(../..)

Getting Started

Checkout the logging-start tag.

```
git checkout logging-start
```

Have a look at the `MoviesBean` and `AlbumsController` classes. These two classes are logging messages to Standard Output and Standard Error.

Let's introduce a logger so we can configure these outputs.

Introducing `SLF4J`

`SLF4J` (https://www.slf4j.org/) is introduced automatically by the Spring Boot starter, thus there are no changes needed in your `pom.xml`.

Create a logger field in both classes using the class `org.slf4j.Logger`

```
private final Logger logger = LoggerFactory.getLogger(this.getClass())
;
```

In `AlbumsController`, replace

```
System.out.println("Uploading cover for album with id " + albumId);
```

with

```
logger.debug("Uploading cover for album with id {}", albumId);
```

Notice that we avoid concatenating the `String` to display. The formatting will only occur if `debug` log level is enabled.

Replace

```
e.printStackTrace();
```

with

```
logger.error("There was an error while uploading album cover", e);
```

This way, you can configure the logger to output the error however you like.

In `MoviesBean`, replace the output on Standard Error using `logger.debug`. Make sure to not use `String` concatenation.

Configuring log levels

You can configure log levels using `application.yml`.

Levels are:

- `ERROR`
- `WARN`
- `INFO`
- `DEBUG`
- `TRACE`

Only statements with a log level at or above the currently configured log level will be logged. For instance, when `TRACE` is enabled, anything from `TRACE` and above is displayed. When `ERROR` is enabled, only `ERROR` shows up.

For example if I want to see `DEBUG` logs for `moviefun`, I can set the following in `application.yml`,

```
logging.level:
    root: INFO
    org.superbiz.moviefun: DEBUG
```

with `root` being the default log level.

If you would like to change log levels of a deployed application on Cloud Foundry, you can set an environment variable, then restage the application.

For example:

```
cf set-env moviefun _JAVA_OPTIONS -Dlogging.level.org.springframework=DEBUG
```

Log levels exercise

How would you configure the logging levels in `application.yml` so that:

- You can see the exception in `AlbumsController`.
- You cannot see the debug log in `AlbumsController`.
- You can see the debug log in `MoviesBean`.

Logging to an external service on Pivotal Cloud Foundry

Open a free account on Paper Trail (https://papertrailapp.com/)

- Enter an email and password
- Click `Subscribe`
- Click `Setup a new system`
- Capture the host and port at the top of the page. For example: `logs5.papertrailapp.com:13071`
- Create a user provided service on PCF, and bind it to your moviefun application:[*]

```
cf create-user-provided-service paper-trail -l syslog-tls://logs5.papertrailapp.com:13071
cf bind-service moviefun paper-trail
cf restage moviefun
```

[*] *Change the protocol of your papertrail url to syslog-tls to handle encrypted logging*

Check the logs on the Paper Trail website.

Assignment

Once you are done with the lab and the application is deployed and working on PCF, you can submit the assignment using the `submitReplatformingLogging` gradle task. It requires you to provide the `movieFunUrl` project property. For example:

```
cd ~/workspace/assignment-submission
./gradlew submitReplatformingLogging -PmovieFunUrl=http://my-movie-fun
.cfapps.io
```

(https://pivotal.io)

course version: 1.5.3