

🔑 1f399a2299 ▾

⋮

one-java-agent / one-java-agent-plugin / src / main / java / com / alibaba / oneagent / utils / IOUtils.java / <> Jump to ▾



hengyunabc refactor spy module

🕒 History

👤 1 contributor

162 lines (138 sloc) | 3.91 KB

⋮

```

1  package com.alibaba.oneagent.utils;
2
3  import java.io.BufferedInputStream;
4  import java.io.BufferedOutputStream;
5  import java.io.ByteArrayOutputStream;
6  import java.io.Closeable;
7  import java.io.File;
8  import java.io.FileOutputStream;
9  import java.io.IOException;
10 import java.io.InputStream;
11 import java.io.OutputStream;
12 import java.io.Reader;
13 import java.io.Writer;
14 import java.net.URLClassLoader;
15 import java.util.Enumeration;
16 import java.util.zip.ZipEntry;
17 import java.util.zip.ZipFile;
18
19 /**
20  *
21  * @author hengyunabc 2018-11-06
22  *
23  */
24 public class IOUtils {
25
26     public static String toString(InputStream inputStream) throws IOException {
27         ByteArrayOutputStream result = new ByteArrayOutputStream();
28         byte[] buffer = new byte[1024];

```

```

29         int length;
30         while ((length = inputStream.read(buffer)) != -1) {
31             result.write(buffer, 0, length);
32         }
33         return result.toString("UTF-8");
34     }
35
36     public static void copy(InputStream in, OutputStream out) throws IOException {
37         byte[] buffer = new byte[1024];
38         int len;
39         while ((len = in.read(buffer)) != -1) {
40             out.write(buffer, 0, len);
41         }
42     }
43
44     /**
45      * @return a byte[] containing the information contained in the specified
46      *         InputStream.
47      * @throws java.io.IOException
48      */
49     public static byte[] getBytes(InputStream input) throws IOException {
50         ByteArrayOutputStream result = new ByteArrayOutputStream();
51         copy(input, result);
52         result.close();
53         return result.toByteArray();
54     }
55
56     public static IOException close(InputStream input) {
57         return close((Closeable) input);
58     }
59
60     public static IOException close(OutputStream output) {
61         return close((Closeable) output);
62     }
63
64     public static IOException close(final Reader input) {
65         return close((Closeable) input);
66     }
67
68     public static IOException close(final Writer output) {
69         return close((Closeable) output);
70     }
71
72     public static IOException close(final Closeable closeable) {
73         try {
74             if (closeable != null) {
75                 closeable.close();
76             }
77         } catch (final IOException ioe) {

```

```

78         return ioe;
79     }
80     return null;
81 }
82
83 // support jdk6
84 public static IOException close(final ZipFile zip) {
85     try {
86         if (zip != null) {
87             zip.close();
88         }
89     } catch (final IOException ioe) {
90         return ioe;
91     }
92     return null;
93 }
94
95 public static IOException close(URLClassLoader urlClassLoader) {
96     try {
97         if (urlClassLoader instanceof Closeable) {
98             urlClassLoader.close();
99         }
100    } catch (IOException e) {
101        return e;
102    }
103    return null;
104 }
105
106 public static void unzip(String zipFile, String extractFolder) throws IOException {
107     File file = new File(zipFile);
108     ZipFile zip = null;
109     try {
110         int BUFFER = 1024 * 8;
111
112         zip = new ZipFile(file);
113         String newPath = extractFolder;
114
115         new File(newPath).mkdir();
116         Enumeration<? extends ZipEntry> zipFileEntries = zip.entries();
117
118         // Process each entry
119         while (zipFileEntries.hasMoreElements()) {
120             // grab a zip file entry
121             ZipEntry entry = (ZipEntry) zipFileEntries.nextElement();
122             String currentEntry = entry.getName();
123
124             File destFile = new File(newPath, currentEntry);
125             // destFile = new File(newPath, destFile.getName());
126             File destinationParent = destFile.getParentFile();

```

```
127
128 // create the parent directory structure if needed
129 destinationParent.mkdirs();
130
131 if (!entry.isDirectory()) {
132     BufferedInputStream is = null;
133     BufferedOutputStream dest = null;
134     try {
135         is = new BufferedInputStream(zip.getInputStream(en
136             int currentByte;
137         // establish buffer for writing file
138         byte data[] = new byte[BUFFER];
139
140         // write the current file to disk
141         FileOutputStream fos = new FileOutputStream(destFi
142         dest = new BufferedOutputStream(fos, BUFFER);
143
144         // read and write until last byte is encountered
145         while ((currentByte = is.read(data, 0, BUFFER)) !=
146             dest.write(data, 0, currentByte);
147     }
148     dest.flush();
149 } finally {
150     close(dest);
151     close(is);
152 }
153
154 }
155
156 }
157 } finally {
158     close(zip);
159 }
160
161 }
162 }
```