## 统一认证平台设计

1 对外提供认证接口

2 提供对外了licence管理

3 对内进行虚拟登录使用统一的token

认证方式： RSA算法是一种非对称加密算法，公钥和私钥的非对称加密方式如果认证通过默认有访问权限，权限由超级管理员配置

提供保存对方公钥的配置界面，

键值对方式

key:对方系统的标识

value:对方公钥文件

验证成功之后本系统提供一个虚拟用户登录，生成token，保存到Redis中设置有效期，并同步保存到cookie里面，对方凭认证信息，就可以访问相应的资源

而本系统的管理员也可以对虚拟用户赋权，进行管理

3 公钥保存的数据库表设计

### public\_key公钥信息管理表

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表名 | | public\_key | | | | | | | | | |
| 表说明 | | 公钥信息管理表 | | | | | | | | | |
| 表空间 | |  | | | | | | | | | |
| 存储特性 | | 默认 | | | | | | | | | |
| 字段描述 | | | | | | | | | | | |
| 序号 | 字段名 | | 英文名 | 字段类型 | | | 空否 | 备注 | | 约束 | |
| 1 | id | | ID | int(11) | | | 否 | 自增主键id | |  | |
| 2 | system\_sign | | system\_sign | varchar(50) | | | 否 | 系统唯一标识 | |  | |
| 3 | public\_key\_encryption | | public\_key\_encryption | varchar(3000) | | | 否 | 系统公钥 | |  | |
| 4 | certificate\_issued | | certificate\_issued | varchar(1) | | | 否 | 证书是否下发  0未下发  1已下发 | | |  |
| 5 | certificate\_path | | certificate\_path | varchar(1000) | | | 否 | 证书地址 | | |  |
| 6 | status | | status | varcher(1) | | | 否 | 公钥状态  0 未生效  1 已生效  2 已撤销 | | |  |
| 7 | valid\_time | | valid\_time | time | | | 否 | 有效时间 | | |  |
| 18 | create\_name | | CREATE\_NAME | varchar(50) | | | 是 | 创建人 | | |  |
| 19 | create\_time | | CREATE\_TIME | datetime | | | 是 | 创建时间 | | |  |
| 20 | update\_name | | UPDATE\_NAME | varchar(50) | | | 是 | 修改人 | | |  |
| 21 | update\_time | | UPDATE\_TIME | datetime | | | 是 | 修改时间 | | |  |
| 索引段描述 |  | |  |  | | |  |  | | |  |
| 索引段名1 |  | | 属性 | Unique | | |  |  | | |  |
| 表空间 |  | |  |  | | |  |  | | |  |
| 存储特性 | 默认 | |  |  | | |  |  | | |  |
| 序号 | 字段名 | | 升序/降序 | 备注 | | |  |  | | |  |
| 1 |  | |  |  | | |  |  | | |  |
|  |  | |  |  | | |  |  | | |  |
|  |  | |  |  | | |  |  | | |  |
|  |  | |  |  | | |  |  | | |  |
|  |  | |  |  | | |  |  | | |  |
|  | | | | | | | | | | | |
|  | |  | | | |  | |  | | | |
|  | |  | | | | | | | | | |
|  | |  | | | | | | | | | |
|  |  | | | |  | | | |  | | |
|  |  | | | |  | | | |  | | |

实现思路：

1 校验license的有效期

2 客户端首先由自己的私钥机密，访问本系统的认证接口

3 系统使用客户保存的公钥解密，如果签名可以对得上就是验证通过

4 服务端生成特殊的Token并保存在Redis和对方的cookie中，用于统一登录