

日志文件压缩 SDK

日志文件压缩SDK用于压缩文本文件（txt、json、log、xml、sql、csv...），使它们变得更小以便于保存在存储设备中或在网络中传输。

SDK 基于 Java，支持Linux 和 Windows。

将 SDK Jar 导入项目

将给定的 JAR 文件导入您的 Java 项目，按照您的 IDE 的说明进行操作。

这是在 Maven 中导入的示例

- 运行此命令以安装 SDK

```
mvn install:install-file -Dfile=<path-to-sdk-jar> -DgroupId=org.example -
DartifactId=lfcr -Dversion=1.0-SNAPSHOT -Dpackaging=jar
```

- 在项目中导入SDK

```
<dependency>
  <groupId>org.example</groupId>
  <artifactId>lfcr</artifactId>
  <version>1.0-SNAPSHOT</version>
</dependency>
```

将 SDK 导入项目后，您可以使用以下命令将其导入 java 文件：

```
import org.logfilecompression.LogFileCompression;
```

如何使用

1. 初始化 SDK.

- 使用 API 密钥 (在线模式)
 - `LogFileCompression lfc = new LogFileCompression("API_KEY_HERE");`
- 使用密钥文件 (离线模式)
 - 如果没有找到密钥文件，将生成一个新的密钥文件，其必须上传到 `http://120.79.36.189:8000/request/license/`. 请求获得批准后，您可以下载批准的密钥文件并使用它，如下所示。
 - 从默认密钥文件位置
 - `LogFileCompression lfc = new LogFileCompression(true);`
 - 从不同的密钥文件位置

- `LogFileCompression lfc = new LogFileCompression(true, "23c3b3a6-8271-4b26-820b-0282788fa87b.key");`

输入

SDK 输入参数是:

压缩: `org.logfilecompression.LogFileCompression.compress`

- 文件输入路径
- 文件输出路径
- 压缩级别（根据内存使用和压缩等级有4个压缩级别）。

解压: `org.logfilecompression.LogFileCompression.decompress`

- 文件输入路径
- 文件输出路径

输出

日志文件的输出路径是输入时可以控制的一个变量

使用示例

```
import static org.junit.jupiter.api.Assertions.assertTrue;

import java.io.File;
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.security.InvalidAlgorithmParameterException;
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import java.util.Objects;
import javax.crypto.NoSuchPaddingException;
import org.junit.jupiter.api.Test;
import org.logfilecompression.LogFileCompression;

public class TestFileCompress {

    @Test
    public void testFileEncryptionDecryption()
        throws IOException, InterruptedException, NoSuchPaddingException,
        NoSuchAlgorithmException, InvalidKeyException, URISyntaxException,
        InvalidAlgorithmParameterException {
        String compressMode = "4";
        File outputFile = File.createTempFile("compressApi", "");
        URI tempFilePath =
            Objects.requireNonNull(TestFileCompress.class.getResource("/test.txt"))
```

```
        .toURI());
    String inputFilePath = Paths.get(tempFilePath).toString();
    String outputFilePath = Objects.requireNonNull(outputFile.getAbsolutePath());
    LogFileCompression lfc = new LogFileCompression("API_KEY");
    // LogFileCompression lfc = new LogFileCompression(true);
    // LogFileCompression lfc = new LogFileCompression(true, "path to license key
file");
    lfc.compress(inputFilePath, outputFilePath, compressMode);
    assertTrue(outputFile.exists());
    System.out.println("Original file size: " +
Files.size(Paths.get(inputFilePath)));
    System.out.println("Compressed file size: " +
Files.size(outputFile.toPath()));
    lfc.decompress(outputFilePath, "test.lfc");
    outputFile.delete();
    }
}
```