

游戏开发

大多数游戏 通信都是 udp

我们是做 web系统开发， websocket tcp

WebSocket是一种在单个TCP连接上进行全双工通信的协议。

WebSocket通信协议于2011年被IETF定为标准RFC 6455，并由RFC7936补充规范。WebSocket API 也被W3C定为标准。

WebSocket使得客户端和服务端之间的数据交换变得更加简单，允许服务端主动向客户端推送数据。在WebSocket API中，浏览器和服务器只需要完成一次握手，两者之间就直接可以创建持久性的连接，并进行双向数据传输。

知识点

- 服务端可以主动推消息
 - 服务端如何实现 主动推送消息
- 一次握手后状态始终保持连接
- 节省带宽和性能
- 消息吞吐量大
- websocket 消息是异步的
- 所有的消息都在一个 连接里处理

SpringBoot整合Websocket

SpringBoot 使用websocket 非常简单

因为 springboot 已经整合过websocket 了， 我们直接使用继承过来就行了

```
1 <dependency>
2     <groupId>org.springframework.boot</groupId>
3     <artifactId>spring-boot-starter-websocket</artifactId>
4 </dependency>
5
6 @Bean
7 public ServerEndpointExporter getEndpointExporter(){
8     return new ServerEndpointExporter();
9 }
10
```

WebSocket 关键点

1. open 事件 当一个客户端连接到服务端的时候， 触发

2. close 事件 当一个客户端和服务端 断开连接的时候
3. onMessage 当服务端收到一个 客户端消息的时候
4. onError 当发生异常的情况下会触发
5. 以上事件 是发生在 一个连接的情况下

实现一个DEMO案例

```
1 Gateway::sendToCurrentClient({'type':"welcome","id":'.$_SESSION['id'].'});
2 GateWay::sendToAll(json_encode(array('type'=>'closed', 'id'=>$_SESSION['id'])));
3 switch($message_data['type'])
4 {
5     case 'login':
6         break;
7     // 更新用户
8     case 'update':
9         // 转播给所有用户
10         Gateway::sendToAll(json_encode(
11             array(
12                 'type'      => 'update',
13                 'id'        => $_SESSION['id'],
14                 'angle'     => $message_data["angle"]+0,
15                 'momentum' => $message_data["momentum"]+0,
16                 'x'         => $message_data["x"]+0,
17                 'y'         => $message_data["y"]+0,
18                 'life'      => 1,
19                 'name'      => isset($message_data['name']) ? $message_data['name'] :
20 'Guest.'.$_SESSION['id'],
21                 'authorized' => false,
22             )
23         ));
24         return;
25     // 聊天
26     case 'message':
27         // 向大家说
28         $new_message = array(
```

```
28         'type'=>'message',
29         'id'  =>$_SESSION['id'],
30         'message'=>$message_data['message'],
31     );
32     return Gateway::sendToAll(json_encode($new_message));
33 }
```

需要用到的 依赖包

```
1 <dependency>
2     <groupId>org.springframework.boot</groupId>
3     <artifactId>spring-boot-starter-thymeleaf</artifactId>
4 </dependency>
5
6 <dependency>
7     <groupId>com.alibaba</groupId>
8     <artifactId>fastjson</artifactId>
9     <version>2.0.23</version>
10 </dependency>
```

thymeleaf 配置

```
1 spring.thymeleaf.mode: HTML
2 spring.thymeleaf.encoding: UTF-8
3 spring.thymeleaf.prefix: classpath:/templates/
4 spring.thymeleaf.suffix: .html
5 spring.mvc.static-path-pattern=/static/**
```