✅ Log4j – Java Logging Framework

🔹 What is Log4j?

Log4j (Logging for Java) is a **Java-based logging framework** provided by Apache to help developers record log messages from applications.

🔧 Logging means:

Writing messages (e.g., errors, flow, debug info) to the **console**, **file**, or **external systems**.

🔸 Why Do We Use Log4j?

Need	Reason
Troubleshooting	Helps trace issues in code during dev or production
Monitoring	Tracks what's happening inside your application
Debugging	Easily find where the bug occurs by reading logs
No System.out.println()	Better control, formatting, and log level support

---

🧱 How Log4j Works

1. **Logger** – Main class used to generate logs.
2. **Appender** – Where to send logs (Console, File, DB).
3. **Layout** – Format for logs (pattern, JSON, etc.).
4. **Configuration file** – XML, JSON, or properties file to define log rules.

📁 Example log4j.properties file:

properties

CopyEdit

log4j.rootLogger=DEBUG, console, file

```
# Console Appender
log4j.appender.console=org.apache.log4j.ConsoleAppender
log4j.appender.console.layout=org.apache.log4j.PatternLayout
log4j.appender.console.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss} %-5p %c - %m%n

# File Appender
log4j.appender.file=org.apache.log4j.FileAppender
log4j.appender.file.File=logs/app.log
log4j.appender.file.Append=true
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%d %-5p %c - %m%n
```

📌 **Example Java Code Using Log4j**

```
java
CopyEdit
import org.apache.log4j.Logger;

public class StudentService {
   static Logger logger = Logger.getLogger(StudentService.class);

   public void saveStudent(Student student) {
      logger.info("Saving student: " + student.getName());
      try {
         // save logic here
      } catch (Exception e) {
         logger.error("Failed to save student", e);
      }
```

```
    }
}
```

🔑 Log Levels in Log4j

Level	Use Case
TRACE	Most detailed (method entry, exit, etc.)
DEBUG	Debugging info (variables, flow)
INFO	General info (e.g., "User saved")
WARN	Warnings (e.g., "Low disk space")
ERROR	Error conditions (e.g., "File not found")
FATAL	Critical errors causing shutdown

You can configure what level of logs you want to see in production vs development.

🧠 Real-Life Example

Imagine you're building a Spring Boot Student API:

- When a student is saved, you log:
  "Student saved successfully: Ram"
- When the DB fails, you log:
  "ERROR: Database connection failed"

This helps when you check logs during production bugs or performance issues.

✅ Benefits of Log4j

Benefit	Description
Flexible	Supports console, file, DB, email, remote
Configurable	Change log levels without touching code
Lightweight	Very fast and efficient

Benefit	Description
Industry standard	Widely used and stable
Layouts	Custom format support (patterns, JSON, etc.)

❌ Drawbacks

Drawback	Note
Verbose setup	Requires config file
Security concern (Log4Shell)	Older versions had a serious vulnerability (fixed in Log4j2)
Not part of Java core	Needs external dependency

🔐 Log4Shell (Security Warning)

In Dec 2021, a major vulnerability was found in **Log4j 1.x and early 2.x**.

✅ **Always use the latest Log4j2 version**
or consider safer alternatives like **SLF4J + Logback**.

💉 Interview Questions on Log4j

1. What is Log4j and why is it used?
2. How is Log4j better than System.out.println()?
3. What are log levels in Log4j?
4. What is an Appender in Log4j?
5. How do you configure Log4j in a project?
6. What is Log4Shell?
7. What's the difference between Log4j and Logback?

🧱 Maven Dependency for Log4j2

xml

CopyEdit

```
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-core</artifactId>
  <version>2.20.0</version>
</dependency>
```

Also add:

xml

CopyEdit

```
<dependency>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-api</artifactId>
  <version>2.20.0</version>
</dependency>
```

🆚 **Bonus: Log4j vs SLF4J vs Logback**

Feature	Log4j	Logback	SLF4J
Logging Framework	✅	✅	❌ (just an API)
Modern & Active	❌ (use Log4j2)	✅	✅
XML/JSON Config	✅	✅	N/A
Used with Spring Boot	❌	✅ (Logback is default)	✅

✅ Spring Boot uses **SLF4J + Logback** by default.
But you can replace it with Log4j2 if needed.