

Please visit website: <http://cxyroad.com>

SpringBoot整合WebSocket两步曲

=====

* 首先需要创建一个websocket处理器，该类需要继承TextWebSocketHandler并重写里面的方法

...

```
import org.springframework.web.socket.CloseStatus;
import org.springframework.web.socket.WebSocketMessage;
import org.springframework.web.socket.WebSocketSession;
import org.springframework.web.socket.handler.TextWebSocketHandler;
```

```
public class WebSocketHandler extends TextWebSocketHandler {

    /**
     * 连接成功调用
     */
    @Override
    public void afterConnectionEstablished(WebSocketSession session)
    throws Exception {
        super.afterConnectionEstablished(session);
    }

    /**
     * 收到消息时调用
     */
    @Override
    public void handleMessage(WebSocketSession session,
    WebSocketMessage<?> message) throws Exception {
        super.handleMessage(session, message);
    }

    /**
     * 关闭连接时调用
     */
    @Override
    public void afterConnectionClosed(WebSocketSession session,
    CloseStatus status) throws Exception {
        super.afterConnectionClosed(session, status);
    }

    /**
     * 发生错误时调用
     */
}
```

```

    */
    @Override
    public void handleTransportError(WebSocketSession session,
    Throwable exception) throws Exception {
        super.handleTransportError(session, exception);
    }
}

```

...

创建好websocket处理器后添加配置类，将websocket处理器注入容器

...

```

import org.springframework.context.annotation.Configuration;
import org.springframework.web.socket.config.annotation.*;
import com.piim.handler.ChatWebSocketHandler;

```

```

@Configuration
@EnableWebSocket
public class WebSocketConfig implements WebSocketConfigurer {

    @Override
    public void registerWebSocketHandlers(WebSocketHandlerRegistry
registry) {
        //配置在线聊天处理器
        registry.addHandler(new WebSocketHandler(), "/chat")
            .setAllowedOrigins("*");
    }
}

```

...

* 问题

1. WebSocket文本消息长度限制，如图片转的base64无法发送

原因：Http请求的大小默认大小为0.8kb

解决方法：在配置文件中添加该配置

...

#限制Tomcat接收的HTTP请求的大小，包括WebSocket消息的大小限制，修改为10m

server:

```
max-http-header-size: 10485760
```

...

2. WebSocket中无法使用mapper和service，注入为null

原因：spring 默认管理的是单例，所以只会注入一次 service。当新用户进入聊天时，系统又会创建一个新的 websocket 对象，spring 管理的都是单例，不会给第二个 websocket 对象注入 service，所以导致只要是用户连接创建的 websocket 对象，都不能再注入，mapper同理
解决方法：

在WebSocket处理器添加spring上下文对象并创建对应的set方法

...

//解决无法注入service和mapper问题

```
private static ApplicationContext applicationContext;
```

```
public static void setApplicationContext(ApplicationContext  
applicationContext) {
```

```
    WebSocketHandler.applicationContext = applicationContext;  
}
```

...

并在启动类中将spring上下文对象注入

...

//解决websocketServer无法注入mapper问题

```
SpringApplication springApplication = new  
SpringApplication(PilmApplication.class);
```

```
ConfigurableApplicationContext configurableApplicationContext =  
springApplication.run(args);
```

```
ChatWebSocketHandler.setApplicationContext(configurableApplicationC  
ontext);
```

...

通过getBean获取service和mapper对象

...

```
MessageService messageService =  
applicationContext.getBean(MessageService.class);
```

...

原文链接: <https://juejin.cn/post/7376925694612799538>