Guided Exercise: Configuring Record Handlers

In this lab, you will create a handler for files that rotate based on size and implement an application that uses this handler to log messages.

Resources	
Files:	**/home/student/JB248/labs/standalone** **/home/student/JB248/labs/logging-handlers**
App URL:	**http://localhost:8080/logtest**
Resources	**/home/student/JB248/labs/logging-handlers/logtest.war** **/home/student/JB248/labs/logging-handlers/add_sizerotating_log.cli**

Results

You should be able to create a file handler that rotates based on size and see application generated log messages in the EAP server log files.

before you start

Before beginning the guided exercise, run the following command to verify that EAP has been installed to /opt/jboss-eap-7.0, that no EAP instances are running, and that the previous guided exercise has completed, as well as to download the logtest.war application:

```
[student@workstation ~]$ lab logging-handlers setup
```

1. Start the standalone EAP server.

Use the following command to start an EAP instance using the /home/student/JB248/labs/standalone folder as the base directory:

```
[student@workstation ~]$ cd /opt/jboss-eap-7.0/bin
[student@workstation bin]$ ./standalone.sh \
-Djboss.server.base.dir=/home/student/JB248/labs/standalone/
```

Before proceeding, wait for the server to finish starting up.

2. You will use a JBoss EAP CLI script file to create a new size-rotating file-handler and implement the logtest.war file. Briefly review the add_sizerotating_log.cli file located in the /home/student/JB248/labs/logging-handlers directory.

2.1. The commands in the add_sizerotating_log.cli file should be as follows describes below:

```
batch
```

```
/subsystem=logging/size-rotating-file-handler=FILE_BY_SIZE_ROTATING/:add\ (file={"path"=>"production-
 server.log",\ "relative-to"=>"jboss.server.log.dir"},\
 formatter="%d{HH:mm:ss,SSS} %-5p [%c] (%t)
%s%E%n",\ level=INFO,max-backup-index=3,name=FILE_BY_SIZE_ROTATING,
\ rotate-size=1m)


/subsystem=logging/logger=com.redhat.training.view:add\
 (category=com.redhat.training.view,handlers=["FILE_BY_SIZE_ROTATING"])

deploy /home/student/JB248/labs/logging-handlers/logtest.war

run-batch
```

The script will configure the logging subsystem to use the handler named FILE_BY_SIZE_ROTATING to get all the logs generated by the com.redhat.training.view category (representing a Java package in which the logging source code will be executed), and will capture all logs generated with the INFO level in a file named production-server.log, which will be located in the /home/student/JB248/labs/standalone/log folder.

After the log file reaches 1 MB in size, the logging subsystem will rotate the log file to a new log file with a numbered suffix. There will be a maximum of three (3) log files before the contents of the log files are iteratively overwritten.

The application named logtest is deployed to the server. It is a Java web application with all the source code in the com.redhat.training.view package.

3. Run the CLI script in a new terminal window:

```
[student@workstation ~]$ cd /opt/jboss-eap-7.0/bin
[student@workstation bin]$ ./jboss-cli.sh -c \ --file=/home/
student/JB248/labs/logging-handlers/add_sizerotating_log.cli The batch executed successfully
```

Analyze the above message to verify if the batch commands were executed successfully. If you see errors during execution, you can try inserting the commands line by line into the CLI request and debugging the errors.

4. Use the following CLI commands to verify that the handler has been added correctly:

```
[student@workstation bin] ./jboss-cli.sh --connect
[standalone@localhost:9990] /subsystem=logging/\ size-rotating-
file-handler=FILE_BY_SIZE_ROTATING:read-resource
```

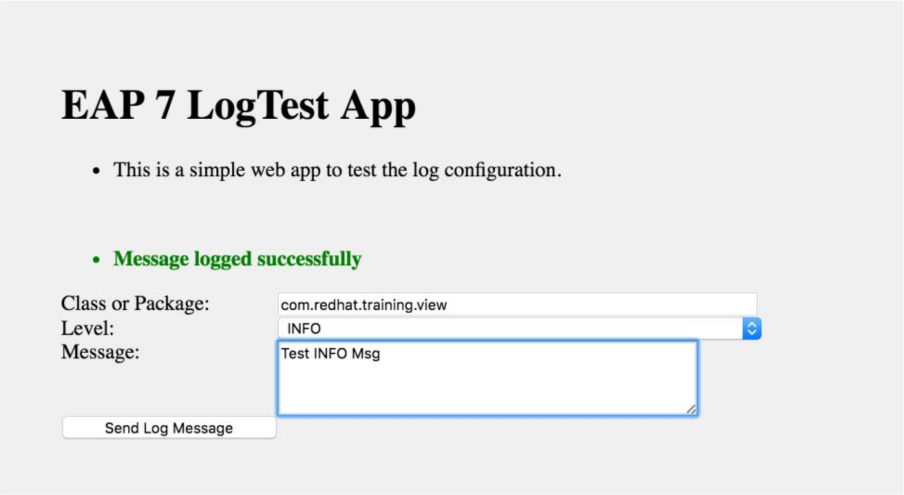
The result should appear as follows:

```
{
        "outcome" => "success", "result"
        => { "append" =>
                true, "autoflush" =>
                true, "enabled" => true,
                "encoding" => undefined,
                "file" => {
                        "relative-to" => "jboss.server.log.dir", "path" =>
                        "production-server.log"
                },
                "filter" => undefined, "filter-
                spec" => undefined, "formatter" =>
                "%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n", "level" => "INFO", "max-
                backup-index" => 3,
                "name" =>
                "FILE_BY_SIZE_ROTATING", "named-
                formatter" => undefined, "rotate-on-boot"
                => false, "rotate-size" => "1024m",
                "suffix" => undefined

        }
}
```

5. Test the registry.

5.1. Navigate to http://127.0.0.1:8080/logtest/ on the workstation virtual machine to access the logtest application.



EAP 7 LogTest App

- This is a simple web app to test the log configuration.

- Message logged successfully

Class or Package: com.redhat.training.view
Level: INFO
Message: Test INFO Msg

5.2. • Enter com.redhat.training.view in the Class or package field.

• Select the INFO level from the Level dropdown list.

• Enter the text Test INFO Msg in the Message field.

Click Send Log Messages to send the log message to the handler you created in the previous step.

5.3. Open a new terminal window to view the log file in /home/ student/JB248/labs/standalone/log/production-server.log and verify that you can see the messages logged by the application.

```
[student@workstation ~]$ tail \ -f /home/
student/JB248/labs/standalone/log/production-server.log 01:52:05,392 INFO
[com.redhat.training.view] (default task-10) Test INFO Msg
```

5.4. Send a few more Application INFO messages and verify that the messages appear in the log file.

5.5. Change the Class or Package field to com.redhat.training in the logtest application and verify that the messages do NOT appear in the log file. This is because the handler is configured to only process log messages from the com.redhat.training.view package by the logger. However, the messages will appear in the console window in which you started EAP, since the CONSOLE handler is a child process of the ROOT logger, which logs all INFO level messages from any package.

6. Perform cleaning.

6.1. Undeploy the logtest.war application:

```
[standalone@localhost:9990 /] undeploy logtest.war
```

6.2. Exit the EAP CLI:

```
[standalone@localhost:9990 /] exit
```

6.3. Stop the EAP instance by pressing Ctrl+C in the terminal window that is running EAP.

Stop the tail command by pressing Ctrl+C in the terminal window that is running the tail command.

This concludes the guided exercise.