Echo 的发帖操作是怎么做的

原创 飞天小牛肉 [飞天小牛肉](javascript:void(0);)

**飞天小牛肉**

微信号 CS-Wiki

功能介绍 InfoQ & 阿里云签约作者，分享原创技术干货和成长经验

2021-04-07[原文](https://mp.weixin.qq.com/s?__biz=MzI0NDc3ODE5OQ==&mid=2247486251&idx=2&sn=2cad3c5a1b6a9ffa5ff921e0697c3329&chksm=e959de93de2e57854597b02601e18dcfa5a1b39760ea09500f69a6d18100e4233812e6ecbed7&scene=178&cur_album_id=1744497649518493697#rd) 发表于

收录于合集 #Echo 学习教程 18个

**浅谈 Ajax**

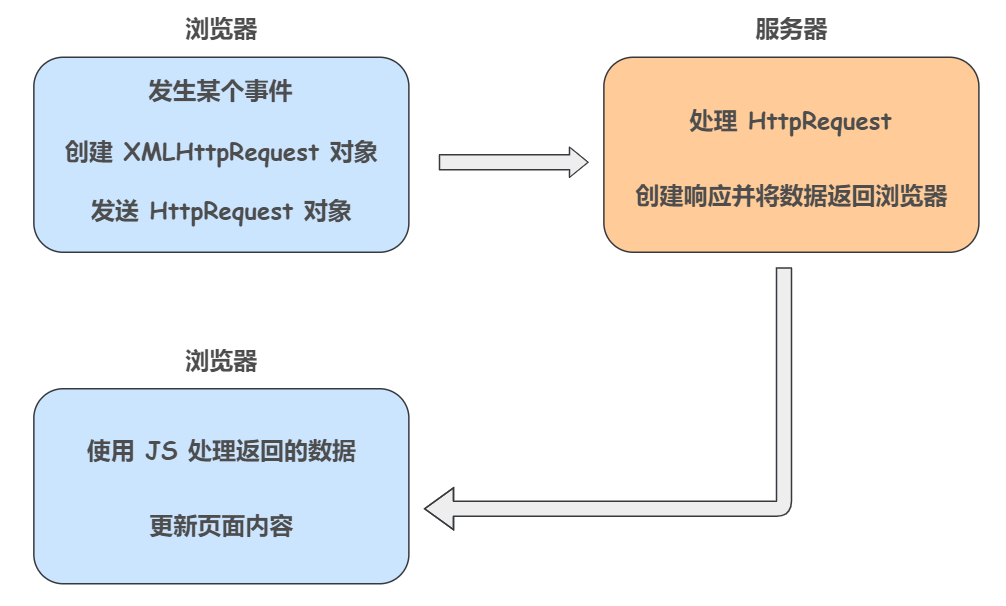
首先，各位不妨想一想，在平常开发中，我们是怎么在前端跟后端之间进行数据交互的？

最常用最原始的，form 表单。通过 form 表单以 post/get 方式提交数据，当你点击 submit 按钮时，浏览器会把你在 input 里面输入的数据提交到 form 表单中的 action 这个路径。

但是这种方式在某些情况下，对用户来说并不友好。因为在进行提交时，页面会发生跳转或刷新，我给帖子点了个赞你页面还需要刷新一下？显然这样用户体验不好。

为此，Ajax 应运而生。Ajax 的全称是 Asynchronous JavaScript and XML（**异步** JavaScript+XML），它并不是一种新的编程语言，而是一种使用现有标准的新方法。它依赖的是现有的 CSS/HTML/Javascript，而其中最核心的依赖是浏览器提供的 XMLHttpRequest 对象。这个对象为向服务器发送请求和解析服务器返回的响应提供了流畅的接口，使得浏览器可以发出 HTTP 请求与接收 HTTP 响应，**实现在页面不刷新（局部刷新）的情况下和服务端进行数据交互**。

Ajax 的工作原理大概是这样的：



**浅谈同步、异步、阻塞、非阻塞**

本部分内容参考知乎「怎样理解阻塞非阻塞与同步异步的区别？- https://www.zhihu.com/question/19732473」问题下的高赞回答「卢毅luis - https://www.zhihu.com/people/svjoke」

上面我们说到 Ajax 是异步的，那么所谓同步和异步，它们其实是两种不同的**消息通信机制**，我们以客户端（调用者）和服务端（被调用者）之间的通信为例：

* **同步**，就是指客户端调用服务端的某个东西时，在没有得到调用结果之前，该调用就不会返回。也就是说客户端必须等到这个调用返回结果才能继续往后执行；
* **异步**，和同步相反，调用在发出之后，这个调用就直接返回了，所以没有返回结果。换句话说，当客户端发送出了一个异步调用后，它不会立刻得到结果，而是在未来的某个时间，服务端通过状态、通知来通知客户端你的这个异步调用成功了，或者也可以通过回调函数来处理这个异步调用的返回结果。

举个通俗的例子：

你打电话问书店老板有没有某本书，如果是同步通信机制，书店老板会说，"您稍等，我查一下"，然后开始查啊查，等查好了（可能是 5 秒，也可能是一天）告诉你结果（返回结果）；

而异步通信机制，书店老板直接告诉你 "我查一下啊，查好了打电话给你"，然后直接挂电话了（不返回结果）。然后查好了，他会主动打电话给你。在这里老板通过 "回电" 这种方式来回调。

另外，关于同步和异步，需要区别于阻塞和非阻塞，这几个概念经常容易混淆。阻塞和非阻塞关注的是**客户端在等待调用结果时的状态**：

* **阻塞调用**，是指调用结果返回之前，客户端的当前线程会被挂起，这个调用线程只有在获取到服务端的调用结果之后才能继续运行；
* **非阻塞调用**，就是说即使客户端的线程无法立即获取到服务端的调用结果，这个线程也不会被阻塞，它可以继续去做其他的事情。

还是上面的例子，你打电话问书店老板有没有某本书，如果是阻塞式调用，你会一直把自己 "挂起"，直到得到这本书有没有的结果；

如果是非阻塞式调用，你不管老板有没有告诉你，你自己就先去做别的事情了， 当然你也要偶尔过几分钟 check 一下老板这边有没有返回结果。

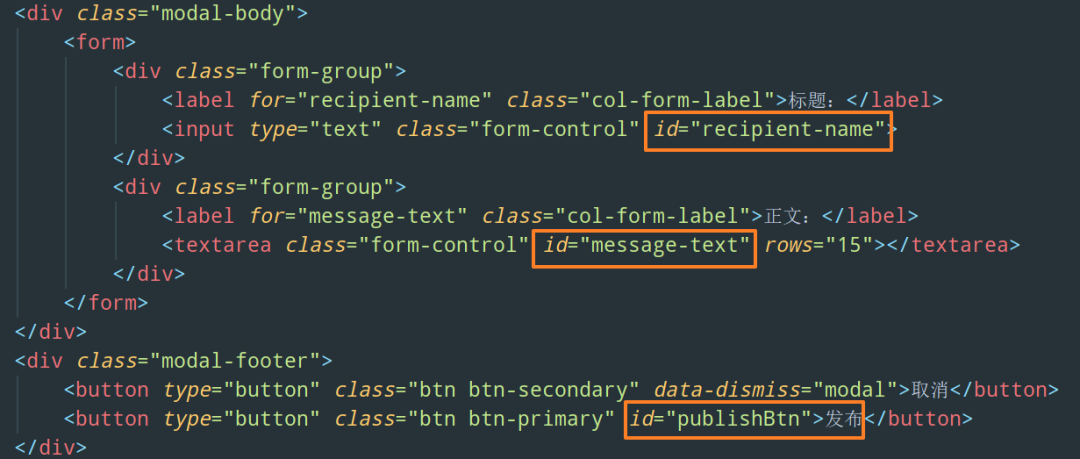
**发帖功能解析**

事实上，使用 JS 编写 Ajax 代码并不容易，因为不同的浏览器对 Ajax 的实现并不相同。这意味着我们必须编写额外的代码对浏览器进行测试。不过，jQuery 团队解决了这个难题，我们只需要一行简单的代码，就可以实现 Ajax  功能，这里就不再详细赘述了。Echo 中当然也使用了 jQuery，我们来看看在 Echo 中发帖操作是如何发送异步请求的：



function(data) 就是回调函数，是 Ajax 在请求成功后自动调用的，参数 data 就是服务端返回的这个异步请求的值。

Ajax 会根据我们指定的 url /discuss/add 来找到对应的 Controller 方法，通过 id 选择器获取用户输入的数据，封装成 JSON 字符串发送过去（即帖子的标题和内容：{"title": title, "content": content}），这些数据会被自动传入到 Controller 的方法形参中。Controller 方法调用完成后，Ajax 会执行回调函数，获取 Controller 返回结果并执行相应操作。



使用 Ajax 异步提交代替传统的 form 表单提交的好处在于，使用异步方式与服务器通信，不需要打断用户的操作，具有更加迅速的响应能力，使得用户体验更好。

我们来看看 Controller 层的方法：



getJSONString 是我们自己写的一个工具类中的方法，通过阿里开源的 fastjson 将服务端返回的消息封装成 JSON 格式的字符串：



真正的发帖操作在 Service 层，其实就是一个插入数据库的操作，目前做的还比较简单，帖子的内容只能是普通的文本，后面会考虑支持 MarkDown 的。另外，这里有一个**过滤敏感词**的操作，涉及**前缀树**的设计与使用，后续会单开一篇文章详细讲解。

DiscussPostService.addDiscussPost：



**小结**

本来准备这章把首页帖子列表是怎么实现的也讲了，但是发现还涉及分页这个模块，由于分页功能是我们自己动手实现的，没有借助框架，所以还是有必要花点篇幅讲一下，就放在下一篇文章吧。

点击进入小牛肉的留言板



* 博主小硕在读，深耕 Java，目前在维护一个教程类仓库 CS-Wiki「Gitee 官方推荐项目，现已 1.6k+ star，仓库地址：https://gitee.com/veal98/CS-Wiki」，公众号上的文章也会在此同步更新，欢迎各位前来交流学习。
* 准备春招秋招的小伙伴可以参考我的这个论坛项目 Echo「Gitee 官方推荐项目，现已 600+ star，仓库地址：https://gitee.com/veal98/Echo」。配套教程正在同步更新中，公众号后台回复 "Echo" 即可免费获取。
* 另外，虽然现在本号仍然很小，不过我还是建了一个交流群『小牛肉和它的小伙伴们』，感兴趣的各位可以下方扫码加我微信回复 "进群"，我拉你进群：





### 精选留言

暂无...