



SpringBoot入门 (十三) WebSocket使用

本文记录在SpringBoot中使用WebSocket。

一 什么是WebSocket

WebSocket是基于TCP协议的一种网络协议，它实现了浏览器与服务器全双工通信，支持客户端和服务端之间相互发送信息。在有WebSocket之前，如果服务端数据发生了改变，客户端想知道的话，只能采用定时轮询的方式去服务端获取，这种方式很大程度上增大了服务器端的压力，有了WebSocket之后，如果服务端数据发生改变，可以立即通知客户端，客户端就不用轮询去换取，降低了服务器的压力。目前主流的浏览器都已经支持WebSocket协议了。

WebSocket使用ws和wss作资源标志符，它们两个类似于http和https，wss是使用TSL的ws。主要有4个事件：

- onopen 创建连接时触发
- onclose 连接断开时触发
- onmessage 接收到信息时触发
- onerror 通讯异常时触发

二 简单使用示例

SpringBoot对WebSocket也做了支持，需要使用的话引入依赖所需要的包spring-boot-starter-websocket就可以了。我们利用它可以双向通信的特性来实现一个简单的聊天室功能。主要功能如下

- 1 用户在浏览器端进入聊天室(创建WebSocket连接);
- 2 用户端发送消息到服务端(客户端像服务端发信息);
- 3 服务端将消息转发到客户端(服务端向客户端发信息);
- 4 用户退出聊天室(断开WebSocket连接)。

修改pom.xml文件，引入WebSocket的支持

```
1 <dependency>
2     <groupId>org.springframework.boot</groupId>
3     <artifactId>spring-boot-starter-web</artifactId>
4 </dependency>
5 <dependency>
6     <groupId>org.springframework.boot</groupId>
7     <artifactId>spring-boot-starter-websocket</artifactId>
8 </dependency>
```

修改application.properties文件，配置服务相关信息

```
1 server.port=8090
2 server.servlet.context-path=/websocket
```

要使用WebSocket，我们需要在启动来开启对WebSocket的支持功能，使用@EnableWebSocket注解

公告

昵称： 梓&艺
园龄： 1年11个月
粉丝： 50
关注： 4
[+加关注](#)

< 2020年12月 >						
日	一	二	三	四	五	六
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

积分与排名

积分 - 23679

排名 - 42051

随笔分类

spring(2)

springboot入门实战(14)

常见问题(1)

设计模式(5)

杂七杂八(2)

随笔档案

2019年4月(2)

2019年2月(6)

2019年1月(10)

最新评论

1. Re:SpringBoot入门 (十三) WebSocket 使用

文章很棒，为java实现websocket提供了很好的技术思路。不过在正式的项目中，我一般是用第三方websocket框架来做，稳定性、实时性有保证的多。【GoEasy】算是比较不错的一个解决方案。...

--EmmaLan

2. Re:SpringBoot入门 (十三) WebSocket 使用

@ 队长给我球。 不应该吧，我用的2.0.5.RELEASE...

--梓&艺

3. Re:SpringBoot入门 (九) MQ使用

对于纯小白而言，RabbitMQ还以为直接导入jar包就可以用了，结果全部代码写好，才发现连接失败，根本就还没有服务器啊，怎么会连接成功呢？（手动滑稽）

--捆板牛肉汤

4. Re:SpringBoot入门 (十三) WebSocket 使用

博主 用的是springboot 哪个版本？
我是1.5.8release 但是缺提示找不到@EnableWebSocket 这个类？

--队长给我球。

5. Re:SpringBoot入门 (九) MQ使用

@ 努力的小白菜SpringBoot使用得还是Spring的东西，只是把与其他框架的整合统一了处理，方便开发者使用。...

--梓&艺

```
1 @SpringBootApplication
2 @EnableWebSocket
3 public class WebSocketApplication {
4
5     public static void main(String[] args) {
6         SpringApplication.run(WebSocketApplication.class, args);
7     }
8
9     /**
10      * 初始化Bean，它会自动注册使用了 @ServerEndpoint 注解声明的 WebSocket endpoint
11      * @return
12      */
13     @Bean
14     public ServerEndpointExporter serverEndpointExporter(){
15         return new ServerEndpointExporter();
16     }
17
18 }
```

聊天时，我们需要知道有哪些用户在线，所以我们创建一个工具类来记录在线用户和实现服务端向客户端发送消息

```
1 public class WebSocketUtil {
2
3     /**
4      * 模拟存储    在线用户
5      */
6     public static final Map<String, Session> USERS_ONLINE = new ConcurrentHashMap<>();
7
8     /**
9      * 向所有在线用户发送消息(遍历 向每一个用户发送)
10      * @param message
11      */
12     public static void sendMessageToAllOnlineUser(String message){
13         USERS_ONLINE.forEach((username, Session) -> sendMessage(Session, message));
14     }
15
16     /**
17      * 向指定用户发送消息
18      * @param session 用户session
19      * @param message 发送消息内容
20      */
21     private static void sendMessage(Session session, String message) {
22         if (session == null) {
23             return;
24         }
25
26         final RemoteEndpoint.Basic basic = session.getBasicRemote();
27         if (basic == null) {
28             return;
29         }
30
31         try {
32             basic.sendText(message);
33         } catch (IOException e) {
34             e.printStackTrace();
35         }
36     }
37
38 }
```

服务端类，使用@ServerEndpoint("")说明服务端监听此地址的消息

```
1 @Controller
2 @ServerEndpoint("/chat/{username}") //说明创建websocket的endpoint
3 public class ChatServerEndpoint {
4
5     /**
6      * 访问聊天室页面
7      * @return
8      */
9     @GetMapping("/chatPage")
10     public String chatPage(){
11         return "chat.html";
12     }
13
14 }
```

WebSocket有4个事件，我们对每个事件做监听，使用对应的注解即可实现监听

onopen 在连接创建(用户进入聊天室)时触发

```
1  @OnOpen
2      public void openSession(@PathParam("username") String username, Session session){
3          //存储用户
4          WebSocketUtil.USERS_ONLINE.put(username, session);
5          //向所有在线用户发送用户上线通知消息
6          String message = "["+username+"]进入聊天室";
7          System.out.println(message);
8          WebSocketUtil.sendMessageToAllOnlineUser(message);
9      }
```

onclose 在连接断开(用户离开聊天室)时触发

```
1  @OnClose
2      public void closeSession(@PathParam("username") String username, Session session){
3          //删除用户
4          WebSocketUtil.USERS_ONLINE.remove(username);
5          //向所有在线用户发送用户下线通知消息
6          String message = "["+username+"]离开了聊天室";
7          System.out.println(message);
8          WebSocketUtil.sendMessageToAllOnlineUser(message);
9          //下线后关闭session
10         try {
11             session.close();
12         } catch (IOException e) {
13             e.printStackTrace();
14         }
15     }
```

onmessage 在接收到消息时触发

```
1  @OnMessage
2      public void onMessage(@PathParam("username") String username, String message){
3          //向聊天室中的人发送消息
4          message = "["+username+"]: " + message;
5          System.out.println(message);
6          WebSocketUtil.sendMessageToAllOnlineUser(message);
7      }
```

onerror 在连接发生异常时触发

```
1  @OnError
2      public void sessionError(Session session, Throwable throwable){
3          try {
4              session.close();
5          } catch (IOException e) {
6              e.printStackTrace();
7          }
8          System.out.println("WebSocket连接发生异常, message:"+throwable.getMessage());
9      }
```

聊天室页面

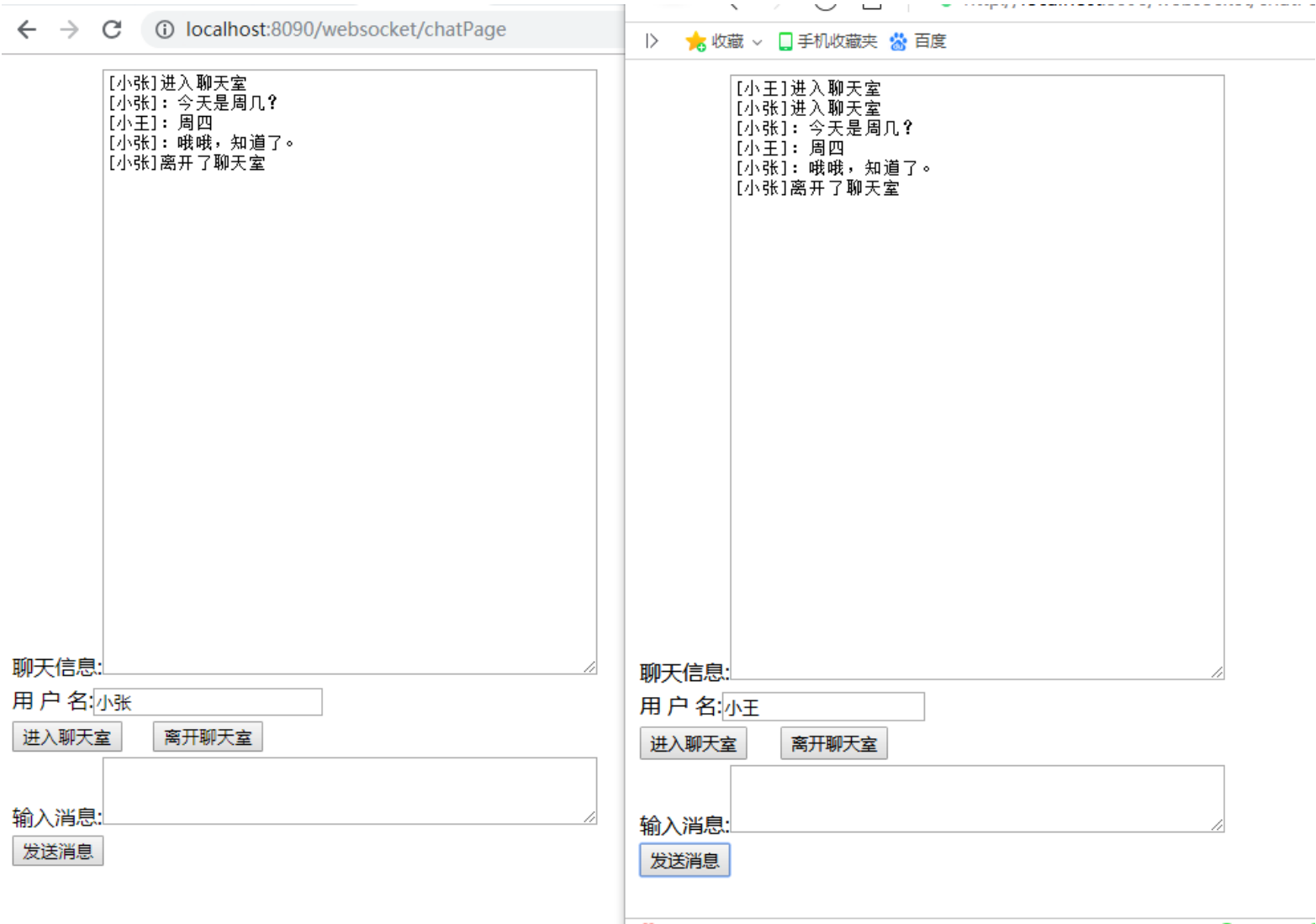
```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <script src="jquery-1.9.1.min.js"></script>
7  </head>
8  <body>
9  <table>
10     <tr><td>
11         <label for="messageArea">聊天信息:</label><textarea id="messageArea" cols="50" rows="30"></textarea></br>
12     </td></tr>
13
14     <tr><td>
15         <label for="username">用 户 名:</label><input type="text" id="username" /></br>
16     </td></tr>
17
18     <tr><td>
19         <input type="button" id="joinRoomBtn" value="进入聊天室" />
20         <input type="button" id="leaveRoomBtn" value="离开聊天室" /></br>
21     </td></tr>
22
23     <tr><td>
24         <label for="sendMessage">输入消息:</label><textarea id="sendMessage" cols="50" rows="3"></textarea></br>
25     </td></tr>
26
```

```
27      <tr><td>
28          <input type="button" id="sendBtn" value="发送消息" />
29      </td></tr>
30  </table>
31
32  </body>
33  </html>
```

WebSocket客户端连接服务端及相关事件方法

```
1  <script>
2      $(function(){
3          var websocket;
4          var url = 'ws://localhost:8090/websocket/chat/';
5          //进入聊天室
6          $('#joinRoomBtn').click(function(){
7              var username = $('#username').val();
8              websocket = new WebSocket(url+username);
9              websocket.onopen = function (){
10                  console.log('websocket连接创建。。。');
11              }
12              websocket.onclose = function(){
13                  $('#messageArea').append([''+username+'离开了聊天室\n']);
14              }
15              websocket.onmessage = function(event){
16                  $('#messageArea').append(event.data + '\n');
17              }
18              websocket.onerror = function (event) { console.log('websocket连接异常。。。');
19              }
20          });
21          //退出聊天室
22          $('#leaveRoomBtn').click(function(){
23              if(websocket){
24                  //关闭连接
25                  websocket.close();
26              }
27          });
28          //发送消息
29          $('#sendBtn').click(function(){
30              var msg = $('#sendMessage').val();
31              websocket.send(msg);
32              $('#sendMessage').val('');
33          });
34      });
35  </script>
```

启动项目，访问聊天室页面，先输入用户名加入聊天室，再发送消息及退出聊天室，为了效果，打开两个浏览器，如下图



由于个人能力有限，难免有错误之处，敬请读者指正，不胜感激。

分类: [springboot入门实战](#)

好文要顶

关注我

收藏该文

梓&艺

关注 - 4

粉丝 - 50

2

推荐

0

反对

+加关注

« 上一篇: [SpringBoot入门 \(十二\) 定时任务](#)

» 下一篇: [SpringBoot入门 \(十四\) Security安全控制](#)

posted @ 2019-02-15 10:00 梓&艺 阅读(4688) 评论(3) 编辑 收藏

评论列表

- #1楼 2019-02-18 15:59 队长给我球。

博主 用的是springboot 哪个版本？
我是1.5.8release 但是缺提示找不到@EnableWebSocket 这个类？

支持(0) 反对(0)
- #2楼 [楼主] 2019-02-22 09:01 梓&艺

@ 队长给我球。
不应该吧，我用的2.0.5.RELEASE

支持(0) 反对(0)
- #3楼 2020-06-16 16:48 EmmaLan

文章很棒，为java实现websocket提供了很好的技术思路。不过在正式的项目中，我一般是用第三方websocket框架来做，稳定性、实时性有保证的多。【GoEasy】算是比较不错的一个解决方案。

支持(0) 反对(2)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

登录后才能发表评论，立即 [登录](#) 或 [注册](#)， [访问](#) [网站首页](#)

- 【推荐】News: 大型组态、工控、仿真、CADGIS 50万行VC++源码免费下载
- 【推荐】博客园 & 陌上花开HIMMR 给单身的程序员小哥哥助力脱单啦~
- 【推荐】了不起的开发者，挡不住的华为，园子里的品牌专区
- 【推荐】未知数的距离，毫秒间的传递，声网与你实时互动
- 【推荐】阿里云爆款特惠，精选爆款产品低至0.95折



- 相关博文:
- [SpringBoot-Web开发](#)
 - [SpringBoot整合web三大组件](#)
 - [SpringBoot学习（四）开发web应用](#)
 - [SOCKET](#)
 - [基于springboot的web项目最佳实践](#)
 - » [更多推荐...](#)



最新 IT 新闻:

- 这些看起来很厉害的云游戏平台 居然在暗地里搞起了盗版
- 「软件生态」对平板来说到底意味着什么？
- 张文宏：艾滋病病毒比新冠病毒更狡猾
- 大神着手为M1 Mac原生移植Linux：内核之父曾放言不可能完成
- 圆圈在动吗？这个骗过人眼的动图火了
- » 更多新闻...