

第30讲 | 如何定制合适的开发协议?

2018-08-02 蔡能

从0开始学游戏开发

[进入课程 >](#)



讲述：蔡能

时长 08:48 大小 4.04M



什么是开发协议？说得简单一点，就是一种客户端和服务端端的网络沟通协议（Protocol）。

广义上说，协议是计算机各种设备之间沟通的桥梁。比如网络之间需要协议，ping 一个网站是否通顺也需要协议，广播地址也需要协议。我们甚至可以说键盘鼠标操作事件也需要协议，Dubbo 架构也需要协议沟通等等。

从狭义上说，协议指的就是网络协议。比如在网络游戏中，客户端和服务端端之间的内容交互，就需要网络协议；在 Web 网站中，前端和后端的交互，也需要协议；再比如，邮件服务的网络交互也需要协议的交互等等。可以说，任何与网络相关的服务都少不了协议的支撑。

在游戏开发中，我们可以自定义一套自己的开发协议，也可以把现成的开发协议拿来使用。具体怎么做呢？我们先来看现在网上用得比较多的几种协议。

三种最流行的开发协议

XML

XML 几乎是网络上最早出现的传输协议之一。在最早的 Web 开发中，XML 可以作为网络协议，也可以用作配置文件。比如某些游戏或者某些应用的配置文件，都可以使用 XML 来编写。

从人类的角度讲，它的可读性比较强，解析也比较方便。我们先来看几种解析方式。

解析方式是这些协议被程序理解的一种方式，按照这种方式解析，和我后面要说的自定义协议的解析和剖析结合起来，乃前后呼应之奇效。

1. PULL 方式：PULL 解析是一种专门为安卓设备解析 XML 文件设计的解析方式。这种解析方式更适用于移动设备。PULL 解析和我们下面要说的 SAX 解析比较类似，不一样的地方是 PULL 读取 XML 文件后，触发相应的事件调用方法返回的是 number。另外，PULL 方式可以控制程序的解析停止位置，不需要全部解析，可以随时停止。
2. SAX 方式：SAX (Simple API for XML) 采用事件驱动型方式。语法解析从发现元素开始、元素结束、文本、文档的开始或结束等，就开始事件的发送，程序员编写响应这些事件的代码，就可以直接将数据保存下来。所以优点是，不需要载入整个文档，资源占用比较少。

SAX 解析器的代码比 DOM 解析器的代码更精简，一般应用于 Applet 技术。缺点就是，这种解析方式并不持久，等事件消息过去后，如果你没有保存数据，那么数据就丢失了；从事件的消息中我们只能得到文本数据，但不知道这个文本属于哪个元素。但是，我们只需 XML 文档的少量内容就可以解析出我们所需的数据，所以耗费的内存更少。

3. DOM 方式：DOM (Document Object Model) 是最传统的解析方式。解析器读入整个文档，然后在内容中构建一个 XML 的树结构，使用代码就可以操作这个树结构。优点是整个文档树在内存中，便于操作；而且支持删除、修改、重排等多种功能。缺点是将整个文档调入内存比较浪费计算机的时间和空间，但是如果一旦解析了文档，还需多次访问这些数据的话，这种方式就可以起到作用了。

JSON

其实，目前 XML 已经不太流行，取而代之的是 JSON。JSON 是一种轻量级的数据交换格式。它用完全独立于编程语言的文本格式来存储和表示数据。

比之 XML，它看起来更加简洁和清晰，层次结构分明；JSON 易于阅读和编写，在程序方面，也易于机器解析和生成，同时也提升了网络传输效率。这些优点使得 JSON 很快在程序员中流行起来，成为理想的数据交换语言。

它也是移动端比较常见的网络传输协议。相对于前面所说的 XML 格式，它更为简单，体积更小，加之对网络流量和内存的需求更小，所以 JSON 比 XML 更适合移动端的使用。

我们来看一下 JSON 的几种流行的解析程序库。

1. Gson 是谷歌开源的一种解析方法，使用 Java 编写，你可以通过提供的 JAR 文件包，使用静态方法直接将 JSON 字符串解析成 Java 对象，这样使用起来简单方便。
2. FastJSON 是阿里开源的一个解析 JSON 数据的类库。
3. JSONObject 也是一个解析 JSON 字符串的 Java 类。第二、第三这两种用的人都比较少，我就不多介绍了。

当然，支持别的语言的库也有很多，由于 JSON 比较流行，所以各种语言都有其支持的类库版本，比如 Python、C++、Ruby 等等。

ProtoBuf

ProtoBuf 全称 Google Protocol Buffer，是谷歌公司开发的内部混合语言数据标准。目前正在使用的有接近五万种报文格式定义和超过一万两千多个 proto 文件。它们都用于 RPC 系统和持续数据存储的系统。

这是一种轻便、高效的结构化数据存储格式，可以用于结构化数据的序列化操作。它很适合用作数据存储或 RPC 数据交换格式。可以用于通讯协议、数据存储等领域。由于是独立的协议系统，所以它和开发语言、运行平台都没有关系，可以用在扩展的序列化结构数据格式。目前提供了 C++、Java、Python、Ruby、Go 等语言的开发接口 API。

ProtoBuf 方便的地方在于，它有一款编译器可以将 proto 后缀的协议文件，编译成 C++、Java、Python 等语言的源代码。你可以直接看到和利用这些源代码，且不需要自己去做解析，所以不同语言之间使用 ProtoBuf 的协议规范都是一样的，但是有一个问题是，ProtoBuf 存储的文件格式是二进制的，由于是二进制的，所以程序员需要调试其保存的内

容就有点麻烦，当然这可能只是对于某些人来说的瑕疵吧，对于大部分人来讲，方便性还是大于瑕疵的。

ProtoBuf 的编码风格是这样的，花括号的使用类似 C/C++、Java。数据类型的命名方式使用驼峰命名，比如 `DataType`、`NewObject`。字段的变量小写并使用下划线连接，类似 GNU 规范，比如 `proto_buf`、`user_name`。枚举类型使用大写，并使用下划线连接，比如 `MY_HOME`，`BEST_FRIEND`。

Protobuf 并不是针对大型数据设计的，Protobuf 对于 1M 以下的 message 有很高的效率，但是当 message 大于 1M 的时候，Protobuf 的效率会逐步降低。

如何自己定义协议包？


我们讲完了三种目前最流行的开发协议，接下来我们要讲讲如何自己定义协议包。

我们所说的**协议包**，是在 **TCP 和 UDP 传输之上的协议包**，也就是**通过字符串的形式发送的协议包**。这些协议包在客户端和服务端之间做了约定，也就是说，客户端和服务端都能通过拿到协议包来进行解包操作，并且进行一系列的逻辑运算并返回结果，当然结果也是协议包的形式发送出去。

一个好的协议不仅能节约网络带宽，也能让接收端快速拿到和解析需要的内容。设计协议包，必须保证**安全性**和**完整性**。

为了保证完整性，接收方需要知道协议的长度，或者知道协议的尾部在哪里。


我们可以给协议最末尾添加分隔符，该分隔符需要特殊字符。不能被传输的内容所混淆，又要能达到方便接收方辨认，因此，该特殊字符需要具有唯一性。比如我们可以将 `“!@#$”` 这四个字符做为分隔符，那么协议看起来可能是这样：

 复制代码

1 [协议头][协议体][协议结尾分隔符]

你可能要问了，在传输的过程中，我知道了协议长度，不需要协议头，只需要协议长度就可以？是的。因为有了协议长度，协议尾部有没有分隔符就不重要了。如果我们固定好输出协

议长度的字节数，就可以忽略协议头。在这种情况下，协议看起来像是这样：

 复制代码


```
1 [协议长度 2 字节][协议体]
```

这样简单地就能定义整个协议的内容。

在读取的时候，我们只需要读取开头的两个字节，转换为一个 short 的长度，或者四个字节一个 int 的长度，在第三个字节开始就是协议体。让程序开始计算长度，如果长度少于协议长度所定义的长度，那就继续接收，如果接收长度超过协议所定义的长度，切割协议体，并将下一段开始的协议存储到内存中留待下一次取出。这种方式是最方便的。

我们在保证协议完整性的同时，也要保证协议不被破坏和篡改，也就是所说的安全性。在这种情况下，最直接的方式，就可以将协议内容进行加密。比如 SHA-256 或者 AES 等等加密方式将内容加密，随后传输过去，最简单的做法就是将密码在客户端和服务端协商好就可以了。

看起来可能是这个样子：

 复制代码

```
1 [协议长度 2 字节][加密协议体]
```

小结

这节内容差不多了，我们总结一下。我和你介绍了这几个内容。

我介绍了三种的开发协议 XML、JSON 和 ProtoBuf，以及它们对应的解析方式。XML 是网络上最早出现的传输协议之一。

游戏或应用的配置文件，都可以使用 XML 来编写，但是目前 XML 已经不太流行，取而代之的是 JSON。

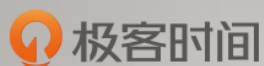
ProtoBuf 适合用作数据存储或 RPC 数据交换格式，缺点是保存比较麻烦，但是总体来讲还是比较方便的。

自己定义协议包需要考虑完整性和安全性。接收方需要知道协议的长度，或者知道协议的尾部在哪里，就可以保证协议包的完整性。而最直接的给协议包加密，就可以保证安全性。

最后，给你留一个小问题吧。

在自定义协议中，如果使用添加协议结尾的方式来做协议，如何才能保证协议结尾分割字符串不和协议本身的二进制内容重复？

欢迎留言说出你的看法。我在下一节的挑战中等你！



从 0 开始学游戏开发

你的游戏开发入门第一课

蔡能

原网易游戏引擎架构师
资深游戏底层技术专家



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 第29讲 | 如何选择合适的开发语言？

下一篇 第31讲 | 热点剖析（八）：谈谈移动游戏的未来发展

精选留言 (3)

写留言



以往

2018-09-11



请问老师怎么保证唯一性?

用随机生成的字符串替换 “!@#\$” 可行吗

展开 ▾



以往

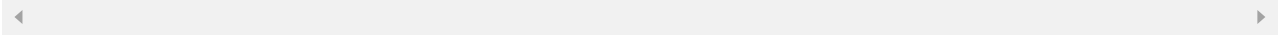
2018-09-06



内容中出现 “!@#\$” 时可以用\\!\\@\\#\\\$这种形式来转义

展开 ▾

作者回复: 也可以但并不能保证唯一性



以往

2018-09-03



当传输内容中出现结尾字符时, 对其进行转义

展开 ▾

作者回复: 说的简单了点

