# 从第三方拿数据，需要确定数据拿到后，需要的数据结构，可以创建空对象，拿到想要的数据，再添加到这个对象中，组成我们想要的结构

# 关于学子书院评论的分页

分页的的使用在追书神器的API中通过**start**和**limit**参数实现的

其中:

start指**从第几条记录开始返回**，其从**0**开始编号

limit指**返回的数量**

此时声明data中的变量**page**，代表**当前页码**

只有到达页面底部时，page变量才累加，然后计算出start的参数值，并且以该值为URL的构成部分，向服务器发送异步的请求

Start=(page-1)\*pagesize

1.不能无限发送请求

2.初始情况下执行的异步请求与向下滚动时执行的异步请求基本是相同的，所以最好封装成自定义函数

# 

# 2.WebSocket

## 2.1 概述

WebSocket是一种**网络通信协议**，其最大的特点是：**实现实时的双向通讯**，即服务器可以主动推送消息到客户端，客户端也可以向服务器发送请求。

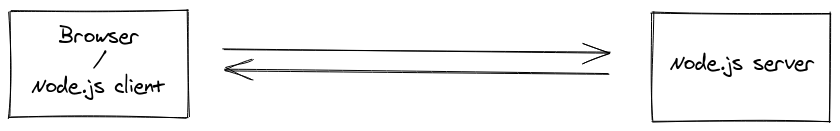
WebSocket协议产生2008年，2011年成为国际标准。

**Socket.io**是一个WebSocket库，其提供**实时的、双向的数据通讯**，并且**抹平了跨平台的技术细节**。

https://socket.io/

socket.io包括：

* 浏览器
* Node.js服务器



## 2.2 WebSocket优点

* **没有同源策略的限制，**客户端可以与任何服务器之间进行通讯
* **数据格式轻量，通讯高效**
* WebSocket协议的前缀是**ws**(如果加密的话，则为**wss**)

### 2.2 安装

Node.js服务器

npm install --save socket.io

客户端

直接**在HTML文件中通过<script>标签引入socket.io的客户端脚本文件即可**

<script src="https://cdnjs.cloudflare.com/ajax/libs/socket.io/2.3.0/socket.io.js">   
</script>

只要调用的socket.io.js脚本文件，系统将自动暴露名称为io的函数

## 客户端API

emit()方法

**emit()**方法用于**实现客户端向服务器广播事件**，其语法结构是：

**client.emit("事件名称"[,数据])**

**on()**方法用于实现**根据指定的事件来注册一个函数**，其语法结构是：

**client.on("事件名称",回调函数)**

## 服务器API

· on()方法

**on()**方法用于实现**根据指定的事件来注册一个函数**，其语法结构是：

**server.on("事件名称",回调函数)**

emit()方法

**emit()**方法用于**实现服务器向客户端广播事件**，其语法结构是：

**server.emit("事件名称"[,数据])**