认证和权限方案

修改历史：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 修改日期 | 修改内容 | 修改人 | 备注 | 版本 |
| 1 | 2018-01-10 | 初稿 | 高松 |  | 1.0.0 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 一 问题

1. 当前系统中，权限部分使用了身份这个概念进行管理，一个身份对应了一整套接口权限，对于系统中用户身份单一的情况下，没有问题；而当系统中的用户存在多种身份，每个身份对应的接口集合有差异时，仅仅依靠身份信息很难对权限进行扩展和控制。
2. 当前系统中，上层业务服务（如登录，注册）对于下层基础服务的使用比较混乱；用户和经纪人的数据还可以进一步分离
3. 对第三方认证接入的用户没有预留存储属性

实际的例子

1. 经纪人登录网站，针对网站用户的发布房源和签合同接口，不允许经纪人用户访问，也就出现了不同身份对应的接口集合存在差异。
2. 用户和经纪人的注册登录现在是一套代码实现，部分耦合。

# 二 主要技术点

拆分和解耦。

将现在的认证和权限系统拆解为更加细化的认证和授权部分，并联合api网关和token统一实现整体系统的认证和权限方案。

所有的认证方式是从第三方认证，从而更加方便地接入第三方认证

# 三 整体结构图



图中，绿色代表页面服务，蓝色代表非业务服务，黄色代表业务服务。

# 四 逻辑说明

认证和权限系统处理的信息分为认证信息，用户信息，角色信息，接口信息（包含服务实例信息）。

1. 用户信息属于业务性质比较强的信息，其他信息非业务性质比较强。



1. 信息之间的关联性如上图所示，图中每个箭头代表着两类信息的关联，会有单独的服务（或者接口）处理
2. 信息来源。
3. URI，服务实例自动推送
4. 角色，设计阶段需要定义系统中的角色和权限范围（接口集合），并在指定界面进行配置
5. 认证，用户注册
6. 用户信息，用户注册
7. 信息处理流程
8. 设计阶段
9. 制定角色和权限范围方案
10. 运行阶段
11. URI将自动采集。
12. 再按照1）中指定的角色方案在角色配置界面添加角色
13. 配置角色和URI的映射
14. 将角色标识配置到用户登录注册相关服务中
15. 当用户注册时，用户从认证服务申请认证id，将用户的认证id和角色标识映射
16. 当用户登录时，根据用户的认证id获取到角色，URI信息，再与请求的URI进行对比是否有权限。

# 五 服务说明

## 认证服务

认证服务提供用户基本信息的服务，和业务无关。例如，认证id，认证密码

预计提供的接口如下：

1. 生成认证id （可以指定密码和随机生成密码）
2. 认证输入的认证id和认证密码是否匹配
3. 提供认证id列表（翻页或者范围）

## 授权服务

授权服务是一个复合服务，整合其他服务提供授权相关服务

授权服务主要功能如下：

1. 接收各个服务推送的URI、URI名称、服务名称

接口：

1. 接收URI信息接口
2. 角色配置，管理业务逻辑上的用户角色概念

接口：

1. 角色信息的增删改查
2. 授权分配，结合角色服务提供角色，将角色和URI进行映射

接口：

1. 绑定/解绑URI和角色的映射关系
2. 查询URI和角色的映射关系
3. 认证id和角色关联，用户注册时，将认证id和角色进行映射
4. 绑定/解绑认证id和角色的映射关系
5. 查询认证id和角色的映射关系

## Token服务

Token服务提供了用户登录凭证，用户访问权限凭证。

Token和请求的安全处理相关。

预计提供的接口如下：

1. 生成token
2. token和登录的认证id，以及绑定角色的认证id的映射
3. 根据token查询认证id和绑定角色的认证id
4. token和客户端标识映射
5. token和用户标识映射

# 六 流程图

认证和授权主要融合在3个流程中

1. 注册
2. 登录
3. 权限验证

## 注册流程



## 登录流程



## 权限验证流程



# 七 总结

通过将授权服务提取出来，将权限细化到各个角色上，来解决系统中的用户存在多种身份的问题；每一种认证方式对应各个业务系统的登录功能和注册功能都需要进行单独开发，开发工作量变大，但是逻辑不再重合，利于后期维护。

在整体结构图中，一些服务上标识出了当前系统中使用的数据表的名称，并没有出现一个数据表跨越多个服务的现象，说明之前的数据库设计基本满足这个方案设计的要求。所以之后在详细设计的阶段，需要补充新增服务或者逻辑的数据模型。