Commons Logging is a simple and general-purpose logging interface that provides an abstractio
private static final Log log = LogFactory.getLog(LoggingExample.class);
over different logging frameworks.
It allows developers to write logging code that is independent of the underlying logging
public static void main(String[] args) {
implementation, making it easier to switch between different logging libraries without changing th
log.debug('Debug message');
source code.
log.info('Info message');
log.warn('Warning message');
Key Features:
log.error('Error message');
- **Abstraction Layer:** Commons Logging provides a common interface for various logging
log.fatal('Fatal error message');
frameworks like Log4j, java.util.logging, etc.
}
- **Simple to Use:** It allows for quick integration and usage across applications.
}
- **Flexible Configuration:** You can configure the underlying logging framework at runtime witho
```
recompiling your application.
### Conclusion:
### How to Use Commons Logging:
Commons Logging is an essential tool for developers who want a flexible logging solution that ca
1. **Add Dependency:** Include the Commons Logging dependency in your project.
adapt to various logging frameworks. By following the guidelines and examples provided, you ca
2. **Create a Logger Instance:** Use LogFactory to create a logger instance.
effectively integrate logging into your Java applications.
3. **Log Messages:** Use different logging levels such as debug, info, warn, error, and fatal to lo
messages.
### Example Code:
```java
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
public class LoggingExample {