主要代码逻辑

pom.xml

applicationContext.xml dispatcher-servlet.xml

dao-config.xml

mysql-config.xml

beans-config.xml

web层

dao层

service层

**web层：**

包路径：com.demo.web.controller

UserController：处理URL和具体handler的映射关系

这个controller下有三个方法

* **public String create ()**

映射到创建用户的url，

调用userService.createUser

* **public String borrow**

映射到借款的url

调用userService.fromAtoB

* **public String payback**

映射到还款的url

调用userService.fromAtoB

借款和还款调用相同的接口，有一个type参数，用来区分业务逻辑的不同

**service层：**

userService.createUser

直接调用dao做单次数据库插入操作，创建新的用户，用户ID即为该表的主键ID

userService.fromAtoB

当传入type表示借款动作时，调用私有的borrow方法

业务逻辑为：

1.验证当前余额是否大于借出金额

2.在relation表中，插入一条转账信息

3.更新借款人与被借款人的账户余额信息

userService.fromAtoB

当传入type表示还款动作时，调用私有的payback方法

业务逻辑为：

1.验证当前余额是否大于借出金额

2.获取当前借款人还款数是否超出当前欠款

调用dao获取借款人A向B的总转账款，记为已经还的总还款

调用dao获取借款人B向A的总转账款，记为已经借出的总借出款

计算本次还款是否超出剩余欠款

3.在relation表中，插入一条转账信息

私有的borrow方法和payback方法均依赖于多条数据库操作的同时执行成功，且要求该期间无其他人插入、更新数据，将数据库事务隔离级别设置为最高的串行化