Tomcat Proxies

Overview

> **Notes:**
> - This document **only** applies to systems that run cPanel & WHM version 76 and later.
> - You **must** possess `root`-level access to create an Apache Tomcat® proxy.

This document explains how to configure a proxy for Tomcat in EasyApache 4.

Configure a proxy

> **Notes:**
> - When you assign Tomcat access to a cPanel user, the script assigns two ports to the user. These port assignments reside in the `/etc/cpanel/cpuser_port_authority.json` file and the user's `/root/ea-tomcat85/conf/server.xml` files.
>   - The `/root/ea-tomcat85/conf/server.xml` file contains the ports on which the Apache JServ Protocol (AJP) support module listens. You **must** use a port on which AJP listens. For more information about AJP, read Apache's AJP documentation.
> - The following examples are **not** comprehensive. Many methods exist to proxy from Apache® to Tomcat.

To configure a proxy for EasyApache 4, create a virtual host include file for any domain for which you wish to proxy an application to Tomcat. Add your include file to the appropriate location in the /etc/apache2/ directory. For more information about Apache include files and their locations, read our Modify Apache Virtual Hosts with Include Files documentation.

After you create or edit an Apache include `userdata` file, you must rebuild the `httpd.conf` file and restart Apache for the changes to take effect.

To rebuild the `httpd.conf` file, run the following script:

```
/usr/local/cpanel/scripts/rebuildhttpdconf
```

To restart Apache, run the following script:

```
/usr/local/cpanel/scripts/restartsrv_httpd
```

LiteSpeed Web Server

To use LiteSpeed with Tomcat, your server **must** use the Tomcat instance's assigned port. LiteSpeed is **not** compatible with the `ProxyPassMatch` directive and will **not** proxy from the assigned port to port `80`.

Configure an app for AJP

To configure an app to use AJP, your entry might resemble the following example:

> **Note:**

```
<IfModule proxy_ajp_module>
 ProxyPass "/docs" "ajp://127.0.0.1:11111/docs"
</IfModule>
```

In the above example, if you navigated to `http://example.com/docs`, the server would proxy the response to the `docs` Tomcat application.

Set up a subdomain proxy

To set up a subdomain proxy, your entry might resemble the following example:

Note:
In this example, `/store` represents the Tomcat application and `11111` represents the user's AJP port number.

```
<IfModule proxy_ajp_module>
 ProxyPass "/" "ajp://127.0.0.1:11111/store"
</IfModule>
```

In the above example, if you navigated to `http://store.example.com`, the server would proxy the response to the `store` Tomcat application.

Mimic EasyApache 3 Tomcat proxy behavior in EasyApache 4

You can mimic EasyApache 3's proxy behavior in a private instance and run applications from the user's document root.

Warning:
We **strongly** recommend that you do **not** mimic EasyApache 3's behavior. If you create a proxy for Tomcat to a user's document root, a malicious user could bypass the Apache access permissions for the files located in that directory. If you choose to perform this action, you **must** limit your Tomcat ports to **only** the users that require access.

We also **strongly** recommend that you disable listening on any ports that you do not intend to use. For example, if you will only use HTTP, we recommend that you disable the AJP port. Or, if you will only proxy to AJP, we recommend that you disable the HTTP port.

To configure this behavior, perform the following steps:

- Add a `Host` entry to the `~/ea-tomcat85/conf/server.xml` file.
- Create an include file to act as a proxy for the appropriate domain with the `mod_proxy_ajp` Apache module.

Your include file would resemble the following example, where `11111` represents the user's AJP port:

Note:
In this example, `11111` represents the user's AJP port number.

```
<IfModule proxy_ajp_module>
 ProxyPassMatch "^/(.\.jsp|.\.do|(?:./)?servlets?/.)$"
"ajp://127.0.0.1:11111/$1"
</IfModule>
```

This method ensures that the application functions similarly to EasyApache 3, except it now runs as the user.

Additional documentation

- Tomcat
- Tomcat Private Instances

Content by label

There is no content with the specified labels

- Tomcat Private Instances
- Tomcat
- Tomcat Proxies
- EasyApache 4 Release Notes
- EasyApache 4 Recommendations

- WHM API 1 Functions - ea4_tomcat85_rem
- WHM API 1 Functions - ea4_tomcat85_list
- WHM API 1 Functions - ea4_tomcat85_add
- WHM API 1 Functions - php_set_vhost_versions
- WHM API 1 Functions - ea4_metainfo