

Logging

For compartmentalization of logs in your environment or to provide Tiny with more succinct feedback around the behavior of your deployed [Textbox.io](#) Server Components (e.g. for support purposes), you may want to write out the [Textbox.io](#) logs to a specific file.

To write the [Textbox.io](#) logs to a specific file, you'll need to perform the following steps:

Step 1 - Create a logging configuration XML file

The [Textbox.io](#) services use the [Logback](#) logging format.

For easy implementation, here is a sample XML configuration with a tokenized value you can populate:

Where `{LOG_LOCATION}` is the location and name of the file you would like to write the logs to (e.g. `/tmp/textboxio_services.log`)

```
<configuration>

  <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
    <encoder>
      <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
    </encoder>
  </appender>

  <appender name="FILE" class="ch.qos.logback.core.FileAppender">
    <file>{LOG_LOCATION}</file>
    <encoder>
      <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</pattern>
    </encoder>
  </appender>

  <!-- This results in all ephox logging going to file.
       Change/uncomment this part here if spellchecking specific logging is required -->
  <logger name="com.ephox" level="INFO"/>
  <!-- <logger name="com.ephox.honey" level="INFO"/> -->

  <root level="INFO">
    <appender-ref ref="FILE" />
    <!-- If you want logging to go to the container as well uncomment
         the following line -->
    <!-- <appender-ref ref="STDOUT" /> -->
  </root>

</configuration>
```

Step 2 - Add the logging configuration to your classpath

Much like how your `application.conf` services configuration file is added to the classpath (see Step 3 of the [Spell Checking Install Guide](#)), you'll need to follow a similar pattern to add your logging configuration XML to the classpath.

```
-Dlogback.configurationFile={LOG_LOCATION}
```