编写与中智组织人员集成的Web Service

2014年8月22日

17:02

在组织和人员数据准备好后，需要编写访问组织机构和授权的API，用于工作流人员等功能。

原来的实现接口，请参考以下实现：

MCSWebApp\PermissionCenterServices\Services\OGUReaderService.asmx

MCSWebApp\PermissionCenterServices\Services\AppSecurityCheckService.asmx

PermissionCenterServices比较核心，我们需要搭建一个新的Web工程，也实现这个两个Web Service。

需要注意的是，这是传统的Web Service，不是WCF。这个新的Web工程，有可能需要引用流程平台以及高铁一号线两边的Assembly才能完成（主要是缓存相关）。

先看看OGUReaderService.asmx需要实现的方法：

|  |  |
| --- | --- |
| 方法名 | 描述 |
| SignInCheck | 人员的认证。其参数就是登录名和口令，我们可以永远返回True |
| RemoveAllCache | 工作流平台在机构人员访问的两端都有缓存，这个API就是在数据调整后，发送缓存失效的UDP包。一般在运维层面有可能有个界面去执行一下此操作。这里面就执行了一个函数SCCacheHelper.InvalidateAllCache(); |
| GetObjectsDetail | 获得某个对象的详细信息，其传入参数是对象的不同ID以及对象的类型。返回的是DataSet。对象的类型包括组织、人员和群组 |
| GetOrganizationChildren | 得到一个组织下的子对象，子对象的类型包括组织、人员和群组，返回包含对象集合的DataSet |
| QueryOGUByCondition3 | 根据对象的名称等信息模糊查询。返回包含对象集合的DataSet |
| GetUsersInGroups | 查询群组中的人员，相当于获取群组的Members。返回包含人员集合的DataSet |
| GetGroupsOfUsers | 查询某些人从属的群组，相当于获取用户的MemberOf。返回包含群组的DataSet |
| GetSecretariesOfLeaders | 获取领导们的秘书们。返回包含人员集合的DataSet |
| GetLeadersOfSecretaries | 获取秘书们的领导们。返回包含人员集合的DataSet |

这里有一个需要注意的地方，整棵组织机构树的根是内置在系统中的，隐含的，它的名称是“根组织”，GUID是“e588c4c6-4097-4979-94c2-9e2429989932”。如果涉及到组织根组织的查询，需要返回此对象。

上面个接口，很多是返回DataSet的，其实每个DataSet，仅仅包含一个DataTable，因此下面介绍这些DataSet的结构。

通用组织、人、群组的DataTable如下表所示，如果没有特别说明，是允许为空的。

参考代码02.Develop\MCSWebApp\PermissionCenterServices\Tables\OguObjectTableBuilder.cs

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 默认值 | 说明 |
| OBJECTCLASS | 字符串 |  | 对象的类型，不允许为空。值为"USERS"、"GROUPS"、"ORGANIZATIONS" |
| PERSON\_ID | 字符串 |  | 员工ID |
| POSTURAL | 整型 |  | 没用 |
| RANK\_NAME | 字符串 |  | 用户的Title |
| STATUS | 整型 | 1 | 对象的状态。1表示正常，3表示被删除 |
| ALL\_PATH\_NAME | 字符串 |  | 如果属于某个组织，则是组织的ALL\_PATH\_NAME加上自己的OBJ\_NAME。 |
| GLOBAL\_SORT | 字符串 |  | 先用空串试试 |
| ORIGINAL\_SORT | 字符串 |  | 先用空串试试 |
| DISPLAY\_NAME | 字符串 |  | 对象的显示名称，例如“牛小超” |
| OBJ\_NAME | 字符串 |  | 对象的名称，会出现在ALL\_PATH\_NAME中。例如当前对象是会计“牛小超”，那么其ALL\_PATH\_NAME就是“集团总部\财务部\牛小超”，这个名称不应该随便改动 |
| LOGON\_NAME | 字符串 |  | 对于人员而言，就是登录名，和下面的CODE\_NAME相同，可以为空，仅影响登录 |
| PARENT\_GUID | 字符串 |  | 对象的父对象ID，36位的GUID，例如人员所属的组织ID。如果为空，则表示根组织 |
| GUID | 字符串 |  | 对象的唯一ID，36位的GUID。不允许为空 |
| INNER\_SORT | 字符串 |  | 对象在组织内的排序号，一般是6位数字，不足六位用0补齐。影响对象在组织内的排序 |
| DESCRIPTION | 字符串 |  | 对象的描述信息 |
| RANK\_CODE | 字符串 | 10 | 对象的级别，例如人的级别或组织的级别，默认值“10”即可 |
| ORG\_CLASS | 整型 | 128 | 组织的类别，写0也可以。 |
| CUSTOMS\_CODE | 字符串 |  | 没用 |
| ORG\_TYPE | 整型 | 2 | 组织的类型。2表示一般部门 |
| E\_MAIL | 字符串 |  | 用户的邮件地址。有时组织或群组也有 |
| ATTRIBUTES | 整型 |  | 对象的属性 |
| SIDELINE | 整型 | 0 | 人员是否是兼职的，0表示不是，1表示是 |
| CODE\_NAME | 整型 |  | 对于人员来说，等同于LOGON\_NAME。对于其它对象，可以为空 |
| VERSION\_START\_TIME | 日期 |  | 没用 |

上面的Table是查询任何对象都返回的字段。

下面是人员Table的扩展字段。注意，是在上面通用字段基础上增加出来的扩展字段。

请参考代码02.Develop\MCSWebApp\PermissionCenterServices\Tables\OguUserTableBuilder.cs

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 默认值 | 说明 |
| START\_TIME | 日期 |  | 没用 |
| END\_NAME | 日期 |  | 没用 |
| MODIFY\_TIME | 日期 |  | 没用 |
| CREATE\_TIME | 日期 |  | 没用 |
| OUSYSDISTINCT1 | 字符串 |  | 没用 |
| OUSYSDISTINCT2 | 字符串 |  | 没用 |
| OUSYSCONTENT1 | 字符串 |  | 没用 |
| OUSYSCONTENT2 | 字符串 |  | 没用 |
| OUSYSCONTENT3 | 字符串 |  | 没用 |
| FIRST\_NAME | 字符串 |  | 名 |
| LAST\_NAME | 字符串 |  | 姓 |
| IC\_CARD | 字符串 |  | 没用 |
| PWD\_TYPE\_GUID | 字符串 |  | 没用 |
| USER\_PWD | 字符串 |  | 没用 |
| CREATE\_TIME1 | 日期 |  | 没用 |
| MODIFY\_TIME1 | 日期 |  | 没用 |
| AD\_COUNT | 整型 | 0 |  |
| SYSDISTINCT1 | 字符串 |  | 没用 |
| SYSDISTINCT2 | 字符串 |  | 没用 |
| SYSCONTENT1 | 字符串 |  | 没用 |
| SYSCONTENT2 | 字符串 |  | 没用 |
| SYSCONTENT3 | 字符串 |  | 没用 |
| PINYIN | 字符串 |  | 用户名称拼音，没有也可以 |
| SORT\_ID | 整型 |  | 没用 |
| NAME | 字符串 |  | 对应于通用结构的OBJ\_NAME |
| VISIBLE | 整型 | 1 | 没用 |
| RANK\_CLASS | 字符串 | 0 | 没用 |
| AccountDisabled | 布尔 | False | 账户是否禁用 |
| PasswordNotRequired | 布尔 | False | 是否不需要密码 |
| DontExpirePassword | 布尔 | False | 密码永不过期 |
| AccountInspires | 日期 |  | 账户生效日期，登录时才有用 |
| AccountExpires | 日期 |  | 账户失效日期，登录时才有用 |
| Address | 字符串 |  | 用户的地址 |
| MP | 字符串 |  | 用户的手机 |
| WP | 字符串 |  | 用户的座机 |
| OtherMP | 字符串 |  | 用户其它的手机，逗号分割的一串号码 |
| CompanyName | 字符串 |  | 公司名称 |
| DepartmentName | 字符串 |  | 部门名称 |
| PhotoTimestamp | 字符串 |  | 头像标识 |
| Sip | 字符串 |  | Lync地址 |

下面是组织（Organization）Table的扩展字段，注意，是在上面通用字段基础上增加出来的扩展字段。

请参考代码02.Develop\MCSWebApp\PermissionCenterServices\Tables\OguOrganizationTableBuilder.cs

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 默认值 | 说明 |
| CHILDREN\_COUNTER | 整型 |  | 没用 |
| NAME | 字符串 |  | 对应于通用结构的OBJ\_NAME |
| MODIFY\_TIME | 日期 |  | 没用 |
| CREATE\_TIME | 日期 |  | 没用 |
| SORT\_ID | 整型 |  | 没用 |
| SYSDISTINCT1 | 字符串 |  | 没用 |
| SYSDISTINCT2 | 字符串 |  | 没用 |
| SYSCONTENT1 | 字符串 |  | 没用 |
| SYSCONTENT2 | 字符串 |  | 没用 |
| SYSCONTENT3 | 字符串 |  | 没用 |
| VISIBLE | 整型 | 1 | 没用 |
| RANK\_CLASS | 字符串 | 0 | 没用 |

群组（Group）没有扩展字段。

关于对象的类型。

在Web Service的方法参数中，会碰到对象类型参数（strObjType）。我们目前如此定义的：

对象类型分为以下几种：

|  |  |  |
| --- | --- | --- |
| USERS | 2 | 用户 |
| ORGANIZATIONS | 1 | 组织 |
| GROUPS | 4 | 群组 |
| SIDELINE | 8 | 是否兼职 |

这些类型可以组合在一起，使用逗号或分号分割。例如我们希望查询出组织或人员，那么这个参数为“USERS,ORGANIZATIONS”。如果使用值来表示，则可以将这几个整数或在一起使用。

关于对象的ID类型（iSoc）。

在某些查询时，我们按照对象不同的ID来查询，这些ID的类型描述如下：

|  |  |  |
| --- | --- | --- |
| 名称 | 值 | 描述 |
| GUID | 1 | 对象的GUID。每一种对象都有 |
| FullPath | 5 | 对象的全路径。我们的权限中心中全部都有 |
| LogonName | 6 | 对象的登录名。目前只有人员有 |

下面逐一介绍比较重要的方法：

DataSet GetObjectsDetail(string strObjType, string strObjValues, int iSoc, string strParentValues, int iSoco, string strExtAttrs)

得到若干个对象的详细信息，返回多行数据。

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| strObjType | 字符串 | 需要返回对象的类型，请参阅上面关于对象类型的说明。如果为空，则表示所有类型的对象 |
| strObjValues | 字符串 | 需要查询的对象的ID，多个对象的ID使用逗号分隔。ID的类型取决于iSoc参数 |
| iSoc | 整型 | 请参考上面的对象的ID类型 |
| strParentValues | 字符串 | 对象所属的父对象。目前忽略此参数 |
| iSoco | 整型 | 忽略此参数 |
| strExtAttrs | 字符串 | 忽略此参数 |
| 返回值 | DataSet | DataSet中包含一个DataTable，每一行包含一个对象。如果人员兼职于两个组织，则返回两个对象。这里需要注意的是，如果请求查询的对象类型（strObjValues）是单一的类型，则DataTable的字段是上面描述的具体字段定义；如果是多个类型，则返回的是通用的字段定义 |

DataSet GetOrganizationChildren(string strOrgValues, int iSoc, int iLot, int iLod, int iDepth, string strOrgRankCodeName, string strUserRankCodeName, string strHideType, string strAttrs)

得到某几个组织下的子对象，返回多行数据。

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| strOrgValues | 字符串 | 父对象的ID集合，多个ID间使用逗号分隔 |
| iSoc | 整型 | 父对象的ID类型。请参考上面的对象的ID类型 |
| iLot | 整型 | 需要查询的子对象类型，如果为空，则表示所有类型的对象。请参阅上面关于对象类型的说明 |
| iLod | 整型 | 是否查询出删除的对象。1表示正常对象，其它表示包含被删除的对象 |
| iDepth | 整型 | 0表示递归查询出所有的子对象。1表示仅查询一级 |
| strOrgRankCodeName | 字符串 | 忽略此参数 |
| strUserRankCodeName | 字符串 | 忽略此参数 |
| strHideType | 字符串 | 忽略此参数 |
| strAttrs | 字符串 | 忽略此参数 |
| 返回值 | DataSet | DataSet中包含一个DataTable，每一行包含一个对象。根据iLot，如果是单一的类型，则DataTable的字段是上面描述的具体字段定义；如果是多个类型，则返回的是通用的字段定义 |

DataSet QueryOGUByCondition3(string strOrgValues, int iSoc, string strLikeName, bool bLike, string strAttr, int iListObjType, int iLod, int iDep, string strHideType, int rtnRowLimit)

模糊查询某个组织下的对象。用于界面上模糊查询对象。

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| strOrgValues | 字符串 | 父对象的ID集合，多个ID间使用逗号分隔 |
| iSoc | 整型 | 父对象的ID类型。请参考上面的对象的ID类型 |
| strLikeName | 字符串 | 需要查询字符串，例如“牛小超” |
| bLike | 布尔 | 是否是Like性质的查询。目前前端传递过来是false |
| strAttr | 字符串 | 忽略此参数 |
| iListObjType | 整型 | 要求查询的对象类型，如果为空，则表示所有类型的对象。请参阅上面关于对象类型的说明 |
| iLod | 整型 | 是否查询出删除的对象。1表示正常对象，其它表示包含被删除的对象 |
| iDep | 整型 | 0表示递归查询出所有的子对象。1表示仅查询一级 |
| strHideType | 字符串 | 忽略此参数 |
| rtnRowLimit | 整型 | 返回的行数。如果是-1，则返回虽有行 |
| 返回值 | DataSet | DataSet中包含一个DataTable，每一行包含一个对象。根据iLot，如果是单一的类型，则DataTable的字段是上面描述的具体字段定义；如果是多个类型，则返回的是通用的字段定义 |

DataSet GetUsersInGroups(string strGroupValues, int iSocg, string strAttrs, string strOrgValues, int iSoco, string strUserRankCodeName, int iLod)

查询群组下的人员。在中智的场景下，应该是查询角色实例。

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| strGroupValues | 字符串 | 群组的ID，多个ID之间使用逗号分隔 |
| iSocg | 整型 | 群组对象的ID类型。请参考上面的对象的ID类型 |
| strAttrs | 字符串 | 忽略此参数 |
| strOrgValues | 字符串 | 忽略此参数。本意是群组所属的组织 |
| iSoco | 整型 | 忽略此参数 |
| strUserRankCodeName | 字符串 | 忽略此参数 |
| iLod | 整型 | 是否查询出删除的对象。1表示正常对象，其它表示包含被删除的对象 |
| 返回值 | DataSet | DataSet中包含一个DataTable，每一行包含一个对象，使用用户的字段定义 |

DataSet GetGroupsOfUsers(string strUserValues, int iSocu, string strParentValue, int iSoco, string strAttrs, int iLod)

得到人员所属的群组。在中智的场景下，应该是查询人员所属的角色实例。

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| strUserValues | 字符串 | 人员的ID，多个ID之间使用逗号分隔 |
| iSocu | 整型 | 人员对象的ID类型。请参考上面的对象的ID类型 |
| strParentValue | 字符串 | 忽略此参数 |
| iSoco | 整型 | 忽略此参数。本意是群组所属的组织 |
| strAttrs | 字符串 | 忽略此参数 |
| iLod | 整型 | 是否查询出删除的对象。1表示正常对象，其它表示包含被删除的对象 |
| 返回值 | DataSet | DataSet中包含一个DataTable，每一行包含一个对象，使用用户的字段定义 |

DataSet GetSecretariesOfLeaders(string strLeaderValues, int iSoc, string strAttrs, int iLod)

得到领导们对应的秘书们。在中智，返回空集合即可。

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| strLeaderValues | 字符串 | 领导（人员）的ID，多个ID之间使用逗号分隔 |
| iSocu | 整型 | 人员对象的ID类型。请参考上面的对象的ID类型 |
| strAttrs | 字符串 | 忽略此参数 |
| iLod | 整型 | 是否查询出删除的对象。1表示正常对象，其它表示包含被删除的对象 |
| 返回值 | DataSet | DataSet中包含一个DataTable，每一行包含一个对象，使用用户的字段定义 |

DataSet GetLeadersOfSecretaries(string strSecValues, int iSoc, string strAttrs, int iLod)

得到秘书们所对应的领导们。在中智，返回空集合即可。

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| strSecValues | 字符串 | 秘书（人员）的ID，多个ID之间使用逗号分隔 |
| iSoc | 整型 | 人员对象的ID类型。请参考上面的对象的ID类型 |
| strAttrs | 字符串 | 忽略此参数 |
| iLod | 整型 | 是否查询出删除的对象。1表示正常对象，其它表示包含被删除的对象 |
| 返回值 | DataSet | DataSet中包含一个DataTable，每一行包含一个对象，使用用户的字段定义 |

再看看AppSecurityCheckService.asmx需要实现的方法：

|  |  |
| --- | --- |
| 方法名 | 描述 |
| GetApplications | 得到所有的应用 |
| GetRoles | 得到某个应用下的角色 |
| GetFunctions | 得到某个应用下的功能（权限） |
| GetFunctionsRoles | 得到某些应用所对应的角色 |
| GetRolesUsers | 得到某个应用下，某几个角色中包含的人员。由于在权限中心下，角色可以包含人员、组织和群组，因此这个方法是返回最终的人员（递归查找）。 |
| GetUserRoles | 得到某个用户在某个应用下所属的角色 |
| GetUserPermissions | 得到某个用户在某个应用下所拥有的功能（权限） |
| GetChildrenInRoles | 查询某个角色下所包含的对象。由于在权限中心下，角色可以包含人员、组织和群组，因此这里就是返回这些直接包含的对象。在中智，最多只有人员。 |
| RemoveAllCache | 工作流平台在机构人员访问的两端都有缓存，这个API就是在数据调整后，发送缓存失效的UDP包。一般在运维层面有可能有个界面去执行一下此操作。这里面就执行了一个函数SCCacheHelper.InvalidateAllCache(); |

在权限中心下，应用包含角色和权限，而角色包含人员、组织和群组；角色和权限（功能）是多对多的关系。

应用、角色、功能对象都有其唯一的ID（GUID），同时它们都有用于助记的代码名称。因此，我们在配置信息中经常使用“应用代码：角色代码”来表示一个角色。

在中智的授权体系下，角色实例的结构和权限中心不同。我们需要找一种映射关系，用来解决流程管理员这样的全局角色。例如，我们可以在根组织下创建一组角色实例。整个体系下就是一个应用，就是根组织；所有工作流平台使用的角色就在下面定义。

而权限（功能）在中智不涉及到，因此API一律返回为空。

下面介绍一下授权方法中所涉及到的DataSet的结构：

应用、角色和权限都有公用的字段定义，参考代码：02.Develop\MCSWebApp\PermissionCenterServices\Tables\AppObjectTableBuilderBase.cs。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 默认值 | 说明 |
| ID | 字符串 |  | 对象的唯一标识，36位的字符串，不允许为空 |
| NAME | 字符串 |  | 对象的名称，不允许为空 |
| CODE\_NAME | 字符串 |  | 对象的代码名称，一般是便于记忆的。有唯一性的要求，不允许为空，在中智，可以与ID相同 |
| DESCRIPTION | 字符串 |  | 对象的描述信息 |
| SORT\_ID | 整型 |  | 排序号，一般用于显示 |

应用对象的字段定义，是在公共属性基础上进行了扩展，参考代码：02.Develop\MCSWebApp\PermissionCenterServices\Tables\ApplicationTableBuilder.cs。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 默认值 | 说明 |
| RESOURCE\_LEVEL | 字符串 |  | 可以忽略 |
| CHILDREN\_COUNT | 整型 |  | 可以忽略 |
| ADD\_SUBAPP | 字符串 |  | 可以忽略 |
| USE\_SCOPE | 字符串 |  | 可以忽略 |
| INHERITED\_STATE | 整型 | 0 | 可以忽略 |

角色对象的字段定义，是在公共属性基础上进行了扩展，参考代码：02.Develop\MCSWebApp\PermissionCenterServices\Tables\RoleTableBuilder.cs。

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 默认值 | 说明 |
| ALLOW\_DELEGATE | 字符串 |  | 可以忽略 |

权限对象没有扩展字段。

下面介绍一些比较重要的方法：

DataSet GetApplications()

得到所有的应用。

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| 返回值 | DataSet | 返回上面提到的应用的集合。在中智，可以返回组织架构的根组织 |

DataSet GetRoles(string appCodeName, RightMaskType rightMask)

得到某个应用下所有的角色。

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| appCodeName | 字符串 | 应用的代码名称。在中智，可以使用对象GUID来表示 |
| rightMask | RightMaskType | 可以忽略 |
| 返回值 | DataSet | 返回上面提到的角色的集合。在中智，可以是根组织下的角色实例 |

DataSet GetFunctions(string appCodeName, RightMaskType rightMask)

得到某个应用下所有的权限（功能）。

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| appCodeName | 字符串 | 应用的代码名称。在中智，可以使用对象GUID来表示 |
| rightMask | RightMaskType | 可以忽略 |
| 返回值 | DataSet | 返回上面提到的权限的集合。在中智应该返回空的DataTable（没有行数据） |

DataSet GetFunctionsRoles(string appCodeName, string funcCodeNames)

得到某个应用下，某几个权限（功能）所属于的角色。

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| appCodeName | 字符串 | 应用的代码名称。在中智，可以使用对象GUID来表示 |
| funcCodeNames | 字符串 | 权限的代码名称。多个代码名称使用逗号分隔 |
| 返回值 | DataSet | 返回上面提到的角色的集合。在中智应该返回空的DataTable（没有行数据） |

DataSet GetRolesUsers(string orgRoot, string appCodeName, string roleCodeNames, DelegationMaskType delegationMask, SidelineMaskType sidelineMask, string extAttr)

返回某个角色下的所有人员。

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| orgRoot | 字符串 | 可以忽略 |
| appCodeName | 字符串 | 应用的代码名称。在中智，可以使用对象GUID来表示 |
| roleCodeNames | 字符串 | 角色的代码名称。多个代码名称使用逗号分隔 |
| delegationMask | DelegationMaskType | 可以忽略 |
| sidelineMask | SidelineMaskType | 可以忽略 |
| extAttr | 字符串 | 可以忽略 |
| 返回值 | DataSet | 返回之前提到的人员的DataTable |

DataSet GetUserRoles(string userValue, string appCodeName, UserValueType userValueType, RightMaskType rightMask, DelegationMaskType delegationMask)

返回某个应用下，人员所属的角色。

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| userValue | 字符串 | 用户的ID，ID的类型取决于userValueType参数 |
| appCodeName | 字符串 | 应用的代码名称。在中智，可以使用对象GUID来表示 |
| userValueType | UserValueType | userValue参数所表示的ID类型，请参考表格后的代码。由于历史原因，这里和前面组织机构人员中的ID类型还不一样 |
| rightMask | RightMaskType | 可以忽略 |
| delegationMask | DelegationMaskType | 可以忽略 |
| 返回值 | DataSet | 返回上面提到的角色的集合 |

public enum UserValueType

{

// Summary:

//     登录名

LogonName = 1,

//

// Summary:

//     人员全路径

AllPath = 2,

//

// Summary:

//     人员编号

PersonID = 3,

//

// Summary:

//     IC卡号

ICCode = 4,

//

// Summary:

//     人员Guid值

Guid = 8,

//

// Summary:

//     根据唯一索引查询(为配合南京海关统一平台切换，新增加字段ID[自增唯一字段])

Identity = 16,

}

DataSet GetUserPermissions(string userValue, string appCodeName, UserValueType userValueType, RightMaskType rightMask, DelegationMaskType delegationMask)

返回用户在某个应用下所拥有的权限。

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| userValue | 字符串 | 用户的ID，ID的类型取决于userValueType参数 |
| appCodeName | 字符串 | 应用的代码名称。在中智，可以使用对象GUID来表示 |
| userValueType | UserValueType | userValue参数所表示的ID类型，请参考表格后的代码。由于历史原因，这里和前面组织机构人员中的ID类型还不一样 |
| rightMask | RightMaskType | 可以忽略 |
| delegationMask | DelegationMaskType | 可以忽略 |
| 返回值 | DataSet | 返回上面提到的权限的集合。在中智，返回空的DataTable |

DataSet GetChildrenInRoles(string orgRoot, string appCodeName, string roleCodeNames, bool doesMixSort, bool doesSortRank, bool includeDelegate)

返回某个应用下角色中直接包含的对象。

|  |  |  |
| --- | --- | --- |
| 参数名 | 类型 | 说明 |
| orgRoot | 字符串 | 可以忽略 |
| appCodeName | 字符串 | 应用的代码名称。在中智，可以使用对象GUID来表示 |
| roleCodeNames | 字符串 | 角色的代码名称。多个角色使用逗号分隔 |
| doesMixSort | 布尔 | 可以忽略 |
| doesSortRank | 布尔 | 可以忽略 |
| includeDelegate | 布尔 | 可以忽略 |
| 返回值 | DataSet | 返回上面提到的通用的组织和人员的DataTable，字段按照通用结构来定义 |

如何验证实现的Web Service的正确性。

除了必要的单元测试，我们可以通过一个测试页面进行测试：

<http://localhost/MCSWebApp/PermissionCenterServices/CommandConsole/command.aspx>

在测试之前，需要修改config目录下oguAndPermissions.config中的web serivce地址。

计算机生成了可选文字:
文 # 〔 F 〕 
考 〔 E 〕 
胄 输 入 命 令 
令 帮 助 
getAIlCh11dren ： 
查 音 (V) 
收 戾 （ 匍 
工 具 (T) 
助 〔 H 〕 
执 行 （ E } 
getAlICh11dren 
getApp11cat10ns ： getAppI £ 以 t10n5 
getGrouplJsers getGroupUsers (groupFuIIPath) 
getCb)ectByFuI Path ： getOb3eCtByFuIIPath 
getCb)eCtByID ： getCb)eCtByID 
getPerf111SSIonRelat1veRoIes getPernISS10nReIat1veRoIes 
(perm1SS10nCOdeNElTVe) 
getPerm1SSIons ： getPer"1SS10ns fappCodeName) 
getRoles · getRoles (appCodeName) 
getRoot ： getROOt

可以键入一些命令进行测试。最典型的上来就执行getRoot，以及getUser。

这些测试正常后，再去运行流程编辑界面进行验证。