### 完整代码行号解析（1-656行）

#### ​****第1-9行：依赖库导入****​

* ​**功能**​：导入程序运行所需的核心库和自定义模块
* ​**关键内容**​：
  1. 异步框架：asyncio
  2. WebSocket通信库：websockets
  3. JSON数据处理：json
  4. 时间处理：time和datetime
  5. 日志系统：logging
  6. 自定义数据库服务模块：包含5个核心服务类
  7. 类型注解支持：用于代码静态检查

#### ​****第11-16行：日志系统配置****​

* ​**功能**​：建立日志记录机制
* ​**配置参数**​：
  + 日志级别：INFO
  + 日志格式：时间戳-模块名-日志级别-消息内容
  + 全局日志记录器初始化

#### ​****第18-31行：ChatServer类初始化****​

* ​**功能**​：服务器核心组件初始化
* ​**创建对象**​：
  + 数据库管理器
  + 认证服务
  + 好友服务
  + 群组服务
  + 消息服务
* ​**核心数据结构**​：
  + 在线用户字典（用户ID→WebSocket连接）
  + 用户群组订阅字典

#### ​****第33-39行：JSON序列化辅助方法****​

* ​**功能**​：处理JSON序列化中的datetime对象
* 将datetime对象转换为ISO格式字符串

#### ​****第41-51行：在线用户广播****​

* ​**功能**​：向所有用户广播当前在线用户列表
* 从数据库查询用户信息
* 构建广播消息格式
* 遍历在线用户发送

#### ​****第53-80行：连接处理主循环****​

* ​**功能**​：客户端连接生命周期管理
* ​**核心流程**​：
  1. 新连接日志记录
  2. 消息接收循环
  3. JSON消息解析
  4. 根据action类型路由处理
  5. 异常捕获和处理
  6. 连接断开清理

#### ​****第82-88行：错误处理****​

* ​**功能**​：统一错误消息格式
* 标准化错误响应结构
* 简化错误发送流程

#### ​****第90-116行：用户注册处理****​

* ​**流程**​：
  1. 获取用户名密码
  2. 调用认证服务注册
  3. 注册成功加入在线用户
  4. 广播在线用户更新
  5. 发送注册响应

#### ​****第118-161行：用户登录处理****​

* ​**流程**​：
  1. 用户凭证验证
  2. 加入在线用户字典
  3. 广播用户状态变更
  4. 推送离线消息
  5. 发送好友和群组列表
  6. 广播在线用户更新

#### ​****第163-183行：社交关系数据获取****​

* ​**包含功能**​：
  + 好友列表推送
  + 群组列表推送
* ​**流程**​：
  1. 调用相应服务获取数据
  2. 构建标准化响应
  3. 定向发送给用户

#### ​****第185-201行：连接断开处理****​

* ​**清理流程**​：
  1. 从在线用户字典移除
  2. 清除订阅关系
  3. 通知好友下线状态
  4. 广播在线用户更新

#### ​****第203-216行：好友状态通知****​

* ​**功能**​：通知好友用户在线状态变化
* 获取好友列表
* 遍历在线好友发送状态变更

#### ​****第218-242行：定向消息发送****​

* ​**安全机制**​：
  1. 检查连接有效性
  2. 异常处理
  3. 关闭连接清理
* 使用自定义JSON序列化

#### ​****第244-269行：离线消息处理****​

* ​**功能**​：用户登录后推送离线消息
* ​**包含**​：
  + 私聊消息获取
  + 群组消息获取
  + 消息格式化发送

#### ​****第271-356行：数据查询服务****​

* ​**包含功能**​：
  + 好友列表获取
  + 群组列表获取
  + 消息历史查询
* ​**消息历史处理**​：
  + 分页支持
  + 发送者信息补充
  + 响应格式化

#### ​****第358-473行：消息系统核心****​

* ​**消息发送主流程**​：
  1. 参数验证
  2. 消息持久化存储
  3. 根据接收类型路由
* ​**私聊处理**​：
  + 双向消息发送（发送方/接收方）
  + 方向标识添加
* ​**群聊处理**​：
  + 群成员获取
  + 发送者排除
  + 群广播

#### ​****第475-512行：群组创建****​

* ​**流程**​：
  1. 参数验证
  2. 调用群组服务创建
  3. 通知创建者
  4. 通知初始成员
  5. 更新群组列表

#### ​****第514-586行：好友系统****​

* ​**包含功能**​：
  + 好友请求发送
  + 好友请求接受/拒绝
* ​**通知机制**​：
  + 请求发送通知
  + 请求处理结果通知双方

#### ​****第588-627行：用户资料管理****​

* ​**昵称更新流程**​：
  1. 数据库更新
  2. 用户自身通知
  3. 好友通知
* 实时更新所有相关方

#### ​****第629-646行：待处理请求获取****​

* ​**功能**​：获取待处理的好友请求
* 调用好友服务查询
* 结果格式化返回

#### ​****第649-656行：服务器启动入口****​

* ​**启动流程**​：
  1. 创建服务器实例
  2. 绑定0.0.0.0:8765监听
  3. 日志记录启动信息
  4. 启动异步事件循环
  5. 永久运行

### 系统架构特点总结

​**核心机制**​：

* + 基于WebSocket的实时双向通信
  + 异步非阻塞处理模型
  + JSON格式消息协议
  + 动作类型(action)路由机制

​**功能模块**​：

* + 用户认证管理（注册/登录/状态）
  + 实时消息系统（私聊/群聊/离线）
  + 社交关系管理（好友/群组）
  + 数据查询服务
  + 系统状态广播

​**错误处理**​：

* + 全局异常捕获
  + 连接异常自动清理
  + 统一错误响应格式
  + 详细日志记录

​**扩展性设计**​：

* + 模块化服务架构
  + 松耦合组件设计
  + 标准动作路由接口
  + 类型注解增强可维护性

整个系统采用分层架构设计，通过WebSocket协议实现高效实时通信，结合异步编程模型保证高并发性能，为即时通讯应用提供完整、稳定、可扩展的基础框架。