

PBer 如何理解“流”

今年，许多省份开始新医保接口开发。由于后台全是 java 开发，官方给的文档也是以 java 术语为准。于是：

- 接口说明中声明的输入为输入报文中 INPUT 属性内容，输出为输出报文中 OUTPUT 属性内容。除文件上传下载交易（【9101】、【9102】）外，所有交易都应该有输入输出报文。文件上传下载交易对应文件以流式数据传输。

这样的表述就很常见。

那么，我们来了解一下什么是“流”。

“流”在语言中的关键字是 **stream**。我们知道，无论哪个语言，都要处理数据，而常见的数据大概有这么几种：内存、文件、网络传输、设备读取和发送。

在 PB 里面，内存数据实际上就是我们声明的各种变量或者对象；文件操作经常用的是 `fileopen fileread filewrite fileclose`；网络传输通常由接口自己完成，我们不需要关注细节；设备读取和发送，通常会有 SDK 给我们调用，个别也可以用文件操作函数直接与端口打交道。以上这些数据操作，概括一些规律出来，就过程而言有：打开、传输、关闭，就目标而言有：读取（接收）和存储（发送）。那我们是不是就可以这样理解：这是一条数据的小溪，它可以向内部流进来，也可以向外部流出去。流，其实就是这个意思。只不过，PB 没有流这样的对象设计，我们平时更多的是关注结果，并不关注过程。

Java 中的流有诸多对象，它，例如 `InputStream` 和 `OutputStream` 是输入输出流，输入流用于处理读取（接收），输出流用于存储（发

送)。当然，**java** 中的流还有它的一些特性。只不过，对我们 **PBer** 来说，我们不必关心它的那些具体用法。**Java** 的 **stream** 对象，可以在过程中进行处理，并且对流进行封装达到对各种来源数据统一接口的目的。例如，继承输入流进行解密，继承输出流进行加密，那么后续使用这两个输入和输出的继承对象时，网络上传、文件保存时数据直接就可以加密，而在数据下载、文件读取时直接就自动解密了。达到不同数据源、数据方向的接口统一。而 **PB** 中就没这么方便了，不过也不是不能解决，我们可以有函数对数据进行加密和解密，只不过数据操作过程中需要我们去调用它罢了。

那么文档中提到的这些，我们怎么“翻译”成 **PBer** 能理解的用语呢？很简单：接收数据和发送数据、下载文件上传文件。而这些数据，**java** 是存在字节流里，我们 **PB** 是放在 **blob** 这个大二进制变量里面。