

这是请求报文的格式：

```
<method> <request-URL> <version>
<headers>

<entity-body>
```

这是响应报文的格式（注意，只有起始行的语法有所不同）：

```
<version> <status> <reason-phrase>
<headers>

<entity-body>
```

下面是对各部分的简要描述。

- 方法（method）

客户端希望服务器对资源执行的动作。是一个单独的词，比如 GET、HEAD 或 POST。本章稍后将详细介绍方法。

- 请求 URL（request-URL）

命名了所请求资源，或者 URL 路径组件的完整 URL。如果直接与服务器进行对话，只要 URL 的路径组件是资源的绝对路径，通常就不会有什么问题——服务器可以假定自己是 URL 的主机 / 端口。第 2 章详细地介绍了 URL 的语法。

- 版本（version）

报文所使用的 HTTP 版本，其格式看起来是这样的：

```
HTTP/<major>.<minor>
```

其中主要版本号（major）和次要版本号（minor）都是整数。本章稍后会详细说明 HTTP 的版本问题。

- 状态码（status-code）

这三位数字描述了请求过程中所发生的情况。每个状态码的第一位数字都用于描述状态的一般类别（“成功”、“出错”等）。本章稍后提供了 HTTP 规范定义的状态码及其含义的完整列表。

- 原因短语（reason-phrase）

数字状态码的可读版本，包含行终止序列之前的所有文本。本章稍后提供了 HTTP 规范定义的所有状态码的原因短语示例。原因短语只对人类有意义，因此，比如说，尽管响应行 HTTP/1.0 200 NOT OK 和 HTTP/1.0 200 OK 中原因短语的含义不同，但同样都会被当作成功指示处理。