**UPMS接入文档**

修订历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 版本号 | 作者 | 修订章节 | 修订原因 | 修订日期 | 审阅人 |
| 1.0 | 詹俊鹏 |  | 建档 | 2018/06/06 |  |
|  |  |  |  |  |  |

版权声明

未经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档的部分或全部，并以任何形式传播。若发现上述或其他侵权行为，本公司将依法律追究法律责任。

目录

[一、单点登录机制简述 3](#_Toc516073625)

[二、UPMS接入流程 3](#_Toc516073626)

[2.1 UPMS处理简要流程图 3](#_Toc516073627)

[2.2 接入步骤 5](#_Toc516073628)

[三、业务系统权限控制 8](#_Toc516073629)

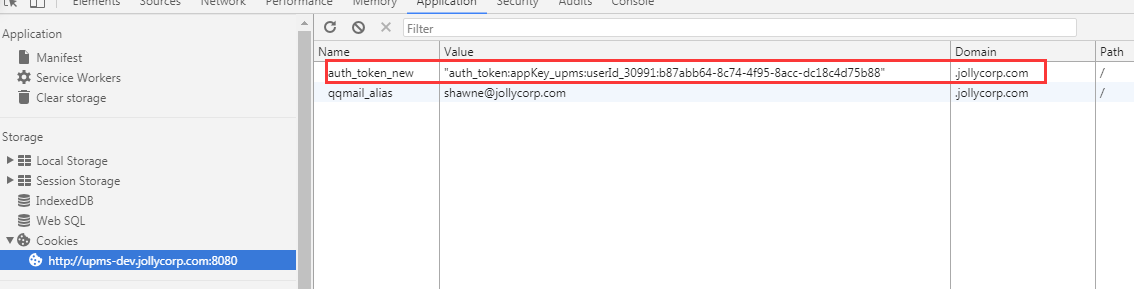
[3.1 菜单可见性 8](#_Toc516073630)

[3.2 按钮可见性 8](#_Toc516073631)

# 一、单点登录机制简述

需要接入单点登录的系统统称业务系统BusiSystem，

统一权限管理系统简称UPMS

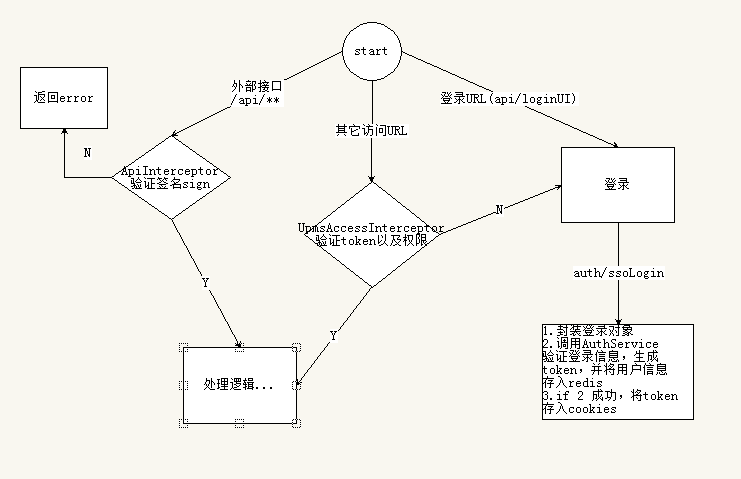
BusiSystem和UPMS需要使用相同的主域名，如（**jollycorp.com:8080**），UPMS处理了登录的业务逻辑，将登录后的用户生成token存储在前端的cookies中，如下图：

并且把用户信息存储在redis中（key:value -> token: AuthUser）。

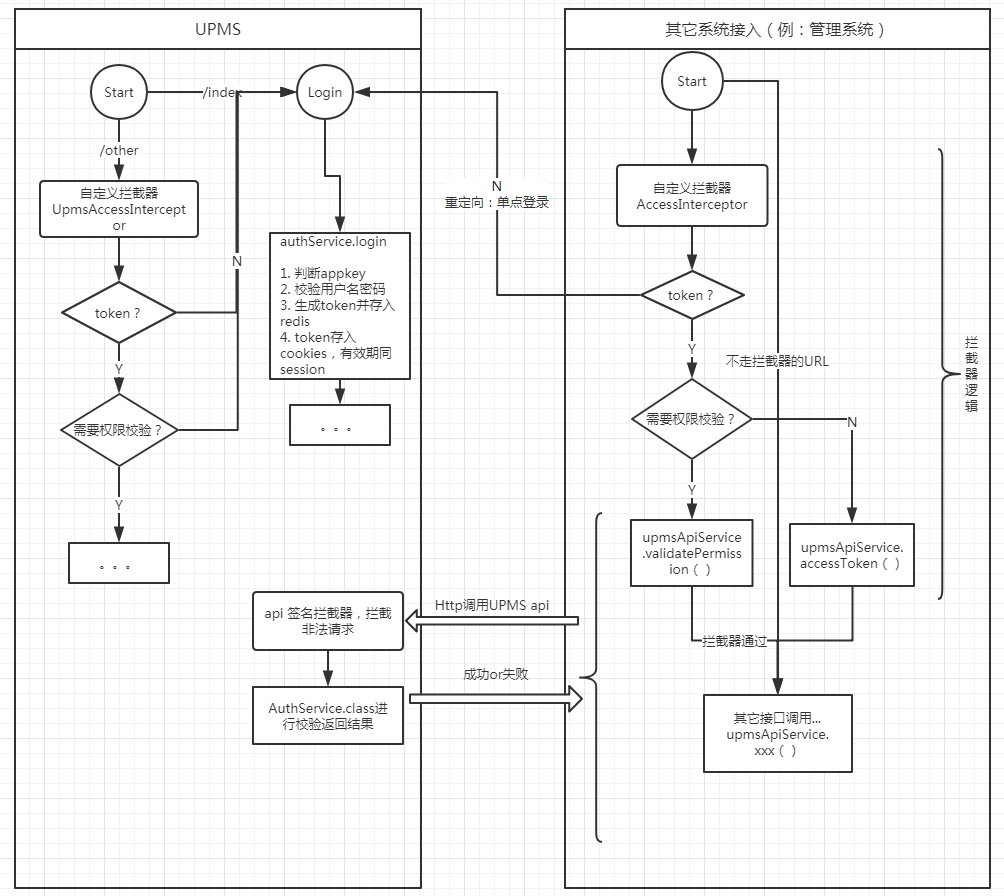
而BusiSystem如果需要进行单点登录，需要自定义拦截器，判断当前cookies中是否存在token，如果不存在token或者token校验（通过http调用UPMS api）或者根据token权限校验失败（通过http调用UPMS api），则拦截器处理拦截到UPMS的登录页面，由UPMS负责登录以及token的管理。

# 二、UPMS接入流程

## 2.1 UPMS处理简要流程图



如上图：UPMS处理登录，由于UPMS和BusiSystem公用同一个主域名，所以通过UPMS登入存放的cookies，其它业务系统也可以访问到，并且UPMS提供了一些api对外开放了一些接口为BusiSystem提供了一些权限获取、校验、用户信息等基础服务。



## 2.2 接入步骤

###### 1. 引入依赖，upms-facade包提供了操作相关API的一些封装。

<**dependency**>  
 <**groupId**>com.jolly</**groupId**>  
 <**artifactId**>upms-facade</**artifactId**>  
 <**version**>1.0-SNAPSHOT</**version**>  
 <**exclusions**>  
 <**exclusion**>  
 <**groupId**>org.slf4j</**groupId**>  
 <**artifactId**>slf4j-api</**artifactId**>  
 </**exclusion**>  
 </**exclusions**>  
</**dependency**>

###### 2. 添加系统配置

其中接口调用地址使用ip，防止调用被解析到外网访问。

*#统一权限管理系统地址，域名***upms.login.url**=**http://upms-dev.jollycorp.com:8080***#统一权限管理系统地址，IP地址***upms.api.url**=**http://172.31.0.91:8080**

###### 3. 添加spring bean

<**bean id="upmsApiService" class="com.jolly.upms.facade.service.impl.UpmsApiServiceImpl"**>  
 <**property name="baseUrl" value="${upms.api.url}"**/>  
</**bean**>

###### 4. 添加spring 拦截器



以上配置中needValidatePermUrls（需要验证权限的url）以及<mvc:exclude-mapping path="xx"/>(不需要拦截的链接)需要根据业务系统自身业务自己配置

accessInterceptor拦截器实现逻辑：

step1：

拦截需要验证token的 URL

step2：

if：URL需要验证权限

BaseResponseDTO respDTO = **upmsApiService**.validatePermission(reqDTO);

判断respDTO返回信息

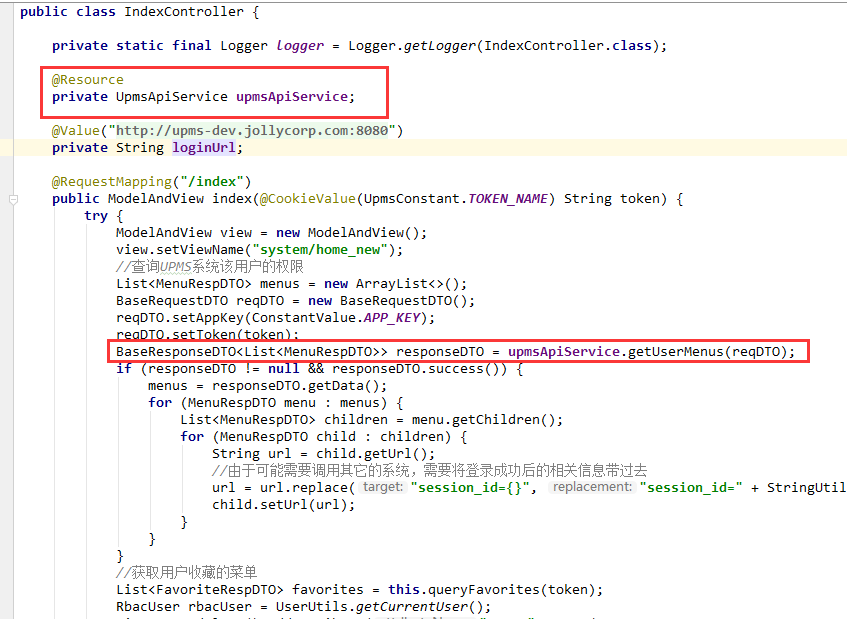
else:

BaseResponseDTO respDTO = **upmsApiService**.accessToken(reqDTO);

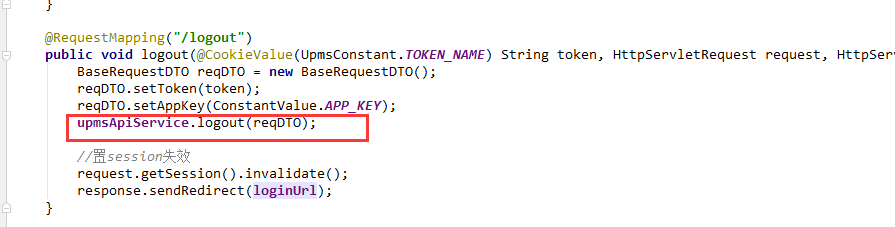
判断respDTO返回信息

###### 5. 应用示例

1. ERP系统获取菜单列表



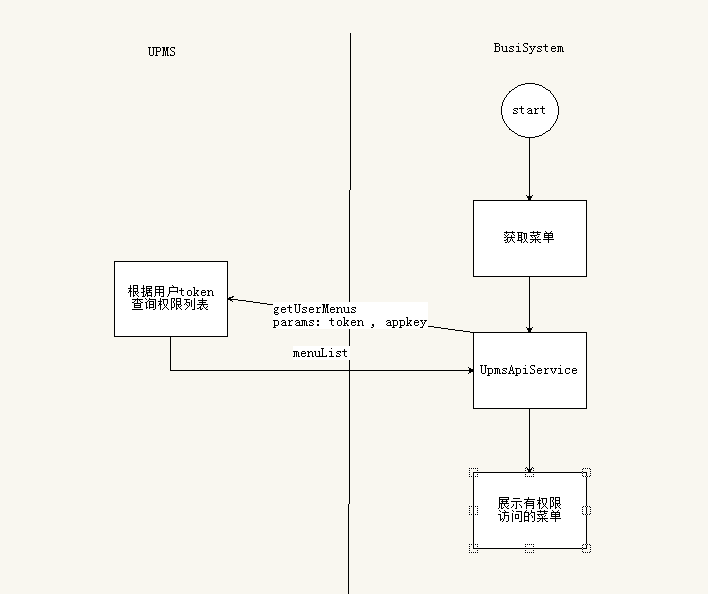
2. 登出



# 三、业务系统权限控制

## 3.1 菜单可见性

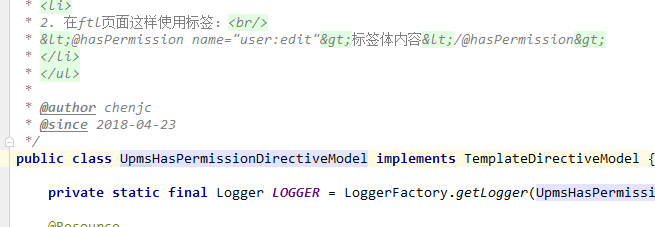
通过UpmsApiService.getMenus接口获取当前用户拥有的权限菜单，并根据自身业务进行展示



## 3.2 按钮可见性

不同于对菜单权限的控制，按钮的权限推荐通过以下方式接入：

在依赖的upms-facade包中，提供了一个权限控制的FreeMarker标签类



1. 在spring配置文件配置的**freemakerCongfig标签中，添加以下配置：**



<**property name="freemarkerVariables"**>  
 <**map**>  
 <**entry key="hasPermission"**>  
 <**bean class="com.jolly.upms.facade.tag.UpmsHasPermissionDirectiveModel"**>  
 <**property name="appKey" value="${appKey}"**/>  
 <**property name="baseUrl" value="${upms.api.url}"**/>  
 </**bean**>  
 </**entry**>  
 </**map**>  
</**property**>

2. 在freemarker页面上使用<@hasPermission name="权限名">标签即可控制按钮权限,如下：

**<@hasPermission name**=**"user:edit">**标签体内容**</@hasPermission>**