

便后期的分布式扩展。在后期，通信服务器需要从单体服务器扩展到多节点服务集群。为了在分布式集群的应用场景下，在后期 sessionId 可以确保全局唯一，使用 String 类型就比较方便。

### 9.5.3 SessionMap 会话管理器

一台服务器需要接受几万/几十万的客户端连接。每一条连接都对应到一个 ServerSession 实例，服务器需要对这些大量的 ServerSession 实例进行管理。

SessionMap 负责管理服务器端所有的 ServerSession。它有一个线程安全的 ConcurrentHashMap 类型的成员 map，保持 sessionId 到服务器端 ServerSession 的映射。

```
package com.crazymakercircle.imServer.server;
//...
@Slf4j
@Data
public final class SessionMap {
    private ConcurrentHashMap<String, ServerSession> map =
        new ConcurrentHashMap<String, ServerSession>();
    //增加会话对象
    public void addSession(String sessionId, ServerSession s) {
        map.put(sessionId, s);
        log.info("用户登录:id= " + s.getUser().getUid()
            + " 在线总数: " + map.size());
    }
    //获取会话对象
    public ServerSession getSession(String sessionId) {
        if (map.containsKey(sessionId)) {
            return map.get(sessionId);
        } else {
            return null;
        }
    }
    //删除会话
    public void removeSession(String sessionId) {
        if (!map.containsKey(sessionId)) {
            return;
        }
        ServerSession s = map.get(sessionId);
        map.remove(sessionId);
        Print.tcfo("用户下线:id= " + s.getUser().getUid()
            + " 在线总数: " + map.size());
    }
    //.....省略不是太重要的方法
}
```

在调用 ServerSession.bind() 绑定时，在 SessionMap 中增加一次映射：

```
public ServerSession bind() {
    //...
    SessionMap inst().addSession(getSessionId(), this);
    //...
}
```

在调用 ServerSession.unbind() 解除绑定时，在 SessionMap 中减少一次映射：