1. **注册**

**Url：** http://domain/register

**输入参数**

Values:{

“recInfo”:{

name:{value:”1234”},

account:{value:”asdf”},

Password:{value:”asdf”},

},

“method”:”0”,

}

**步骤：**

1. 检查用户状态（NO\_SESS/NOT\_LOGIN/LOGIN)
2. 检查输入参数中part的格式和值
3. 检查输入参数是否正确
4. 如果有unique的字段，预先检查；如果有外键，还需要检查docStatus（记录是否成功创建）

4.1 如果有成功创建记录，返回重复错误

4.2 如果记录状态是PENDING，则删除记录以及可能存在的外键记录（user，sugar，**User\_friend\_group**都插入记录，但是没有更新user的状态为DONE），然后重新开始创建过程。

5. 将当前account插入usedAccount，并设置lastAccountUpdateDate

1. 产生sugar
2. 用sugar加密password，替换原始的password
3. 对server产生的字段进行检查（dev only，pro not use）
4. 将用户信息加入users，置docStatus=PENDING。
5. 将sugar加入userSugar
6. 更新users，将docStatus=DONE
7. 使用新生成记录的objectId，在**User\_friend\_group**（朋友分组）中新建一个默认记录

{name:’我的好友’, userId: objectId, firendsInGroup:[]}

**登录(login)**

**Url：** http://domain/login

1. 采用modal：2个input（用户名和密码），一个checkbox（记住密码）

2. db：同“注册”

3. 输入参数

Values:{

“recordInfo”:{

name:{value:”1234”},

Acount:{value:’123’},

Password:{value:”asdf”},

},

“curColl”:”users”,

}

4. 步骤

4.1 检查格式

4.2 检查值

4.3 从usersSugar读取sugar，对密码进行加密，然后读取users中的password，进行比较。

4.4 如果比较一致，，产生sessionId，并存入redis

1. **修改**

**Url：** http://domain/register

**输入参数**

Values:{

“recodInfo”:{

name:{value:”1234”},

account:{value:”asdf”},

Password:{value:”asdf”},

},

“method”:”2”,

}

**步骤：**

1. 检查用户状态（NO\_SESS/NOT\_LOGIN/LOGIN)
2. 检查输入参数中part的格式和值
3. 检查输入参数是否正确
4. 如果输入数据有password，从session中获得userId，然后在db中获得sugar，进行hash操作，并赋给输入数据
5. 根据session中的userId，在db中获得对应的记录（docStatus必须为done）
6. 输入数据和db记录比较，如果是相同的，直接删除字段（因为无需更新到db中）
7. 如果有unique的字段，预先检查；如果有外键，还需要检查docStatus（记录是否成功创建）
8. 如果有account，首先判断account是否已经存在usedAccount，是，结束。否，usedAccount的长度是否超出定义，是，将开头的元素删除直接小于定义；否，将当前account插入usedAccount，并设置lastAccountUpdateDate
9. 产生sugar
10. 用sugar加密password，替换原始的password
11. 对server产生的字段进行检查（dev only，pro not use）
12. 将用户信息加入users，置docStatus=PENDING。
13. 将sugar加入userSugar
14. 更新users，将docStatus=DONE
15. 使用新生成记录的objectId，在**User\_friend\_group**（朋友分组）中新建一个默认记录