author instance	
<a href="http://author.best-comics.net">author.best-comics.net</a>	via <b>Apache</b> and AJP connector
<a href="http://author.best-comics.net:8080">author.best-comics.net:8080</a>	via Tomcat and HTTP connector
<a href="http://localhost:8080">localhost:8080</a>	
public instance	
<a href="http://www.best-comics.net">www.best-comics.net</a>	via <b>Apache</b> and AJP connector
<a href="http://www.best-comics.net:8081">www.best-comics.net:8081</a>	via Tomcat and HTTP connector
<a href="http://www.best-vinyl.net">www.best-vinyl.net</a>	via <b>Apache</b> and AJP connector
<a href="http://www.best-vinyl.net:8081">www.best-vinyl.net:8081</a>	via Tomcat and HTTP connector
<a href="http://localhost:8081">localhost:8081</a>	