通过使用ThreadLocal来绑定connection到每个线程特有的map中，在这个线程中运行的dao都可以来获取这个connection，所以就可以通过一个connection来做事务了，写一个filter过滤所有的请求，做成事务

@WebFilter("/\*")

**public** **class** TranactionFilter **extends** HttpFilter{

@Override

**public** **void** doFilter(HttpServletRequest request, HttpServletResponse response, FilterChain chain)

**throws** IOException, ServletException {

Connection connection=**null**;

**try** {

//1.获取连接

connection=JDBCTools.*getConnection*();

//2.绑定连接

ConnectionContext.*getSingleton*().bind(connection);

//3.开启事务

connection.setAutoCommit(**false**);

//4.做事务

chain.doFilter(request, response);

//5.提交事务

connection.commit();

}**catch** (Exception e) {

//6.回滚事务

**try** {

connection.rollback();

} **catch** (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

}**finally** {

//7.解除绑定

ConnectionContext.*getSingleton*().remove();

//8.关闭连接

JDBCTools.*close*(connection);

}

}

}

ConnectionContext类写成单例模式

**package** tools;

**import** java.sql.Connection;

**public** **class** ConnectionContext {

**private** ThreadLocal<Connection> threadLocal=**new** ThreadLocal<>();

**private** ConnectionContext() {}

**private** **static** ConnectionContext *singleton*=**new** ConnectionContext();

**public** **static** ConnectionContext getSingleton() {

**return** *singleton*;

}

//绑定到ThreadLocal(每个线程独有的map，只有在这个线程中才能获取)

**public** **void** bind(Connection connection) {

threadLocal.set(connection);

}

**public** Connection get() {

**return** threadLocal.get();

}

**public** **void** remove() {

threadLocal.remove();

}

}