

Introduction to Java Server Pages (JSP)

Martin Pola

1 Installing Tomcat

To run our Java-based websites, we need a web server with JSP support, and one option is the *Tomcat* web server. Visit tomcat.apache.org and download the core as a compressed ZIP folder. The downloaded file will probably have a name similar to `apache-tomcat-9.0.30.zip`, possibly with another version in the 9.x.y series.

Extract the compressed ZIP folder somewhere on our computer and navigate to it. Among all the files and folders, you should find the following:

- `apache-tomcat-9.0.30`
 - `bin`
 - `shutdown.bat`
 - `shutdown.sh`
 - `startup.bat`
 - `startup.sh`
 - `conf`
 - `catalina.properties`
 - `webapps`
 - `ROOT`
 - `WEB-INF`

Open a terminal and use `cd` to navigate to `apache-tomcat-9.0.30`. Start Tomcat by running `bin\startup.bat` if you're using Windows or the `.sh` equivalent if you're on a Unix-like system. Make sure your server is up and running by visiting `http://127.0.0.1:8080/` in your web browser.

To be successful, you should see a page with this text:

If you're seeing this, you've successfully installed Tomcat. Congratulations!

2 Print something dynamic

In the `webapps/ROOT` folder, create a new file called `hello.jsp`. Copy and paste the contents from [this sample hello.jsp](#) and save the file.

In your web browser, visit `http://127.0.0.1:8080/hello.jsp`. Note that you will not have to compile your JSP file; Tomcat will automatically do that for you if it hasn't been compiled already. Tomcat will also re-compile your JSP file if the date of the JSP file is more recent than the date of the compiled version.

3 Preparing for classes

While we in the `hello.jsp` file seem to have forgot everything about classes and methods, like if we had one single `main` method, it is still possible to use classes with JSP. That could be useful if you for example want to load or store blog posts – for that purpose we could have a `BlogPost` class.

We will store all custom classes in the `webapps/WEB-INF` folder. In order for Tomcat to find them, we will have to modify the `conf/catalina.properties` file. Open that file and find the line that starts with `common.loader=`.

At the very end of that line, add this:

```
, "${catalina.base}/webapps/ROOT/WEB-INF"
```

Restart Tomcat by shutting it down and starting it again in your terminal:

```
$ bin\shutdown.bat
$ bin\startup.bat
```

(Use normal slashes instead of backslashes if you're on a Unix-like system.)

4 Using a custom class

Now we will create a class to represent entries (posts) in a blog, and then use it to print the entries.

1. In the folder `webapps/ROOT/WEB-INF`, create a new folder called `blog`.
2. Inside that folder, create a new file called `BlogPost.java`.
3. Populate the file with contents from our [sample BlogPost.java](#).
4. Navigate to the folder and compile the file with `javac`.

After compilation, your `webapps/ROOT/WEB-INF/blog` folder should contain the files `BlogPost.java` and `BlogPost.class`.

To use the class, go to `webapps/ROOT` and put the contents of [our blog.jsp](#) in that directory, in a new file called `blog.jsp`.

Finally, visit the website: `http://127.0.0.1:8080/blog.jsp`