# 管理员（Admin\_user）

## Create

**备注：2种用户，1：type=administrator 2： type=normal。Administrator自动创建，normal由administrator创建。**

所以create只对normal生效

**url:** <http://domain/admin/> POST

**输入参数：**

Values:{

RecordInfo:{

name

password:

type: 1 必定是normal

priority:[]

}

method:0

}

**备注：**

**步骤：**

1. 检查输入参数是否正确

**2.** 检查是否有session（用户是否登录）

3. 用户type是否为0（**只有administrator能够创建**）

4. 创建新的记录

5. 获取记录的object\_id，然后生成一个随机字符，在sugar中创建新纪录

6. 使用随机字符对password加密，然后更新password字段

## Update

**url:** <http://domain/admin/> POST

**输入参数：**

Values:{

RecordInfo:{

Name:

password:

priority:[]

},

recordId: admin\_user\_object\_id

method:1

}

**备注：**

**步骤：**

1.检查输入参数是否正确

**2.** 检查是否有session（用户是否登录）

3. 用户type是否为0（administrator更改），或者userId是否等于recdorId（自己改）

4. 如果password存在，

4.1生成一个随机字符，

4.2 根据recordId在sugar中更新对应用户的sugar字段。

4.3 使用新随机字符对新password加密，更新password字段

5. 根据传入参数更新记录

## Search

Admin可以读取所有用户信息，normal只能读取自己的信息（并作修改）

**url:** <http://domain/admin/> POST

**输入参数：**

Values:{

method:2

}

**备注：**

**步骤：**

1.检查输入参数是否正确

**2.** 检查是否有session（用户是否登录）

3. 根据用户id在adminUser中查找对应的记录

4. 如果用户查找到，且type＝0（administrator），返回所有记录；否则，只返回自己的记录

## Delete

只有Admin才可以执行删除操作

**url:** <http://domain/admin/> POST

**输入参数：**

Values:{

recIdArr:[]

method:3

}

**备注：**

**步骤：**

1.检查输入参数是否正确

**2.** 检查是否有session（用户是否登录）

3. 根据用户id在adminUser中查找对应的记录，检查type是否为0

4. 对recIdArr中的object\_id进行删除（设置dDate）

# 管理员做出的处罚（Admin\_penalize）

## Create

**url:** <http://domain/penalize/> POST

**输入参数：**

Values:{

RecordInfo:{

punishedId：

reason:

penalizeType

duration： 单位：天

}

method:0

}

**备注：**

**步骤：**

1. 检查输入参数是否正确

**2.** 检查是否有session（用户是否登录）

3. 管理员是否有权发起处罚

4. creatorId=当前用户id，创建新的记录