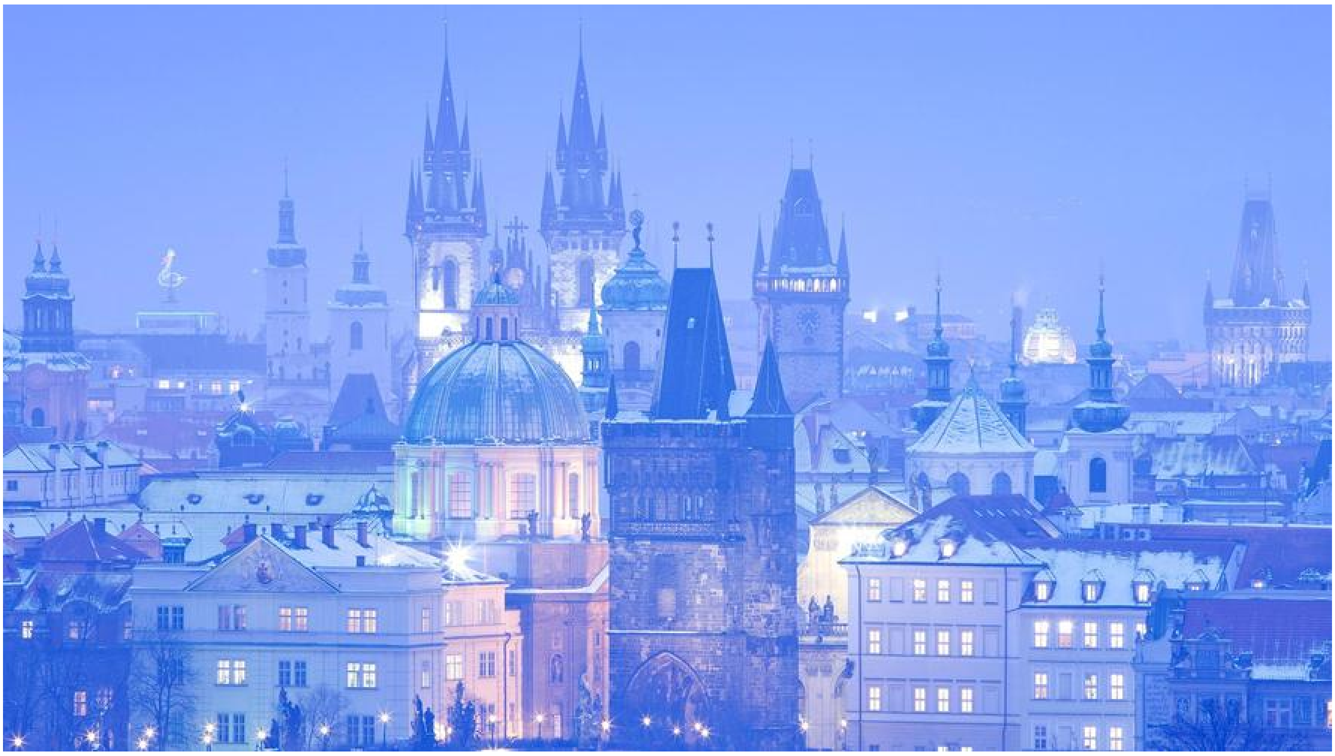
InpurStream 和 String 相互转换

作者：someone36976

原文链接：https://ld246.com/article/1544536705756

来源网站：链滴



InputStream转换为String

- 1、使用 IOUtils.toString (From Apache Utils)

```
String result = IOUtils.toString(inputStream, StandardCharsets.UTF_8);
```

- 2、使 CharStreams (From Guava)

```
String result = CharStreams.toString(new InputStreamReader(inputStream, Charsets.UTF_8));
```

- 3、使用 Scanner (JDK)

```
Scanner s = new Scanner(inputStream).useDelimiter("\\A");
String result = s.hasNext() ? s.next() : "";
```

- 4、使用 Stream API (Java 8).

请注意: 这种方式转换不同的行分隔符 (例如 \r\n) 为 \n.

```
String result = new BufferedReader(new InputStreamReader(inputStream)).lines().collect(Collectors.joining("\n"));
```

- 5、使用 parallel Stream API (Java 8).

请注意: 这种方式转换不同的行分隔符 (例如 \r\n) 为 \n.

```
String result = new BufferedReader(new InputStreamReader(inputStream)).lines().parallel().collect(Collectors.joining("\n"));
```

- 6、使用 InputStreamReader and StringBuilder (JDK)

原文链接：InpurStream 和 String 相互转换

```
final int bufferSize = 1024;
final char[] buffer = new char[bufferSize];
final StringBuilder out = new StringBuilder();
Reader in = new InputStreamReader(inputStream, "UTF-8");
for (; ; ) {
    int rsz = in.read(buffer, 0, buffer.length);
    if (rsz < 0)
        break;
    out.append(buffer, 0, rsz);
}
return out.toString();
```

- 7、使用 StringWriter and IOUtils.copy (Apache Commons)

```
StringWriter writer = new StringWriter();
IOUtils.copy(inputStream, writer, "UTF-8");
return writer.toString();
```

- 8、使用 ByteArrayOutputStream and inputStream.read (JDK)

```
ByteArrayOutputStream result = new ByteArrayOutputStream();
byte[] buffer = new byte[1024];
int length;
while ((length = inputStream.read(buffer)) != -1) {
    result.write(buffer, 0, length);
}
// StandardCharsets.UTF_8.name() > JDK 7
return result.toString("UTF-8");
```

- 9、使用 BufferedReader (JDK).

请注意: 这种方式换转换行分隔符(例如\n\r)为系统的line.separator(例如Windows系统为\r\n).

```
String newLine = System.getProperty("line.separator");
BufferedReader reader = new BufferedReader(new InputStreamReader(inputStream));
StringBuilder result = new StringBuilder();
boolean flag = false;
for (String line; (line = reader.readLine()) != null; ) {
    result.append(flag? newLine: "").append(line);
    flag = true;
}
return result.toString();
```

- 10、使用 Using BufferedInputStream and ByteArrayOutputStream (JDK)

```
BufferedInputStream bis = new BufferedInputStream(inputStream);
ByteArrayOutputStream buf = new ByteArrayOutputStream();
int result = bis.read();
while(result != -1) {
    buf.write((byte) result);
    result = bis.read();
}
// StandardCharsets.UTF_8.name() > JDK 7
return buf.toString("UTF-8");
```

原文链接：InpurStream 和 String 相互转换

- 11、使用 Using inputStream.read() and StringBuilder (JDK).

请注意: 这种方式处理Unicode字符有问题

```
int ch;
StringBuilder sb = new StringBuilder();
while((ch = inputStream.read()) != -1)
    sb.append((char)ch);
reset();
return sb.toString();
```

注意事项

1. 方法 4, 5 and 9 会转换行分隔符
2. 方法11对于处理Unicode字符有问题
3. 方法8的效率是最高的
4. 原文地址

String转换为InputStream

- 1、使用 ByteArrayInputStream（JDK）

```
InputStream is = new ByteArrayInputStream(str.getBytes());
```

- 2、使用 IOUtils.toInputStream（Apache Common）

```
InputStream targetStream = (str, StandardCharsets.UTF_8.name());
```

- 3、使用 ReaderInputStream（Google Guava）

```
InputStream targetStream = new ReaderInputStream(CharSource.wrap(str).openStream(), StandardCharsets.UTF_8.name());
```

原文链接：InpurStream 和 String 相互转换