



Apache LOG4J



Trình bày: Nhóm 4 - Cookies

Nhóm 4 - Cookies

MSSV	Họ tên	Chức vụ	% Công việc
19127038	Phạm Trọng Vinh Khuê	Nhóm trưởng	34
19127081	Trần Đại Hoàng Trung	Thành viên	33
19127311	Trần Thanh Tùng	Thành viên	33

Nội dung

1

Giới thiệu chung

2

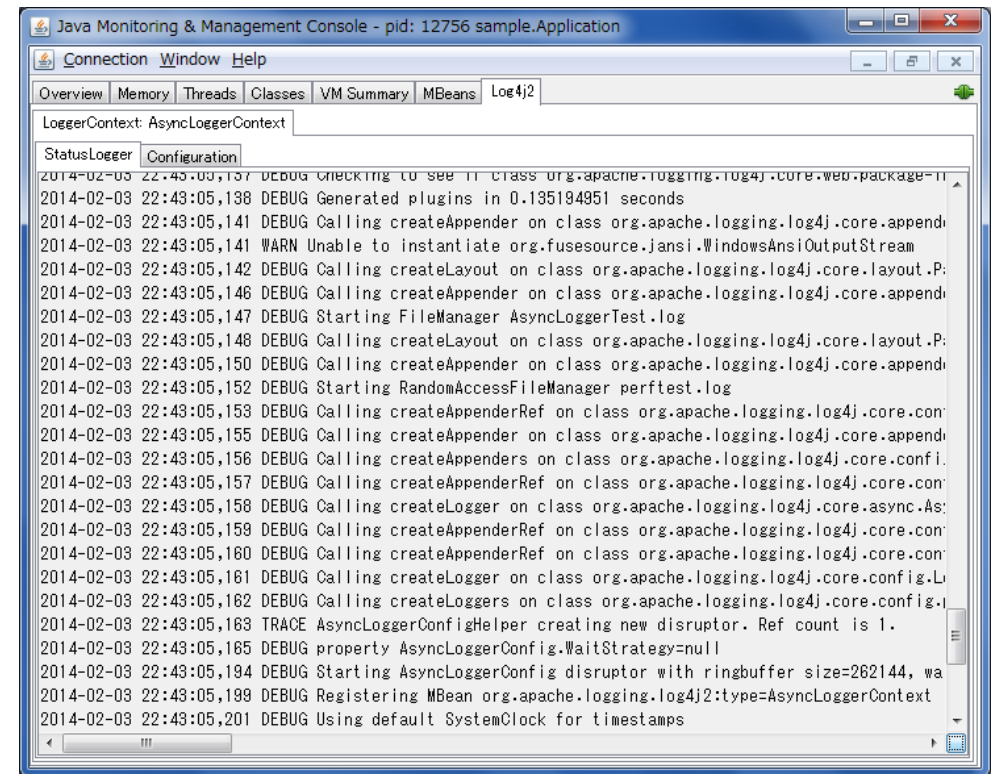
Hướng dẫn sử dụng

1 - Giới thiệu chung

Log4J, cung cấp bởi Tập đoàn Apache, là một trình dùng để ghi log cho các phần mềm Java. Nó theo dõi và ghi nhận những thông tin, sự kiện xảy ra trong quá trình thực thi chương trình.



(Nguồn: Wikipedia)



(Nguồn: logging.apache.org)

2 – Hướng dẫn sử dụng

Bước 1: Tải trình Log4j về máy.

- Truy cập đường dẫn:
<https://logging.apache.org/log4j/2.x/>
- Vào phần *Download*.
- Chọn link ở hàng thứ 1 hoặc hàng thứ 2 ở cột *Mirrors*, sau đó ấn vào link ở ngay bên dưới mục *HTTP*.



The screenshot shows the Apache Log4j 2 download page. At the top, there is a logo with a feather and the URL <http://logging.apache.org>. Below the logo, the text "Logging Services™" is displayed. A banner indicates "Last Published: 2021-03-06 | Version: 2.14.1". On the left, a sidebar menu lists various links: "About", "Download" (highlighted), "Javadoc", "Maven, Ivy, Gradle", "Artifacts", "Runtime Dependencies", "Changelog", "FAQ", "Performance", "Articles and Tutorials", and "Security". The main content area is titled "Download Apache Log4j 2" and states that Apache Log4j 2 is distributed under the Apache License, version 2.0. It explains that the link in the Mirrors column should display a list of available mirrors with a default checksum and signature are links to the originals on the main distribution server. Below this, a table lists the distribution and mirrors for Apache Log4j 2. The table has two columns: "Distribution" and "Mirrors". The rows are: "Apache Log4j 2 binary (tar.gz)" with mirror "apache-log4j-2.14.1-bin.tar.gz", "Apache Log4j 2 binary (zip)" with mirror "apache-log4j-2.14.1-bin.zip", "Apache Log4j 2 source (tar.gz)" with mirror "apache-log4j-2.14.1-src.tar.gz", and "Apache Log4j 2 source (zip)" with mirror "apache-log4j-2.14.1-src.zip". The first two rows are highlighted with an orange border.

http://logging.apache.org

Logging Services™

Last Published: 2021-03-06 | Version: 2.14.1

APACHE LOG4J™ 2

About

Download

Javadoc

Maven, Ivy, Gradle

Artifacts

Runtime Dependencies

Changelog

FAQ

Performance

Articles and Tutorials

Security

Download Apache Log4j 2

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

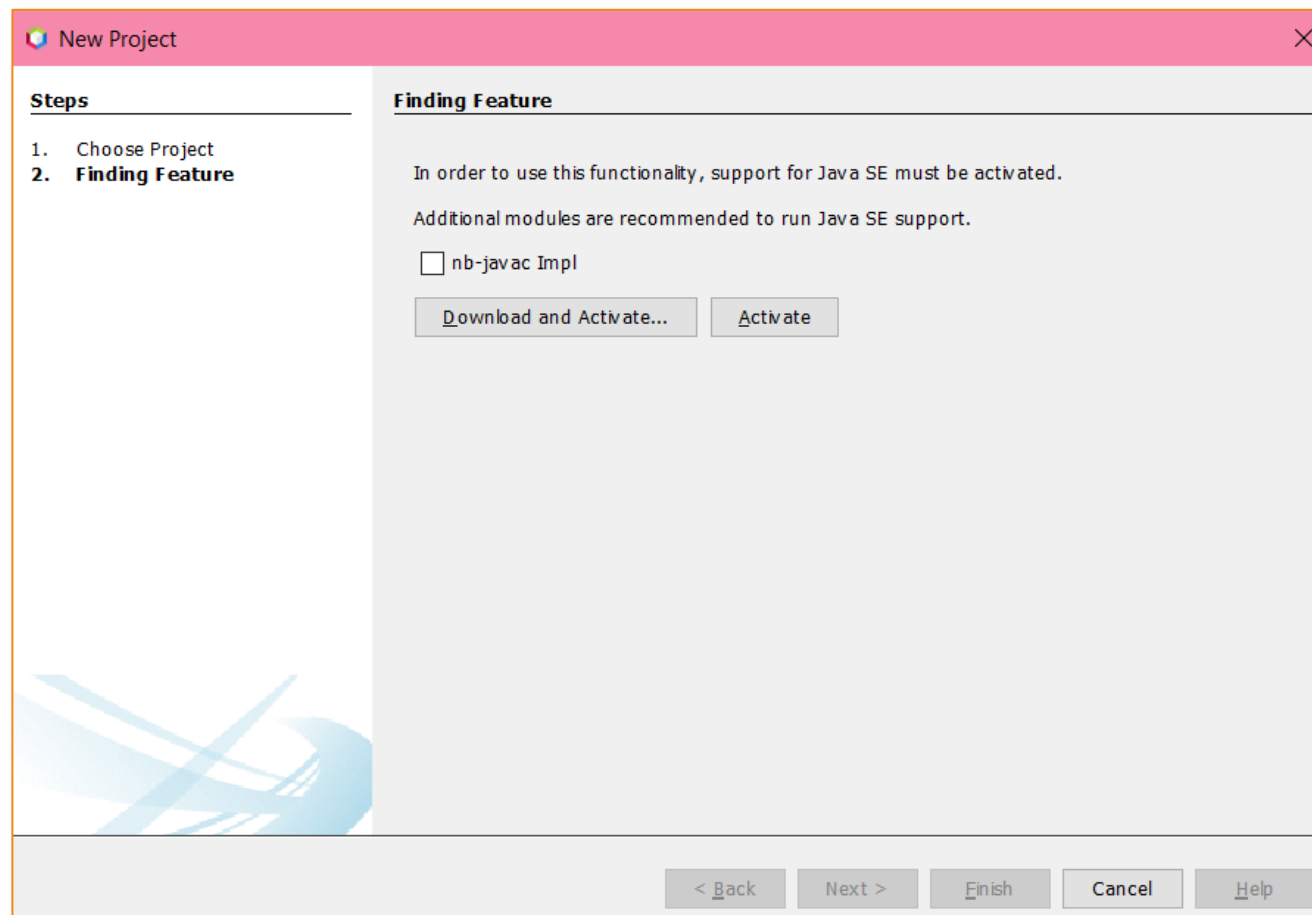
The link in the Mirrors column should display a list of available mirrors with a default checksum and signature are links to the originals on the main distribution server.

Distribution	Mirrors
Apache Log4j 2 binary (tar.gz)	apache-log4j-2.14.1-bin.tar.gz
Apache Log4j 2 binary (zip)	apache-log4j-2.14.1-bin.zip
Apache Log4j 2 source (tar.gz)	apache-log4j-2.14.1-src.tar.gz
Apache Log4j 2 source (zip)	apache-log4j-2.14.1-src.zip

2 – Hướng dẫn sử dụng

Bước 2.1: Tạo project (NetBeans).

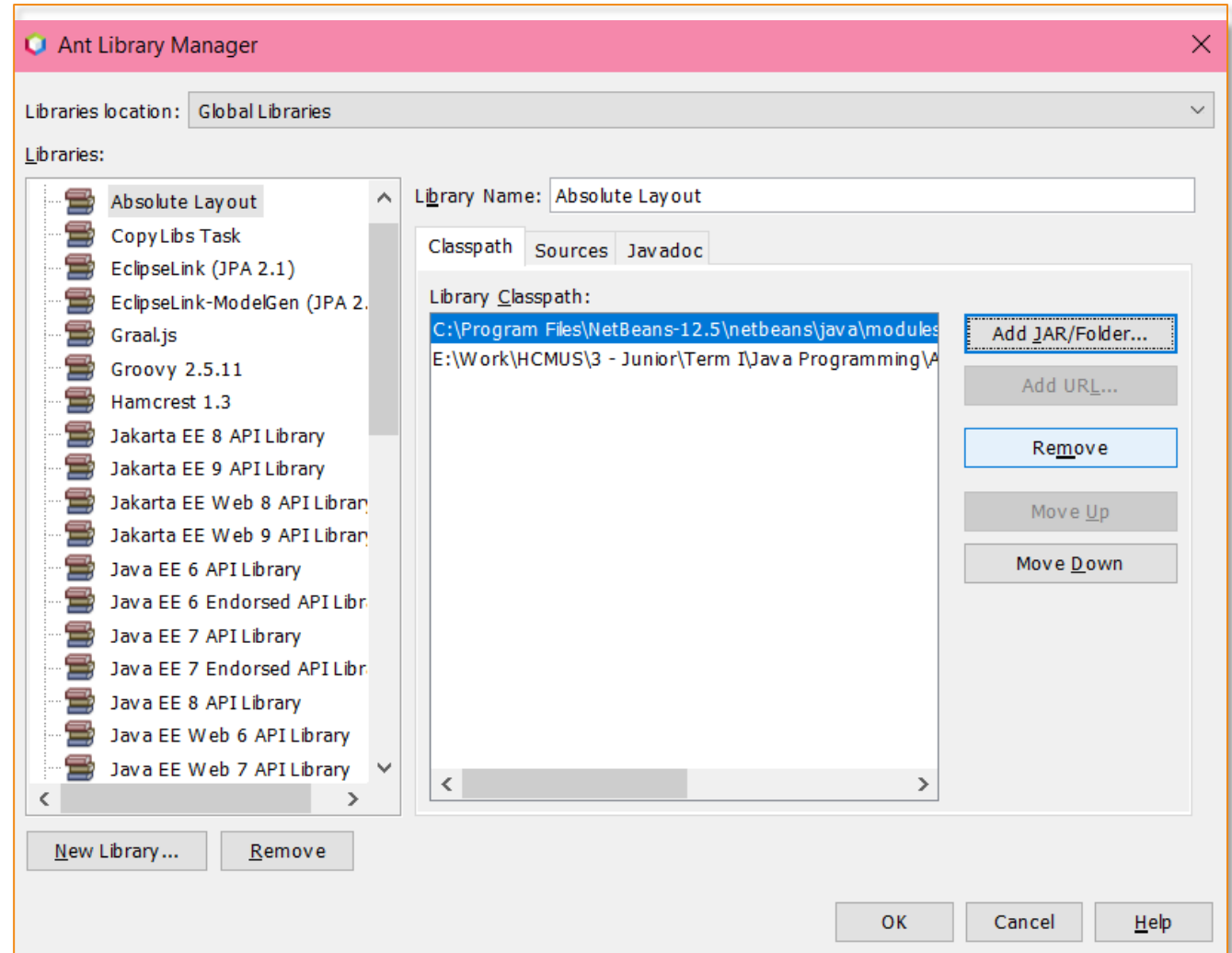
- Chọn *File > New > Project*.
- Chọn *Java with Maven* ở cột *Categories*, và *Java Application* cột *Projects*, và ấn *Next*.
- Đối với lần đầu tạo Project, chờ mục *Finding Feature* tải xong, và chọn *Download and Activate*. Sau đó cài đặt Java SE.
- Nhập thông tin và chọn *Finish*.



2 – Hướng dẫn sử dụng

Bước 3.1: Thêm thư viện (NetBeans).

- Chọn *Tools > Libraries*.
- Chọn *Add JAR/Folder*.
- Chọn thư mục vừa giải nén và ấn *OK* và *Apply*.



2 – Hướng dẫn sử dụng

Bước 2.2: Tạo project (Eclipse).

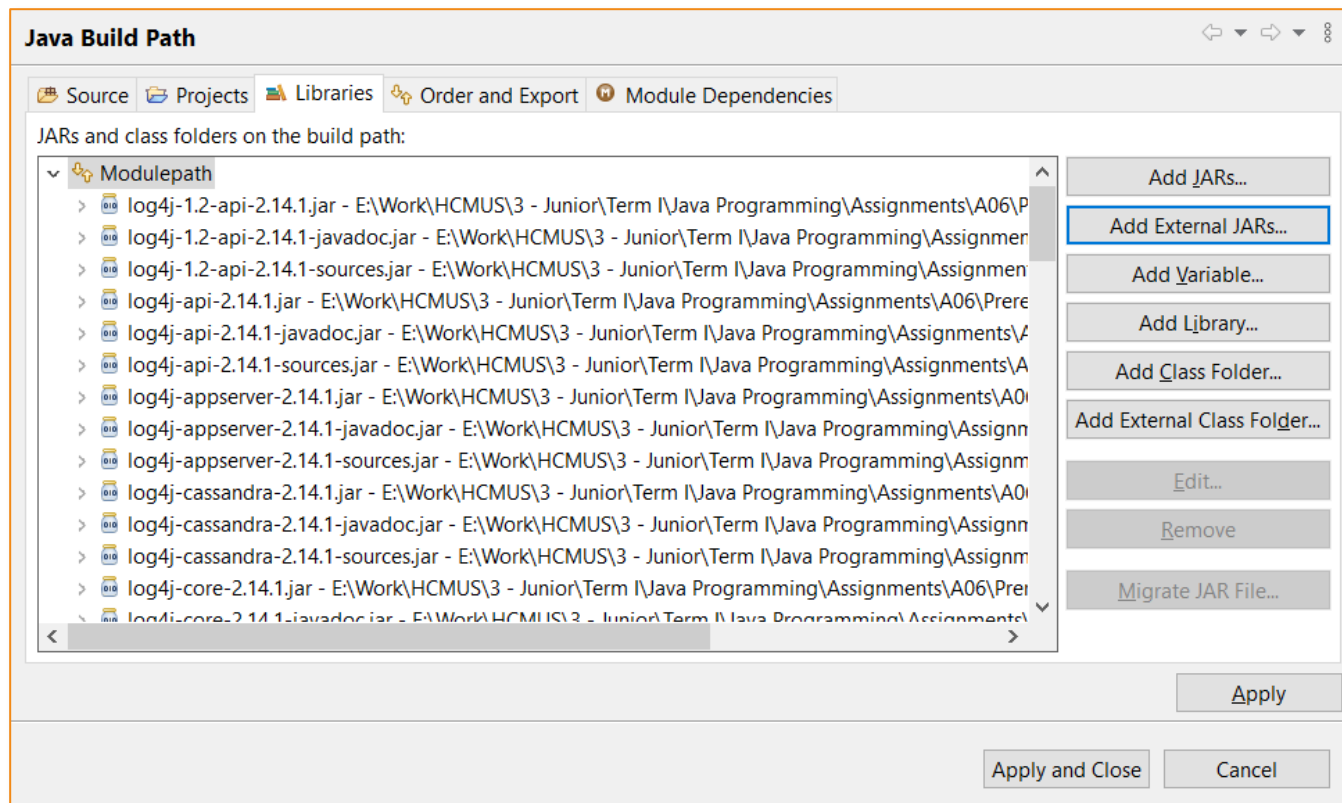
- Chọn *File > New > Java Project*.
- Nhập thông tin và tùy chỉnh theo ý muốn bản thân. Sau đó chọn *Finish*.

The screenshot shows the 'New Java Project' dialog box in Eclipse. The title bar is pink and says 'New Java Project'. The main title is 'Create a Java Project'. Below it, a subtitle says 'Create a Java project in the workspace or in an external location.' There is a folder icon in the top right corner. The dialog is divided into several sections: 'Project name' with a text field containing 'Log4JExample'; 'Use default location' with a checked checkbox and a 'Location' text field containing 'E:\Work\Others\EclipseWS\Log4JExample' and a 'Browse...' button; 'JRE' section with three radio buttons: 'Use an execution environment JRE:' (selected), 'Use a project specific JRE:', and 'Use default JRE 'jdk-11.0.12.7-hotspot' and workspace compiler preferences'. The first option has a dropdown menu showing 'JavaSE-11' and 'jdk-11.0.12.7-hotspot'. There is a 'Configure JREs...' link. 'Project layout' section with two radio buttons: 'Use project folder as root for sources and class files' and 'Create separate folders for sources and class files' (selected). There is a 'Configure default...' link. 'Working sets' section with an unchecked checkbox 'Add project to working sets' and a 'New...' button. Below it is a 'Working sets:' dropdown and a 'Select...' button. 'Module' section with a checked checkbox 'Create module-info.java file'. At the bottom, there is a question mark icon, and four buttons: '< Back', 'Next >', 'Finish' (highlighted with a blue border), and 'Cancel'.

2 – Hướng dẫn sử dụng

Bước 3.2: Thêm thư viện (Eclipse).

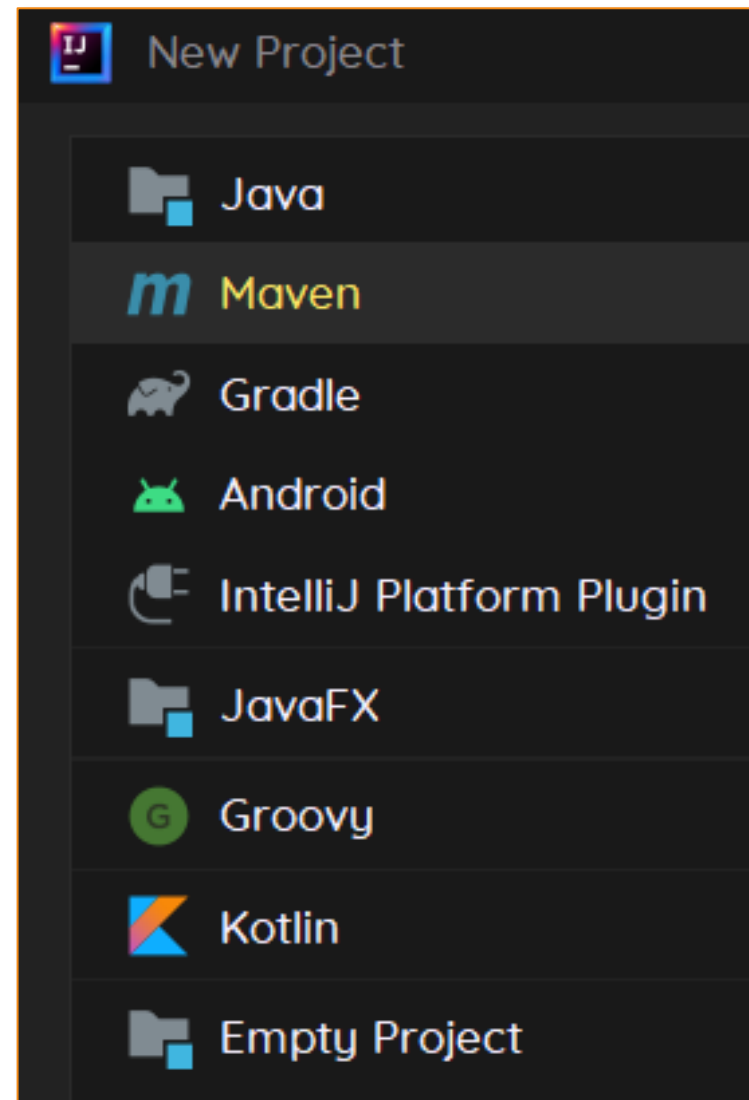
- Chuột phải tên Project, chọn *Build Path > Configure Build Path*.
- Chọn tab *Libraries* và *Modulepath*, sau đó chọn *Add External JARs...*
- Chọn các file JAR vừa giải nén và ấn *Apply*.



2 – Hướng dẫn sử dụng

Bước 2.3: Tạo project (IntelliJ IDEA).

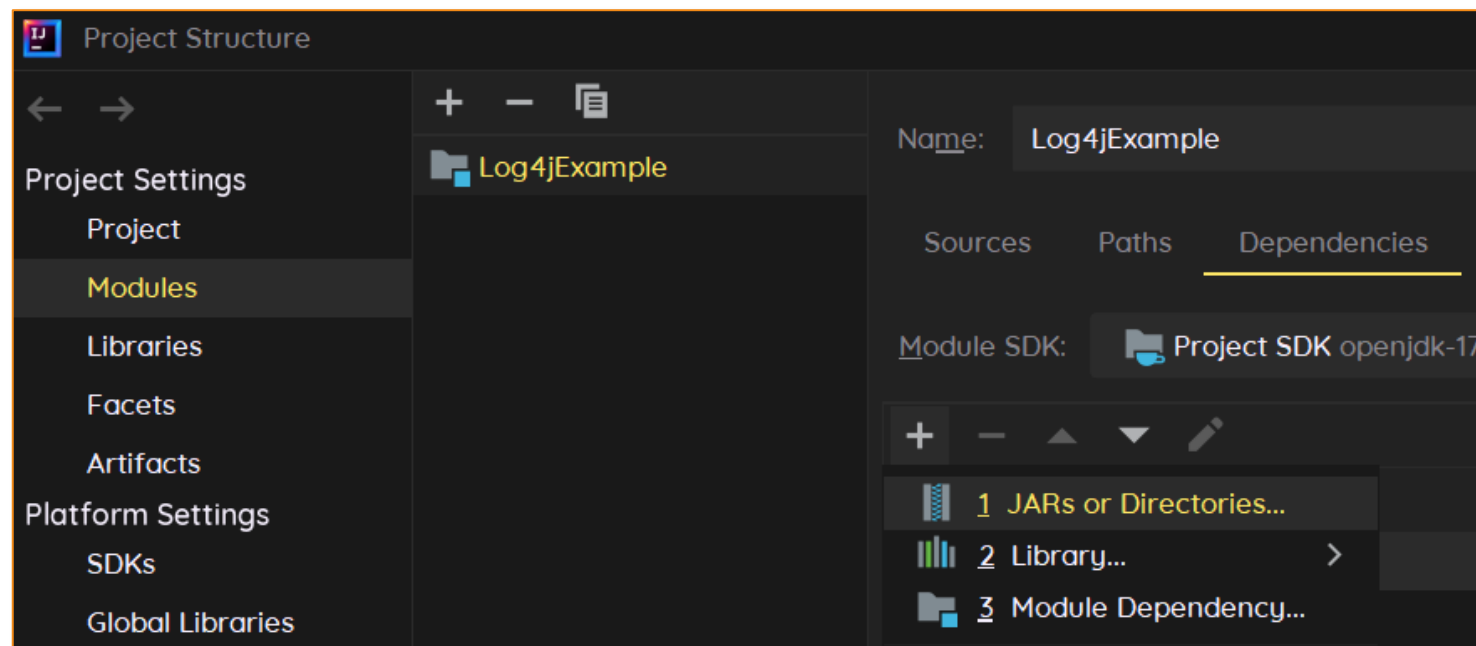
- Khi chưa mở project: Chọn *New Project*.
- Khi đã mở project: Chọn *File > New > Project*.
- Chọn *Maven* và ấn *Next*.
- Đặt tên cho Project và chọn đường dẫn cho Project.
- Ấn *Finish*.



2 – Hướng dẫn sử dụng

Bước 3.3: Thêm thư viện(IntelliJ IDEA).

- Chọn *File > Project Structure*.
- Chọn *Modules* bên trái và chọn tab *Dependencies*.
- Ấn dấu **+** và chọn *JARs or Directories*.
- Chọn thư mục vừa giải nén và ấn *OK* và *Apply*.



2 – Hướng dẫn sử dụng

Bước 4: Cấu hình Log4J bằng log4j2.properties.

Khi không phát hiện file cấu hình nào trong đường dẫn, Log4j sẽ tự động ghi log theo thiết lập mặc định.

Để tự cấu hình cho Log4J theo ý của bản thân, ta cần thêm vào đường dẫn của chương trình 1 file cấu hình để chỉ cho Log4J biết nên ghi log như thế nào.

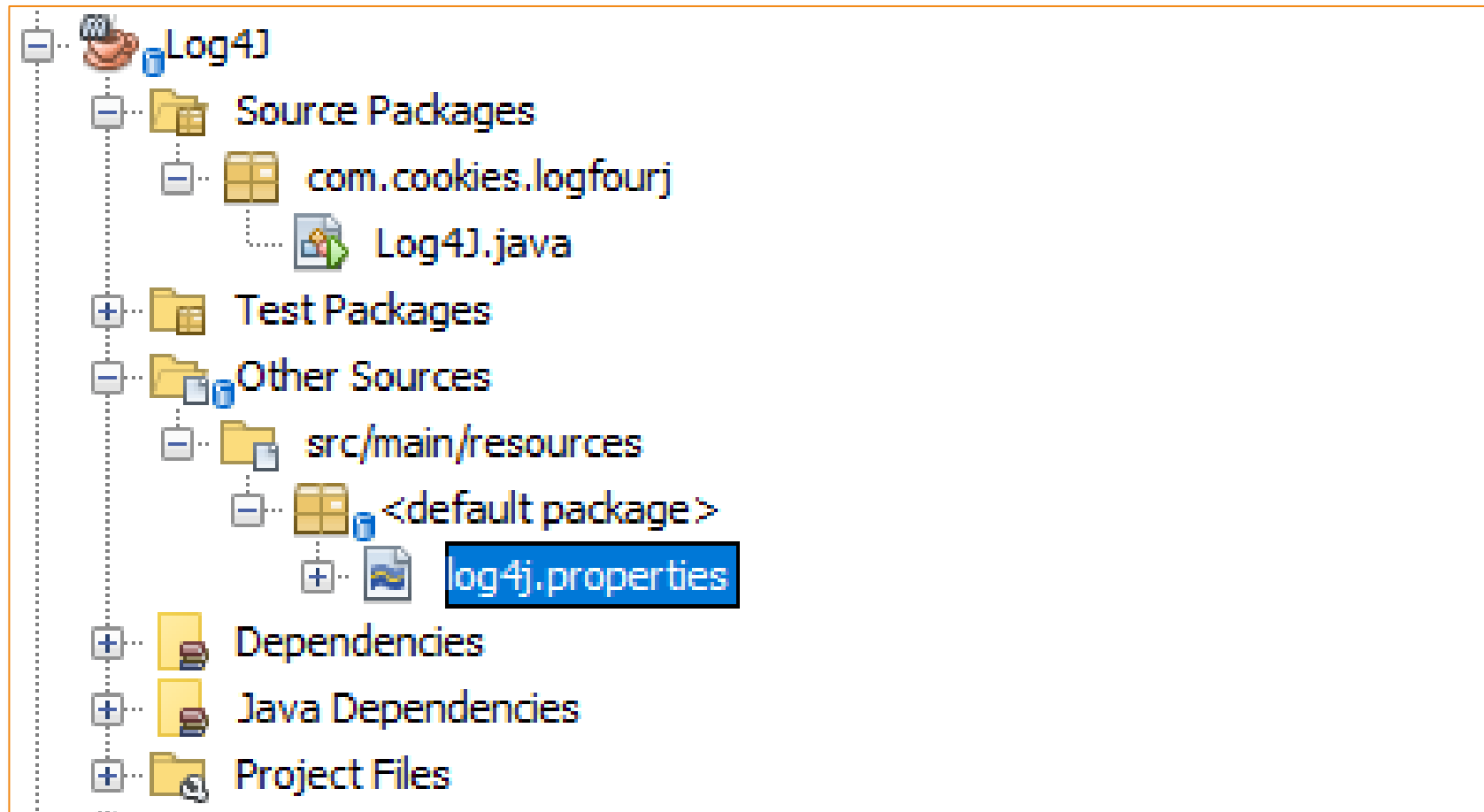
Có 4 định dạng chính của file cấu hình: .properties, .yaml, .json và .xml

Đối với những chương trình đơn giản, hoặc với những người mới làm quen với Java, sử dụng log4j.properties là phương pháp phù hợp và hiệu quả nhất.

2 – Hướng dẫn sử dụng

Bước 4 (tt): Cấu hình Log4J bằng log4j2.properties.

Trước tiên, tạo file log4j2.properties trong đường dẫn src/main/resources.



2 – Hướng dẫn sử dụng

Bước 4 (tt): Cấu hình Log4J bằng log4j2.properties.

Log4J sử dụng Logger và Appender làm 2 thành phần chính để ghi log. Trong đó:

- Logger xử lý việc bắt các sự kiện và chuyển cho Appender tương ứng
- Appender nhận các sự kiện do Logger chuyển đến và ghi lại ở một điểm đến (console, file,...)

Trong file log4j.properties, ta sẽ cấu hình các Logger và Appender để có thể cho ra ghi các sự kiện theo ý muốn bản thân

2 – Hướng dẫn sử dụng

Bước 4 (tt): Cấu hình Log4J bằng log4j2.properties.

Ví dụ sau là một cấu hình giúp ghi các sự kiện ra màn hình console.

```
1  status = error
2  dest = err
3  name = PropertiesConfig
4
5  # appender config
6  appender.console.type = Console
7  appender.console.name = STDOUT
8  appender.console.layout.type = PatternLayout
9  appender.console.layout.pattern = %d %p %C{1.} [%t] %m%n
10 appender.console.filter.threshold.type = ThresholdFilter
11 appender.console.filter.threshold.level = warn
12
13 rootLogger.level = info
14 rootLogger.appenderRef.stdout.ref = STDOUT
```

2 – Hướng dẫn sử dụng

Bước 4 (tt): Cấu hình Log4J bằng log4j2.properties.

Trong đó:

- appender.console.type : loại của Appender, trong ví dụ này là Console
- appender.console.name : tên của Console Appender
- appender.console.layout : thiết kế cách Console Appender sẽ ghi log
- appender.console.filter : cách lọc event của Console Appender
- rootLogger.level : level tối thiểu của của rootLogger
- rootLogger.appenderRef : chọn Appender sẽ được sử dụng cho Logger (rootLogger)

Lưu ý: Bất kì cấu hình nào cũng cần phải có rootLogger. Nếu chỉ có 1 Logger thì đó phải là rootLogger.

2 – Hướng dẫn sử dụng

Bước 4 (tt): Cấu hình Log4J bằng log4j2.properties.

```
5 package com.cookies.logfourj;
6 import org.apache.logging.log4j.LogManager;
7 import org.apache.logging.log4j.Logger;
8
9 /**
10  *
11  * @author ptvin
12  */
13 public class Log4J {
14     private static final Logger logger = LogManager.getLogger(Log4J.class);
15
16     public static void main( String[] args ) {
17         System.out.println( "A log was written!" );
18         logger.trace("This is a TRACE level log!");
19         logger.debug("This is a DEBUG level log!");
20         logger.info("This is an INFO level log!");
21         logger.warn("This is a WARN level log!");
22         logger.error("This is an ERROR level log!");
23         logger.fatal("This is a FATAL level log!");
24     }
25 }
```

Output - Run (Log4J) x

```
-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ logfourj ---
A log was written!
2021-11-21 22:17:54,312 WARN c.c.l.Log4J [main] This is a WARN level log!
2021-11-21 22:17:54,317 ERROR c.c.l.Log4J [main] This is an ERROR level log!
2021-11-21 22:17:54,318 FATAL c.c.l.Log4J [main] This is a FATAL level log!
-----
BUILD SUCCESS
```

2 – Hướng dẫn sử dụng

Bước 5: Log sự kiện ra file.

Thay vì ghi log ra console, chúng ta có thể ghi log ra file riêng bằng cách cấu hình file log4j2.properties. Với file log4j2.properties ở phần trước, ta thêm phần cấu hình cho RollingFileAppender như sau:

```
11
12 appender.rolling.type = RollingFile
13 appender.rolling.name = RollingFile
14 appender.rolling.fileName = ${filename}
15 appender.rolling.filePattern = target/rolling2/test1-%d{MM-dd-yy-HH-mm-ss}-%i.log
16 appender.rolling.layout.type = PatternLayout
17 appender.rolling.layout.pattern = %d %p %C{1.} [%t] %m%n
18 appender.rolling.policies.type = Policies
19 appender.rolling.policies.time.type = TimeBasedTriggeringPolicy
20 appender.rolling.policies.time.interval = 2
21 appender.rolling.policies.time.modulate = true
22 appender.rolling.policies.size.type = SizeBasedTriggeringPolicy
23 appender.rolling.policies.size.size=100MB
24 appender.rolling.strategy.type = DefaultRolloverStrategy
25 appender.rolling.strategy.max = 5
26
27 logger.rolling.name = com.cookies.logfourj
28 logger.rolling.level = debug
29 logger.rolling.additivity = false
30 logger.rolling.appenderRef.rolling.ref = RollingFile
31
```

2 – Hướng dẫn sử dụng

Bước 5 (tt): Log sự kiện ra file.

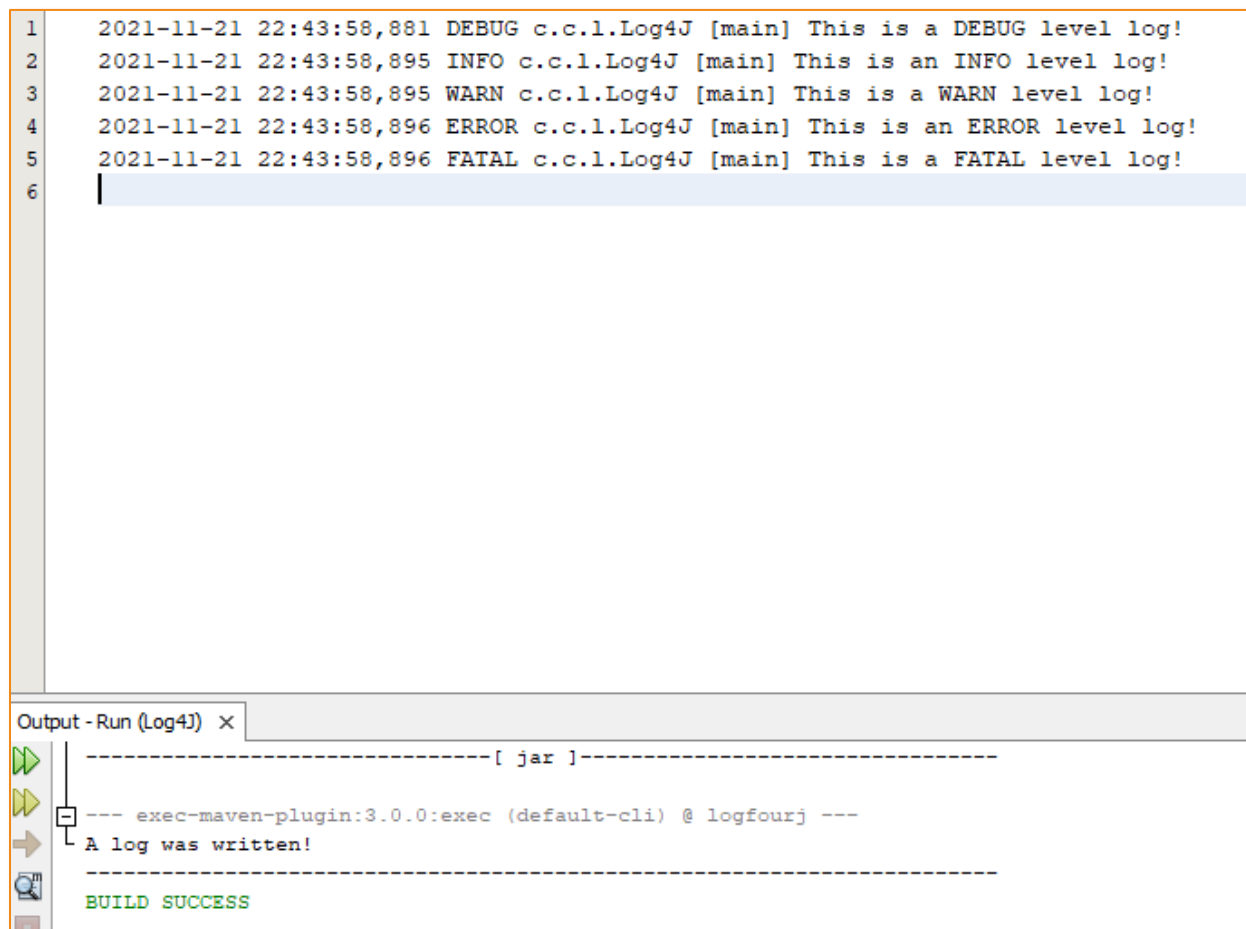
Trong đó:

- `appender.rolling.type` : loại của Appender, trong ví dụ này là `RollingFile`
- `appender.rolling.name` : tên của `RollingFile Appender`
- `appender.rolling.fileName` : tên của file mà Appender sẽ ghi ra
- `appender.rolling.filePattern` : định dạng tên file khi chia file log
- `appender.rolling.layout` : thiết kế cách `RollingFile Appender` sẽ ghi log
- `appender.rolling.policies` : chính sách chia file log
- `appender.rolling.strategy`: chiến lược chia file log
- `logger.rolling.name` : tên package mà logger sẽ hoạt động
- `logger.rolling.level` : level tối thiểu của logger
- `logger.rolling.additivity` : tính lặp lại của logger, dùng để tránh việc logger và `rootLogger` lặp lại cùng 1 nội dung
- `logger.rolling.appenderRef` : chọn Appender sẽ được sử dụng cho Logger

2 – Hướng dẫn sử dụng

Bước 5 (tt): Log sự kiện ra file.

File log và màn hình sau khi chạy chương trình.



The screenshot displays an IDE interface. The main editor window shows a log file with five entries, each on a new line. The entries are: 1. 2021-11-21 22:43:58,881 DEBUG c.c.l.Log4J [main] This is a DEBUG level log! 2. 2021-11-21 22:43:58,895 INFO c.c.l.Log4J [main] This is an INFO level log! 3. 2021-11-21 22:43:58,895 WARN c.c.l.Log4J [main] This is a WARN level log! 4. 2021-11-21 22:43:58,896 ERROR c.c.l.Log4J [main] This is an ERROR level log! 5. 2021-11-21 22:43:58,896 FATAL c.c.l.Log4J [main] This is a FATAL level log! The sixth line is empty. Below the editor, the 'Output - Run (Log4J)' window is open, showing the execution of the 'jar' command. The output text is: --- exec-maven-plugin:3.0.0:exec (default-cli) @ logfourj --- A log was written! BUILD SUCCESS

```
1 2021-11-21 22:43:58,881 DEBUG c.c.l.Log4J [main] This is a DEBUG level log!
2 2021-11-21 22:43:58,895 INFO c.c.l.Log4J [main] This is an INFO level log!
3 2021-11-21 22:43:58,895 WARN c.c.l.Log4J [main] This is a WARN level log!
4 2021-11-21 22:43:58,896 ERROR c.c.l.Log4J [main] This is an ERROR level log!
5 2021-11-21 22:43:58,896 FATAL c.c.l.Log4J [main] This is a FATAL level log!
6
```

Output - Run (Log4J) x

```
-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ logfourj ---
A log was written!
BUILD SUCCESS
```

2 – Hướng dẫn sử dụng

Bước 6: Chia file log theo thời gian và dung lượng.

Như đã giới thiệu, trong phần cấu hình RollingFile Appender, chúng ta có thể dùng `appender.rolling.policies` và `appender.rolling.strategy` để chia file log theo tùy ý.

Nguyên lý chia file:

- File log được xác định trong phần `appender.rolling.fileName` sẽ liên tục ghi cho đến khi bị buộc tách file (được quy định trong phần `appender.rolling.policies`)
- Phần log trong file trên sẽ được lưu vào một file khác theo định dạng quy định trong phần `appender.rolling.filePattern`
- Sau khi đã lưu trữ xong, file log được xác định trong phần `appender.rolling.fileName` sẽ bắt đầu ghi lại từ đầu, và tương tự lặp lại cho các lần ngắt tiếp theo

2 – Hướng dẫn sử dụng

Bước 6 (tt): Chia file log theo thời gian và dung lượng.

Ví dụ ở phần trước đã giới thiệu 2 kiểu chia file cơ bản, đó là theo thời gian và theo dung lượng.

```
14 appender.rolling.fileName = ${filename}
15 appender.rolling.filePattern = target/rolling2/test1-%d{MM-dd-yy-HH-mm-ss}-%i.log
16 appender.rolling.layout.type = PatternLayout
17 appender.rolling.layout.pattern = %d %p %C{1.} [%t] %m%n
18 appender.rolling.policies.type = Policies
19 appender.rolling.policies.time.type = TimeBasedTriggeringPolicy
20 appender.rolling.policies.time.interval = 2
21 appender.rolling.policies.time.modulate = true
22 appender.rolling.policies.size.type = SizeBasedTriggeringPolicy
23 appender.rolling.policies.size.size=100MB
24 appender.rolling.strategy.type = DefaultRolloverStrategy
25 appender.rolling.strategy.max = 5
```

2 – Hướng dẫn sử dụng

Bước 6 (tt): Chia file log theo thời gian và dung lượng.

Trong đó:

- `appender.rolling.policies.type` : chọn chính sách dùng để chia file
- `appender.rolling.policies.time.type` : chọn chính sách để chia file theo thời gian
- `appender.rolling.policies.time.interval` : thời lượng cho 1 file, đơn vị là thành phần nhỏ nhất được qui định trong thành phần thời gian của `appender.rolling.filePattern`
- `appender.rolling.policies.time.modulate` : xác định có chia file ngay vào thời điểm ranh giới giữa các khoảng thời lượng với nhau hay không
- `appender.rolling.policies.size.type` : chọn chính sách để chia file theo dung lượng
- `appender.rolling.policies.size.size` : dung lượng tối đa của 1 file log
- `appender.rolling.strategy.type` : chọn chiến lược chia file
- `appender.rolling.strategy.max` : số file log tối đa được lưu trữ

2 – Hướng dẫn sử dụng

Bước 6 (tt): Chia file log theo thời gian và dung lượng.

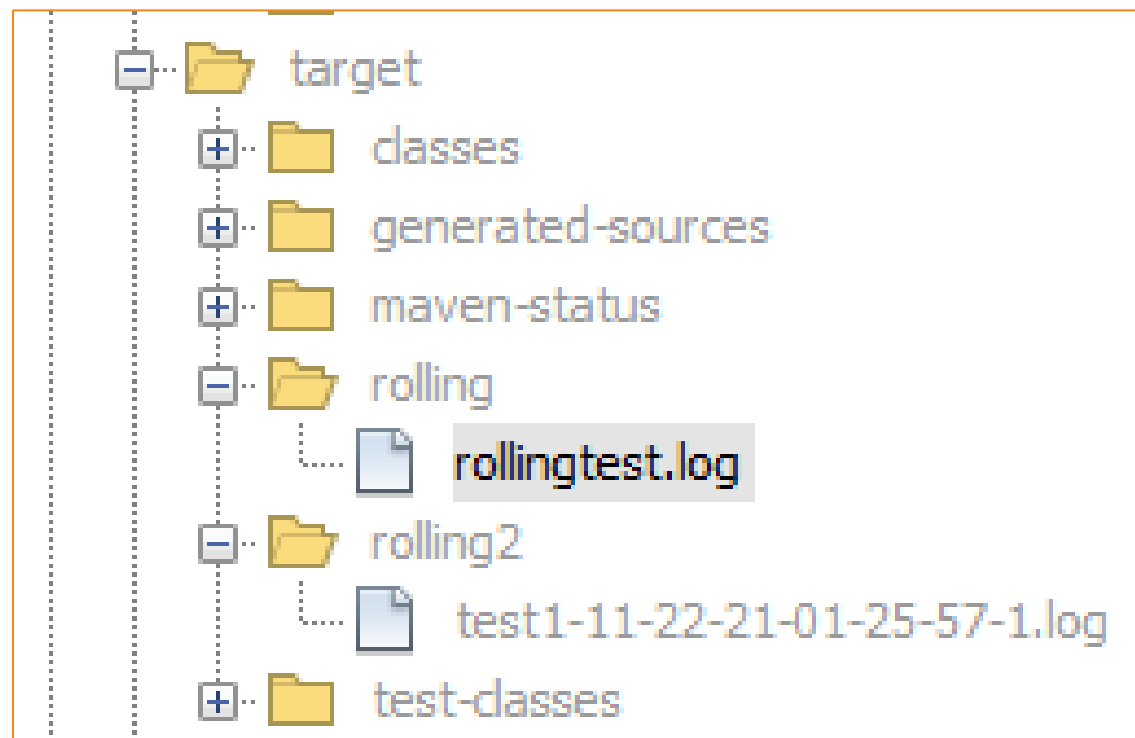
Để kiểm chứng, chúng ta chạy chương trình ví dụ sau.

```
public class Log4J {  
    private static final Logger logger = LogManager.getLogger(Log4J.class);  
  
    public static void main( String[] args ) {  
        System.out.println( "A log was written!" );  
        logger.trace("This is a TRACE level log!");  
        logger.debug("This is a DEBUG level log!");  
        logger.info("This is an INFO level log!");  
        logger.warn("This is a WARN level log!");  
        try {  
            Thread.sleep(5000);  
        } catch (InterruptedException ex) {  
            System.out.println("InterruptedException");  
        }  
        logger.error("This is an ERROR level log!");  
        logger.fatal("This is a FATAL level log!");  
    }  
}
```


2 – Hướng dẫn sử dụng

Bước 6 (tt): Chia file log theo thời gian và dung lượng.

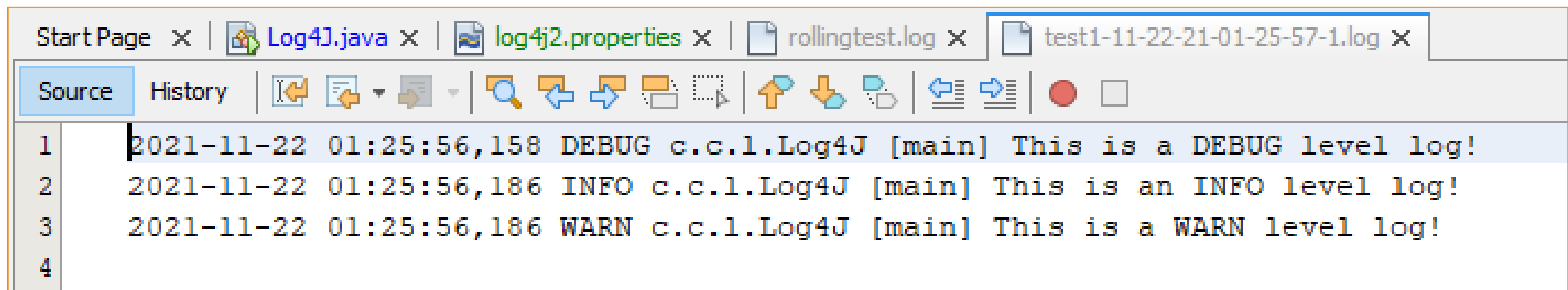
Ta sẽ được các file sau:



2 – Hướng dẫn sử dụng

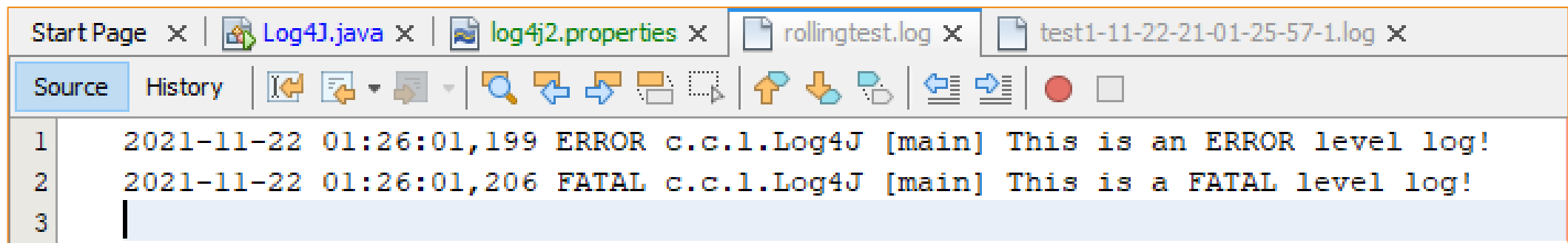
Bước 6 (tt): Chia file log theo thời gian và dung lượng.

Chi tiết các file như sau:



The screenshot shows an IDE window with several tabs: Start Page, Log4J.java, log4j2.properties, rollingtest.log, and test1-11-22-21-01-25-57-1.log. The active tab is test1-11-22-21-01-25-57-1.log, which displays the following log entries:

```
1 2021-11-22 01:25:56,158 DEBUG c.c.1.Log4J [main] This is a DEBUG level log!
2 2021-11-22 01:25:56,186 INFO c.c.1.Log4J [main] This is an INFO level log!
3 2021-11-22 01:25:56,186 WARN c.c.1.Log4J [main] This is a WARN level log!
4
```



The screenshot shows the same IDE window with the same tabs. The active tab is still test1-11-22-21-01-25-57-1.log, which displays the following log entries:

```
1 2021-11-22 01:26:01,199 ERROR c.c.1.Log4J [main] This is an ERROR level log!
2 2021-11-22 01:26:01,206 FATAL c.c.1.Log4J [main] This is a FATAL level log!
3
```

2 – Hướng dẫn sử dụng

Bước 7: Quy định các log level.

Các log level có một giá trị nguyên nhất định để qua đó có thể so sánh các log level với nhau.

Các log level mặc định có các giá trị sau:

Standard Level	intLevel
OFF	0
FATAL	100
ERROR	200
WARN	300
INFO	400
DEBUG	500
TRACE	600
ALL	Integer.MAX_VALUE

2 – Hướng dẫn sử dụng

Bước 7 (tt): Quy định các log level.

Ngoài ra, ta có thể tự tạo log level của bản thân trực tiếp qua chương trình

Ở ví dụ dưới đây, ta tạo một level mới có tên là ALERT, với giá trị là 250.

```
public class Log4J {  
    static final Level ALERT = Level.forName("ALERT", 250);  
    private static final Logger logger = LogManager.getLogger(Log4J.class);  
  
    public static void main( String[] args ) {  
        System.out.println( "A log was written!" );  
        logger.trace("This is a TRACE level log!");  
        logger.debug("This is a DEBUG level log!");  
        logger.info("This is an INFO level log!");  
        logger.warn("This is a WARN level log!");  
        logger.log(ALERT, "This is an ALERT level log!");  
        logger.error("This is an ERROR level log!");  
        logger.fatal("This is a FATAL level log!");  
    }  
}
```

2 – Hướng dẫn sử dụng

Bước 7 (tt): Quy định các log level.

Sau khi chạy chương trình với file log4j2.properties với cấu hình ghi file tương tự như phần trước, ta có file log như sau:

```
2021-11-22 02:17:21,771 DEBUG c.c.l.Log4J [main] This is a DEBUG level log!
2021-11-22 02:17:21,775 INFO c.c.l.Log4J [main] This is an INFO level log!
2021-11-22 02:17:21,775 WARN c.c.l.Log4J [main] This is a WARN level log!
2021-11-22 02:17:21,775 ALERT c.c.l.Log4J [main] This is an ALERT level log!
2021-11-22 02:17:21,776 ERROR c.c.l.Log4J [main] This is an ERROR level log!
2021-11-22 02:17:21,776 FATAL c.c.l.Log4J [main] This is a FATAL level log!
```

Tài liệu tham khảo

<https://en.wikipedia.org/wiki/Log4j>

<https://www.jetbrains.com/help/idea/library.html#define-library>

<https://logging.apache.org/log4j/2.x/manual/configuration.html>

<https://logging.apache.org/log4j/2.x/manual/appenders.html>

<https://logging.apache.org/log4j/2.x/manual/customloglevels.html>

<https://howtodoinjava.com/log4j2/log4j2-properties-example/>

<https://www.ibm.com/support/pages/setting-log4j-log-file-rollover-frequency>