账号接口：

AccountWriteService

AccountReadService

账号IP接口，白名单等

AccountIpService

<dependency>  
 <groupId>com.dtdream.gamma</groupId>  
 <artifactId>gamma-utils</artifactId>  
 </dependency>  
  
<dependency>  
 <groupId>com.dtdream.gamma</groupId>  
 <artifactId>gamma-privilege</artifactId>  
</dependency>

/\*  
 \*  
 \* ------------------------------------------------------------------  
 \* Copyright 2017 Hangzhou DtDream Technology Co.,Lt d. All rights reserved.  
 \* ------------------------------------------------------------------  
 \* Product: 用户中心  
 \* Module Name: organization-api  
 \* Date Created: 17-10-30 下午2:26  
 \* Description:  
 \* ------------------------------------------------------------------  
 \* Modification History  
 \* DATE Name Description  
 \* ------------------------------------------------------------------  
 \* 17-10-30 下午2:13 linrizeng  
 \* ------------------------------------------------------------------  
 \* /  
 \*/  
  
package com.dtdream.gamma.organization.service;  
  
import com.dtdream.commons.common.request.Paging;  
import com.dtdream.commons.common.request.Response;  
import com.dtdream.commons.user.BaseUser;  
import com.dtdream.gamma.account.model.AccountCategory;  
import com.dtdream.gamma.account.model.RouteGroup;  
import com.dtdream.gamma.login.data.EcpLoginUser;  
import com.dtdream.gamma.login.data.LoginUser;  
import com.dtdream.gamma.organization.dto.\*;  
import com.dtdream.gamma.organization.model.Employee;  
import com.dtdream.gamma.organization.model.Institution;  
import com.dtdream.gamma.organization.model.OrgDepartment;  
import com.dtdream.gamma.tree.dto.Tree;  
import com.dtdream.gamma.tree.model.Node;  
import com.dtdream.vision.client.Export;  
  
import javax.annotation.Nullable;  
import javax.validation.constraints.NotNull;  
import java.util.List;  
  
public interface EmployeeReadService {  
 */\*\*  
 \* 通过组织id和账号id分页获取用户信息  
 \** ***@param*** *pageNo 分页数  
 \** ***@param*** *pageSize 每页大小  
 \** ***@param*** *orgId 组织id  
 \** ***@param*** *uid 账号id  
 \** ***@param*** *mobile 手机  
 \** ***@param*** *email 邮箱  
 \** ***@param*** *name 名称  
 \** ***@param*** *keyword 过滤关键字  
 \** ***@return*** *用户分页  
 \*/* Response<Paging<Employee>> pagingEmpByOrgIdAndAccountId(Integer pageNo, Integer pageSize, Long orgId,Long uid,  
 @Nullable String mobile, @Nullable String email, @Nullable String displayName, @Nullable String keyword);  
  
 Response<List<Long>> findIdsByDisplayNameAndMerchantsId(String displayName,Long merchantsId);  
  
 Response<Long> getEmpOrgId(Long accountId, Long orgId);  
 */\*\*  
 \* 通过账户名获取登录用户信息  
 \** ***@param*** *accountId 账户id  
 \** ***@return*** *登录用户信息  
 \*/* @Deprecated  
 Response<EcpLoginUser> getLoginUser(Long accountId);  
  
 Response<List<DepNodeDto>> getEmpDepByAccountIdAndOrgId(Long accountId,Long orgId);  
 */\*\*  
 \* 已废弃，新方法请参考 getEmpInsByAccountIdAndInsId  
 \* 获取雇员的机构简要信息  
 \** ***@param*** *accountId 雇员账号id  
 \** ***@return*** *机构信息  
 \*/* @Deprecated  
 Response<InsNodeDto> getEmpIns(Long accountId);  
  
 */\*\*  
 \* 已废弃，新接口参考 getEmpDepByAccountIdAndInsId  
 \* 获取雇员的部门信息  
 \** ***@param*** *accountId 雇员账号id  
 \** ***@return*** *部门列表  
 \*/* @Export(paramNames = "accountId")  
 @Deprecated  
 Response<List<DepNodeDto>> getEmpDep(Long accountId);  
  
 */\*\*  
 \* 已废弃，  
 \* 通过用户id查找员工信息  
 \** ***@param*** *accountId 账户id  
 \** ***@return*** *员工信息  
 \*/* @Export(paramNames = {"accountId"})  
 @Deprecated  
 Response<Employee> getEmpByAccountId(Long accountId);  
   
 */\*\*  
 \* 通过账号id和机构id获取雇员信息  
 \** ***@param*** *accountId  
 \** ***@param*** *insId  
 \** ***@return*** *\*/* Response<Employee> getEmpByAccountIdAndInsId(Long accountId,Long insId);  
  
 */\*\*  
 \* 已废弃，新接口参考：getEmpDetailByAccountIdAndInsId  
 \* 根据账户id查找员工详情信息  
 \** ***@param*** *accountId 账户id  
 \** ***@return*** *雇员详情  
 \*/* @Deprecated  
 Response<EmpDetail> getEmpDetailByAccountId(Long accountId);  
   
 */\*\*  
 \* 根据账号ID和机构Id获取部门详情  
 \** ***@param*** *accountId  
 \** ***@param*** *insId  
 \** ***@return*** *\*/* Response<List<DepNodeDto>> getEmpDepByAccountIdAndInsId(Long accountId,Long insId);  
  
 */\*\*  
 \* 根据账号ID和机构Id获取机构信息  
 \** ***@param*** *accountId  
 \** ***@param*** *insId  
 \** ***@return*** *\*/* Response<InsNodeDto> getEmpInsByAccountIdAndInsId(Long accountId,Long insId);  
   
 */\*\*  
 \* 根据账号ID和机构ID获取关联的部门信息  
 \** ***@param*** *accountId  
 \** ***@param*** *insId  
 \** ***@return*** *\*/* Response<List<OrgDepartment>> getDepartmentsByAccountIdAndInsId(Long accountId,Long insId);  
   
 */\*\*  
 \* 根据账号id和机构id获取雇员详情  
 \** ***@param*** *accountId 账号Id  
 \** ***@param*** *insId 机构Id  
 \** ***@return*** *\*/* Response<EmpDetail> getEmpDetailByAccountIdAndInsId(Long accountId,Long insId);  
   
 */\*\*  
 \* 获取当前雇员详细信息  
 \** ***@param*** *loginUser 当前用户  
 \** ***@return*** *雇员详情  
 \*/* @Export(paramNames = {"user"})  
 @Deprecated  
 Response<EmpDetail> getCurrentEmpDetail(LoginUser loginUser);  
  
 */\*\*  
 \* 通过组织id分页获取用户信息  
 \** ***@param*** *pageNo 分页数  
 \** ***@param*** *pageSize 每页大小  
 \** ***@param*** *orgId 组织id  
 \** ***@param*** *mobile 手机  
 \** ***@param*** *email 邮箱  
 \** ***@param*** *name 名称  
 \** ***@param*** *keyword 过滤关键字  
 \** ***@return*** *用户分页  
 \*/* @Export(paramNames = {"pageNo", "pageSize", "orgId", "mobile", "email", "name", "keyword"})  
 Response<Paging<Employee>> pagingEmpByOrgId(Integer pageNo,  
 Integer pageSize,  
 Long orgId,  
 @Nullable String mobile,  
 @Nullable String email,  
 @Nullable String name,  
 @Nullable String keyword);  
  
 */\*\*  
 \* 通过组织id获取用户信息  
 \** ***@param*** *orgId 组织id  
 \** ***@param*** *mobile 手机  
 \** ***@param*** *email 邮箱  
 \** ***@param*** *name 名称  
 \** ***@param*** *keyword 过滤关键字  
 \** ***@return*** *用户信息  
 \*/* @Export(paramNames = {"orgId", "mobile", "email", "name", "keyword"})  
 Response<List<Employee>> listEmpByOrgId(Long orgId,  
 @Nullable String mobile,  
 @Nullable String email,  
 @Nullable String name,  
 @Nullable String keyword);  
  
 */\*\*  
 \* 通过组织id分页获取该节点以及子孙节点所有的用户信息  
 \** ***@param*** *pageNo 分页数  
 \** ***@param*** *pageSize 每页大小  
 \** ***@param*** *orgId 组织id  
 \** ***@param*** *mobile 手机  
 \** ***@param*** *email 邮箱  
 \** ***@param*** *name 名称  
 \** ***@param*** *keyword 过滤关键字  
 \** ***@return*** *用户分页  
 \*/* @Export(paramNames = {"pageNo", "pageSize", "orgId", "mobile", "email", "name", "keyword"})  
 Response<Paging<Employee>> pagingEmpRecur(Integer pageNo, Integer pageSize, Long orgId,  
 @Nullable String mobile,  
 @Nullable String email,  
 @Nullable String name,  
 @Nullable String keyword);  
  
 */\*\*  
 \* 通过组织id获取该节点以及子孙节点所有的用户信息  
 \** ***@param*** *orgId 组织id  
 \** ***@param*** *mobile 手机  
 \** ***@param*** *email 邮箱  
 \** ***@param*** *name 名称  
 \** ***@param*** *keyword 过滤关键字  
 \** ***@return*** *用户信息  
 \*/* @Export(paramNames = {"orgId", "mobile", "email", "name", "keyword"})  
 Response<List<Employee>> listEmpRecur(Long orgId,  
 @Nullable String mobile,  
 @Nullable String email,  
 @Nullable String name,  
 @Nullable String keyword);  
  
 */\*\*  
 \* 通过用户类别获取用户信息  
 \** ***@param*** *category 组织id  
 \** ***@param*** *mobile 手机  
 \** ***@param*** *email 邮箱  
 \** ***@param*** *name 名称  
 \** ***@param*** *keyword 过滤关键字  
 \** ***@return*** *用户信息  
 \*/* @Export(paramNames = {"pageNo", "pageSize", "category", "mobile", "email", "name", "keyword"})  
 Response<Paging<Employee>> pagingEmpByCategory(Integer pageNo,  
 Integer pageSize,  
 String category,  
 @Nullable String mobile,  
 @Nullable String email,  
 @Nullable String name,  
 @Nullable String keyword);  
 */\*\*  
 \* 通过用户类别获取用户信息  
 \** ***@param*** *category 组织id  
 \** ***@param*** *mobile 手机  
 \** ***@param*** *email 邮箱  
 \** ***@param*** *name 名称  
 \** ***@param*** *keyword 过滤关键字  
 \** ***@return*** *用户信息  
 \*/* @Export(paramNames = {"category", "mobile", "email", "name", "keyword"})  
 Response<List<Employee>> listEmpByCategory(String category,  
 @Nullable String mobile,  
 @Nullable String email,  
 @Nullable String name,  
 @Nullable String keyword);  
 */\*\*  
 \* 通过账户ID查找所关联的基础用户类别（level为0）  
 \** ***@param*** *accountId 账户id  
 \** ***@return*** *用户类别列表  
 \*/* Response<List<AccountCategory>> getUserBaseCategories(Long accountId);  
  
 */\*\*  
 \* 获取当前用户的根categories  
 \** ***@param*** *baseUser 当前用户  
 \** ***@return*** *用户类别信息  
 \*/* @Export(paramNames = {"baseUser"})  
 Response<List<AccountCategory>> getCurrentUserBaseCategories(@NotNull BaseUser baseUser);  
  
 */\*\*  
 \* 获取用户所有的用户类别  
 \** ***@param*** *accountId 账户id  
 \** ***@return*** *用户类别列表  
 \*/* Response<List<AccountCategory>> getUserCategories(Long accountId);  
  
 */\*\*  
 \* 通过账户ID查找所关联的基础用户类别编码（level为0）  
 \** ***@param*** *accountId 账户id  
 \** ***@return*** *用户类别编码列表  
 \*/* Response<List<String>> getUserBaseCategoryCodes(Long accountId);  
  
 */\*\*  
 \* 通过用户ID查找所关联的用户类别编码列表  
 \** ***@param*** *accountId 账户id  
 \** ***@return*** *用户类别编码列表  
 \*/* Response<List<String>> getUserAllCategories(Long accountId);  
  
 */\*\*  
 \* 获取用户当前portal的具体用户类别  
 \** ***@param*** *accountId 账户ID  
 \** ***@param*** *portal 当前portal  
 \** ***@return*** *用户类别编码列表  
 \*/* Response<List<String>> getAccountCategoryByPortal(Long accountId, String portal);  
  
 */\*\*  
 \* 获取部门的员工以及全机构员列表  
 \** ***@param*** *depId 部门id  
 \** ***@return*** *本机构所有雇员以及本部门的雇员  
 \*/* @Export(paramNames = {"depId"})  
 Response<EmpInDepAndNot> getEmpInDepAndNot(Long depId);  
  
 */\*\*  
 \* 导出指定组织下的用户  
 \** ***@param*** *insId 机构id  
 \** ***@return*** *雇员列表  
 \*/* Response<List<EmpImportDto>> exportEmp(Long insId);  
   
   
}