

W3Cschool

编程入门教程 开源文档 编程实例 速查词典 问答 更多

输入关键字搜索...

首页 HTML CSS Javascript JQuery Bootstrap PHP Java Python MySQL SQL Json 微信小程序 在线工具 极客导航 更多

目录

HTML5 教程

开始学习HTML5

HTML5 浏览器支持

HTML5 新元素

HTML5 Canvas

HTML5 内联 SVG

HTML5 MathML

HTML5 拖放

HTML5 地理定位

HTML5 Video(视频)

HTML5 Audio(音频)

HTML5 Input 类型

HTML5 表单元素

HTML5 表单属性

HTML5 语义元素

HTML5 Web 存储

HTML5 Web SQL

HTML5 应用程序缓存

HTML5 Web Workers

HTML5 SSE

HTML5 WebSocket

HTML5 代码规范

HTML5 参考手册

HTML5 标签

HTML5 <!DOCTYPE> ...

HTML5 <a> 标签

HTML5 <area> 标签

HTML5 <abbr> 标签

HTML5 <applet> 标签

HTML5 <article>标签

HTML5 <aside> 标签

HTML5 <address> 标签

HTML5 <acronym> 标签

HTML5 标签

HTML5 <basefont> 标签

HTML5 <base> 标签

HTML5 <blockquote> ...

HTML5
 标签

HTML5 <big> 标签

HTML5 <button> 标签

HTML5 <bdo> 标签

HTML5 <center> 标签

HTML5 <command> ...

HTML5 <col> 标签

HTML5 <colgroup> 标签

HTML5 <caption> 标签

HTML5 <strong...

HTML5 标签

HTML5 <dl> 标签

HTML5 标签

HTML5 <dd> 标签

HTML5 <details> 标签

HTML5 <div> 标签

HTML5 <dir> 标签

HTML5 <embed> 标签

HTML5 <frameset> 标签

HTML5 <figcaption> ...

HTML5 <form> 标签

HTML5 <frame> 标签

HTML5 <figure> 标签

HTML5 <footer> 标签

HTML5 教程

开始学习HTML5

HTML5 浏览器支持

HTML5 新元素

HTML5 Canvas

HTML5 内联 SVG

HTML5 MathML

HTML5 拖放

HTML5 地理定位

HTML5 Video(视频)

HTML5 Audio(音频)

HTML5 Input 类型

HTML5 表单元素

HTML5 表单属性

HTML5 语义元素

HTML5 Web 存储

HTML5 Web SQL

HTML5 应用程序缓存

HTML5 Web Workers

HTML5 SSE

HTML5 WebSocket

HTML5 代码规范

HTML5 参考手册

HTML5 标签

HTML5 <!DOCTYPE> ...

HTML5 <a> 标签

HTML5 <area> 标签

HTML5 <abbr> 标签

HTML5 <applet> 标签

HTML5 <article>标签

HTML5 <aside> 标签

HTML5 <address> 标签

HTML5 <acronym> 标签

HTML5 标签

HTML5 <basefont> 标签

HTML5 <base> 标签

HTML5 <blockquote> ...

HTML5
 标签

HTML5 <big> 标签

HTML5 <button> 标签

HTML5 <bdo> 标签

HTML5 <center> 标签

HTML5 <command> ...

HTML5 <col> 标签

HTML5 <colgroup> 标签

HTML5 <caption> 标签

HTML5 <strong...

HTML5 标签

HTML5 <dl> 标签

HTML5 标签

HTML5 <dd> 标签

HTML5 <details> 标签

HTML5 <div> 标签

HTML5 <dir> 标签

HTML5 <embed> 标签

HTML5 <frameset> 标签

HTML5 <figcaption> ...

HTML5 <form> 标签

HTML5 <frame> 标签

HTML5 <figure> 标签

HTML5 <footer> 标签

← HTML5 SSE

HTML5 代码规范 →

当前阅读

不用敲代码，快速构建数据应用

我要纠错

HTML5 WebSocket

阅读 (11765) 收藏 赞 (3) 分享 新版预览

HTML5 WebSocket

WebSocket是HTML5开始提供的一种在单个 TCP 连接上进行全双工通讯的协议。

在WebSocket API中，浏览器和服务器只需要做一个握手的动作，然后，浏览器和服务器之间就形成了一条快速通道。两者之间就直接可以数据互相传送。

浏览器通过 JavaScript 向服务器发出建立 WebSocket 连接的请求，连接建立以后，客户端和服务器端就可以通过 TCP 连接直接交换数据。

当你获取 Web Socket 连接后，你可以通过 **send()** 方法来向服务器发送数据，并通过 **onmessage** 事件来接收服务器返回的数据。

以下 API 用于创建 WebSocket 对象。

```
var Socket = new WebSocket(url, [protocol] );
```

以上代码中的第一个参数 url, 指定连接的 URL。第二个参数 protocol 是可选的，指定了可接受的子协议。

WebSocket 属性

以下是 WebSocket 对象的属性。假定我们使用了以上代码创建了 Socket 对象：

属性	描述
Socket.readyState	只读属性 readyState 表示连接状态，可以是以下值： <ul style="list-style-type: none">0 - 表示连接尚未建立。1 - 表示连接已建立，可以进行通信。2 - 表示连接正在进行关闭。3 - 表示连接已经关闭或者连接不能打开。
Socket.bufferedAmount	只读属性 bufferedAmount 已被 send() 放入正在队列中等待传输，但是还没有发出的 UTF-8 文本字节数。

WebSocket 事件

以下是 WebSocket 对象的相关事件。假定我们使用了以上代码创建了 Socket 对象：

事件	事件处理程序	描述
open	Socket.onopen	连接建立时触发
message	Socket.onmessage	客户端接收服务端数据时触发
error	Socket.onerror	通信发生错误时触发
close	Socket.onclose	连接关闭时触发

WebSocket 方法

以下是 WebSocket 对象的相关方法。假定我们使用了以上代码创建了 Socket 对象：

方法	描述
Socket.send()	使用连接发送数据

扩展栏目

html5问答讨论

HTML教程

HTML参考手册

HTML5编程入门

开发工具IDE

编程训练营

Advertisement

全国30座城市140个甲级写字楼办公室任你选

Regus雷格斯服务式办公

分类导航

最新推荐

HTML/CSS

javascript

HTML5 <fieldset> 标签
HTML5 <hgroup> 标签
HTML5 <hr> 标签
HTML5 <h1> 至 <h6>...
HTML5 <header> 标签
HTML5 <html> 标签
HTML5 <head> 标签
HTML5 <i> 标签
HTML5 <iframe> 标签
HTML5 <input> 标签
HTML5 标签
HTML5 <ins> 标签
HTML5 <keygen> 标签
HTML5 <link> 标签
HTML5 标签
HTML5 <legend> 标签
HTML5 <label> 标签
HTML5 <map> 标签
HTML5 <meta> 标签
HTML5 <menu> 标签
HTML5 <meter> 标签
HTML5 <mark> 标签
HTML5 <nav> 标签
HTML5 <noscript> 标签
HTML5 标签
HTML5 <object> 标签
HTML5 <option> 标签
HTML5 <output> 标签
HTML5 <optgroup> 标签
HTML5 <p> 标签
HTML5 <pre> 标签
HTML5 <param> 标签
HTML5 <progress> 标签
HTML5 <q> 标签
HTML5 <rp> 标签
HTML5 <rt> 标签
HTML5 <ruby> 标签
HTML5 <s> 标签
HTML5 <sub> 和 <sup>...
HTML5 标签
HTML5 <small> 标签
HTML5 <style> 标签
HTML5 <strike> 标签
HTML5 <select> 标签
HTML5 <source> 标签
HTML5 <section> 标签
HTML5 <summary> 标...
HTML5 <td> 标签
HTML5 <th> 标签
HTML5 <tt> 标签
HTML5 <tr> 标签
HTML5 <table> 标签
HTML5 <title> 标签
HTML5 <tfoot> 标签
HTML5 <thead> 标签
HTML5 <time> 标签
HTML5 <tbody> 标签
HTML5 <textarea> 标签

拓展阅读

HTML5电子书下载

HTML5开发工具

HTML5小游戏源码

HTML5模板下载

HTML5 video

Socket.close()	关闭连接
<h3>WebSocket 实例</h3> <p>WebSocket 协议本质上是一个基于 TCP 的协议。</p> <p>为了建立一个 WebSocket 连接，客户端浏览器首先要向服务器发起一个 HTTP 请求，这个请求和通常的 HTTP 请求不同，包含了一些附加头信息，其中附加头信息"Upgrade: WebSocket"表明这是一个申请协议升级的 HTTP 请求，服务器端解析这些附加的头信息然后产生应答信息返回给客户端，客户端和服务器的 WebSocket 连接就建立起来了，双方就可以通过这个连接通道自由的传递信息，并且这个连接会持续存在直到客户端或者服务器的某一方主动的关闭连接。</p> <h3>客户端的 HTML 和 JavaScript</h3> <p>目前大部分浏览器支持 WebSocket() 接口，你可以在以下浏览器中尝试实例： Chrome, Mozilla, Opera 和 Safari。</p> <p>w3cschool_websocket.html 文件内容</p> <pre><!DOCTYPE HTML> <html> <head> <meta charset="utf-8"> <title>W3Cschool教程(w3cschool.cn)</title> <script type="text/javascript"> function WebSocketTest() { if ("WebSocket" in window) { alert("您的浏览器支持 WebSocket!"); // 打开一个 web socket var ws = new WebSocket("ws://localhost:9998/echo"); ws.onopen = function() { // Web Socket 已连接上，使用 send() 方法发送数据 ws.send("发送数据"); alert("数据发送中..."); }; ws.onmessage = function (evt) { var received_msg = evt.data; alert("数据已接收..."); }; ws.onclose = function() { // 关闭 websocket alert("连接已关闭..."); }; } else { // 浏览器不支持 WebSocket alert("您的浏览器不支持 WebSocket!"); } } </script> </head> <body> <div id="sse"> 运行 WebSocket </div> </body> </html></pre>	

服务端
数据库
移动端
ASP.NET
Web Service
开发工具

安装 pywebsocket

在执行以上程序前，我们需要创建一个支持 WebSocket 的服务。从 [pywebsocket](#) 下载 mod_pywebsocket ,或者使用 git 命令下载：

```
git clone https://github.com/google/pywebsocket.git
```

mod_pywebsocket 需要 python 环境支持

mod_pywebsocket 是一个 Apache HTTP 的 Web Socket扩展，安装步骤如下：

- 解压下载的文件。
- 进入 **pywebsocket** 目录。
- 执行命令：

```
$ python setup.py build
$ sudo python setup.py install
```

- 查看文档说明：

```
$ pydoc mod_pywebsocket
```

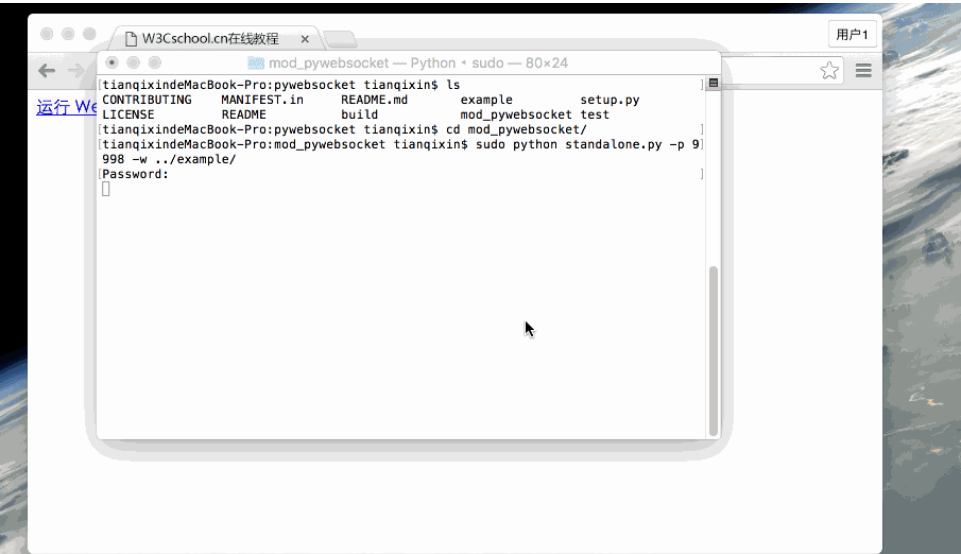
开启服务

在 **pywebsocket/mod_pywebsocket** 目录下执行以下命令：

```
$ sudo python standalone.py -p 9998 -w ../example/
```

以上命令会开启一个端口号为 9998 的服务，使用 -w 来设置处理程序 echo_wsh.py 所在的目录。

现在我们可以 在 Chrome 浏览器打开前面创建的 w3cschool_websocket.html 文件。如果你的浏览器支持 WebSocket(), 点击"运行 WebSocket", 你就可以看到整个流程各个步骤弹出的窗口，流程 Gif 演示：



在我们停止服务后，会弹出 "连接已关闭..."。

网站相关

- 关于我们
- 加入我们
- 免责声明
- 网站地图

内容专区

- 编程入门教程
- 开源文档
- 在线实例
- 文章存档

常用链接

- 使用帮助
- 极客导航
- 编程百科
- w3cschool手机版

联系我们

- 官方QQ群：496712525
- 商务合作QQ：3265790599
- 友情链接QQ：3532917531



微信号：w3cschool