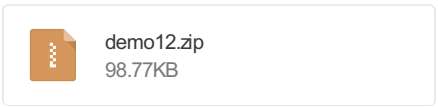
8 Spring Boot Mybatis 事务

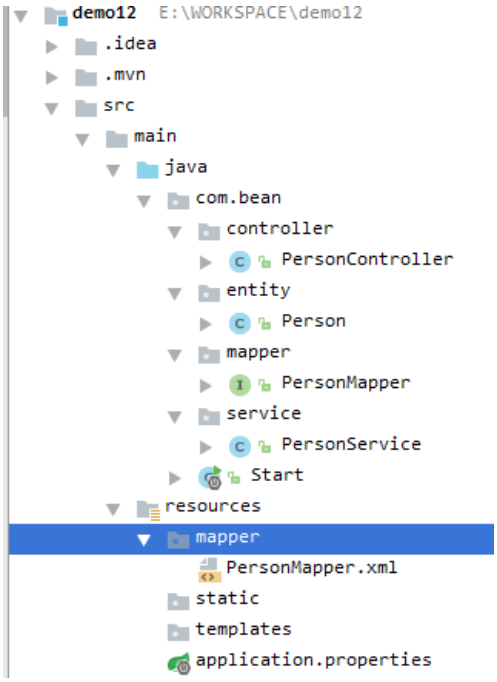
以Mybatis为例讲解事务.

1 启动类添加注释@EnableTransactionManagement

2 控制器方法注释 @Transactional //方法内所有sql都会在一个事务中执行



目录结构:



1 mapper配置文件: 增删改查.

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://
mybatis.org/dtd/mybatis-3-mapper.dtd" >

<mapper namespace="com.bean.mapper.PersonMapper">
  <select id="getPersonList" resultType="com.bean.entity.Person">
    SELECT * FROM person
    <!--mysql 表名区分大小写-->
  </select>
  <select id="getPersonById"  parameterType="int" resultType="com.bean.entity.Person">
    SELECT * FROM person WHERE id = #{id}
  </select>

  <insert id="insertPerson" parameterType="com.bean.entity.Person" useGeneratedKeys="true" keyProperty="id" >
    insert into person (name,passwd,regtime,balance) values (#{param1}, md5(#{param2}), #{param3},#{param4})
    <!--当是多个入参时用param-->
```

```
    </insert>


    <update id="updatePerson" parameterType="com.bean.entity.Person">
        update person set name=#{param2} where id=#{param1}
    </update>

    <update id="transferAcount" parameterType="com.bean.entity.Person">
        update person set balance=#{param2} where id=#{param1}
    </update>

    <delete id="delPerson" parameterType="int" >
        delete from person where id = #{id}
    </delete>


</mapper>
```

2 PersonMapper.java

```
@Repository
@Mapper
public interface PersonMapper {

    public Person getPersonById(int id);

    public List<Person> getPersonList();

    public boolean delPerson(int id);

    public boolean updatePerson(int id,String name);

    public boolean insertPerson(String name,String passwd,String sysdate,double balance);

    public boolean transferAcount(int id,double money);

}
```

3 PersonService.java

```
@Service
public class PersonService {

    @Autowired
    private JdbcTemplate jdbctpl; //idea2016bug:需要修改idea.properties,添加:idea.spring.boot.filter.autoconfig=false

    @Autowired
    private PersonMapper pm;
```

```
    public Person getPerson(int ia){
        return pm.getPersonById(ia);
    }

    public List<Person> getPersonList(){
        return pm.getPersonList();
    }

    public boolean delPerson(int ia){
        return pm.delPerson(ia);
    }

    public boolean updatePerson(int ia,String name){
        return pm.updatePerson(ia,name);
    }


    public boolean insertPerson(String name,String passwd,String sysdate,double balance){
        return pm.insertPerson(name,passwd,sysdate,balance);
    }

    public boolean transferAccount(int id,double money){
        return pm.transferAcount(id,money);
    }
```

4 PersonController.java

```
@RestController
public class PersonController {

    @Autowired
    private PersonService persev;

    @RequestMapping("/all")
    public List<Person> getPeronsList(){
        List<Person> ps = persev.getPersonList();
        return ps;
    }

    //用get把id传进去
    @RequestMapping(value = "/one/{id}",method = RequestMethod.GET)
    public Person getPeron(@PathVariable int id){
        Person ps = persev.getPerson(id);

        return ps;
```

```
}

@RequestMapping(value = "/update/{id}/{name}",method = RequestMethod.GET)
public List<Person> update(@PathVariable int ia,@PathVariable String name){
    persev.updatePerson(ia,name);
    List<Person> ps = persev.getPersonList();
    return ps;
}

@RequestMapping(value = "/insert/{name}/{passwd}",method = RequestMethod.GET)
public List<Person> insert(@PathVariable String name,@PathVariable String passwa){
    double balance = 100.0;
    Date datetime = new Date();
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
    persev.insertPerson(name,passwa,sdf.format(datetime),balance);
    List<Person> ps = persev.getPersonList();
    return ps;
}

@RequestMapping(value = "/del/{id}",method = RequestMethod.GET)
public List<Person> del(@PathVariable int ia){

    persev.delPerson(ia);
    List<Person> ps = persev.getPersonList();
    return ps;
}


/**
 *
 * @param id1 转出帐号
 * @param id2 转进帐号
 * @param money 转账金额
 * @return
 */
@Transactional   //transfer方法内所有sql都会在一个事务中执行
@RequestMapping(value = "/transfer/{id1}/{id2}/{money}",method = RequestMethod.GET)
public List<Person> transfer(@PathVariable int id1,@PathVariable int id2,@PathVariable double money){
    //模拟转账行为,id1转出金额,加入到id2帐号,id2为不存在账户,db报错,id1更新数据回滚.


    Person person1 = persev.getPerson(id1);
    persev.transferAccount(id1,person1.getBalance()-money);

    Person person2 = persev.getPerson(id2);
    persev.transferAccount(id2,person2.getBalance()+money);
```

```
        List<Person> ps = persev.getPersonList();
        return ps;
    }


}
```

5 mysql 内部处理(general.log).

没有查询到id=9的帐号,数据回滚rollback