

2018

Java EE框架整合---Spring整合Mybatis

Java EE framework -Spring整合Mybatis

王磊

研发部

CONTENTS



引入Spring和Mybatis配置文件



Mybatis逆向工程



编写业务逻辑、客户端文件



单元测试

一、创建JavaWeb项目

- 1、新建Java Project
- 2、引入 最小Spring jar包、Mybatis jar包
- 3、植入log4j配置文件
- 4、构建路径

lib

commons-logging-1.2.jar
mybatis-3.4.6.jar
mybatis-generator-core-1.3.7.jar
mybatis-spring-1.3.1.jar
mysql-connector-java-5.1.6-bin.jar
spring-aop-4.3.8.RELEASE.jar
spring-aspects-4.3.8.RELEASE.jar
spring-beans-4.3.8.RELEASE.jar
spring-context-4.3.8.RELEASE.jar
spring-core-4.3.8.RELEASE.jar
spring-expression-4.3.8.RELEASE.jar
spring-jdbc-4.3.8.RELEASE.jar
spring-tx-4.3.8.RELEASE.jar
spring-web-4.3.8.RELEASE.jar

log4j.properties

Mybatis基础 jar包

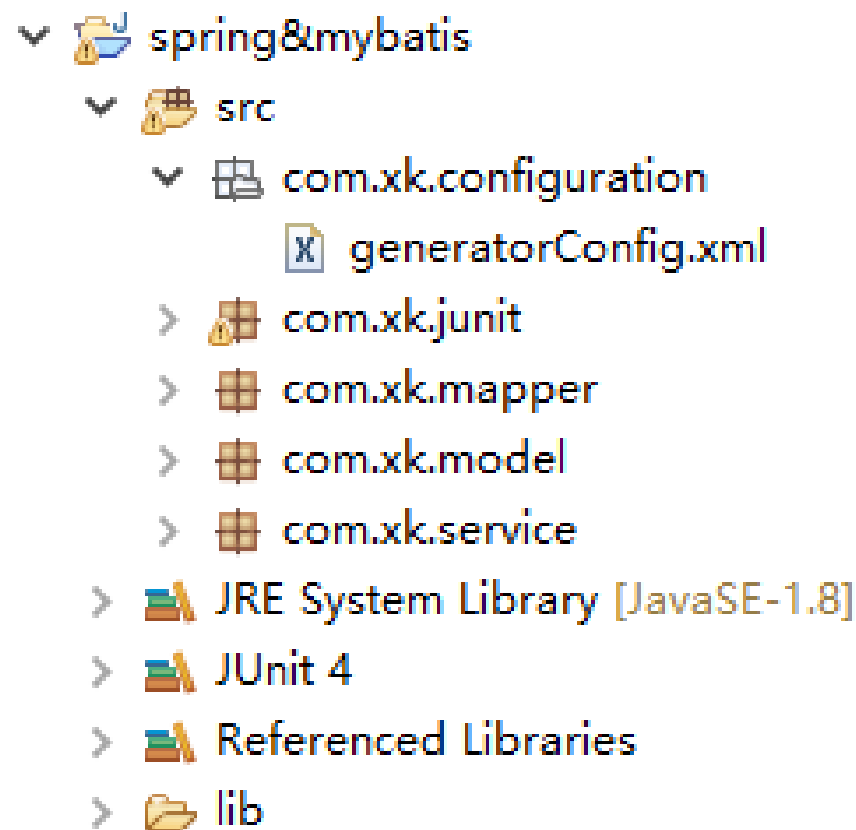
Mybatis逆向工程 jar包

Mybatis结合Spring jar包

日志配置文件

二、创建项目分类包

- 1、创建项目包，以便分类管理
- 2、在com.xk.configuration包中植入generatorConfig.xml配置文件



三、向src中引入Spring和Mybatis配置文件

1、引入applicationContext.xml配置文件

```
<bean id="dataSource"
      class="org.springframework.jdbc.datasource.DriverManagerDataSource">
  <property name="driverClassName">
    <!-- <value>org.gjt.mm.mysql.Driver</value>-->
    <value>com.mysql.jdbc.Driver</value>
  </property>
  <property name="url">
    <value>jdbc:mysql://localhost:3306/usermanager?characterEncoding=utf-8</value>
  </property>
  <property name="username"> <value>root</value> </property>
  <property name="password"> <value>root</value> </property>
</bean>
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
  <property name="dataSource" ref="dataSource"></property>
  <property name="configLocation" value="sqlMapConfig.xml"></property>
</bean>
<bean id="sqlsessionTemplate" class="org.mybatis.spring.SqlSessionTemplate">
  <constructor-arg index="0" ref="sqlSessionFactory" />
</bean>
<bean id="userMapper" class="com.xk.service.UserService">
  <property name="sqlsession" ref="sqlsessionTemplate">
  </property>
</bean>
</bean> <bean id="user" class="com.xk.model.Tuserlogin"></bean>
```

• 三、向src中引入Spring和Mybatis配置文件 •

2、引入sqlMapConfig.xml配置文件




```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE configuration
PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
    <!-- mappers位置 -->
    <mappers>
        <mapper resource="com/xk/mapper/TuserloginMapper.xml"/>
    </mappers>
</configuration>
```




四、使用Mybatis逆向工程

- 1、Mybatis进行逆向工程
- 2、刷新src，出现相应JavaBean和Mapper接口

```
public class Generator {  
    @Test  
    public void create() throws Exception{  
        List<String> warnings = new ArrayList<String>();  
        boolean overwrite = true;  
        File configFile = new File("./src/com/xk/configuration/generatorConfig.xml");  
        ConfigurationParser cp = new ConfigurationParser(warnings);  
        Configuration config = cp.parseConfiguration(configFile);  
        DefaultShellCallback callback = new DefaultShellCallback(overwrite);  
        MyBatisGenerator myBatisGenerator = new MyBatisGenerator(config, callback, warnings);  
        myBatisGenerator.generate(null);  
    }  
}
```

(1)

▼  com.xk.mapper
 >  TuserloginMapper.java
 >  TuserloginMapper.xml

▼  com.xk.model
 >  Tuserlogin.java
 >  TuserloginExample.java

(2)

五、编写业务逻辑类文件

1、编写userService业务逻辑类文件

```
package com.xk.service;
import java.util.List;
public class UserService {
    private SqlSessionTemplate sqlSession;
    public void setSqlsession(SqlSessionTemplate sqlSession) {
        this.sqlSession = sqlSession;
    }
    //根据用户id查找用户
    public Tuserlogin findById(int id){
        TuserloginMapper userMapper = sqlSession.getMapper(TuserloginMapper.class);
        Tuserlogin user = userMapper.selectByPrimaryKey(id);
        return user;
    }
    //查询所有用户信息
    public List<Tuserlogin> findAll(){
        TuserloginMapper userMapper = sqlSession.getMapper(TuserloginMapper.class);
        TuserloginExample example = new TuserloginExample();
        List<Tuserlogin> users = userMapper.selectByExample(example);
        return users;
    }
}
```


五、编写业务逻辑类文件

1、编写userService业务逻辑类文件

```
//根据用户id删除用户
public boolean deleteById(int id) {
    boolean flag = false;
    try {
        TuserloginMapper userMapper = sqlSession.getMapper(TuserloginMapper.class);
        userMapper.deleteByPrimaryKey(id);
        flag = true;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return flag;
}

//删除全体用户信息
public boolean deleteAll() {
    boolean flag = false;
    TuserloginMapper userMapper = sqlSession.getMapper(TuserloginMapper.class);
    try {
        TuserloginExample example = new TuserloginExample();
        userMapper.deleteByExample(example);
        flag = true;
    } catch (Exception e) {
        e.printStackTrace();
    }
    return flag;
}
```

五、编写业务逻辑类文件

1、编写userService业务逻辑类文件

//增加新用户信息

```
public boolean addUser(Tuserlogin user) {  
    boolean flag = false;  
    TuserloginMapper userMapper = sqlSession.getMapper(TuserloginMapper.class);  
    try {  
        userMapper.insert(user);  
        flag = true;  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return flag;  
}
```

//根据用户id修改用户信息

```
public boolean modifyUser(Tuserlogin user) {  
    boolean flag = false;  
    TuserloginMapper userMapper = sqlSession.getMapper(TuserloginMapper.class);  
    userMapper.updateByPrimaryKeySelective(user);  
    return flag;  
}
```

六、编写客户端文件

1、编写client客户端测试类文件

(1) 根据用户id查询用户

```
package com.xk.junit;
import java.util.Iterator;
import java.util.List;
import org.junit.Test;
import org.springframework.context.support.ClassPathXmlApplicationContext;
import com.xk.model.Tuserlogin;
import com.xk.service.UserService;
public class Client {
    ClassPathXmlApplicationContext ctx = new ClassPathXmlApplicationContext
        ("applicationContext.xml");
    UserService userService = (UserService) ctx.getBean("userMapper");
    //根据用户id查找用户
    @Test
    public void testFindById() {
        Tuserlogin user = userService.findById(17);
        System.out.println(user.getUsername()+" "+user.getEmail());
    }
}
```

六、编写客户端文件

- 1、编写client客户端测试类文件
- (2) 查询全体用户
- (3) 根据用户id删除用户

```
//查询所有用户信息
@Test
public void testGetAllUsers() {
    List<Tuserlogin> users = userService.findAll();
    Iterator<Tuserlogin> it = users.iterator();
    while(it.hasNext()) {
        Tuserlogin user = it.next();
        System.out.println(user.getUsername()+" "+user.getEmail());
    }
}

//根据用户id删除用户
@Test
public void testDeleteById() {
    Tuserlogin user = userService.findById(18);
}
```

六、编写客户端文件

1、编写client客户端测试类文件

(4) 删除全体用户信息

(5) 增加用户信息

```
// 删除全体用户信息
```

```
@Test
```

```
public void testDeleteAll() {
```

```
    userService.deleteAll();
```

```
}
```

```
// 增加新用户信息
```

```
@Test
```

```
public void testAddUser() {
```

```
    Tuserlogin user = (Tuserlogin) ctx.getBean("user");
```

```
    user.setUsername("newuser1");
```

```
    user.setPassword("123");
```

```
    user.setGrade(1);
```

```
    user.setEmail("newuser1@126.com");
```

```
    userService.addUser(user);
```

```
}
```

六、编写客户端文件

- 1、编写client客户端测试类文件
- (6) 根据用户id修改用户信息

```
// 修改用户信息
@Test
public void testModifyUser() {
    Tuserlogin user = (Tuserlogin) ctx.getBean("user");
    user.setId(1);
    user.setUsername("newuser1111");
    user.setPassword("1232222222");
    user.setGrade(1);
    user.setEmail("newuser1@126.com");
    userService.modifyUser(user);
}
```

七、逐个进行单元测试

- 1、逐个运行客户端junit单元测试
- 2、观察结果

信息: Loaded JDBC driver: com.mysql.jdbc.Driver
虚拟用户 test@126.com

根据id查询结果

信息: Loaded JDBC driver: com.mysql.jdbc.Driver
admin admin@126.com
虚拟用户 test@126.com
测试用户1 test1@126.com
测试用户2 abc@126.com
李晓明 lixiaom@qq.com
齐峰 qif@163.com
翟敏 zhaim@126.com
司徒 sit@126.com
wang wang@126.com
陈军 chenj@126.com
张恒 zhang@qq.com
刘建军 liu@sina.com.cn
张扬菲 zhangyf@126.com
杨杨 yang@126.com

查询全体用户信息

username	password	grade	email
admin_123	888888	1	admin_123@126.com

根据用户id修改后信息

• 总结 •

- 1、创建JavaWeb项目
- 2、创建项目分类包
- 3、使用Mybatis逆向工程
- 4、编写业务逻辑类文件
- 5、编写客户端文件
- 6、单元测试

