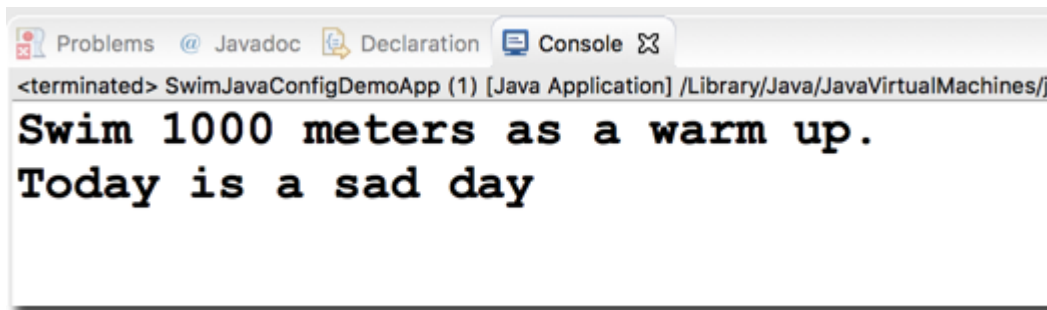


Heads Up - Add Logging Messages in Spring 5.1 - All Java Config Version

The Problem

In Spring 5.1, the Spring Development team changed the logging levels internally. As a result, by default you will no longer see the red logging messages at the INFO level. This is different than in the videos.



The Solution

If you would like to configure your app to show similar logging messages as in the video, you can make the following updates. Note, you will not see the EXACT same messages, since the Spring team periodically changes the text of the internal logging messages. However, this should give you some additional logging data.



The screenshot shows an IDE console window with tabs for Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the following log output:

```
<terminated> SwimJavaConfigDemoApp (1)
Oct 13, 2018 8:04:07 AM org.springframework.bean
FINE: Creating shared instance of singleton bean
Oct 13, 2018 8:04:07 AM org.springframework.bean
FINE: Creating shared instance of singleton bean
Oct 13, 2018 8:04:07 AM org.springframework.bean
FINE: Creating shared instance of singleton bean
Swim 1000 meters as a warm up.
Today is a sad day
Oct 13, 2018 8:04:07 AM org.springframework.cont
FINE: Closing org.springframework.context.annota
```

Overview of the steps

0. Create a logging properties file
1. Create a configuration class to configure the parent logger and console handler

Detailed Steps

0. Create a logging properties file

This properties file will define the logging levels for the application. The props file sets the logger level to FINE. For more detailed logging info, you can set the logging level to level to FINEST. You can read more about the logging levels

at <http://www.vogella.com/tutorials/Logging/article.html>

File: *src/mylogger.properties*

1. root.logger.level=FINE
2. printed.logger.level=FINE

1. Create a configuration class to configure the parent logger and console handler

This class will set the parent logger level for the application context. It will also set the logging level for console handler. The logging levels are loaded from the configuration file using the `@PropertySource` annotation. The fields are injected using the `@Value` annotation. This class also has a `@PostConstruct` method to handle the actual configuration.

File: MyLoggerConfig.java

```
1. package com.luv2code.springdemo;
2.
3. import java.util.logging.ConsoleHandler;
4. import java.util.logging.Level;
5. import java.util.logging.Logger;
6. import java.util.logging.SimpleFormatter;
7.
8. import javax.annotation.PostConstruct;
9.
10. import org.springframework.beans.factory.annotation.Value;
11. import org.springframework.context.annotation.AnnotationConfigApplicationContext;
12. import org.springframework.context.annotation.Configuration;
13. import org.springframework.context.annotation.PropertySource;
14.
15. @Configuration
16. @PropertySource("classpath:mylogger.properties")
17. public class MyLoggerConfig {
18.
19.     @Value("${root.logger.level}")
20.     private String rootLoggerLevel;
21.
22.     @Value("${printed.logger.level}")
23.     private String printedLoggerLevel;
24.
25.     @PostConstruct
26.     public void initLogger() {
27.
28.         // parse levels
29.         Level rootLevel = Level.parse(rootLoggerLevel);
30.         Level printedLevel = Level.parse(printedLoggerLevel);
31.
32.         // get logger for app context
33.         Logger applicationContextLogger = Logger.getLogger(AnnotationConfigA
pplicationContext.class.getName());
34.
35.         // get parent logger
36.         Logger loggerParent = applicationContextLogger.getParent();
37.
38.         // set root logging level
39.         loggerParent.setLevel(rootLevel);
```

```
40.  
41.         // set up console handler  
42.         ConsoleHandler consoleHandler = new ConsoleHandler();  
43.         consoleHandler.setLevel(printedLevel);  
44.         consoleHandler.setFormatter(new SimpleFormatter());  
45.  
46.         // add handler to the logger  
47.         loggerParent.addHandler(consoleHandler);  
48.     }  
49.  
50. }
```

Source code is available at the following link

<https://gist.github.com/darbyluv2code/a49009fe1f92f95a30d2d5f7ac987ce5>

Once you make these updates, then you will be able to see additional logging data. :-)