# 11.6 使用Socket API

我们之前介绍的Http协议和WebSocket协议都属于应用层协议，除了它们，应用层协议还有很多如：SMTP、FTP等，这些应用层协议的实现都是通过Socket API来实现的。其实，操作系统中提供的原生网络请求API是标准的，在C语言的Socket库中，它主要提供了端到端建立链接和发送数据的基础API，而高级编程语言中的Socket库其实都是对操作系统的socket API的一个封装。所以，如果我们需要自定义协议或者想直接来控制管理网络链接、又或者我们觉得自带的HttpClient不好用想重新实现一个，这时我们就需要使用Socket。Flutter的Socket API在dart：io包中，下面我们看一个使用Socket实现简单http请求的示例，以请求百度首页为例：

\_request() async{  
 //建立连接  
 var socket=await Socket.connect("baidu.com", 80);  
 //根据http协议，发送请求头  
 socket.writeln("GET / HTTP/1.1");  
 socket.writeln("Host:baidu.com");  
 socket.writeln("Connection:close");  
 socket.writeln();  
 await socket.flush(); //发送  
 //读取返回内容  
 \_response =await socket.transform(utf8.decoder).join();  
 await socket.close();  
}

可以看到，使用Socket需要我们自己实现Http协议（需要自己实现和服务器的通信过程），本例只是一个简单示例，没有处理重定向、cookie等。本示例完整代码参考示例demo，运行后效果如图11-2所示：

图11-2

可以看到响应内容分两个部分，第一部分是响应头，第二部分是响应体，服务端可以根据请求信息动态来输出响应体。由于本示例请求头比较简单，所以响应体和浏览器中访问的会有差别，读者可以补充一些请求头(如user-agent)来看看输出的变化。