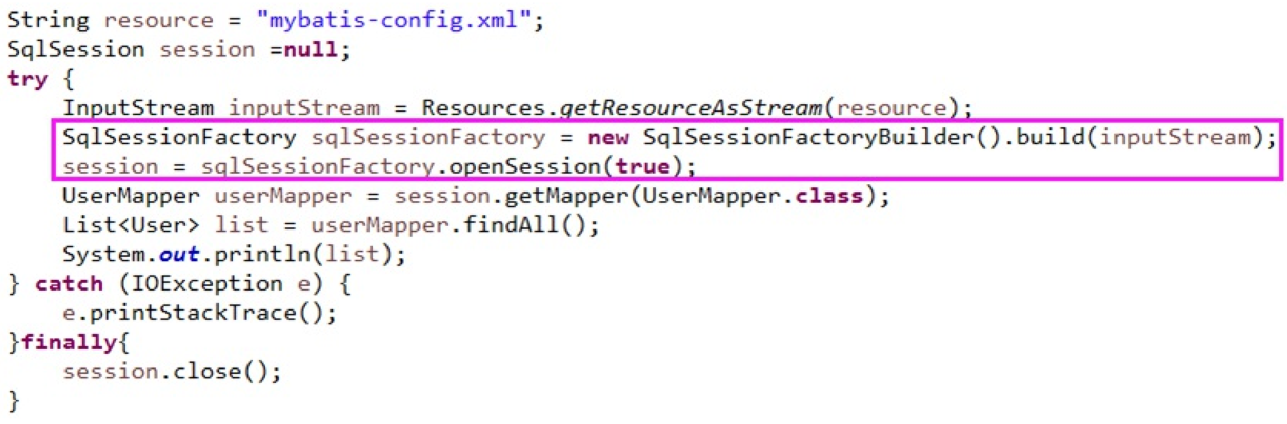
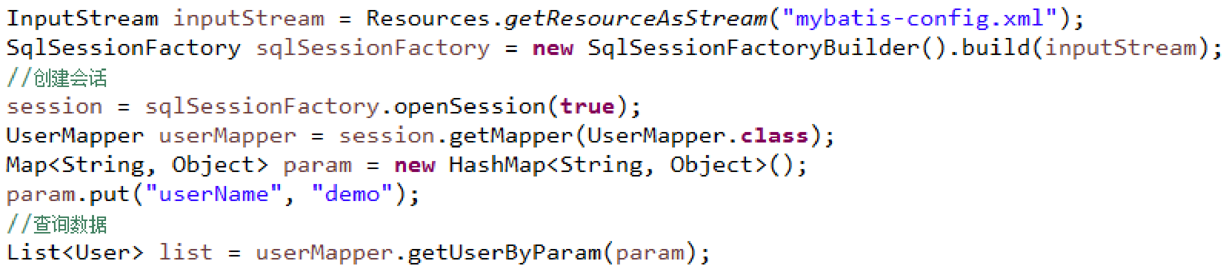
SqlSession



```
String resource = "mybatis-config.xml";
SqlSession session =null;
try {
    InputStream inputStream = Resources.getResourceAsStream(resource);
    SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(inputStream);
    session = sqlSessionFactory.openSession(true);
    UserMapper userMapper = session.getMapper(UserMapper.class);
    List<User> list = userMapper.findAll();
    System.out.println(list);
} catch (IOException e) {
    e.printStackTrace();
}finally{
    session.close();
}
```

每个基于MyBatis的应用都是以一个SqlSessionFactory的实例为中心的。

Sqlsession：包含了面向数据库执行SQL命令所需的所有方法



```
InputStream inputStream = Resources.getResourceAsStream("mybatis-config.xml");
SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(inputStream);
//创建会话
session = sqlSessionFactory.openSession(true);
UserMapper userMapper = session.getMapper(UserMapper.class);
Map<String, Object> param = new HashMap<String, Object>();
param.put("userName", "demo");
//查询数据
List<User> list = userMapper.getUserByParam(param);
```

SqlSessionFactory：

一旦被创建就应该在应用的运行期间一直存在

SqlSession：

每个线程都应该有自己的SqlSession实例

每次收到的HTTP请求，就可以打开一个SqlSession，返回一个响应，就关闭它

通过SqlSession获取映射器Mapper接口的实例

使用完一定要关闭