

图片压缩 SDK

图片压缩 SDK 适用于以不同方式压缩图片，您可以选择格式（JPG、PNG 或 WebP），同时您可以选择压缩级别（图片质量），JPG 和 WebP 有四种压缩级别，PNG 有三种压缩级别，通过对选择不同的级别来减少对图像质量的影响。

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=compression-api -Dversion=1.0-SNAPSHOT -Dpackaging=jar
```

- 在项目中导入 SDK

```
<dependency>
  <groupId>org.example</groupId>
  <artifactId>compression-api</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.compressapi.ImageCompress;
```

输入

SDK 的输入参数是：

- 文件输入路径
- 文件输出路径
- 压缩级别（JPG 和 WebP 有 4 个压缩级别，PNG 有 3 个压缩级别）
- API 密钥/许可证文件

如何使用

1. 初始化 SDK.

- 使用 API 密钥 (在线模式)
 - `ImageCompress ic = new ImageCompress("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：
 - 从默认密钥位置
 - `ImageCompress ic = new ImageCompress(true);`
 - 从不同的密钥位置
 - `ImageCompress ic = new ImageCompress(true, "23c3b3a6-8271-4b26-820b-0282788fa87b.key");`

2. `ic.compress(inputFilePath, outputFilePath, compressionLevel)`

输出

图片文件的输出路径是输入时可以控制的一个变量

使用示例

带有自定义许可证密钥的 JPEG 示例

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

import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.Objects;
import org.compressapi.ImageCompress;
import org.junit.jupiter.api.Test;

public class TestImageCompress {
    @Test
    public void testJpegImageCreation() throws IOException, InterruptedException {
        File outputFile = File.createTempFile("compressApi", ".jpg");
        String inputFilePath =
            new
            File(Objects.requireNonNull(TestImageCompress.class.getResource("/test.jpg")).getFile())
                .getPath();
        String outputFilePath = Objects.requireNonNull(outputFile.getAbsolutePath());
        ImageCompress ic = new ImageCompress(true, "23c3b3a6-8271-4b26-820b-0282788fa87b.key");
        ic.compress(
            inputFilePath, outputFilePath, "L4");
        assertTrue(outputFile.exists());
        System.out.println("Original file size" +
            Files.size(Paths.get(inputFilePath)));
        System.out.println("Compressed file size" + Files.size(outputFile.toPath()));
    }
}
```

```
}

```

具有默认许可证密钥位置的 PNG 示例

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

import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.Objects;
import org.compressapi.ImageCompress;
import org.junit.jupiter.api.Test;

public class TestImageCompress {
    @Test
    public void testPngImageCreation() throws IOException, InterruptedException {
        File outputFile = File.createTempFile("compressApi", ".png");
        String inputFilePath =
            new
File(Objects.requireNonNull(TestImageCompress.class.getResource("/test.png")).getFile())
                .getPath();
        String outputFilePath = Objects.requireNonNull(outputFile.getAbsolutePath());
        ImageCompress ic = new ImageCompress(true);
        ic.compress(inputFilePath, outputFilePath, "L2");
        assertTrue(outputFile.exists());
        System.out.println("Original file size" +
Files.size(Paths.get(inputFilePath)));
        System.out.println("Compressed file size" + Files.size(outputFile.toPath()));
    }
}

```

带有 API 密钥的 WebP 示例

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

import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.Objects;
import org.compressapi.ImageCompress;
import org.junit.jupiter.api.Test;

public class TestImageCompress {
    @Test
    public void testWebpImageCreation() throws IOException, InterruptedException {

```

```
File outputFile = new File("temp.webp");
String inputFilePath =
    new
File(Objects.requireNonNull(TestImageCompress.class.getResource("/test.webp")).get
File())
    .getPath();

String outputFilePath = Objects.requireNonNull(outputFile.getAbsolutePath());
ImageCompress ic = new ImageCompress("API_KEY");
ic.compress(inputFilePath, outputFilePath, "L1");
assertTrue(outputFile.exists());
System.out.println("Original file size" +
Files.size(Paths.get(inputFilePath)));
System.out.println("Compressed file size" + Files.size(outputFile.toPath()));
}
}
```