Automation Testing

TECHNOLOGY

Logging In REST Framework

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main()
{
    cout << "Hello, world!" << endl;
    return 0;
}
```

A Day in the Life of an Automation Test Engineer

Thomas has successfully authenticated two-way SSL authentication using various tools, such as PubSub, OpenSSL, and Postman. Now, however, he wants proper documentation of his changes in the REST Assured project. He knows about logging a bit from his prior lesson on XML Handling and utilities for parsing the XML and JSON responses, so he should only move forward with it. It will make the task less complicated.

To achieve the above goal, he must explore more about logging and learn about the tool Log4J.

Learning Objectives

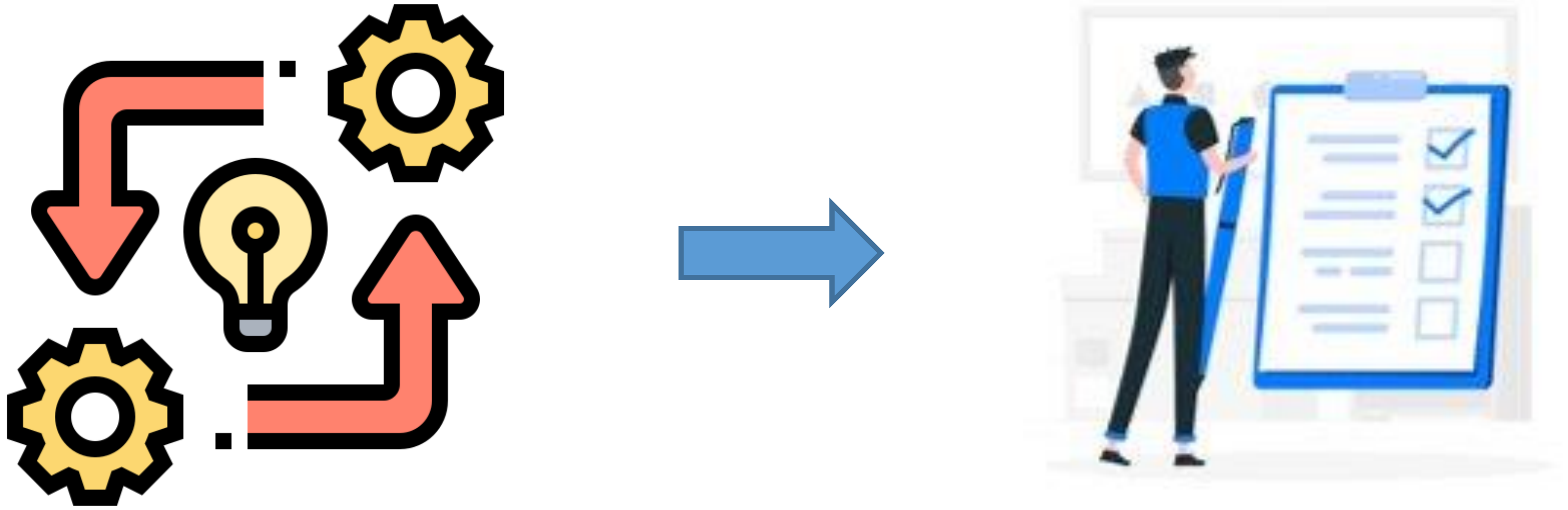
By the end of this lesson, you will be able to:

- Understand logging and its benefits and drawbacks
- Download and setup Log4J
- Understand the architecture of Log4J
- Use Log4J for logging

What Is Logging?

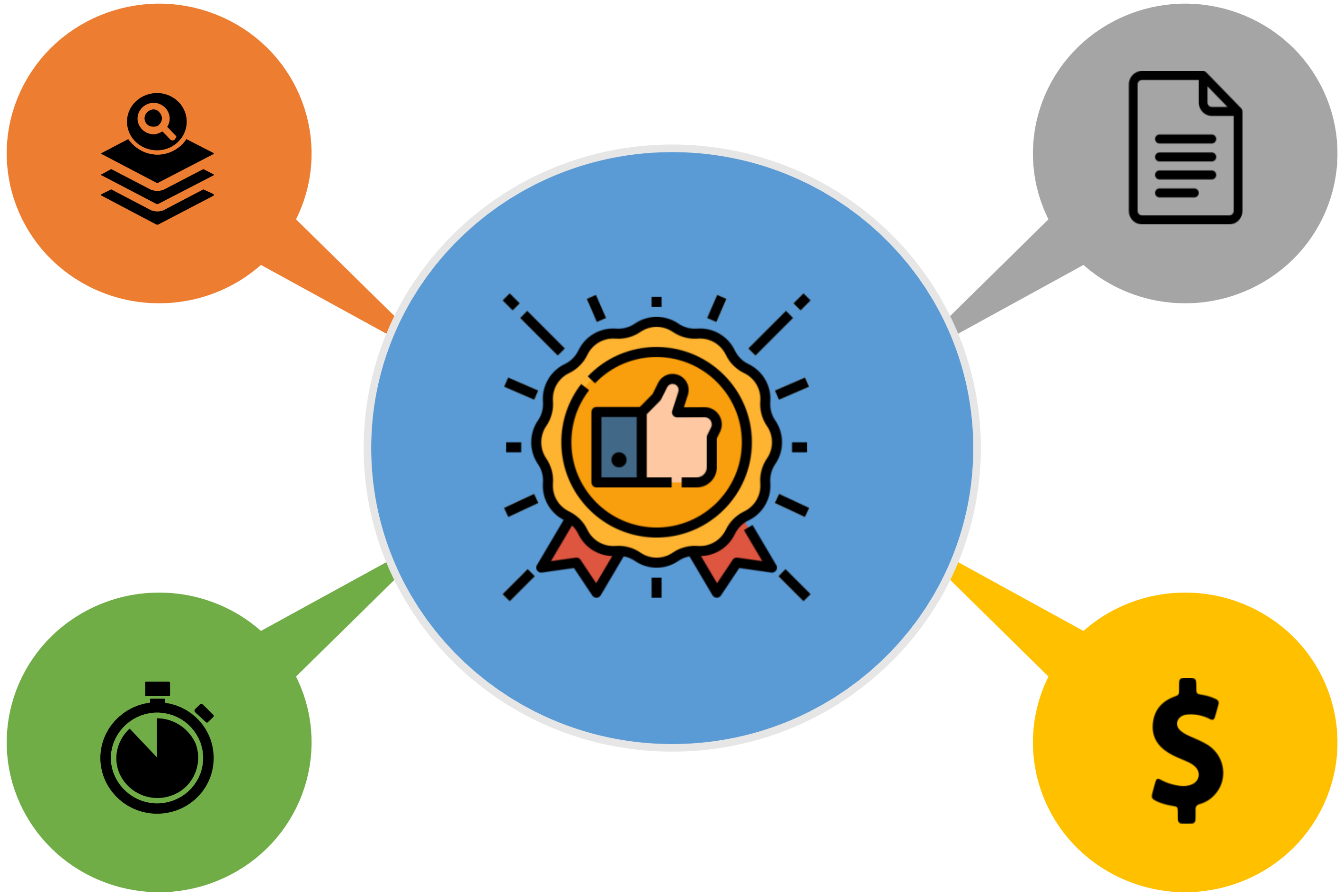
About Logging

An application or software comes across various bugs during its lifecycle. To debug it smoothly, logging is performed.



Advantages of Logging

Advantages of logging are as following:



It helps in diagnosing the problem

It is easy to maintain

It provides quick debugging

It saves time and cost

Disadvantages of Logging

Disadvantages of logging are as following:

It adds runtime overhead

It could lead to confusion, if produced badly

It increases storage requirement

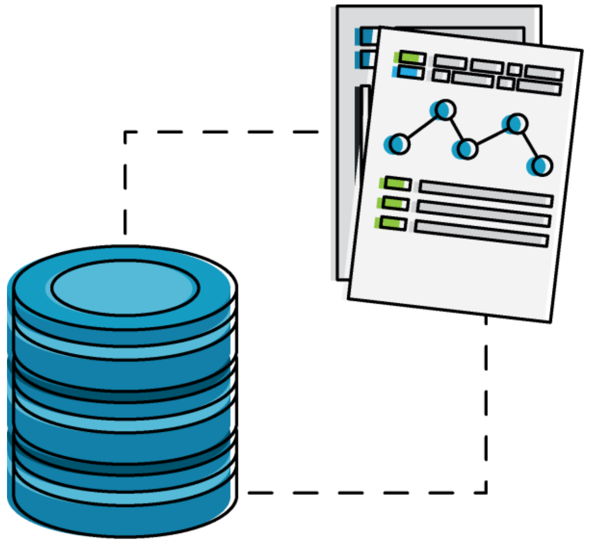
It cannot be added in between

About Log4J

What Is Log4J?

Log4J is an open-source logging API available for JAVA and .NET framework.

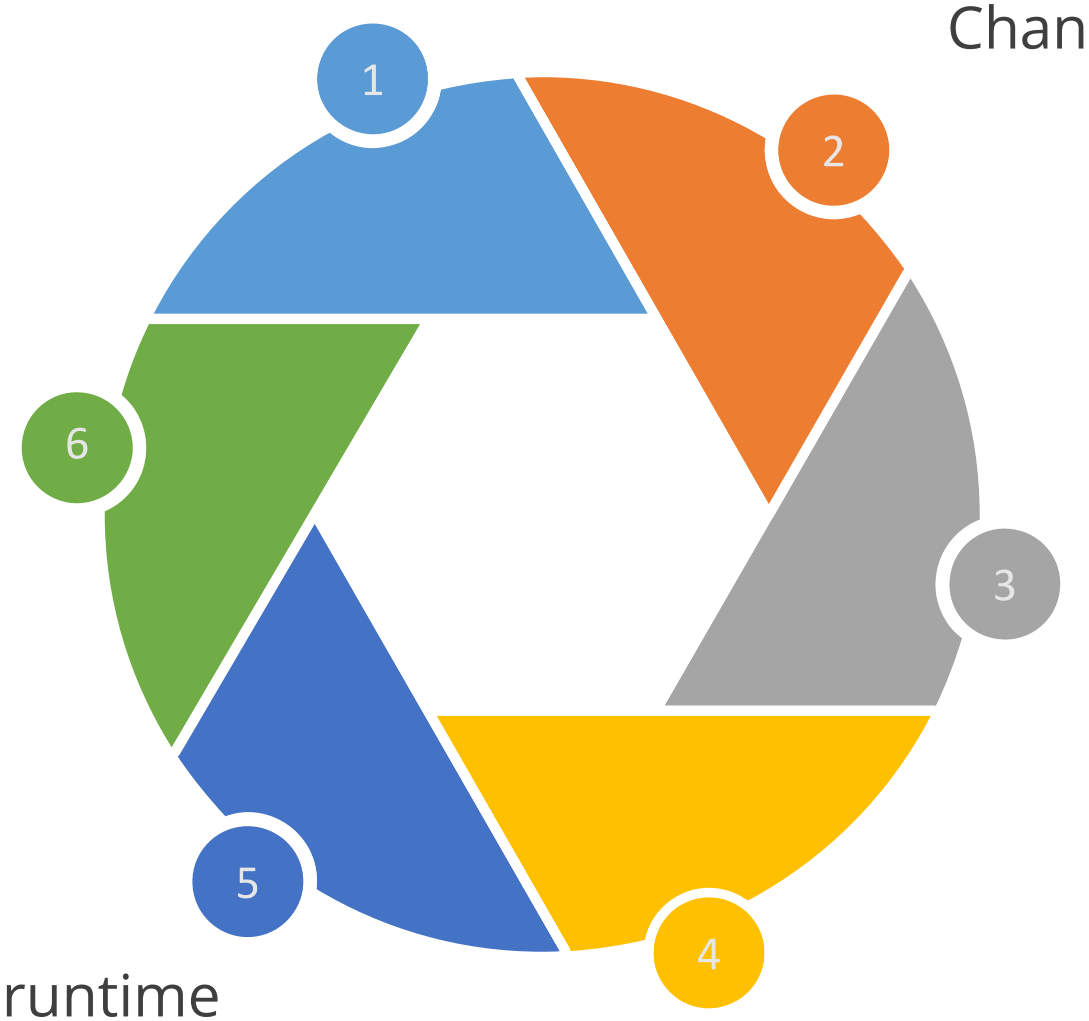
Log4J



Log4J is licensed under Apache.

Features of Log4J

Log4J has the following features:



1 Optimized speed

2 Changeable format of the output

3 Multiple output support

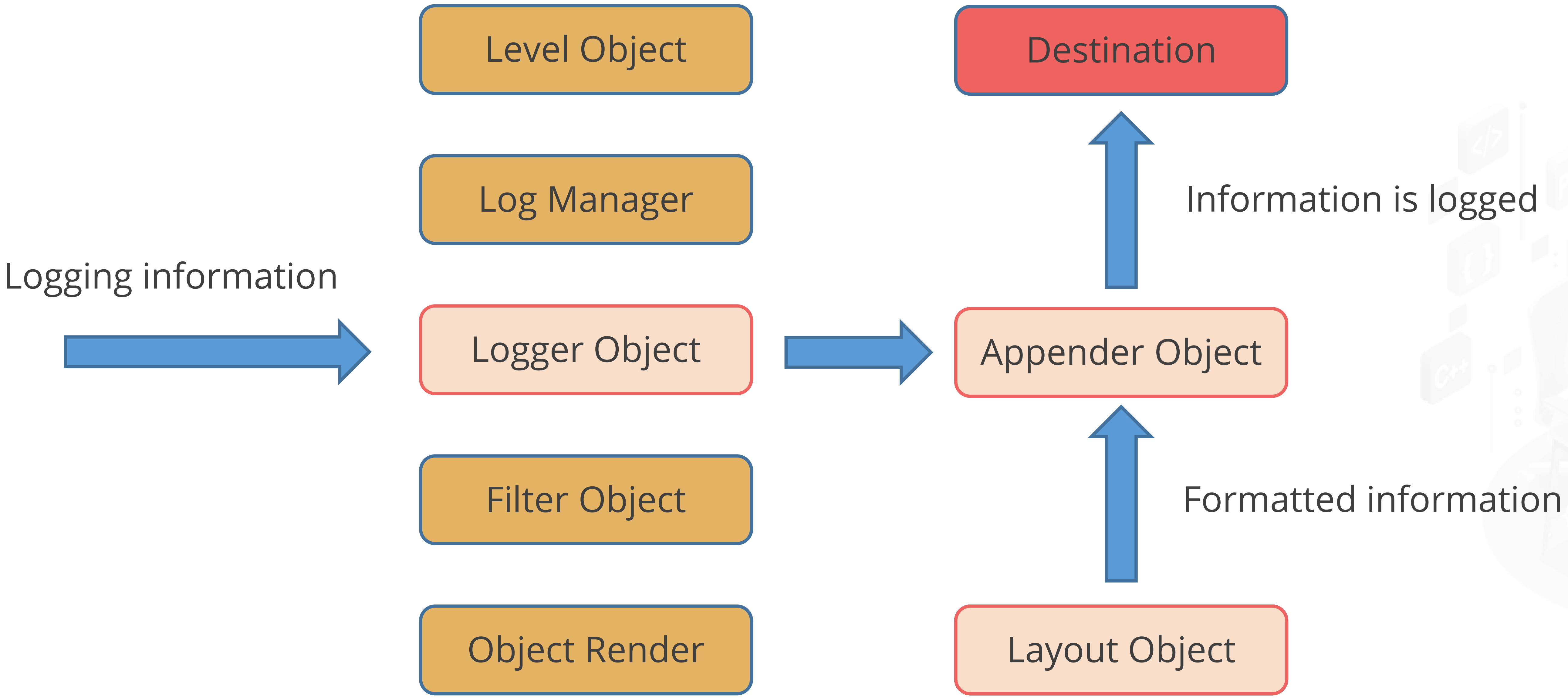
4 Predefined facilities

5 Configure behavior for runtime

6 Internationalization

Architecture of Log4J

An illustration of the Log4J architecture is shown in the flow chart below:



TECHNOLOGY

Various Log4J Objects

Various Log4J Objects

Log4J's architecture is made up of objects that perform various functions:

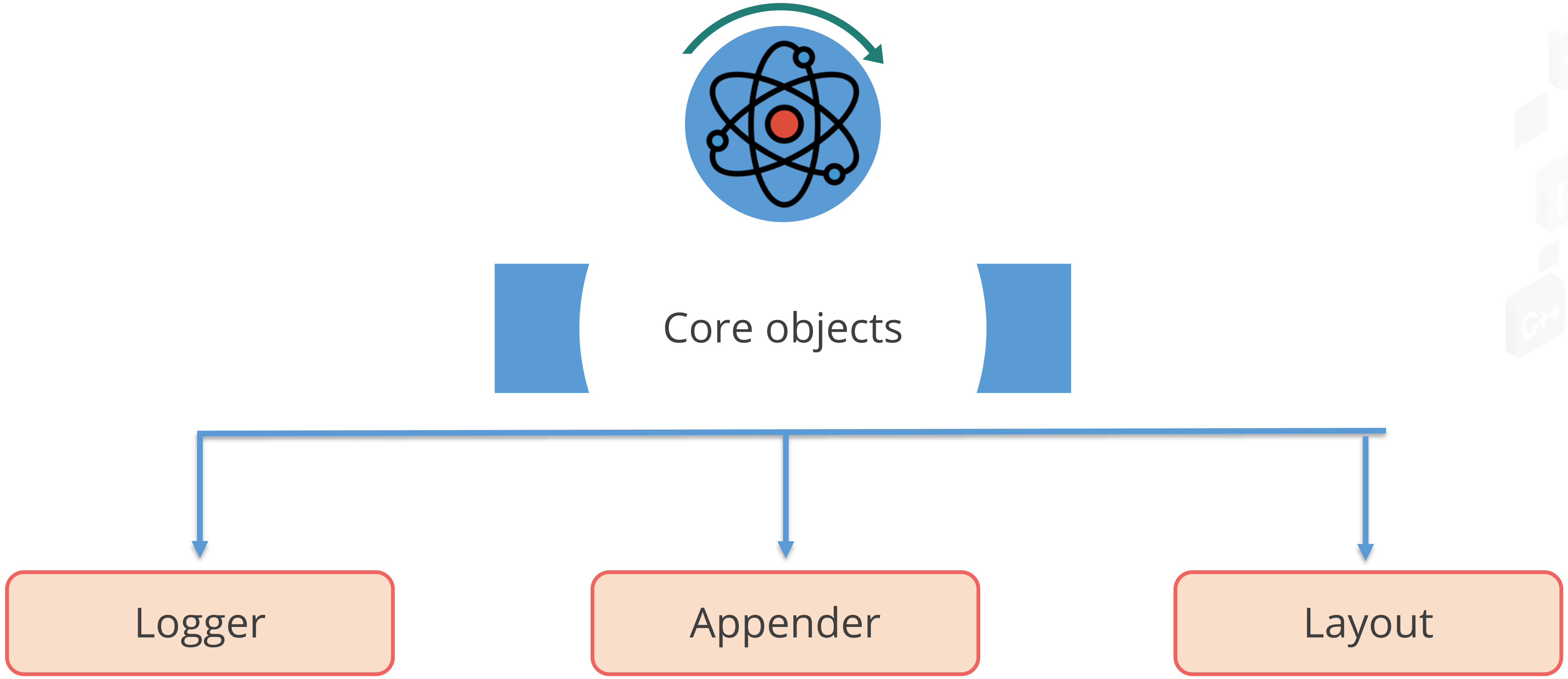
There are two types of objects:

Core objects

Support objects

Core Objects

The **Core objects** are mandatory elements in the Log4J architecture, and the log data passes through these objects.



Core objects

Logger

Appender

Layout

Logger

The **Logger object** comes under the top layer of the architecture.

The Logger object takes the information and stores it in a hierarchical manner.

Logger

Various methods of Logger object which print the status are :

Methods	Description
debug()	It is used for detailed tracing used by the end user.
info()	It is used to log information that makes sense to the end user.
warn()	It is used to log issues with some potential.
error()	It is used to log errors, which can stop execution.
fatal()	It is used to log serious issue that can terminate the program.

Logger

Method for getting Logger object in Java code:

```
static Logger log = Logger.getLogger(YourClassName.class.getName())
```

Appender

The lowest level of the architecture contains the Appender object.

The Appender object publishes the information provided by the Logger to various places.

Appender

There are different implementation classes of Appender in Log4J:

Appender	Description
FileAppender	It is used to append log events to a file and supports RollingFileAppender and DailyRollingFileAppender.
ConsoleAppender	It is used to append log events to System.out using a layout provided by the end user.
JDBCAAppender	It is used for databases.
AMTPAppender	It is used to send emails when events occur.

Appender

There are different implementation classes of Appender in Log4J:

Appender	Description
SocketAppender	It is used when remote storage is required.
SyslogAppender	It is used to send messages to a remote Syslog domain.
TelnetAppender	It is used for sending read-only messages to sockets.
WriterAppender	It is used to append events to an OutputStream as per the user's choice.

Appender

Method for defining Appender object in Java:

```
log4j.appender.FILE=org.apache.log4j.FileAppender
log4j.appender.FILE.File=${log}/log.out
```

Layout

The Layout object offers the Layout object and gives the Appender many formats.

It makes the log in human-readable form.

Layout

There are different layout classes in Log4J:

Layout	Description
SimpleLayout	It prints the output with level and log information.
PatterLayout	It prints the output in a specified pattern by user.
HTMLLayout	It prints the output in HTML table format.

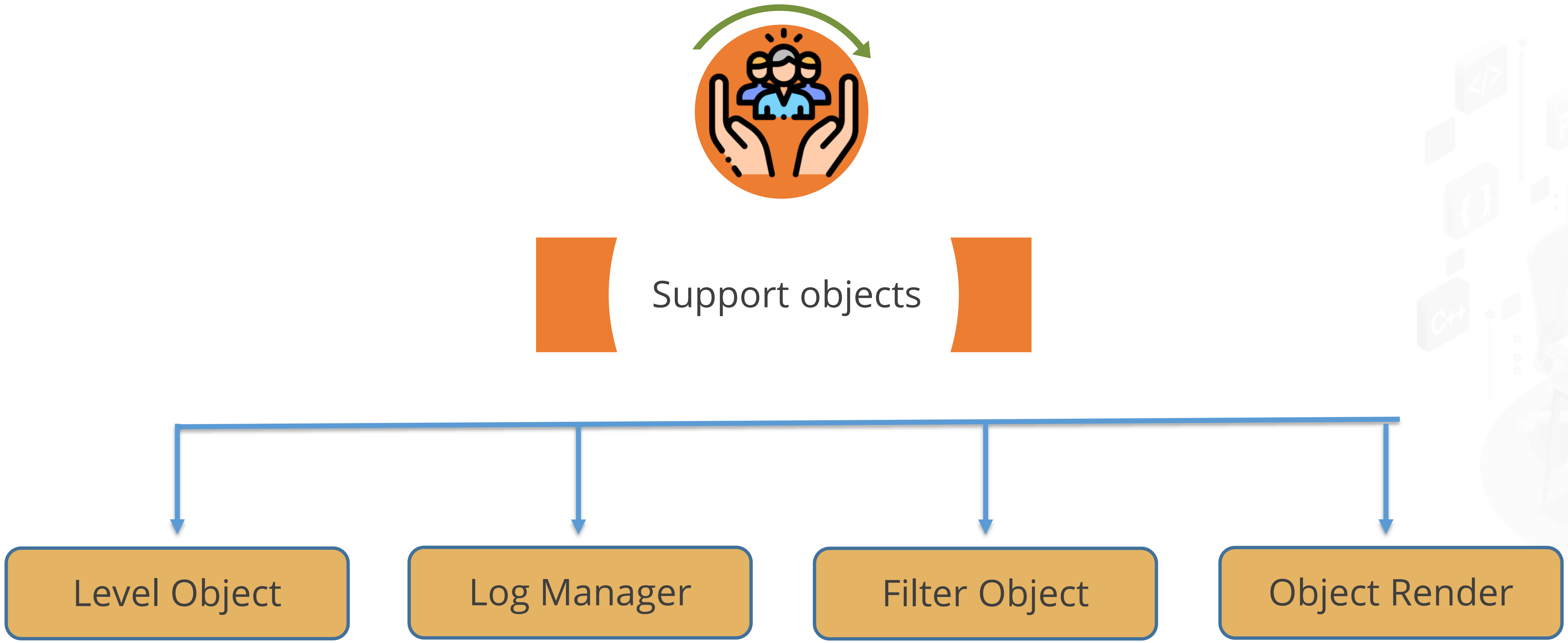
Layout

Method for defining Layout for Appender object in Java:

```
log4j.appender.FILE.layout=org.apache.log4j.PatternLayout
log4j.appender.FILE.layout.conversionPattern=%m%n
```

Support Objects

SLog4J Support objects are optional elements within the architecture, but they perform tasks that support Core objects.



Support objects

- Level Object
- Log Manager
- Filter Object
- Object Render

Level Object

The **Level object** defines the priority and granularity of the logging information based on the values associated with the methods.

Log Manager

The **Log Manager** helps to manage the framework by a file rendered from Log4J. It reads the initial configuration from the properties file.

Filter Object

The **Filter object** decides whether the information should be logged or not. It provides more control over the log information.

Render Object

The **Render object** specializes in converting the log data into strings. These strings can be easily formatted by the layout object.

Steps to Install Log4J

Steps to Install Log4J

Following are the steps to install Log4J in the computer:

Step 1

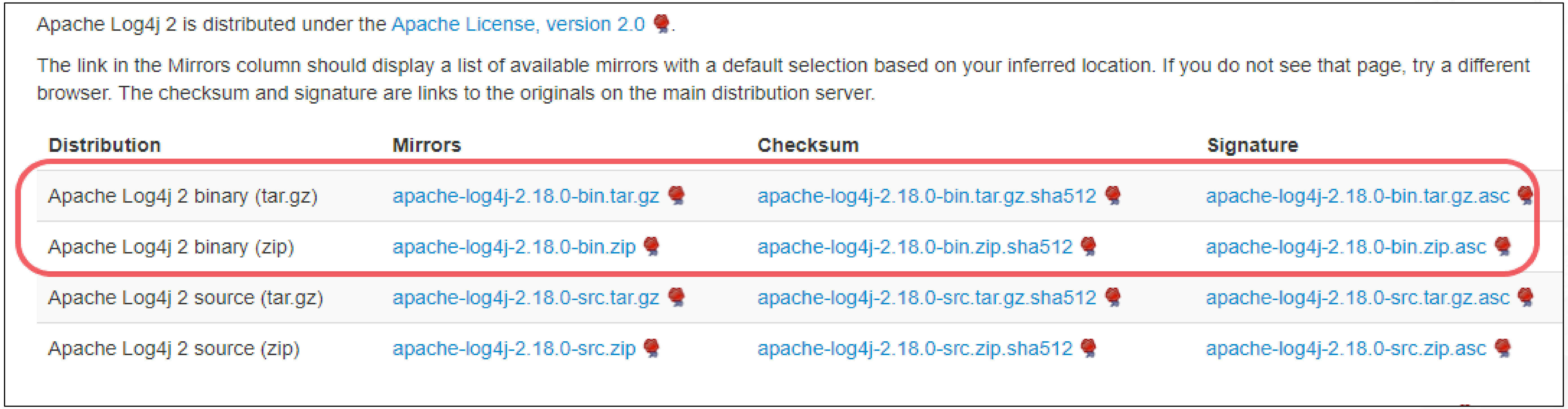
Download the Log4J version according to your operating system

Step 2

Unzip the downloaded file to the desired location

Steps to Install Log4J

Select the Apache Log4J binary files as per the operating system:



Apache Log4j 2 is distributed under the Apache License, version 2.0.

The link in the Mirrors column should display a list of available mirrors with a default selection based on your inferred location. If you do not see that page, try a different browser. The checksum and signature are links to the originals on the main distribution server.

Distribution	Mirrors	Checksum	Signature
Apache Log4j 2 binary (tar.gz)	apache-log4j-2.18.0-bin.tar.gz	apache-log4j-2.18.0-bin.tar.gz.sha512	apache-log4j-2.18.0-bin.tar.gz.asc
Apache Log4j 2 binary (zip)	apache-log4j-2.18.0-bin.zip	apache-log4j-2.18.0-bin.zip.sha512	apache-log4j-2.18.0-bin.zip.asc
Apache Log4j 2 source (tar.gz)	apache-log4j-2.18.0-src.tar.gz	apache-log4j-2.18.0-src.tar.gz.sha512	apache-log4j-2.18.0-src.tar.gz.asc
Apache Log4j 2 source (zip)	apache-log4j-2.18.0-src.zip	apache-log4j-2.18.0-src.zip.sha512	apache-log4j-2.18.0-src.zip.asc

Screenshot courtesy: https://logging.apache.org/log4j/2.x/build.html

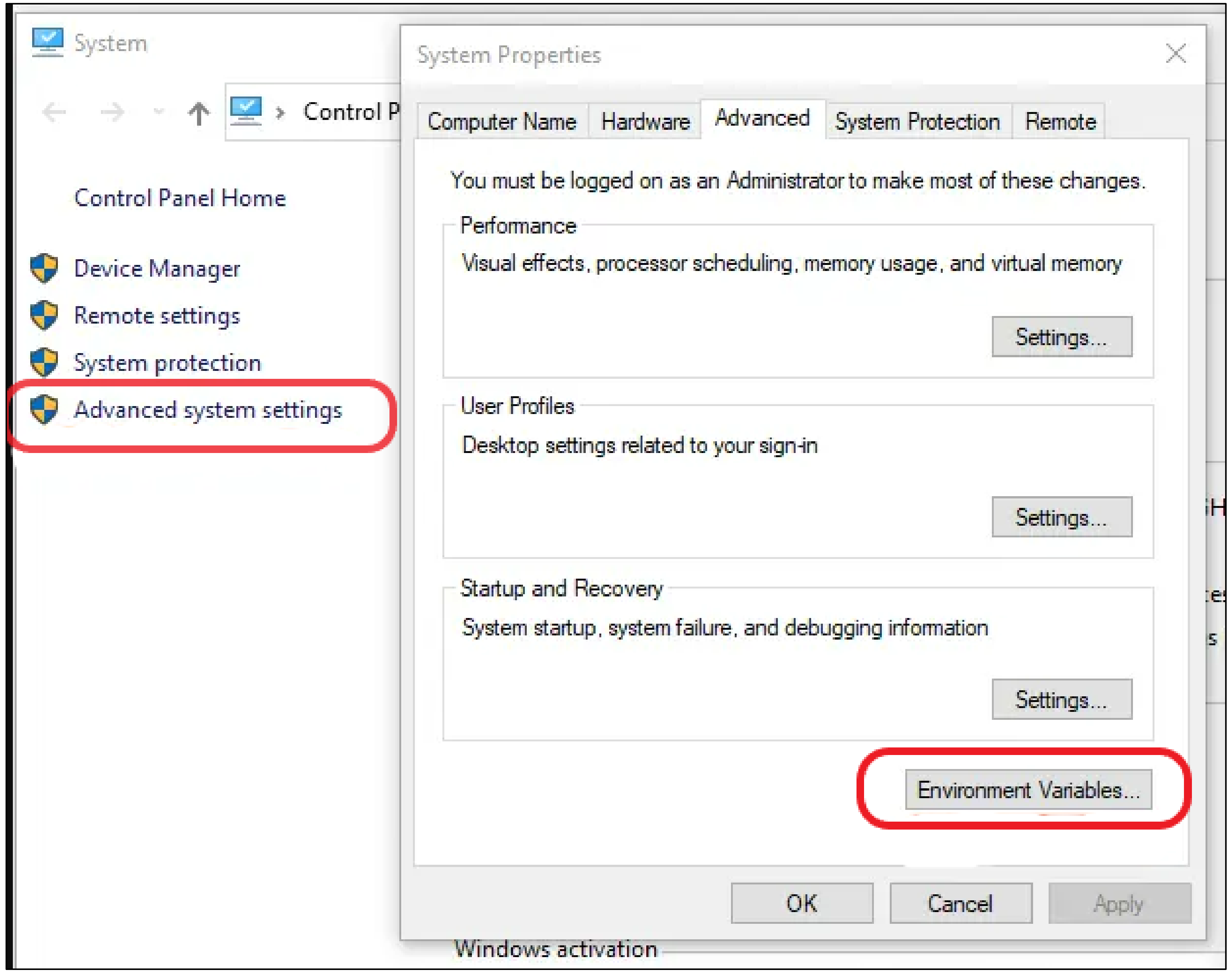
Steps to Install Log4J

Following are the steps to install Log4J in the computer:

✓ Step 3

Set up the environment variables

Steps to Install Log4J



You must be logged on as an Administrator to make most of these changes.

Performance

Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles

Desktop settings related to your sign-in

Startup and Recovery

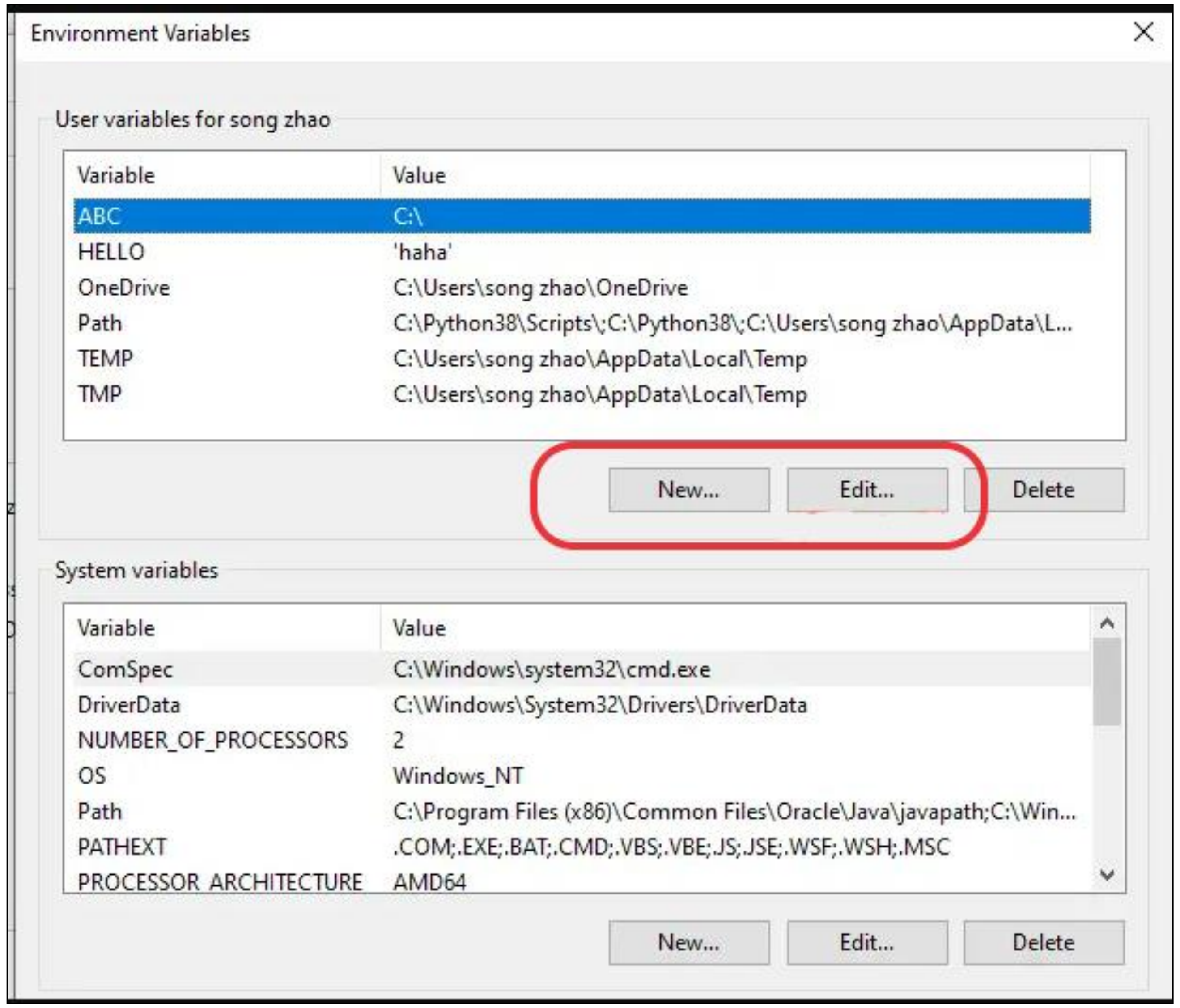
System startup, system failure, and debugging information

Screenshot courtesy: https://www.dev2qa.com/how-to-set-windows-environment-variables/

Advanced systems settings > Environment Variables

Steps to Install Log4J

Add the Path and Classpath of the directory and API JAR file in the Environment Variables window:



User variables for song zhao

Variable	Value
ABC	C:\
HELLO	'haha'
OneDrive	C:\Users\song zhao\OneDrive
Path	C:\Python38\Scripts\;C:\Python38\;C:\Users\song zhao\AppData\L...
TEMP	C:\Users\song zhao\AppData\Local\Temp
TMP	C:\Users\song zhao\AppData\Local\Temp

System variables

Variable	Value
ComSpec	C:\Windows\system32\cmd.exe
DriverData	C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS	2
OS	Windows_NT
Path	C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Win...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	AMD64

Screenshot courtesy: https://www.dev2qa.com/how-to-set-windows-environment-variables/

TECHNOLOGY

Log4J Properties

Log4J Properties

The **log4j.properties** file has runtime configurations.
The file contains:

- Appender information
- Log level information
- Output file names

Log4J Properties Syntax

The **log4j.properties** syntax has the following points:

It has a root level's logger defined as DEBUG which attaches appender named XYZ.

It has validated the name of appender.

It has a layout for appender XYZ.

Log4J Properties Syntax

The Java code syntax for log4j properties:

```
# Defining the root logger with appender XYZ
log4j.rootLogger = DEBUG, XYZ

# Putting the appender named XYZ to be a File appender
log4j.appender.XYZ=org.apache.log4j.FileAppender

# Defining the layout for XYZ appender
log4j.appender.XYZ.layout=org.apache.log4j.PatternLayout
log4j.appender.XYZ.layout.conversionPattern=%m%n
```

Key Takeaways

- The applications can be easily debugged by logging.
- The destination in Log4J can be changed according to the requirement.
- The architecture of Log4J contains Core and Support objects.
- The Log Manager uses the log4j.properties file, which contains all the configurations, to run the program.