

WebSocket

Java-Code

1. Java API for WebSocket

连接WebSocket的几种方式:

- 使用 `@ServerEndpoint` 注解，并且进行配置

```
import javax.websocket.server.ServerEndpoint;
import org.springframework.stereotype.Component;

@Component
@ServerEndpoint(value = "/echoserver/echo")
public class AnnotationalEchoServer {

    @OnMessage
    public String echo(String message) {
        return "I got this (" + message + ") so I am sending it back!";
    }

    @OnOpen
    public void onOpen(Session session) throws InterruptedException
    {
        // 建立连接
    }
}

// -----另外还要添加配置类-----
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.web.socket.server.standard.ServerEndpointExporter;

@Configuration
public class WebSocketConfig{

    // 关键是要把ServerEndpointExporter交给spring管理
```

```

@Bean
public ServerEndpointExporter serverEndpointExporter() {
    return new ServerEndpointExporter();
}

```

- 使用注册器

```

import javax.websocket.Endpoint;
import javax.websocket.EndpointConfig;
import javax.websocket.MessageHandler;
import javax.websocket.Session;
import java.io.IOException;

public class ProgrammaticEchoServer extends Endpoint {

    @Override
    public void onOpen(Session session, EndpointConfig config) {
        final Session s = session;
        s.addMessageHandler(new MessageHandler.Whole<String>() {

            @Override
            public void onMessage(String message) {
                try {
                    s.getBasicRemote().sendText("I got this(" +
message + ") so I am sending it back!");
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });
    }
}

// -----另外还要添加配置类-----
import
cn.com.blumoon.material.delivery.electric.weighter.provider.websock
et.ProgrammaticEchoServer;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import
org.springframework.web.socket.server.standard.ServerEndpointExporte
r;

```

```
import
org.springframework.web.socket.server.standard.ServerEndpointRegistr
ation;

@Configuration
public class WebSocketConfig{

    // 关键是要把ServerEndpointExporter交给spring管理
    @Bean
    public ServerEndpointExporter serverEndpointExporter() {
        return new ServerEndpointExporter();
    }

    @Bean
    public ServerEndpointRegistration echoEndpoint() {
        return new ServerEndpointRegistration("/echo",
        ProgrammaticEchoServer.class);
    }
}
```

2. Spring WebSocket API

- **HandshakeInterceptor**

- 此接口包含2个方法： `beforeHandshake` 和 `afterHandshake`，可实现对握手动作的操作；

- **WebSocketHandler**

- 此接口包含了对连接建立后、连接断开后、传输失败、数据处理等操作；
- 可以考虑继承其子类： `AbstractWebSocketHandler`

3. 参数传递

- url中key=value形式

```
"ws:localhost:82/chat/123?username=ydh";
```

- 获取方式

- `session.getQueryString();`
- `session.getRequestURI().toString();`

- 路径参数

```
// 类注解
@ServerEndpoint("/chat/{param}")

// 方法注解
@OnOpen
public void open(Session session, @PathParam("param")String
param) {
    System.out.println("用户"+param+" 登录");
}
```

- 获取方式

- session.getPathParameters()
- @PathParam