# 第02章-你的第一款Netty应用程序

在本章中，我们将展示如何构建一个基于Netty 的客户端和服务器。应用程序很简单：客户端将消息发送给服务器，而服务器再将消息回送给客户端。但是这个练习很重要，原因有两个。

首先，它会提供一个测试台，用于设置和验证你的开发工具和环境，如果你打算通过对本书的示例代码的练习来为自己将来的开发工作做准备，那么它将是必不可少的。

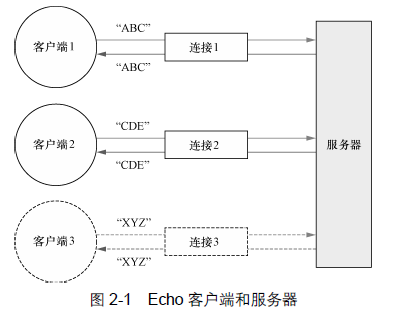
其次，你将获得关于Netty 的一个关键方面的实践经验，即在前一章中提到过的：通过ChannelHandler 来构建应用程序的逻辑。这能让你对Netty API 的深入学习做好准备。

## 2.1 设置开发环境

要编译和运行本书的示例，只需要JDK 和Apache Maven 这两样工具，它们都是可以免费下载的。可以使用纯文本编辑器，但是建议你使用用于Java 的集成开发环境（IDE）。

## 2.2 Netty 客户端/服务器概览

图2-1 从高层次上展示了一个你将要编写的Echo 客户端和服务器应用程序。虽然你的主要关注点可能是编写基于Web 的用于被浏览器访问的应用程序，但是通过同时实现客户端和服务器，你一定能更加全面地理解Netty 的API。



虽然我们已经谈及到了客户端，但是该图展示的是多个客户端同时连接到一台服务器。所能够支持的客户端数量，在理论上，仅受限于系统的可用资源（以及所使用的JDK 版本可能会施加的限制）。

Echo 客户端和服务器之间的交互是非常简单的；在客户端建立一个连接之后，它会向服务器发送一个或多个消息，反过来，服务器又会将每个消息回送给客户端。虽然它本身看起来好像用处不大，但它充分地体现了客户端/服务器系统中典型的请求-响应交互模式。

## 2.3 编写Echo 服务器

所有的Netty 服务器都需要以下两部分。

* 至少一个ChannelHandler—该组件实现了服务器对从客户端接收的数据的处理，即它的业务逻辑。
* 引导—这是配置服务器的启动代码。至少，它会将服务器绑定到它要监听连接请求的端口上。

在本小节的剩下部分，我们将描述Echo 服务器的业务逻辑以及引导代码。