

# \_includes\_wsc55x\_apache\_tomcat\_server\_setup

## Option A: Binding of Static Files and Serving Service Requests by AppServer

To bind only static files and the rest of the service requests such as spelling or grammar checking directly connecting with AppServer:

1. Stop Apache Tomcat before modifying the configuration files.
2. Add the following paths to the **server.xml** file inside the **<Host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true">** tag:

```
<Context docBase="/opt/WSC/WebComponents/WebInterface" path="/wscservice" />
<Context docBase="/opt/WSC/WebComponents/Samples" path="/wscservice/samples" />
```

3. If you plan to host WebSpellChecker and your web app(s) on different domains, you might face the CORS issues. To avoid this, add the **Access-Control-Allow-Headers** to enable cross-origin requests to the Apache Tomcat settings. For this you can use [CORS filter](#).
4. Start Apache Tomcat after modifying the configuration files.

## Option B: Serving both Static Files and Service Requests by Java Application Server (deployment of wscservice.war)

To deploy wscservice.war on Apache Tomcat for binding static files and all service requests to engines such as spelling or grammar checking:

1. Set and check the **WSC\_DIR** environment variable. It must be pointed to the WebSpellChecker Server installation directory (for example, **WSC\_DIR=/opt/WSC** as in example below):

```
export WSC_DIR=/opt/WSC
echo @WSC_DIR
```

2. Deploy **wscservice.war** on your Apache Tomcat Application Server.

- Copy the **wscservice.war** file from **<WSC\_DIR>/J2EEServerComponents** to the Apache Tomcat **/webapps** directory.
- Start Apache Tomcat, or stop and then start if it is already running.



Note that the **wscservice.war** file should be unpacked automatically. Otherwise, unpack it manually.