

枕边书

常怀敬畏之心。

博客圖

[首页](#)

新随笔

联系

管理

网页实时聊天之PHP实现websocket

前言

websocket 作为 HTML5 里一个新的特性一直很受人关注，因为它真的非常酷，打破了 http “请求-响应”的常规思维，实现了服务器向客户端主动推送消息，本文介绍如何使用 PHP 和 JS 应用 websocket 实现一个网页实时聊天室；

以前写过一篇文章讲述如何使用ajax长轮询实现网页实时聊天，见链接：[网页实时聊天之js和jQuery实现ajax长轮询](#)，但是轮询和服务器的 pending 都是无谓的消耗，websocket 才是新的趋势。

最近艰难地“挤”出了一点时间，完善了很早之前做的 websocket “请求-原样返回”服务器，用js完善了下客户端功能，把过程和思路分享给大家，顺便也普及一下 websocket 相关的知识，当然现在讨论 websocket 的文章也特别多，有些理论性的东西我也就略过了，给出参考文章供大家选择阅读。

正文开始前，先贴一张聊天室的效果图（请不要在意CSS渣的页面）：



然后当然是源码：[我是源码链接 - github - 枕边书](#)

websocket

简介

WebSocket 不是一门技术，而是一种全新的协议。它应用 TCP 的 Socket（套接字），为网络应用定义了一个新的重要的能力：客户端和服务端的双全工传输和双向通信。是继 Java applets、XMLHttpRequest、Adobe Flash、ActiveXObject、各类 Comet 技术之后，服务器推送客户端消息的新趋势。

与http的关系

在网络分层上，websocket 与 http 协议都是应用层的协议，它们都是基于 tcp 传输层的，但是 websocket 在建立连接时，是借用 http 的 101 switch protocol 来达到协议转换（Upgrade）的，从 HTTP 协议切换到 WebSocket 通信协议，这个动作协议中称“握手”；

握手成功后，websocket 就使用自己的协议规定的方式进行通讯，跟 http 就没有关系了。

握手

以下是一个我自己的浏览器发送的典型的握手 http 头:

公告

欢迎来到我的博客园~

本人后端开发，对网络编程和 Unix 系统编程有浓厚兴趣。

GitHub: [zhenbianshu](#)

博客原创，首发于Github.io：枕边书，
也会更新一些工作和学习的心得，欢迎
交流关注。

昵称：枕边书

园龄： 4年4个月

荣誉： 推荐博客

粉丝: 446

关注: 24

+加关注

积分与排名

积分 - 182944

排名 - 2644

随笔分类

C/C++(4)

Java(8)

JS(3)

LINUX(9)

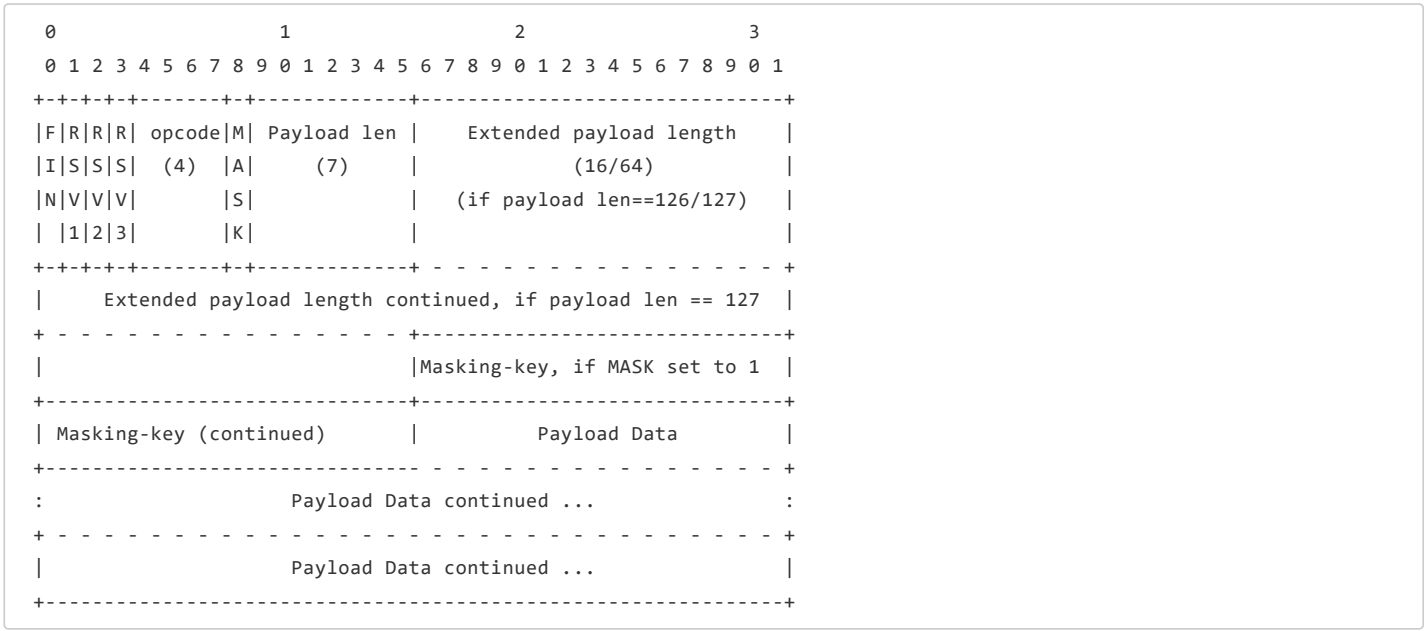
PHP(21)

```
handshake buffer:
Cache-Control: no-cache
Upgrade: websocket
Origin: http://localhost
Sec-WebSocket-Version: 13
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/54.0.2840.98 Safari/537.36
Accept-Encoding: gzip, deflate, sdch, br
Accept-Language: zh-CN,zh;q=0.8,en;q=0.6
Sec-WebSocket-Key: V5CbP0tZalZYat9mE+YHow==
Sec-WebSocket-Extensions: permessage-deflate; client_max_window_bits
```

服务器收到握手请求后，提取出请求头中的“Sec-WebSocket-Key”字段，追回一个固定的字符串‘258EAFa5-E914-47DA-95CA-C5AB0DC85B11’， 然后进行 sha1 加密，最后转换为 base64 编码，作为 key 以“Sec-WebSocket-Accept” 字段返回给客户端，客户端匹配此 key 后，便建立了连接，完成了握手；

数据传输

websocket 有自己规定的数据传输格式，称为 帧（Frame）， 下图是一个数据帧的结构，其中单位为bit：



具体每个字段是什么意思，有兴趣的可以看一下这篇文章 [The WebSocket Protocol 5.数据帧](#) 感觉自己对二进制的操作还不是很灵活，也就没有挑战自己写算法解析数据了，下面的数据帧解析和封装都是使用的网上的算法。

不过，我工作中写支付网关中还是会经常用到数据的进制操作的，这个一定是要仔细研究总结一下的，嗯，先记下。

PHP 实现 websocket 服务器

PHP 实现 websocket 的话，主要是应用 PHP 的 socket 函数库：

PHP 的 socket 函数库跟 C 语言的 socket 函数非常类似，以前翻过一遍 APUE, 所以觉得还挺好理解。在 PHP 手册中看一遍 socket 函数，我想大家也能对 php 的 socket 编程有一定的认识。

下面会在代码中对所用函数进行简单的注释。

文件描述符

忽然提及'文件描述符'，大家可能会有些奇怪。

但作为服务器，是必须要对已经连接的 socket 进行存储和识别的。每一个 socket 代表一个用户，如何关联和查询用户信息与 socket 的对应就是一个问题了，这里便应用了关于文件描述符的一点小技巧。

我们知道 linux 是'万物皆文件'的，C 语言的 socket 的实现便是一个个的'文件描述符'，这个文件描述符一般是打开文件的顺序递增的 int 数值，从 0 一直递增（当然系统是有限制的）。每一个 socket 都对应一个文件，读写 socket 都是操作对应的文件，所以也能像文件系统一样应用 read 和 write 函数。

tips: linux 中，标准输入对应的是文件描述符 0；标准输出对应的文件描述符是 1；标准错误对应的文件描述符是 2；所以我们可以使用 0 1 2对输入输出重定向。

那么类似于 C socket 的 PHP socket 自然也继承了这一点，它创建的 socket 也是类型于 int 值为 4 5 之类的资源类型。我们可以使用 (int) 或 intval() 函数把 socket 转换为一个唯一的ID，从而可以实现用一个 '类索引数组' 来存储 socket 资源和对应的用户信息；

结果类似：

```
$connected_sockets = array(
    (int)$socket => array(
        'resource' => $socket,
        'name' => $name,
        'ip' => $ip,
        'port' => $port,
        ...
    )
)
```

创建服务器socket

下面是一段创建服务器 socket 的代码：

```
// 创建一个 TCP socket，此函数的可选值在官方文档中写得十分详细，这里不再提了
$this->master = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
// 设置IP和端口重用，在重启服务器后能重新使用此端口；
socket_set_option($this->master, SOL_SOCKET, SO_REUSEADDR, 1);
// 将IP和端口绑定在服务器socket上；
```

Profession(6)

WEB(4)

后端(24)

算法(6)

最新评论

1. Re:PHP中的回调函数和匿名函数

o(*^@^*)o

--石海莹

2. Re:空间索引 - 四叉树

楼主，你好。能抽空把怎么释放内存的函数写一下吗？

--人生小覬

3. Re:多线程编程 - PHP 实现

讲解的非常清楚，感谢啊！

--aubert

阅读排行榜

1. 网页实时聊天之PHP实现websocket(73133)

2. 多线程编程 - PHP 实现(57286)

3. PHP的openssl加密扩展使用小结(45699)

4. shell实现SSH自动登陆(31487)

5. 初探PHP多进程(22158)

6. 网页实时聊天之js和jQuery实现ajax长轮询(22147)

7. PHP中的回调函数和匿名函数(18445)

8. 小时到分钟 - 一步步优化巨量关键词的匹配(16465)

9. 空间索引 - 四叉树(12865)

10. 使用PostgreSQL进行中文全文检索(12339)

推荐排行榜

1. 小时到分钟 - 一步步优化巨量关键词的匹配(98)

```
socket_bind($this->master, $host, $port);
// listen函数使主动连接套接口变为被连接套接口，使得此 socket 能被其他 socket 访问，从而实现服务器功能。后面的参数则是自定义的待处理socket
socket_listen($this->master, self::LISTEN_SOCKET_NUM);
```

这样，我们就得到一个服务器 socket，当有客户端连接到此 socket 上时，它将改变状态为可读，那就看接下来服务器的处理逻辑了。

服务器逻辑

这里着重讲一下 `socket_select($read, $write, $except, $tv_sec [, $tv_usec])`:

select 函数使用传统的 select 模型，可读、写、异常的 socket 会被分别放入 \$socket, \$write, \$except 数组中,然后返回 状态改变的 socket 的数目，如果发生了错误，函数将会返回 false.

需要注意的是最后两个时间参数，它们只有单位不同，可以搭配使用，用来表示 socket_select 阻塞的时长，为0时此函数立即返回，可以用于轮询机制。为 NULL 时，函数会一直阻塞下去, 这里我们置 \$tv_sec 参数为null，让它一直阻塞，直到有可操作的 socket 返回。

下面是服务器的主要逻辑：

```
$write = $except = NULL;
$sockets = array_column($this->sockets, 'resource'); // 获取到全部的 socket 资源
$read_num = socket_select($sockets, $write, $except, NULL);

foreach ($sockets as $socket) {
    // 如果可读的是服务器 socket，则处理连接逻辑；
    if ($socket == $this->master) {
        socket_accept($this->master);
        // socket_accept() 接受 请求 “正在 listen 的 socket（像我们的服务器 socket ）” 的连接，并一个客户端 socket，错误时返回 false
        self::connect($client);
        continue;
    }
    // 如果可读的是其他已连接 socket ,则读取其数据,并处理应答逻辑
} else {
    // 函数 socket_recv() 从 socket 中接受长度为 len 字节的数据，并保存在 $buffer 中。
    $bytes = @socket_recv($socket, $buffer, 2048, 0);

    if ($bytes < 9) {
        // 当客户端忽然中断时，服务器会接收到一个 8 字节长度的消息（由于其数据帧机制，8字节的消息我们认为它是客户端异常中断消息）
        $recv_msg = $this->disconnect($socket);
    } else {
        // 如果此客户端还未握手，执行握手逻辑
        if (!$this->sockets[(int)$socket]['handshake']) {
            self::handShake($socket, $buffer);
            continue;
        } else {
            $recv_msg = self::parse($buffer);
        }
    }

    // 广播消息
    $this->broadcast($msg);
}
}
```

这里只是服务器处理消息的基础代码，日志记录和异常处理都略过了，而且还有些数据帧解析和封装的方法，各位也不一定看爱，有兴趣的可以去 github 上支持一下我的源码~~

此外，为了便于服务器与客户端的交互，我自己定义了 json 类型的消息格式，形似：

```
$msg = [
    'type' => $msg_type, // 有普通消息，上下线消息，服务器消息
    'from' => $msg_resource, // 消息来源
    'content' => $msg_content, // 消息内容
    'user_list' => $uname_list, // 便于同步当前在线人数与姓名
];
```

客户端

创建客户端

前端我们使用 js 调用 WebSocket 方法很简单就能创建一个 websocket 连接，服务器会为帮我们完成连接、握手的操作，js 使用事件机制来处理浏览器与服务器的交互：

```
// 创建一个 websocket 连接
var ws = new WebSocket("ws://127.0.0.1:8080");

// websocket 创建成功事件
ws.onopen = function () {
};

// websocket 接收到消息事件
ws.onmessage = function (e) {
    var msg = JSON.parse(e.data);
}

// websocket 错误事件
```

2. Linux - 请允许我静静地后台运行 (26)
3. 多线程编程 - PHP 实现(20)
4. 网页实时聊天之PHP实现websocket(19)
5. 一次 Java 内存泄漏的排查(14)
6. 网页实时聊天之js和jQuery实现ajax长轮询(14)
7. 从 PHP 到 Java(14)
8. shell实现SSH自动登陆(12)
9. linux的“自动化”(11)
10. Gotorch - 多机定时任务管理系统 (11)

```
ws.onerror = function () {  
};
```

发送消息也很简单，直接调用 `ws.send(msg)` 方法就行了。

页面功能

页面部分主要是让用户使用起来方便，这里给消息框 `textarea` 添加了一个键盘监控事件，当用户按下回车键时直接发送消息；

```
function confirm(event) {  
    var key_num = event.keyCode;  
    if (13 == key_num) {  
        send();  
    } else {  
        return false;  
    }  
}
```

还有用户打开客户端时生成一个默认唯一用户名；

然后是一些对数据的解析构造，对客户端页面的更新，这里就不再啰嗦了，感兴趣的可以看源码。

用户名异步处理

这里不得不提一下用户登陆时确定用户名时的小问题，我原来是想在客户端创建一个连接后直接发送用户名到服务器，可是控制台里报出了“`websocket 仍在连接中或已关闭`”的错误信息。

Uncaught DOMException: Failed to execute 'send' on 'WebSocket': Still in CONNECTING state.

考虑到连接可能还没处理好，我就实现了 `sleep` 方法等了一秒再发送用户名，可是错误仍然存在。

后来忽然想到 `js` 的单线程阻塞机制，才明白使用 `sleep` 一直阻塞也是没有用的，利用好 `js` 的事件机制才是正道：于是在服务器端添加逻辑，在握手成功后，向客户端发送握手已成功的信息；客户端先将用户名存入一个全局变量，接收到服务器的握手成功的提醒消息后再发送用户名，于是成功在第一时间更新用户名。

小结

聊天室扩展方向

简易聊天室已经完成，当然还要给它带有希望的美好未来，希望有人去实现：

- 页面美化（信息添加颜色等）
- 服务器识别 '@' 字符而只向某一个 `socket` 写数据实现聊天室的私聊；
- 多进程（使用 `redis` 等缓存数据库来实现资源的共享），可参考我以前的一篇文章：[初探PHP多进程](#)
- 消息记录数据库持久化（`log` 日志还是不方便分析）
- ...

总结

多读些经典书籍还是很有用的，有些东西真的是触类旁通，`APUE/UNP` 还是要再多翻几遍。此外互联网技术日新月异，挑一些自己喜欢的学习一下，跟大家分享一下也是挺舒服的（虽然程序和博客加一块用了至少10个小时...）。

参考：

[websocket协议翻译](#)

[刨根问底 HTTP 和 WebSocket 协议（下）](#)

[学习WebSocket协议—从顶层到底层的实现原理（修订版）](#)

嗯，持续更新。喜欢的可以点个推荐或关注，有错漏之处，请指正，谢谢。


分类： PHP

标签： websocket， PHP， 聊天室

好文要顶

关注我

收藏该文



枕边书
关注 - 24
粉丝 - 446

推荐博客
+加关注

19

0

« 上一篇： PHP中的回调函数和匿名函数

» 下一篇： WEB开发中的字符集和编码

posted @ 2016-11-28 20:43 枕边书 阅读(73133) 评论(63) 编辑 收藏

评论列表

#51楼 2018-03-29 10:46 雪剑无影

请问有源码 吗，分享一下

支持(0)

反对(0)

#52楼 [楼主] 2018-03-29 10:48 枕边书

@ 雪剑无影
正文中有链接的。。。
<https://github.com/zhenbianshu/websocket>

支持(0) 反对(0)

#53楼 2018-03-29 11:52 雪剑无影

@ 枕边书
谢谢分享

支持(0) 反对(0)

#54楼 2018-06-21 09:02 银酱

感谢楼主的无私分享

支持(0) 反对(0)

#55楼 2018-07-04 11:36 deelian

这个站
<http://www.ebolaunion.gq/>
右下角那个东东，原理是不是和你的一样的？

支持(0) 反对(0)

#56楼 2018-07-04 11:37 deelian

@ 雪剑无影
看下这个站右下角的demo
<http://www.ebolaunion.gq/>

支持(0) 反对(0)

#57楼 2018-07-23 15:32 小白大仁

这个new WebSocket("ws://127.0.0.1:8080"); 是怎么配的？是配到server.php的路径吗？

支持(0) 反对(0)

#58楼 [楼主] 2018-07-23 15:36 枕边书

@ 小白大仁
跟 server.php 没关系，配的你选择的本地IP和端口。

支持(0) 反对(0)

#59楼 2018-07-23 15:43 小白大仁

@ 枕边书
那server.php要不要运行？

支持(0) 反对(0)

#60楼 [楼主] 2018-07-23 15:47 枕边书

@ 小白大仁
server.php 就是简易版的 nginx 服务器，当然要运行。

支持(0) 反对(0)

#61楼 2018-11-09 14:42 失忆王子

您好，需要nginx配置端口吗，我配置了nginx的8080，socket就bind就报错了，怎么让握手指到server.php端口

支持(0) 反对(0)

#62楼 2018-11-09 14:46 失忆王子

是否可以给个微信号，希望能请教您问题

支持(0) 反对(0)

#63楼 2019-09-18 11:11 dreameng~

谢谢分享

支持(0) 反对(0)

注册用户登录后才能发表评论，请 登录 或 注册 ， 访问 网站首页。

- 【推荐】超50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库
- 【推荐】开发者上云福利，腾讯云1核4G云服务器11元/月起
- 【推荐】开年盛典，百度智能云1核1G云服务器84元/年

【推荐】像阿里巴巴一样远程办公，这些资源+工具+方案都好用

【推荐】疫情期间如何高效工作？ 阿里巴巴为你提供办公工具和资源

相关博文：

- 网页实时聊天之js和jQuery实现ajax长轮询
 - Nodejs之WebSocket
 - 通过(Node Js|.Net)基于HTML5的WebSocket实现实时视频文字传输(上)
 - WebSocket实现Java后台消息推送
 - js--websocket入门
- » 更多推荐...

疫情期间如何高效工作？ 阿里巴巴为你提供办公工具和资源！

最新 IT 新闻:

- 战疫之下的“空中哨兵们”
 - 俄罗斯自研CPU受阻 说好的8000元一台的采购黄了
 - 速升Android 10，黑客可通过旧版安卓系统蓝牙发送恶意软件
 - 神秘水星迎绝佳观测时机，10日上演今年首场大距
 - 云网络安全创业公司「Netskope」获 3.4 亿美元融资
- » 更多新闻...