**Log 4j:**

* It is open source tool
* Using log4j we can store the flow details in file execution.
* We can use log statements while execution of project.
* Need to configure a “.properties” or .xml” (inside testNG xml we need to create the log4j) file in order to use the Log4j.

**Principles in log4j:**

1. **Logger – What to print**
2. **Appender – Where to save(console, txt file or log file …etc)**
3. **Layout – Design of logging in the appender**
4. **Loggers**: It is responsible for logging information. To implement loggers into a project following steps need to be performed -

* **Create an instance for logger class**: Logger class is a Java-based utility that has got all the generic methods already implemented to use log4j
* **Define the log level**: Primarily there are five kinds of log levels
  1. **All** - This level of logging will log everything ( it turns all the logs on )
  2. **DEBUG – print the debugging information and is helpful in development stage**
  3. **INFO – print informational message that highlights the progress of the application**
  4. **WARN – print information regarding faulty and unexpected system behavior.**
  5. **ERROR – print error message that might allow system to continue**
  6. **FATAL** – print system critical information which are causing the application to crash
  7. **OFF** – No logging

1. **Appenders**: It is used to deliver LogEvents to their destination. It decides what will happen with log information. In simple words, it is used to write the logs in file. Following are few types of Appenders

* ConsoleAppender logs to standard output
* File appender prints logs to some file
* Rolling file appender to a file with maximum size

1. **Layouts**: It is responsible for formatting logging information in different styles.

# Each conversion specifier starts with a percent sign (%) and is followed by optional format modifiers and a conversion character

# %d{dd MMM yyyy HH:mm:ss,SSS} - Used to output the date of the logging event, we can specify date format specifier enclosed between braces

#%d{ISO8601} Formats a Date in the format "yyyy-MM-dd HH:mm:ss,SSS" for example "1999-11-27 15:49:37,459".

# %F - Used to print file name

# %-5p means the priority of the logging event should be left justified to a width of five characters.

# %t - Used to output the name of the thread that generated the logging event.

# %L - Used to output the line number from where the logging request was issued.

# %m - Used to output the application supplied message associated with the logging event.

# %M - Used to output the method name where the logging request was issued.

Example:

**i/p:**

%d{dd:mm:yyyy, hh:mm:ss} %F : %L %M 🡪 %m

**o/p:**

06:09:2017, 08:56:23 filename.java : 87 Methodname 🡪 message

The Logger class provides different methods to handle logging activities. It provides two static methods for obtaining a Logger Object.

* **Public static Logger getRootLogger()**
* **Public static Logger getLogger(String name)**

How to use in script?

Specify your respective properties file in the java execution file.

**PropertyConfigurator.*configure*("nameofthefile.properties")**

**(OR)**

If you r using a xml file

**DomConfigurator.configure(“nameofthefile.xml”)**

In code, we have used "log" as a reference variable referencing getLogger method of Logger Class

L**ogger log = Logger.getLogger("name");**

**(Or)**

L**ogger log = Logger.getLogger(name of the class);**

Use "log" referencing variable and debug method to log the **information** we want.

**log.info(“—information--”);**

**log.error(“—information--”)**

**log.warn(“—information--”)**

**log.debug(“—information--”)**

**o/p:**

name information

name information

**Downloading Log4j Jar**

**Steps to follow:**

1) Go to Apache Logging Services and click **Apache log4j**.

2) Click on “**Download**” on the left side menu.

[*https://logging.apache.org/log4j/1.2/download.html*](https://logging.apache.org/log4j/1.2/download.html)

[*http://www.apache.org/dyn/closer.cgi/logging/log4j/1.2.17/log4j-1.2.17.zip*](http://www.apache.org/dyn/closer.cgi/logging/log4j/1.2.17/log4j-1.2.17.zip)

3) You will always get the latest version here. Click on the ZIP file under **Mirrors** column.

4) Click on the highlighted link at the top of the page.

5) Select the radio button for “**Save File**” and click **OK**. Zip file will be saved on your system with in few seconds.

6) Right click on the Zip file and select “**Extract All**“.

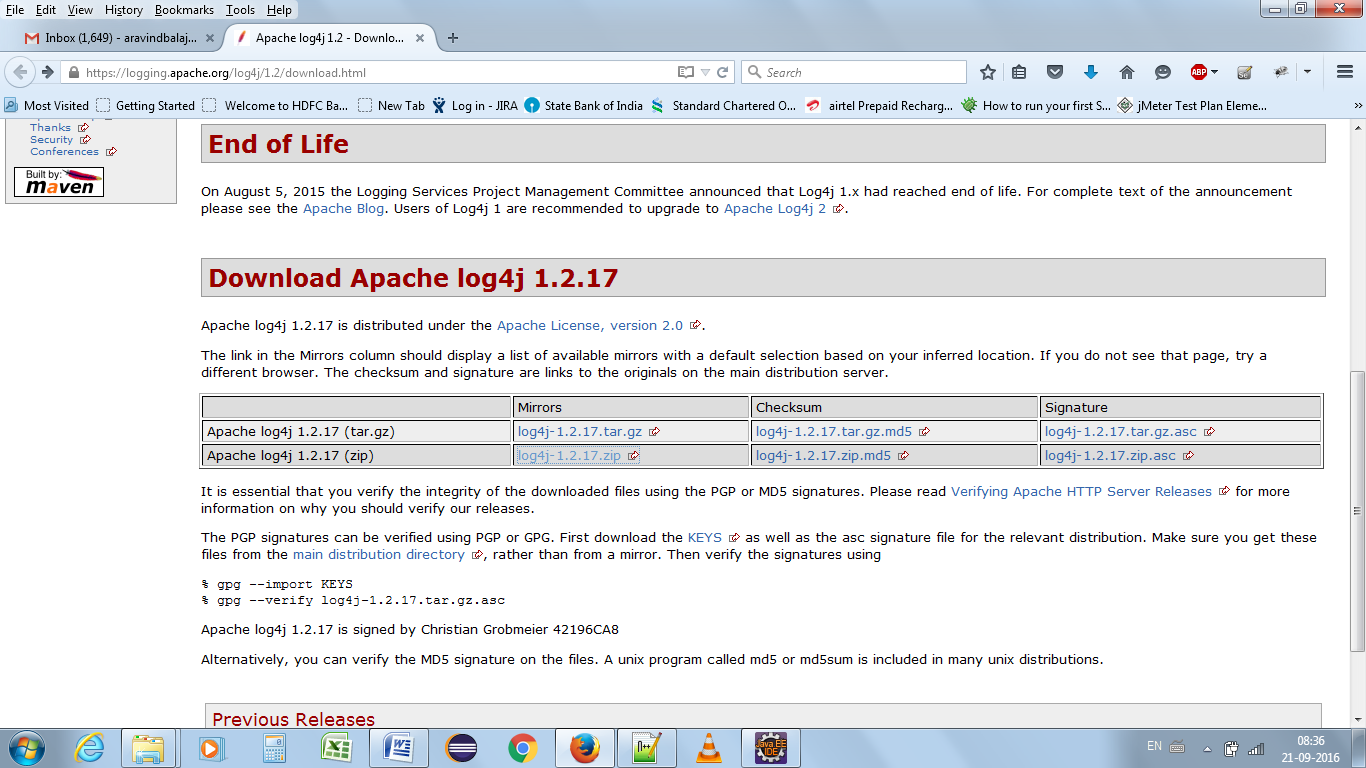
7) Specify the location.

8) Wait for the Extraction to finish.

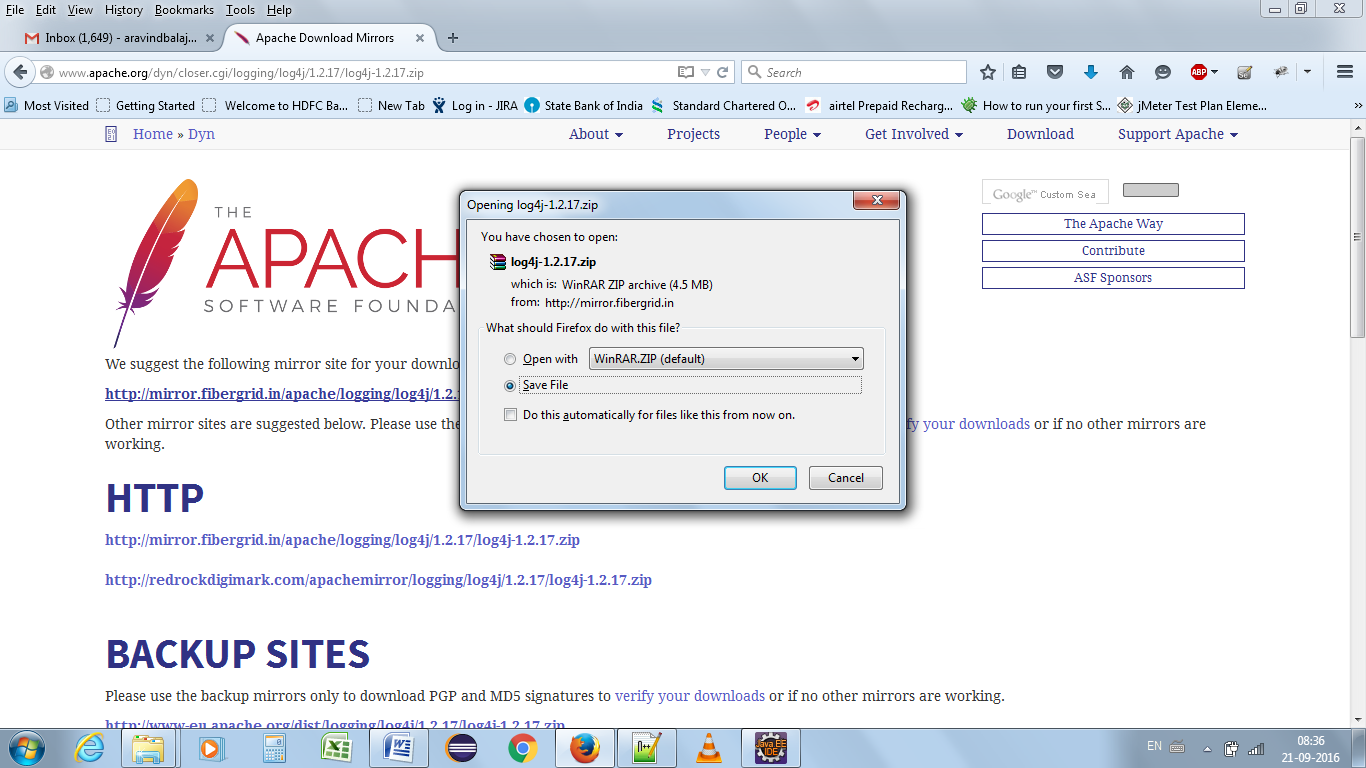
9) Open the Log4j extracted folder from the saved location.

**You are done with downloading the Log4j JAR file, now all you need to do is to add this JAR file to your project and write your first Log4j logging enabled test script.**

Download:







Add only the below jar to build path

