



通用权限管理系统 接口参数说明文档



开发单位：杭州海日涵科技
开发部门：产品部
编制日期：2011 年 04 月



目录

一、	用户帐户接口	17
1.1	用户名是否重复	17
1.2	添加用户	17
1.3	按部门获取部门用户	17
1.4	获取实体	18
1.5	获得用户	18
1.6	按主键获取获得用户	18
1.7	按角色获取用户	18
1.8	查询用户列表	19
1.9	设置用户状态	19
1.10	设置用户的默认角色	19
1.11	批量设置用户的默认角色	20
1.12	单个删除	20
1.13	批量删除	20
1.14	批量做删除标志	21
1.15	获取用户类型	21
1.16	用户是否在某个角色里	21
1.17	更新用户	21
1.18	获取职员角色列表	22
1.19	批量加入到职员	22
1.20	批量移出角色	22
1.21	批量保存	23
1.22	获得用户的组织机构兼职情况	23
1.23	用户帐户添加到组织机构	23
1.24	批量删除	24
1.25	清除用户归属的角色	24



二、	组织机构（部门）管理接口	25
2.1	添加用户到组织机构	25
2.2	添加	25
2.2	获得列表	26
2.3	获得列表	26
2.4	获取内部部门	26
2.5	获取公司列表	26
2.6	获得部门的列表	27
2.7	获取子节点成员	27
2.8	获取父子节点成员	27
2.9	搜索部门	28
2.10	获取实体	28
2.11	更新一个	28
2.12	删除一个	28
2.13	批量删除	29
2.14	批量做删除标志	29
2.15	批量保存	29
2.16	移动数据	30
2.17	批量移动数据	30
2.18	保存组织机构编号	30
2.19	保存组织机构排序顺序	31
三、	角色接口	32
3.1	添加角色	32
3.2	获取列表	32
3.3	按组织机构获取角色	32
3.4	获取实体	33
3.5	编辑	33
3.6	获取角色列表	33



3.7 查询	34
3.8 批量保存	34
3.9 移动数据	34
3.10 批量移动数据	34
3.10 排序顺序	35
3.11 获得角色中的用户主键	35
3.12 用户添加到角色	35
3.13 将用户从角色中移除	36
3.13 删除	36
3.13 批量删除	36
3.14 批量做删除标志	37
3.15 清除角色用户关联	37
四、权限接口	38
4.1 用户是否在指定的角色里	38
4.2 当前用户是否有相应的操作权限	38
4.3 某个用户是否有相应的权限	38
4.4 某个角色是否有相应的权限	39
4.5 当前用户是否超级管理员	39
4.6 某个用户是否超级管理员	39
4.7 获得当前用户的所有权限列表	39
4.8 获得某个用户的所有权限列表	40
4.9 当前用户是否对某个模块有相应的权限	40
4.10 某个用户是否对某个模块有相应的权限	40
4.11 获得用户的数据集权限范围	41
4.12 按某个数据集权限获取组织列表	41
4.13 按某个数据集权限获取组织主键数组	41
4.14 按某个数据集权限获取角色列表	42
4.15 按某个数据集权限获取角色主键数组	42



4.16 按某个权限域获取用户列表	42
4.17 按某个数据集权限获取用户主键数组	43
4.18 按某个权限域获取模块列表	43
4.19 授权权限的权限列表	43
4.20 获取角色权限主键数组	44
4.21 授予角色的权限	44
4.22 授予角色的权限	44
4.23 授予角色的权限	45
4.24 撤消角色的权限	45
4.25 撤消角色的权限	45
4.26 撤消角色的权限	46
4.27 获取角色的某个权限域的组织范围	46
4.28 获取角色的某个权限域的组织范围	46
4.29 获取角色的某个权限域的组织范围	47
4.30 授予角色的某个权限域的组织范围	47
4.31 撤消角色的某个权限域的组织范围	47
4.32 授予角色的某个权限域的组织范围	48
4.33 撤消角色的某个权限域的组织范围	48
4.34 授予角色的某个权限域的组织范围	48
4.35 撤消角色的某个权限域的组织范围	49
4.36 获取角色授权权限列表	49
4.37 授予角色的授权权限范围	49
4.38 授予角色的授权权限范围	50
4.39 清除权限	50
4.40 获取用户权限主键数组	50
4.41 授予用户的权限	51
4.42 授予用户的权限	51
4.43 撤消用户的权限	51



4.44 撤消用户的权限	52
4.45 获取用户的某个权限域的组织范围	52
4.46 设置用户的某个权限域的组织范围	52
4.47 设置用户的某个权限域的组织范围	53
4.48 获取用户的某个权限域的用户范围	53
4.49 设置用户的某个权限域的用户范围	53
4.50 设置用户的某个权限域的用户范围	54
4.51 获取用户的某个权限域的用户范围	54
4.52 设置用户的某个权限域的用户范围	54
4.53 获取用户的某个权限域的用户范围	55
4.54 获取用户授权权限列表	55
4.55 授予用户的授权权限范围	56
4.56 撤消用户的授权权限范围	56
4.57 清除权限	56
4.58 清除权限	57
4.59. 获得用户有权限的模块	57
4.60 获得用户有权限的模块	57
4.61 获得用户模块权限范围主键数组	57
4.62 授予用户模块的权限范围	58
4.63 授予用户模块的权限范围	58
4.64 撤消用户模块的权限范围	58
4.65 撤消用户模块的权限范围	59
4.66 取用户模块权限范围主键数组	59
4.67 授予用户模块的权限范围	59
4.68 授予用户模块的权限范围	60
4.69 撤消用户模块的权限范围	60
4.70 撤消用户模块的权限范围	60
4.71 获取资源权限主键数组	61



4.72 授予资源的权限	61
4.73 撤消资源的权限	61
4.74 获取资源权限范围主键数组	62
4.75 获取数据集权限目标主键	62
4.76 授予资源的权限范围	63
4.77 授予数据集权限	63
4.78 撤消资源的权限范围	63
4.79 撤销数据集权限	64
4.80 清除数据集权限	64
4.81 获取用户的某个资源的权限范围(列表资源).....	65
4.82 获取用户的某个资源的权限范围(树型资源).....	65
五、权限配置的操作接口	66
5.1 获取权限列表	66
5.2 获得列表	66
5.3 添加一个权限	66
5.4 获取一个操作权限	67
5.5 按编码获取一个权限	67
5.6 按编码获取名称	68
5.7 更新	68
5.7 移动权限数据	68
5.8 批量移动权限数据	69
5.9 删除一个权限	69
5.10 批量删除权限	69
5.11 批量做删除标志	69
5.12 保存权限排序顺序	70
六、员工接口	71
6.1 获取内部通讯录	71
6.2 更新通讯地址	71



6.3 更新通讯地址	71
6.4 添加员工	72
6.5 更新员工	72
6.6 获得职员列表	72
6.7 按主键获取	73
6.8 按部门获取部门职员	73
6.9 获得职员列表	73
6.10 获得职员列表	74
6.11 设置职员关联的用户	74
6.12 删除职员关联的用户	74
6.13 单个删除	74
6.14 批量删除	75
6.15 量做删除标志	75
6.16 移动数据	75
6.17 量移动数据	76
6.18 批量保存职员	76
6.19 重新排序数据	76
七、模块接口	77
7.1 获取列表	77
7.2 获取实体	77
7.3 获取名称	77
7.3 添加名称	78
7.4 更新	78
7.5 获得列表	78
7.6 单个删除	79
7.7 批量删除模块	79
7.8 批量做删除标志	79
7.9 移动数据	79



7.10 批量移动数据	80
7.11 批量保存	80
7.12 保存排序顺序	80
7.13 模块权限列表	81
7.14 权限模块列表	81
7.15 模块关联权限	81
7.16 模块关联权限	82
7.17 撤消模块关联权限	82
7.18 撤消模块关联权限	82
八、登录	83
8.1 获得登录用户列表	83
8.2 获得内部员工列表	83
8.3 激活帐户	83
8.4 按唯一识别码登录	84
8.5 按用户名登录	84
8.6 登录	84
8.7 操作员退出应用程序	85
8.8 检查在线状态(服务器专用).....	85
8.9 设置密码	85
8.10 修改密码	86
九、数据库访问通用类标准接口	87
9.1 执行查询	87
9.2 执行查询	87
9.3 执行查询	87
9.4 执行查询	88
9.5 填充数据表	88
9.6 填充数据表	88
十、参数服务接口	89



10.1 获取参数	89
10.2 更新参数设置	89
10.3 获取记录	89
10.4 获取记录	90
10.5 删除	90
10.6 删除	90
10.7 删除	91
10.8 批量删除	91
十一、服务工厂接口定义接口	92
11.1 初始化服务	92
11.2 创建登录服务	92
11.3 创建序列服务	92
11.4 创建用户服务	92
11.5 创建日志服务	92
11.6 创建异常服务	93
11.7 创建权限管理服务	93
11.8 创建部门管理服务	93
11.9 创建选项服务	93
11.10 创建选项明细服务	94
11.11 创建模块服务	94
11.12 创建职员服务	94
11.13 创建角色服务	94
11.14 创建消息服务	94
11.15 创建文件服务	95
11.16 创建目录服务	95
11.17 创建参数服务	95
11.18 创建权限服务	95
11.19 创建名片服务	96



11.20 创建业务数据库服务	96
11.21 创建用户中心数据库服务	96
11.22 创建用当前 workflow 服务	96
11.23 创建工作流审核管理服务	96
11.23 创建工作流管理服务	97
十二、日志服务接口	98
12.1 写入日志	98
12.2 离开时的日志记录	98
12.3 获取日志	98
12.3 重置访问情况	99
12.4 按日期获取日志	99
12.5 按模块获取日志	99
12.6 按职员获取日志	100
12.7 删除日志	100
12.8 批量删除日志	100
12.9 清除日志	101
十三、异常接口	102
13.1 获取列表	102
13.2 批量删除异常	102
13.3 批量删除异常	102
13.4 清除	102
十四、名片管理接口	104
14.1 获取全部列表	104
14.2 获取公开列表	104
14.3 获取用户列表	104
14.4 获取某个用户名片	104
14.5 获取某个用户实体	105
14.5 添加名片	105



14.6 添加名片	105
14.6 更新名片	106
14.5 更新名片	106
14.6 批量删除	106
14.6 批量保存	107
十五、序列(流水号)接口	108
15.1 添加序列流水号	108
15.2 获取全部序列列表	108
15.3 获取序列	108
15.3 获取原序列号	109
15.4 获取新序列号	109
15.5 获取序列号	109
15.6 获取倒序序列号	110
15.7 批量重置	110
15.8 批量删除日志	110
15.9 批量删除权限	111
十六、选项（字典）管理接口	112
16.1 获取列表	112
16.2 获取列表	112
16.3 获取子列表	112
16.4 添加编码	112
16.5 创建表结构	113
16.6 更新编码	113
16.7 删除	113
16.8 批量删除编码	114
16.9 批量移动编码	114
16.10 批量保存编码	114
16.11 获取列表（详细）	115



16.12 获取子列表（详细）	115
16.13 获取下拉框数据（详细）	115
16.14 获取获取下拉框数据（详细）	115
16.15 获取列表（详细）	116
16.16 添加编码（详细）	116
16.17 更新编码（详细）	116
16.18 删除（详细）	117
16.19 批量删除编码（详细）	117
16.20 批量移动编码（详细）	117
16.21 批量保存编码（详细）	118
十七、文件夹及文件接口	119
17.1 获取文件夹列表	119
17.2 获取一条	119
17.3 按目录获取列表	119
17.4 添加文件夹	119
17.5 添加文件夹	120
17.6 更新文件夹	120
17.7 重命名文件夹	121
17.8 查询文件夹	121
17.9 删除文件夹	121
17.10 批量删除文件夹	122
17.11 移动文件夹	122
17.12 批量移动文件夹	122
17.13 批量保存文件夹	122
17.14 文件夹是否已存在	123
17.15 下载文件	123
17.16 上传文件	123
17.17 获取实体	124



17.18	按文件夹获取列表	124
17.19	按文件夹删除文件	124
17.19	添加文件	125
17.20	更新文件	125
17.21	更新文件	126
17.22	重命名文件	126
17.23	查询文件	126
17.24	移动数据	127
17.25	批量移动数据	127
17.26	删除	127
17.27	批量删除	128
17.28	批量保存	128
十八、即时通讯组件接口		129
18.1	获得内部部门（公司的组织机构）	129
18.2	发送即时消息	129
18.3	批量发送即时消息	129
18.4	获取用户的新信息	130
18.5	获取特定用户的新信息	130
18.6	阅读短信	130
18.7	检查在线状态	131
18.8	获取在线用户列表	131
十九、工作日志服务接口		132
19.1	获取一个工作日志列表	132
19.2	获取职员工作日志	132
19.3	更新工作日志	132
19.4	添加工作日志	133
19.5	批量删除	133
19.6	查询审核列表	133



19.7	批量设置审核状态	134
19.8	批量保存	134
19.9	获取项目列表	134
19.10	获取项目全称	135
19.11	检查工作日志	135
19.12	求某天工时之和	135
二十、工作流服务接口		136
20.1	获取工作流步骤定义列表	136
20.2	添加工作流	136
20.3	批量删除	136
20.3	批量保存	137
20.4	获取待审核主键	137
20.5	获取列表	137
20.6	获取监控列表	137
20.7	获取待审核列表	138
20.8	获取审核历史明细	138
20.9	最终审核通过	138
20.10	撤消审批流程中的单据	139
20.11	审核驳回	139
20.12	自动工作流审核通过	139
20.13	开始审核	139
20.14	审核驳回	140
20.15	按角色审核开始	140
20.16	审核开始	141
20.17	下个流程发送给谁	141
20.18	下个流程发送给谁	141
20.19	审核驳回	142
20.20	按用户审核启动	142



20.21	添加 workflow	143
20.22	批量删除组织机构	143
20.23	批量保存	143
20.24	单个删除 workflow	144
20.25	获取 workflow 列表	144
20.26	获取 workflow	144
20.27	批量做删除标志	145
20.28	更新 workflow	145



一、用户帐户接口

1.1 用户名是否重复

```
/// <summary>
/// 用户名是否重复
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="userName">用户名</param>
/// <returns>已存在</returns>
[OperationContract]
bool Exists(BaseUserInfo userInfo, string userName);
```

1.2 添加用户

```
/// <summary>
/// 添加用户
/// </summary>
/// <param name="userInfo">操作员信息</param>
/// <param name="userEntity">用户实体</param>
/// <param name="statusCode">状态码</param>
/// <param name="statusMessage">状态信息</param>
/// <returns>主键</returns>
[OperationContract]
string AddUser(BaseUserInfo userInfo, BaseUserEntity userEntity, out string statusCode,
out string statusMessage);
```

1.3 按部门获取部门用户

```
/// <summary>
/// 按部门获取部门用户
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="departmentId">部门主键</param>
/// <param name="containChildren">含子部门</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByDepartment(BaseUserInfo userInfo, string departmentId, bool
```



```
containChildren);
```

1.4 获取实体

```
/// <summary>
/// 获取实体
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="id">主键</param>
/// <returns>实体</returns>
[OperationContract]
BaseUserEntity GetEntity(BaseUserInfo userInfo, string id);
```

1.5 获得用户

```
/// <summary>
/// 获得用户列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDT(BaseUserInfo userInfo);
```

1.6 按主键获取获得用户

```
/// <summary>
/// 按主键获取
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByIds(BaseUserInfo userInfo, string[] ids);
```

1.7 按角色获取用户

```
/// <summary>
```



```
/// 按角色获取用户
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByRole(BaseUserInfo userInfo, string roleId);
```

1.8 查询用户列表

```
/// <summary>
/// 查询用户列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userName">用户名</param>
/// <param name="auditStates">用户状态</param>
/// <param name="roleIds">角色主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable Search(BaseUserInfo userInfo, string userName, string auditStates, string[]
roleIds);
```

1.9 设置用户状态

```
/// <summary>
/// 设置用户状态
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <param name="auditStates">审核状态</param>
/// <returns>影响行数</returns>
[OperationContract]
int SetUserAuditStates(BaseUserInfo userInfo, string[] ids, AuditStatus auditStatus);
```

1.10 设置用户的默认角色

```
/// <summary>
/// 设置用户的默认角色
/// </summary>
```



```
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="roleId">角色主键</param>
/// <returns>影响的行数</returns>
[OperationContract]
int SetDefaultRole(BaseUserInfo userInfo, string userId, string roleId);
```

1.11 批量设置用户的默认角色

```
/// <summary>
/// 批量设置用户的默认角色
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userIds">用户主键</param>
/// <param name="roleId">角色主键</param>
/// <returns>影响的行数</returns>
[OperationContract]
int BatchSetDefaultRole(BaseUserInfo userInfo, string[] userIds, string roleId);
```

1.12 单个删除

```
/// <summary>
/// 单个删除
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int Delete(BaseUserInfo userInfo, string id);
```

1.13 批量删除

```
/// <summary>
/// 批量删除
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
```



```
int BatchDelete(BaseUserInfo userInfo, string[] ids);
```

1.14 批量做删除标志

```
/// <summary>
/// 批量做删除标志
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int SetDeleted(BaseUserInfo userInfo, string[] ids);
```

1.15 获取用户类型

```
/// <summary>
/// 获取用户类型
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetRoleDT(BaseUserInfo userInfo);
```

1.16 用户是否在某个角色里

```
/// <summary>
/// 用户是否在某个角色里
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="roleCode">角色编号</param>
/// <returns>存在</returns>
[OperationContract]
bool UserInRole(BaseUserInfo userInfo, string userId, string roleCode);
```

1.17 更新用户

```
/// <summary>
```



```
/// 更新用户
/// </summary>
/// <param name="userInfo">操作员信息</param>
/// <param name="userEntity">用户实体</param>
/// <param name="statusCode">状态码</param>
/// <param name="statusMessage">状态信息</param>
/// <returns>影响行数</returns>
[OperationContract]
int UpdateUser(BaseUserInfo userInfo, BaseUserEntity userEntity, out string statusCode,
out string statusMessage);
```

1.18 获取职员角色列表

```
/// <summary>
/// 获取职员角色列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="staffId">职员主键</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetUserRoleIds(BaseUserInfo userInfo, string userId);
```

1.19 批量加入到职员

```
/// <summary>
/// 批量加入到职员
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="staffId">职员主键</param>
/// <param name="addToRoleIds">加入角色主键集</param>
/// <returns>影响的行数</returns>
[OperationContract]
int AddUserToRole(BaseUserInfo userInfo, string userId, string[] addToRoleIds);
```

1.20 批量移出角色

```
/// <summary>
/// 批量移出角色
/// </summary>
/// <param name="userInfo">用户</param>
```



```
/// <param name="staffId">职员主键</param>
/// <param name="removeRoleIds">移出角色主键集</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RemoveUserFromRole(BaseUserInfo userInfo, string userId, string[] removeRoleIds);
```

1.21 批量保存

```
/// <summary>
/// 批量保存
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="dataTable">数据表</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchSave(BaseUserInfo userInfo, DataTable dataTable);
```

1.22 获得用户的组织机构兼职情况

```
/// <summary>
/// 获得用户的组织机构兼职情况
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="userId">用户主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetUserOrganizedDT(BaseUserInfo userInfo, string userId);
```

1.23 用户帐户添加到组织机构

```
/// <summary>
/// 用户帐户添加到组织机构
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="userOrganizeEntity">用户组织关系</param>
/// <param name="statusCode">状态码</param>
/// <param name="statusMessage">状态信息</param>
/// <returns>主键</returns>
[OperationContract]
string AddUserToOrganize(BaseUserInfo userInfo, BaseUserOrganizeEntity
```



```
userOrganizeEntity, out string statusCode, out string statusMessage);
```

1.24 批量删除

```
/// <summary>
/// 批量删除
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchDeleteUserOrganize(BaseUserInfo userInfo, string[] ids);
```

1.25 清除用户归属的角色

```
/// <summary>
/// 清除用户归属的角色
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int ClearUserRole(BaseUserInfo userInfo, string userId);
```




二、组织机构（部门）管理接口

2.1 添加用户到组织机构

```
/// <summary>
/// 添加用户到组织机构
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="organizeEntity">实体</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状消息</param>
/// <returns>主键</returns>
[OperationContract]
string Add(BaseUserInfo userInfo, BaseOrganizeEntity organizeEntity, out string
statusCode, out string statusMessage);
```

2.2 添加

```
/// <summary>
/// 添加
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="parentId">父级主键</param>
/// <param name="code">编号</param>
/// <param name="fullName">名称</param>
/// <param name="categoryId">类别</param>
/// <param name="outerPhone">外线</param>
/// <param name="innerPhone">内线</param>
/// <param name="fax">传真</param>
/// <param name="enabled">有效</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状消息</param>
/// <returns>主键</returns>
[OperationContract]
string AddByDetail(BaseUserInfo userInfo, string parentId, string code, string fullName,
string categoryId, string outerPhone, string innerPhone, string fax, bool enabled, out
string statusCode, out string statusMessage);
```



2.2 获得列表

```
/// <summary>
/// 获得列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDT(BaseUserInfo userInfo);
```

2.3 获得列表

```
/// <summary>
/// 获得列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="parentId">父亲节点主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByParent(BaseUserInfo userInfo, string parentId);
```

2.4 获取内部部门

```
/// <summary>
/// 获取内部部门
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="organizeId">主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetInnerOrganizeDT(BaseUserInfo userInfo, string organizeId);
```

2.5 获取公司列表

```
/// <summary>
/// 获取公司列表
/// </summary>
```



```
/// <param name="userInfo">用户</param>
/// <param name="organizeId">主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetCompanyDT(BaseUserInfo userInfo, string organizeId);
```

2.6 获得部门的列表

```
/// <summary>
/// 获得部门的列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="organizeId">主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDepartmentDT(BaseUserInfo userInfo, string organizeId);
```

2.7 获取子节点成员

```
/// <summary>
/// 获取子节点成员
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="organizeId">主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetChildrenStaffs(BaseUserInfo userInfo, string organizeId);
```

2.8 获取父子节点成员

```
/// <summary>
/// 获取父子节点成员
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="organizeId">主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetParentChildrenStaffs(BaseUserInfo userInfo, string organizeId);
```



2.9 搜索部门

```
/// <summary>
/// 搜索部门
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="organizeId">主键</param>
/// <param name="search">查询字符</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable Search(BaseUserInfo userInfo, string organizeId, string searchValue);
```

2.10 获取实体

```
/// <summary>
/// 获取实体
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="id">主键</param>
/// <returns>实体</returns>
[OperationContract]
BaseOrganizeEntity GetEntity(BaseUserInfo userInfo, string id);
```

2.11 更新一个

```
/// <summary>
/// 更新一个
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="organizeEntity">实体</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状态消息</param>
/// <returns>影响行数</returns>
[OperationContract]
int Update(BaseUserInfo userInfo, BaseOrganizeEntity organizeEntity, out string
statusCode, out string statusMessage);
```

2.12 删除一个

```
/// <summary>
```



```
/// 删除一个
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int Delete(BaseUserInfo userInfo, string id);
```

2.13 批量删除

```
/// <summary>
/// 批量删除
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchDelete(BaseUserInfo userInfo, string[] ids);
```

2.14 批量做删除标志

```
/// <summary>
/// 批量做删除标志
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int SetDeleted(BaseUserInfo userInfo, string[] ids);
```

2.15 批量保存

```
/// <summary>
/// 批量保存
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="dataTable">数据表</param>
/// <returns>影响行数</returns>
[OperationContract]
```



```
int BatchSave(BaseUserInfo userInfo, DataTable dataTable);
```

2.16 移动数据

```
/// <summary>
/// 移动数据
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="organizeId">组织机构主键</param>
/// <param name="parentId">父结点主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int MoveTo(BaseUserInfo userInfo, string organizeId, string parentId);
```

2.17 批量移动数据

```
/// <summary>
/// 批量移动数据
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="organizeIds">组织机构主键</param>
/// <param name="parentId">父结点主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchMoveTo(BaseUserInfo userInfo, string[] organizeIds, string parentId);
```

2.18 保存组织机构编号

```
/// <summary>
/// 保存组织机构编号
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <param name="codes">编号数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchSetCode(BaseUserInfo userInfo, string[] ids, string[] codes);
```



2.19 保存组织机构排序顺序

```
/// <summary>
/// 保存组织机构排序顺序
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchSetSortCode(BaseUserInfo userInfo, string[] ids);
```



三、角色接口

3.1 添加角色

```
/// <summary>
/// 添加角色
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="roleEntity">实体</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状态信息</param>
/// <returns>主键</returns>
[OperationContract]
string Add(BaseUserInfo userInfo, BaseRoleEntity roleEntity, out string statusCode, out
string statusMessage);
```

3.2 获取列表

```
/// <summary>
/// 获取列表
/// </summary>
/// <param name="userInfo">操作员</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDT(BaseUserInfo userInfo);
```

3.3 按组织机构获取角色

```
/// <summary>
/// 按组织机构获取角色
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="organizeId">组织机构主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByOrganize(BaseUserInfo userInfo, string organizeId);
```




3.4 获取实体

```
/// <summary>
/// 获取实体
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="id">主键</param>
/// <returns>实体</returns>
[OperationContract]
BaseRoleEntity GetEntity(BaseUserInfo userInfo, string id);
```

3.5 编辑

```
/// <summary>
/// 编辑
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="roleEntity">实体</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状态信息</param>
/// <returns>影响行数</returns>
[OperationContract]
int Update(BaseUserInfo userInfo, BaseRoleEntity roleEntity, out string statusCode, out string statusMessage);
```

3.6 获取角色列表

```
/// <summary>
/// 获取角色列表
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="ids">角色主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByIds(BaseUserInfo userInfo, string[] ids);
```



3.7 查询

```
/// <summary>
/// 查询
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="organizeId">组织机构主键</param>
/// <param name="searchValue">查询字符串</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable Search(BaseUserInfo userInfo, string organizeId, string searchValue);
```

3.8 批量保存

```
/// <summary>
/// 批量保存
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="roleEntites">角色列表</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchSave(BaseUserInfo userInfo, List<BaseRoleEntity> roleEntites);
```

3.9 移动数据

```
/// <summary>
/// 移动数据
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="id">主键</param>
/// <param name="targetId">目标主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int MoveTo(BaseUserInfo userInfo, string id, string targetId);
```

3.10 批量移动数据

```
/// <summary>
```



```
/// 批量移动数据
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="ids">主键数组</param>
/// <param name="targetId">目标主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchMoveTo(BaseUserInfo userInfo, string[] ids, string targetId);
```

3.10 排序顺序

```
/// <summary>
/// 排序顺序
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="organizeId">组织机构主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int ResetSortCode(BaseUserInfo userInfo, string organizeId);
```

3.11 获得角色中的用户主键

```
/// <summary>
/// 获得角色中的用户主键
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="roleId">角色主键</param>
/// <returns>用户主键</returns>
[OperationContract]
string[] GetRoleUserIds(BaseUserInfo userInfo, string roleId);
```

3.12 用户添加到角色

```
/// <summary>
/// 用户添加到角色
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="roleId">角色主键</param>
/// <param name="addUserIds">用户主键</param>
```



```
/// <returns>影响行数</returns>
[OperationContract]
int AddUserToRole(BaseUserInfo userInfo, string roleId, string[] addUserIds);
```

3.13 将用户从角色中移除

```
/// <summary>
/// 将用户从角色中移除
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="roleId">角色主键</param>
/// <param name="userIds">用户主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int RemoveUserFromRole(BaseUserInfo userInfo, string roleId, string[] userIds);
```

3.13 删除

```
/// <summary>
/// 删除
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="id">主键</param>
/// <returns>数据表</returns>
[OperationContract]
int Delete(BaseUserInfo userInfo, string id);
```

3.13 批量删除

```
/// <summary>
/// 批量删除
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchDelete(BaseUserInfo userInfo, string[] ids);
```



3.14 批量做删除标志

```
/// <summary>
/// 批量做删除标志
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int SetDeleted(BaseUserInfo userInfo, string[] ids);
```

3.15 清除角色用户关联

```
/// <summary>
/// 清除角色用户关联
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int ClearRoleUser(BaseUserInfo userInfo, string roleId);
```



四、权限接口

4.1 用户是否在指定的角色里

```
/// <summary>
/// 01. 用户是否在指定的角色里
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="roleName">角色名称</param>
/// <returns>在角色里</returns>
[OperationContract]
bool IsInRole(BaseUserInfo userInfo, string userId, string roleName);
```

4.2 当前用户是否有相应的操作权限

```
/// <summary>
/// 02. 当前用户是否有相应的操作权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="permissionItemCode">操作权限编号</param>
/// <returns>是否有权</returns>
[OperationContract]
bool IsAuthorized(BaseUserInfo userInfo, string permissionItemCode);
```

4.3 某个用户是否有相应的权限

```
/// <summary>
/// 03. 某个用户是否有相应的权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemCode">操作权限编号</param>
/// <returns>是否有权</returns>
[OperationContract]
bool IsAuthorizedByUser(BaseUserInfo userInfo, string userId, string permissionItemCode);
```



4.4 某个角色是否有相应的权限

```
/// <summary>
/// 04. 某个角色是否有相应的权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="permissionItemCode">权限编号</param>
/// <returns>是否有权限</returns>
[OperationContract]
bool IsAuthorizedByRole(BaseUserInfo userInfo, string roleId, string
permissionItemCode);
```

4.5 当前用户是否超级管理员

```
/// <summary>
/// 05. 当前用户是否超级管理员
/// </summary>
/// <param name="userInfo"></param>
/// <returns></returns>
[OperationContract]
bool IsAdministrator(BaseUserInfo userInfo);
```

4.6 某个用户是否超级管理员

```
/// <summary>
/// 06. 某个用户是否超级管理员
/// </summary>
/// <param name="userInfo"></param>
/// <param name="userId"></param>
/// <returns></returns>
[OperationContract]
bool IsAdministratorByUser(BaseUserInfo userInfo, string userId);
```

4.7 获得当前用户的所有权限列表

```
/// <summary>
/// 07. 获得当前用户的所有权限列表
```



```
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetPermissionDT(BaseUserInfo userInfo);
```

4.8 获得某个用户的所有权限列表

```
/// <summary>
/// 08. 获得某个用户的所有权限列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetPermissionDTByUser(BaseUserInfo userInfo, string userId);
```

4.9 当前用户是否对某个模块有相应的权限

```
/// <summary>
/// 09. 当前用户是否对某个模块有相应的权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="moduleCode">模块编号</param>
/// <returns>是否有权</returns>
[OperationContract]
bool ModuleIsAuthorized(BaseUserInfo userInfo, string moduleCode);
```

4.10 某个用户是否对某个模块有相应的权限

```
/// <summary>
/// 10. 某个用户是否对某个模块有相应的权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="moduleCode">模块编号</param>
/// <returns>是否有权</returns>
[OperationContract]
bool ModuleIsAuthorizedByUser(BaseUserInfo userInfo, string userId, string moduleCode);
```




4.11 获得用户的数据集权限范围

```
/// <summary>
/// 11. 获得用户的数据集权限范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemScopeCode">权限域编号</param>
/// <returns>数据集权限范围</returns>
[OperationContract]
PermissionScope GetUserPermissionScope(BaseUserInfo userInfo, string userId, string
permissionItemScopeCode);
```

4.12 按某个数据集权限获取组织列表

```
/// <summary>
/// 12. 按某个数据集权限获取组织列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemScopeCode">权限域编号</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetOrganizedDTByPermission(BaseUserInfo userInfo, string userId, string
permissionItemScopeCode);
```

4.13 按某个数据集权限获取组织主键数组

```
/// <summary>
/// 13. 按某个数据集权限获取组织主键数组
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemScopeCode">权限域编号</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetOrganizeIdsByPermission(BaseUserInfo userInfo, string userId, string
permissionItemScopeCode);
```



4.14 按某个数据集权限获取角色列表

```
/// <summary>
/// 14. 按某个数据集权限获取角色列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemScopeCode">权限域编号</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetRoleDTByPermission(BaseUserInfo userInfo, string userId, string
permissionItemScopeCode);
```

4.15 按某个数据集权限获取角色主键数组

```
/// <summary>
/// 15. 按某个数据集权限获取角色主键数组
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemScopeCode">权限域编号</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetRoleIdsByPermission(BaseUserInfo userInfo, string userId, string
permissionItemScopeCode);
```

4.16 按某个权限域获取用户列表

```
/// <summary>
/// 16. 按某个权限域获取用户列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemScopeCode">权限域编号</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetUserDTByPermission(BaseUserInfo userInfo, string userId, string
permissionItemScopeCode);
```



4.17 按某个数据集权限获取用户主键数组

```
/// <summary>
/// 17. 按某个数据集权限获取用户主键数组
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemScopeCode">权限域编号</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetUserIdsByPermission(BaseUserInfo userInfo, string userId, string
permissionItemScopeCode);
```

4.18 按某个权限域获取模块列表

```
/// <summary>
/// 18. 按某个权限域获取模块列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemCode">权限域编号</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetModuleDTByPermission(BaseUserInfo userInfo, string userId, string
permissionItemCode);
```

4.19 授权权限的权限列表

```
/// <summary>
/// 19. 有授权权限的权限列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemCode">权限域编号</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetPermissionItemDTByPermission(BaseUserInfo userInfo, string userId, string
permissionItemCode);
```



4.20 获取角色权限主键数组

```
/// <summary>
/// 20. 获取角色权限主键数组
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetRolePermissionIds(BaseUserInfo userInfo, string roleId);
```

4.21 授予角色的权限

```
/// <summary>
/// 21. 授予角色的权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="grantPermissionIds">授予权限数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int GrantRolePermissions(BaseUserInfo userInfo, string roleId, string[]
grantPermissionIds);
```

4.22 授予角色的权限

```
/// <summary>
/// 22. 授予角色的权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="grantPermissionId">授予权限</param>
/// <returns>影响的行数</returns>
[OperationContract]
string GrantRolePermission(BaseUserInfo userInfo, string roleId, string
grantPermissionId);
```



4.23 授予角色的权限

```
/// <summary>
/// 23. 授予角色的权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="grantPermissionId">授予权限</param>
/// <returns>影响的行数</returns>
[OperationContract]
string GrantRolePermissionById(BaseUserInfo userInfo, string roleId, string
grantPermissionId);
```

4.24 撤消角色的权限

```
/// <summary>
/// 24. 撤消角色的权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="grantPermissionIds">授予权限数组</param>
/// <param name="revokePermissionIds">撤消权限数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeRolePermissions(BaseUserInfo userInfo, string roleId, string[]
revokePermissionIds);
```

4.25 撤消角色的权限

```
/// <summary>
/// 25. 撤消角色的权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="revokePermissionId">撤消权限数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeRolePermission(BaseUserInfo userInfo, string roleId, string
revokePermissionId);
```



4.26 撤消角色的权限

```
/// <summary>
/// 26. 撤消角色的权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="revokePermissionId">撤消权限数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeRolePermissionById(BaseUserInfo userInfo, string roleId, string
revokePermissionId);
```

4.27 获取角色的某个权限域的组织范围

```
/// <summary>
/// 27. 获取角色的某个权限域的组织范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="permissionItemId">权限主键</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetRoleScopeUserIds(BaseUserInfo userInfo, string roleId, string
permissionItemCode);
```

4.28 获取角色的某个权限域的组织范围

```
/// <summary>
/// 28. 获取角色的某个权限域的组织范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="permissionItemId">权限主键</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetRoleScopeRoleIds(BaseUserInfo userInfo, string roleId, string
permissionItemCode);
```



4.29 获取角色的某个权限域的组织范围

```
/// <summary>
/// 29. 获取角色的某个权限域的组织范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="permissionItemCode">权限编号</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetRoleScopeOrganizeIds(BaseUserInfo userInfo, string roleId, string
permissionItemCode);
```

4.30 授予角色的某个权限域的组织范围

```
/// <summary>
/// 30. 授予角色的某个权限域的组织范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="permissionItemId">权限主键</param>
/// <param name="grantUserIds">授予用户主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int GrantRoleUserScopes(BaseUserInfo userInfo, string roleId, string permissionItemCode,
string[] grantUserIds);
```

4.31 撤消角色的某个权限域的组织范围

```
/// <summary>
/// 31. 撤消角色的某个权限域的组织范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="permissionItemId">权限主键</param>
/// <param name="revokeUserIds">撤消的用户主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeRoleUserScopes(BaseUserInfo userInfo, string roleId, string
```



```
permissionItemCode, string[] revokeUserIds);
```

4.32 授予角色的某个权限域的组织范围

```
/// <summary>
/// 32. 授予角色的某个权限域的组织范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="permissionItemId">权限主键</param>
/// <param name="grantRoleIds">授予角色主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int GrantRoleRoleScopes(BaseUserInfo userInfo, string roleId, string permissionItemCode,
string[] grantRoleIds);
```

4.33 撤消角色的某个权限域的组织范围

```
/// <summary>
/// 33. 撤消角色的某个权限域的组织范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="permissionItemId">权限主键</param>
/// <param name="revokeRoleIds">撤消的角色主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeRoleRoleScopes(BaseUserInfo userInfo, string roleId, string
permissionItemCode, string[] revokeRoleIds);
```

4.34 授予角色的某个权限域的组织范围

```
/// <summary>
/// 34. 授予角色的某个权限域的组织范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="permissionItemCode">权限主键</param>
/// <param name="grantOrganizeIds">授予组织主键数组</param>
```




```
/// <returns>影响的行数</returns>
[OperationContract]
int GrantRoleOrganizeScopes(BaseUserInfo userInfo, string roleId, string
permissionItemCode, string[] grantOrganizeIds);
```

4.35 撤消角色的某个权限域的组织范围

```
/// <summary>
/// 35. 撤消角色的某个权限域的组织范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="permissionItemCode">权限主键</param>
/// <param name="revokeOrganizeIds">撤消的组织主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeRoleOrganizeScopes(BaseUserInfo userInfo, string roleId, string
permissionItemCode, string[] revokeOrganizeIds);
```

4.36 获取角色授权权限列表

```
/// <summary>
/// 36. 获取角色授权权限列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="permissionItemCode">操作权限项编号</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetRoleScopePermissionIds(BaseUserInfo userInfo, string roleId, string
permissionItemCode);
```

4.37 授予角色的授权权限范围

```
/// <summary>
/// 37. 授予角色的授权权限范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
```



```
/// <param name="permissionItemCode">操作权限编号</param>
/// <param name="grantPermissionIds">授予的权限主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int GrantRolePermissionItemScopes(BaseUserInfo userInfo, string roleId, string
permissionItemCode, string[] grantPermissionIds);
```

4.38 授予角色的授权权限范围

```
/// <summary>
/// 38. 授予角色的授权权限范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="permissionItemCode">操作权限编号</param>
/// <param name="revokePermissionIds">撤销的权限主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeRolePermissionItemScopes(BaseUserInfo userInfo, string roleId, string
permissionItemIdCode, string[] revokePermissionIds);
```

4.39 清除权限

```
/// <summary>
/// 39. 清除权限
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="id">主键</param>
/// <returns>数据表</returns>
[OperationContract]
int ClearRolePermission(BaseUserInfo userInfo, string id);
[OperationContract]
int ClearRolePermissionScope(BaseUserInfo userInfo, string id, string
permissionItemIdCode);
```

4.40 获取用户权限主键数组

```
/// <summary>
/// 40. 获取用户权限主键数组
/// </summary>
```



```
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetUserPermissionIds(BaseUserInfo userInfo, string userId);
```

4.41 授予用户的权限

```
/// <summary>
/// 41. 授予用户的权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="grantPermissionIds">授予权限数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int GrantUserPermissions(BaseUserInfo userInfo, string userId, string[]
grantPermissionIds);
```

4.42 授予用户的权限

```
/// <summary>
/// 42. 授予用户的权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="grantPermissionId">授予权限数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
string GrantUserPermissionById(BaseUserInfo userInfo, string userId, string
grantPermissionId);
```

4.43 撤消用户的权限

```
/// <summary>
/// 43. 撤消用户的权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
```



```
/// <param name="revokePermissionIds">撤销权限数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeUserPermission(BaseUserInfo userInfo, string userId, string[]
revokePermissionIds);
```

4.44 撤销用户的权限

```
/// <summary>
/// 44. 撤销用户的权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="revokePermissionId">撤销权限</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeUserPermissionById(BaseUserInfo userInfo, string userId, string
revokePermissionId);
```

4.45 获取用户的某个权限域的组织范围

```
/// <summary>
/// 45. 获取用户的某个权限域的组织范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemIdCode">权限主键</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetUserScopeOrganizeIds(BaseUserInfo userInfo, string userId, string
permissionItemIdCode);
```

4.46 设置用户的某个权限域的组织范围

```
/// <summary>
/// 46. 设置用户的某个权限域的组织范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
```



```
/// <param name="permissionItemIdCode">权限主键</param>
/// <param name="grantOrganizeIds">授予的组织主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int GrantUserOrganizeScopes(BaseUserInfo userInfo, string userId, string
permissionItemIdCode, string[] grantOrganizeIds);
```

4.47 设置用户的某个权限域的组织范围

```
/// <summary>
/// 47. 设置用户的某个权限域的组织范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemIdCode">权限主键</param>
/// <param name="revokeOrganizeIds">撤消的组织主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeUserOrganizeScopes(BaseUserInfo userInfo, string userId, string
permissionItemIdCode, string[] revokeOrganizeIds);
```

4.48 获取用户的某个权限域的用户范围

```
/// <summary>
/// 48. 获取用户的某个权限域的用户范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemIdCode">权限主键</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetUserScopeUserIds(BaseUserInfo userInfo, string userId, string
permissionItemIdCode);
```

4.49 设置用户的某个权限域的用户范围

```
/// <summary>
/// 49. 设置用户的某个权限域的用户范围
/// </summary>
```



```
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemIdCode">权限主键</param>
/// <param name="grantUserIds">授予的用户主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int GrantUserUserScopes(BaseUserInfo userInfo, string userId, string
permissionItemIdCode, string[] grantUserIds);
```

4.50 设置用户的某个权限域的用户范围

```
/// <summary>
/// 50. 设置用户的某个权限域的用户范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemIdCode">权限主键</param>
/// <param name="revokeUserIds">撤销的用户主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeUserUserScopes(BaseUserInfo userInfo, string userId, string
permissionItemIdCode, string[] revokeUserIds);
```

4.51 获取用户的某个权限域的用户范围

```
/// <summary>
/// 51. 获取用户的某个权限域的用户范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemIdCode">权限主键</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetUserScopeRoleIds(BaseUserInfo userInfo, string userId, string
permissionItemIdCode);
```

4.52 设置用户的某个权限域的用户范围

```
/// <summary>
```



```
/// 52. 设置用户的某个权限域的用户范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemIdCode">权限主键</param>
/// <param name="grantUserIds">授予的用户主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int GrantUserRoleScopes(BaseUserInfo userInfo, string userId, string
permissionItemIdCode, string[] grantRoleIds);
```

4.53 获取用户的某个权限域的用户范围

```
/// <summary>
/// 53. 设置用户的某个权限域的用户范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemIdCode">权限主键</param>
/// <param name="revokeUserIds">撤销的用户主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeUserRoleScopes(BaseUserInfo userInfo, string userId, string
permissionItemIdCode, string[] revokeRoleIds);
```

4.54 获取用户授权权限列表

```
/// <summary>
/// 54. 获取用户授权权限列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetUserScopePermissionIds(BaseUserInfo userInfo, string userId, string
permissionItemIdCode);
```



4.55 授予用户的授权权限范围

```
/// <summary>
/// 55. 授予用户的授权权限范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="grantPermissionIds">授予的权限主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int GrantUserPermissionItemScopes(BaseUserInfo userInfo, string userId, string
permissionItemIdCode, string[] grantPermissionIds);
```

4.56 撤消用户的授权权限范围

```
/// <summary>
/// 56. 撤消用户的授权权限范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="revokePermissionIds">撤消的权限主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeUserPermissionItemScopes(BaseUserInfo userInfo, string userId, string
permissionItemIdCode, string[] revokePermissionIds);
```

4.57 清除权限

```
/// <summary>
/// 57. 清除权限
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="id">主键</param>
/// <returns>数据表</returns>
[OperationContract]
int ClearUserPermission(BaseUserInfo userInfo, string id);
```




4.58 清除权限

```
/// <summary>
/// 58.
/// </summary>
/// <param name="userInfo"></param>
/// <param name="id"></param>
/// <param name="permissionItemIdCode"></param>
/// <returns></returns>
[OperationContract]
int ClearUserPermissionScope(BaseUserInfo userInfo, string id, string
permissionItemIdCode);
```

4.59.获得用户有权限的模块

```
/// <summary>
/// 59. 获得用户有权限的模块
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetModuleDT(BaseUserInfo userInfo);
```

4.60 获得用户有权限的模块

```
/// <summary>
/// 60. 获得用户有权限的模块
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetModuleDTByUser(BaseUserInfo userInfo, string userId);
```

4.61 获得用户模块权限范围主键数组

```
/// <summary>
/// 61. 获取用户模块权限范围主键数组
```



```
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="permissionItemCode">权限编号</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetUserScopeModuleIds(BaseUserInfo userInfo, string userId, string
permissionItemCode);
```

4.62 授予用户模块的权限范围

```
/// <summary>
/// 62. 授予用户模块的权限范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="grantModuleId">授予模块主键</param>
/// <returns>影响的行数</returns>
[OperationContract]
string GrantUserModuleScope(BaseUserInfo userInfo, string userId, string
permissionScopeItemCode, string grantModuleId);
```

4.63 授予用户模块的权限范围

```
/// <summary>
/// 63. 授予用户模块的权限范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="grantModuleIds">授予模块主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int GrantUserModuleScopes(BaseUserInfo userInfo, string userId, string
permissionScopeItemCode, string[] grantModuleIds);
```

4.64 撤消用户模块的权限范围

```
/// <summary>
/// 64. 撤消用户模块的权限范围
```



```
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="revokeModuleId">撤销模块主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeUserModuleScope(BaseUserInfo userInfo, string userId, string
permissionScopeItemCode, string revokeModuleId);
```

4.65 撤销用户模块的权限范围

```
/// <summary>
/// 65. 撤销用户模块的权限范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="revokeModuleIds">撤销模块主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeUserModuleScopes(BaseUserInfo userInfo, string userId, string
permissionScopeItemCode, string[] revokeModuleIds);
```

4.66 取用户模块权限范围主键数组

```
/// <summary>
/// 66. 获取用户模块权限范围主键数组
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="permissionItemCode">权限编号</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetRoleScopeModuleIds(BaseUserInfo userInfo, string roleId, string
permissionItemCode);
```

4.67 授予用户模块的权限范围

```
/// <summary>
/// 67. 授予用户模块的权限范围
```



```
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="grantModuleIds">授予模块主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int GrantRoleModuleScopes(BaseUserInfo userInfo, string roleId, string
permissionItemScopeCode, string[] grantModuleIds);
```

4.68 授予用户模块的权限范围

```
/// <summary>
/// 68. 授予用户模块的权限范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="grantModuleId">授予模块主键</param>
/// <returns>影响的行数</returns>
[OperationContract]
string GrantRoleModuleScope(BaseUserInfo userInfo, string roleId, string
permissionItemScopeCode, string grantModuleId);
```

4.69 撤消用户模块的权限范围

```
/// <summary>
/// 69. 撤消用户模块的权限范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="revokeModuleIds">撤消模块主键数组</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeRoleModuleScopes(BaseUserInfo userInfo, string roleId, string
permissionItemScopeCode, string[] revokeModuleIds);
```

4.70 撤消用户模块的权限范围

```
/// <summary>
/// 70. 撤消用户模块的权限范围
/// </summary>
```



```
/// <param name="userInfo">用户</param>
/// <param name="roleId">角色主键</param>
/// <param name="revokeModuleId">撤消模块主键</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeRoleModuleScope(BaseUserInfo userInfo, string roleId, string
permissionItemScopeCode, string revokeModuleId);
```

4.71 获取资源权限主键数组

```
/// <summary>
/// 71. 获取资源权限主键数组
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="resourceCategory">资源分类</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetResourcePermissionIds(BaseUserInfo userInfo, string resourceCategory,
string resourceId);
```

4.72 授予资源的权限

```
/// <summary>
/// 72. 授予资源的权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="resourceCategory">资源分类</param>
/// <param name="resourceId">资源主键</param>
/// <param name="grantPermissionID">权限主键</param>
/// <returns>影响的行数</returns>
[OperationContract]
int GrantResourcePermission(BaseUserInfo userInfo, string resourceCategory, string
resourceId, string[] grantPermissionIds);
```

4.73 撤消资源的权限

```
/// <summary>
/// 73. 撤消资源的权限
/// </summary>
/// <param name="userInfo">用户</param>
```



```
/// <param name="resourceCategory">资源分类</param>
/// <param name="resourceId">资源主键</param>
/// <param name="revokePermissionId">权限主键</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokeResourcePermission(BaseUserInfo userInfo, string resourceCategory, string
resourceId, string[] revokePermissionIds);
```

4.74 获取资源权限范围主键数组

```
/// <summary>
/// 74. 获取资源权限范围主键数组
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="resourceCategory">资源分类</param>
/// <param name="resourceId">资源主键</param>
/// <param name="targetCategory">目标类别</param>
/// <param name="permissionItemCode">权限编号</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetPermissionScopeTargetIds(BaseUserInfo userInfo, string resourceCategory,
resourceId, string targetCategory, string permissionItemCode);
```

4.75 获取数据集权限目标主键

```
/// <summary>
/// 75. 获取数据集权限目标主键
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="resourceCategory">资源类别</param>
/// <param name="targetResourceId">资源主键</param>
/// <param name="targetResourceCategory">目标资源</param>
/// <param name="permissionItemCode">权限编号</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetPermissionScopeResourceIds(BaseUserInfo userInfo, string resourceCategory,
string targetResourceId, string targetResourceCategory, string permissionItemCode);
```



4.76 授予资源的权限范围

```
/// <summary>
/// 76. 授予资源的权限范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="resourceCategory">资源分类</param>
/// <param name="resourceId">资源主键</param>
/// <param name="targetCategory">目标类别</param>
/// <param name="grantTargetIds">目标主键数组</param>
/// <param name="permissionItemId">权限主键</param>
/// <returns>影响的行数</returns>
[OperationContract]
int GrantPermissionScopeTargets(BaseUserInfo userInfo, string resourceCategory, string
resourceId, string targetCategory, string[] grantTargetIds, string permissionItemId);
```

4.77 授予数据集权限

```
/// <summary>
/// 77. 授予数据集权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="resourceCategory">资源类别</param>
/// <param name="resourceIds">资源主键数组</param>
/// <param name="targetCategory">目标资源类别</param>
/// <param name="grantTargetId">目标资源主键</param>
/// <param name="permissionItemId">权限主键</param>
/// <returns>影响行数</returns>
int GrantPermissionScopeTarget(BaseUserInfo userInfo, string resourceCategory, string[]
resourceIds, string targetCategory, string grantTargetId, string permissionItemId);
```

4.78 撤消资源的权限范围

```
/// <summary>
/// 78. 撤消资源的权限范围
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="resourceCategory">资源分类</param>
/// <param name="resourceId">资源主键</param>
```



```
/// <param name="targetCategory">目标类别</param>
/// <param name="revokeTargetIds">目标主键数组</param>
/// <param name="permissionItemId">权限主键</param>
/// <returns>影响的行数</returns>
[OperationContract]
int RevokePermissionScopeTargets(BaseUserInfo userInfo, string resourceCategory,
string resourceId, string targetCategory, string[] revokeTargetIds, string
permissionItemId);
```

4.79 撤销数据集权限

```
/// <summary>
/// 79. 撤销数据集权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="resourceCategory">资源类别</param>
/// <param name="resourceIds">资源主键数组</param>
/// <param name="targetCategory">目标分类</param>
/// <param name="revokeTargetId">目标主键</param>
/// <param name="permissionItemId">权限主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int RevokePermissionScopeTarget(BaseUserInfo userInfo, string resourceCategory,
string[] resourceIds, string targetCategory, string revokeTargetId, string
permissionItemId);
```

4.80 清除数据集权限

```
/// <summary>
/// 80. 清除数据集权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="resourceCategory">资源类别</param>
/// <param name="resourceId">资源主键</param>
/// <param name="targetCategory">目标类别</param>
/// <param name="permissionItemId">权限主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int ClearPermissionScopeTarget(BaseUserInfo userInfo, string resourceCategory, string
resourceId, string targetCategory, string permissionItemId);
```




4.81 获取用户的某个资源的权限范围(列表资源)

```
/// <summary>
/// 81. 获取用户的某个资源的权限范围(列表资源)
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="targetCategory">目标类别</param>
/// <param name="permissionItemCode">权限编号</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetResourceScopeIds(BaseUserInfo userInfo, string userId, string
targetCategory, string permissionItemCode);
```

4.82 获取用户的某个资源的权限范围(树型资源)

```
/// <summary>
/// 82. 获取用户的某个资源的权限范围(树型资源)
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="targetCategory">目标类别</param>
/// <param name="permissionItemCode">权限编号</param>
/// <param name="childrens">是否含子节点</param>
/// <returns>主键数组</returns>
[OperationContract]
string[] GetTreeResourceScopeIds(BaseUserInfo userInfo, string userId, string
targetCategory, string permissionItemCode, bool childrens);
```



五、权限配置的操作接口

5.1 获取权限列表

```
/// <summary>
/// 获得权限列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDT(BaseUserInfo userInfo);
```

5.2 获得列表

```
/// <summary>
/// 获得列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="parentId">父亲节点主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByParent(BaseUserInfo userInfo, string parentId);
```

5.3 添加一个权限

```
/// <summary>
/// 添加一个权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="permissionItemEntity">权限实体</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状消息</param>
/// <returns>数据表</returns>
[OperationContract]
string Add(BaseUserInfo userInfo, BasePermissionItemEntity permissionItemEntity, out
string statusCode, out string statusMessage);

/// <summary>
```



```
/// 按明细添加一个操作权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="code">编号</param>
/// <param name="fullName">名称</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状消息</param>
/// <returns>主键</returns>
[OperationContract]
string AddByDetail(BaseUserInfo userInfo, string code, string fullName, out string
statusCode, out string statusMessage);
```

5.4 获取一个操作权限

```
/// <summary>
/// 获取一个操作权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>实体</returns>
[OperationContract]
BasePermissionItemEntity GetEntity(BaseUserInfo userInfo, string id);
```

5.5 按编码获取一个权限

```
/// <summary>
/// 按编码获取一个权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="code">编号</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状消息</param>
/// <returns>实体</returns>
[OperationContract]
BasePermissionItemEntity GetByCode(BaseUserInfo userInfo, string code, out string
statusCode, out string statusMessage);
```



5.6 按编码获取名称

```
/// <summary>
/// 按编号获取名称
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="code">编号</param>
/// <returns>实体</returns>
[OperationContract]
BasePermissionItemEntity GetByEntityCode(BaseUserInfo userInfo, string code);
```

5.7 更新

```
/// <summary>
/// 更新
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="permissionItemEntity">权限实体</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状态消息</param>
/// <returns>影响行数</returns>
[OperationContract]
int Update(BaseUserInfo userInfo, BasePermissionItemEntity permissionItemEntity, out
string statusCode, out string statusMessage);
```

5.7 移动权限数据

```
/// <summary>
/// 移动权限数据
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <param name="parentId">父结点主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int MoveTo(BaseUserInfo userInfo, string id, string parentId);
```



5.8 批量移动权限数据

```
/// <summary>
/// 批量移动权限数据
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键组</param>
/// <param name="parentId">父结点主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchMoveTo(BaseUserInfo userInfo, string[] ids, string parentId);
```

5.9 删除一个权限

```
/// <summary>
/// 删除一个权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int Delete(BaseUserInfo userInfo, string id);
```

5.10 批量删除权限

```
/// <summary>
/// 批量删除权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchDelete(BaseUserInfo userInfo, string[] ids);
```

5.11 批量做删除标志

```
/// <summary>
/// 批量做删除标志
```



```
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int SetDeleted(BaseUserInfo userInfo, string[] ids);

/// <summary>
/// 批量保存权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="dataTable">权限数据表</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchSave(BaseUserInfo userInfo, DataTable dataTable);
```

5.12 保存权限排序顺序

```
/// <summary>
/// 保存权限排序顺序
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchSetSortCode(BaseUserInfo userInfo, string[] ids);
```



六、员工接口

6.1 获取内部通讯录

```
/// <summary>
/// 获取内部通讯录
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="organizeId">组织机构主键</param>
/// <param name="search">查询内容</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetAddressDT(BaseUserInfo userInfo, string organizeId, string searchValue);
```

6.2 更新通讯地址

```
/// <summary>
/// 更新通讯地址
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="staffEntity">实体</param>
/// <returns>影响行数</returns>
[OperationContract]
int UpdateAddress(BaseUserInfo userInfo, BaseStaffEntity staffEntity, out string
statusCode, out string statusMessage);
```

6.3 更新通讯地址

```
/// <summary>
/// 更新通讯地址
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="staffEntites">实体</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchUpdateAddress(BaseUserInfo userInfo, List<BaseStaffEntity> staffEntites, out
string statusCode, out string statusMessage);
```



6.4 添加员工

```
/// <summary>
/// 添加员工
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="staffEntity">实体</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状消息</param>
/// <returns>主键</returns>
[OperationContract]
string AddStaff(BaseUserInfo userInfo, BaseStaffEntity staffEntity, out string
statusCode, out string statusMessage);
```

6.5 更新员工

```
/// <summary>
/// 更新员工
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="staffEntity">实体</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状消息</param>
/// <returns>影响行数</returns>
[OperationContract]
int UpdateStaff(BaseUserInfo userInfo, BaseStaffEntity staffEntity, out string
statusCode, out string statusMessage);
```

6.6 获得职员列表

```
/// <summary>
/// 获得职员列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDT(BaseUserInfo userInfo);

[OperationContract]
```




```
BaseStaffEntity GetEntity(BaseUserInfo userInfo, string id);
```

6.7 按主键获取

```
/// <summary>
/// 按主键获取
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByIds(BaseUserInfo userInfo, string[] ids);
```

6.8 按部门获取部门职员

```
/// <summary>
/// 按部门获取部门职员
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="DepartmentId">部门主键</param>
/// <param name="containChildren">含子部门</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByDepartment(BaseUserInfo userInfo, string departmentId, bool containChildren);
```

6.9 获得职员列表

```
/// <summary>
/// 获得职员列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="organizeId">组织机构主键</param>
/// <param name="containChildren">含子部门</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByOrganize(BaseUserInfo userInfo, string organizeId, bool containChildren);
```



6.10 获得职员列表

```
/// <summary>
/// 获得职员列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="organizeId">组织机构主键</param>
/// <param name="search">查询</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable Search(BaseUserInfo userInfo, string organizeId, string searchValue);
```

6.11 设置职员关联的用户

```
/// <summary>
/// 设置职员关联的用户
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="staffId">职员主键</param>
/// <param name="userId">用户主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int SetStaffUser(BaseUserInfo userInfo, string staffId, string userId);
```

6.12 删除职员关联的用户

```
/// <summary>
/// 删除职员关联的用户
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="staffId">职员主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int DeleteUser(BaseUserInfo userInfo, string staffId);
```

6.13 单个删除

```
/// <summary>
```



```
/// 单个删除
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int Delete(BaseUserInfo userInfo, string id);
```

6.14 批量删除

```
/// <summary>
/// 批量删除
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchDelete(BaseUserInfo userInfo, string[] ids);
```

6.15 量做删除标志

```
/// <summary>
/// 批量做删除标志
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int SetDeleted(BaseUserInfo userInfo, string[] ids);
```

6.16 移动数据

```
/// <summary>
/// 移动数据
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="organizeIds">主键</param>
/// <param name="organizeId">组织机构主键</param>
/// <returns>影响行数</returns>
```



[OperationContract]

int MoveTo(BaseUserInfo userInfo, string id, string organizeId);

6.17 量移动数据

/// <summary>

/// 批量移动数据

/// </summary>

/// <param name="userInfo">用户</param>

/// <param name="organizeIds">主键数组</param>

/// <param name="organizeId">组织机构主键</param>

/// <returns>影响行数</returns>

[OperationContract]

int BatchMoveTo(BaseUserInfo userInfo, string[] ids, string organizeId);

6.18 批量保存职员

/// <summary>

/// 批量保存职员

/// </summary>

/// <param name="userInfo">用户</param>

/// <param name="dataTable">数据表</param>

/// <param name="organizeId">组织机构主键</param>

/// <returns>影响行数</returns>

[OperationContract]

int BatchSave(BaseUserInfo userInfo, DataTable dataTable);

6.19 重新排序数据

/// <summary>

/// 重新排序数据

/// </summary>

/// <param name="userInfo">用户</param>

/// <returns>影响行数</returns>

[OperationContract]

int ResetSortCode(BaseUserInfo userInfo);



七、模块接口

7.1 获取列表

```
/// <summary>
/// 获取列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDT(BaseUserInfo userInfo);
```

7.2 获取实体

```
/// <summary>
/// 获取实体
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="id">主键</param>
/// <returns>实体</returns>
[OperationContract]
BaseModuleEntity GetEntity(BaseUserInfo userInfo, string id);
```

7.3 获取名称

```
/// <summary>
/// 获取名称
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="code">编号</param>
/// <returns>名称</returns>
[OperationContract]
string GetFullNameByCode(BaseUserInfo userInfo, string code);
```



7.3 添加名称

```
/// <summary>
/// 添加
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="moduleEntity">实体</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状消息</param>
/// <returns>主键</returns>
[OperationContract]
string Add(BaseUserInfo userInfo, BaseModuleEntity moduleEntity, out string statusCode,
out string statusMessage);
```

7.4 更新

```
/// <summary>
/// 更新
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="moduleEntity">实体</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状消息</param>
/// <returns>影响行数</returns>
[OperationContract]
int Update(BaseUserInfo userInfo, BaseModuleEntity moduleEntity, out string statusCode,
out string statusMessage);
```

7.5 获得列表

```
/// <summary>
/// 获得列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="parentId">父结点主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByParent(BaseUserInfo userInfo, string parentId);
```



7.6 单个删除

```
/// <summary>
/// 单个删除
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int Delete(BaseUserInfo userInfo, string id);
```

7.7 批量删除模块

```
/// <summary>
/// 批量删除模块
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <param name="parentId">父结点主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchDelete(BaseUserInfo userInfo, string[] ids);
```

7.8 批量做删除标志

```
/// <summary>
/// 批量做删除标志
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int SetDeleted(BaseUserInfo userInfo, string[] ids);
```

7.9 移动数据

```
/// <summary>
/// 移动数据
```



```
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="organizeId">组织机构主键</param>
/// <param name="parentId">父结点主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int MoveTo(BaseUserInfo userInfo, string moduleId, string parentId);
```

7.10 批量移动数据

```
/// <summary>
/// 批量移动数据
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="organizeIds">组织机构主键数组</param>
/// <param name="parentId">父结点主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchMoveTo(BaseUserInfo userInfo, string[] moduleIds, string parentId);
```

7.11 批量保存

```
/// <summary>
/// 批量保存
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="dataTable">数据表</param>
/// <param name="parentId">父结点主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchSave(BaseUserInfo userInfo, DataTable dataTable);
```

7.12 保存排序顺序

```
/// <summary>
/// 保存排序顺序
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
```




```
/// <returns>影响行数</returns>
[OperationContract]
int SetSortCode(BaseUserInfo userInfo, string[] ids);
```

7.13 模块权限列表

```
/// <summary>
/// 模块权限列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="moduleId">模块主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetPermissionDT(BaseUserInfo userInfo, string moduleId);
```

7.14 权限模块列表

```
///<summary>
/// 权限模块列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="permissionItemId">权限主键</param>
/// <returns>数据表</returns>
[OperationContract]
string[] GetIDsByPermission(BaseUserInfo userInfo, string permissionItemId);
```

7.15 模块关联权限

```
/// <summary>
/// 模块关联权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="moduleId">模块主键</param>
/// <param name="permissionItemIds">权限主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchAddPermissions(BaseUserInfo userInfo, string moduleId, string[]
permissionItemIds);
```



7.16 模块关联权限

```
/// <summary>
/// 模块挂接权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="permissionItemId">权限主键</param>
/// <param name="moduleIds">模块主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchAddModules(BaseUserInfo userInfo, string permissionItemId, string[]
moduleIds);
```

7.17 撤消模块关联权限

```
/// <summary>
/// 撤消模块挂接权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="moduleId">模块主键</param>
/// <param name="permissionItemIds">权限主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchDletePermissions(BaseUserInfo userInfo, string moduleId, string[]
permissionItemIds);
```

7.18 撤消模块关联权限

```
/// <summary>
/// 撤消模块挂接权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="permissionItemId">权限主键</param>
/// <param name="modulesIds">模块主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchDleteModules(BaseUserInfo userInfo, string permissionItemId, string[]
modulesIds);
```



八、登录

8.1 获得登录用户列表

```
/// <summary>
/// 获得登录用户列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetUserDT(BaseUserInfo userInfo);
```

8.2 获得内部员工列表

```
/// <summary>
/// 获得内部员工列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns></returns>
[OperationContract]
DataTable GetStaffUserDT(BaseUserInfo userInfo);
```

8.3 激活帐户

```
/// <summary>
/// 激活帐户
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="openId">唯一识别码</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状消息</param>
/// <returns>用户实体</returns>
[OperationContract]
BaseUserInfo AccountActivation(BaseUserInfo userInfo, string openId, out string
statusCode, out string statusMessage);
```



8.4 按唯一识别码登录

```
/// <summary>
/// 按唯一识别码登录
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="openId">唯一识别码</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状消息</param>
/// <returns>用户实体</returns>
[OperationContract]
BaseUserInfo LogOnByOpenId(BaseUserInfo userInfo, string openId, out string statusCode,
out string statusMessage);
```

8.5 按用户名登录

```
/// <summary>
/// 按用户名登录
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userName">用户名</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状消息</param>
/// <returns>用户实体</returns>
[OperationContract]
BaseUserInfo LogOnByUserName(BaseUserInfo userInfo, string userName, out string
statusCode, out string statusMessage);
```

8.6 登录

```
/// <summary>
/// 登录
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userName">用户名</param>
/// <param name="password">密码</param>
/// <param name="ipAddress">IP地址</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状消息</param>
```



```
/// <returns>登录实体类</returns>
[OperationContract]
BaseUserInfo UserLogOn(BaseUserInfo userInfo, string userName, string password, out
string statusCode, out string statusMessage);
```

8.7 操作员退出应用程序

```
/// <summary>
/// 操作员退出应用程序
/// </summary>
/// <param name="userInfo">用户</param>
[OperationContract]
void OnExit(BaseUserInfo userInfo);
```

8.8 检查在线状态(服务器专用)

```
/// <summary>
/// 检查在线状态(服务器专用)
/// </summary>
/// <returns>离线人数</returns>
[OperationContract]
int ServerCheckOnline();
```

8.9 设置密码

```
/// <summary>
/// 设置密码
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userIds">被设置的用户主键</param>
/// <param name="password">新密码</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状态消息</param>
/// <returns>影响行数</returns>
[OperationContract]
int SetPassword(BaseUserInfo userInfo, string[] userIds, string password, out string
statusCode, out string statusMessage);
```



8.10 修改密码

```
/// <summary>
/// 修改密码
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="oldPassword">原始密码</param>
/// <param name="newPassword">新密码</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状消息</param>
/// <returns>影响行数</returns>
[OperationContract]

int ChangePassword(BaseUserInfo userInfo, string oldPassword, string newPassword, out
string statusCode, out string statusMessage);
```



九、数据库访问通用类标准接口

9.1 执行查询

```
/// <summary>
/// 执行查询
/// </summary>
/// <param name="commandText">查询</param>
/// <returns>影响行数</returns>
[OperationContract]
int ExecuteNonQuery(BaseUserInfo userInfo, string commandText);
```

9.2 执行查询

```
/// <summary>
/// 执行查询
/// </summary>
/// <param name="commandText">查询</param>
/// <param name="dbParameters">参数集</param>
/// <returns>影响行数</returns>
[OperationContract]
int ExecuteNonQuery(BaseUserInfo userInfo, string commandText, DbParameter[]
dbParameters);
```

9.3 执行查询

```
/// <summary>
/// 执行查询
/// </summary>
/// <param name="commandText">sql查询</param>
/// <returns>Object</returns>
[OperationContract]
object ExecuteScalar(BaseUserInfo userInfo, string commandText);
```



9.4 执行查询

```
/// <summary>
/// 执行查询
/// </summary>
/// <param name="commandText">sql查询</param>
/// <param name="dbParameters">参数集</param>
/// <returns>Object</returns>
[OperationContract]
object ExecuteScalar(BaseUserInfo userInfo, string commandText, DbParameter[]
dbParameters);
```

9.5 填充数据表

```
/// <summary>
/// 填充数据表
/// </summary>
/// <param name="commandText">查询</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable Fill(BaseUserInfo userInfo, string commandText);
```

9.6 填充数据表

```
/// <summary>
/// 填充数据表
/// </summary>
/// <param name="commandText">sql查询</param>
/// <param name="dbParameters">参数集</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable Fill(BaseUserInfo userInfo, string commandText, DbParameter[] dbParameters);
```




十、参数服务接口

10.1 获取参数

```
/// <summary>
/// 获取参数
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="categoryId">类别主键</param>
/// <param name="flagID">标志主键</param>
/// <param name="paramFlagCode">编码</param>
/// <returns>数据集</returns>
[OperationContract]
string GetParameter(BaseUserInfo userInfo, string categoryId, string parameterId,
string parameterCode);
```

10.2 更新参数设置

```
/// <summary>
/// 更新参数设置
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="categoryId">类别主键</param>
/// <param name="parameterId">标志主键</param>
/// <param name="parameterCode">编码</param>
/// <param name="parameterContent">参数内容</param>
/// <returns>影响行数</returns>
[OperationContract]
int SetParameter(BaseUserInfo userInfo, string categoryId, string parameterId, string
parameterCode, string parameterContent);
```

10.3 获取记录

```
/// <summary>
/// 获取记录
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="categoryId">类别主键</param>
```



```
/// <param name="flagID">标志主键</param>
/// <param name="paramFlagCode">编码</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByParameter(BaseUserInfo userInfo, string categoryId, string
parameterId);
```

10.4 获取记录

```
/// <summary>
/// 获取记录
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="categoryId">类别主键</param>
/// <param name="parameterId">标志主键</param>
/// <param name="parameterCode">编码</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTPParameterCode(BaseUserInfo userInfo, string categoryId, string
parameterId, string parameterCode);
```

10.5 删除

```
/// <summary>
/// 删除
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="categoryId">类别主键</param>
/// <param name="flagID">标志主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int DeleteByParameter(BaseUserInfo userInfo, string categoryId, string parameterId);
```

10.6 删除

```
/// <summary>
/// 删除
/// </summary>
/// <param name="userInfo">用户</param>
```



```
/// <param name="categoryId">类别主键</param>
/// <param name="parameterId">标志主键</param>
/// <param name="parameterCode">标志编号</param>
/// <returns>影响行数</returns>
[OperationContract]
int DeleteByParameterCode(BaseUserInfo userInfo, string categoryId, string parameterId,
string parameterCode);
```

10.7 删除

```
/// <summary>
/// 删除
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int Delete(BaseUserInfo userInfo, string id);
```

10.8 批量删除

```
/// <summary>
/// 批量删除
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchDelete(BaseUserInfo userInfo, string[] ids);
```



十一、服务工厂接口定义接口

11.1 初始化服务

```
/// <summary>  
/// 初始化服务  
/// </summary>  
void InitService();
```

11.2 创建登录服务

```
/// <summary>  
/// 创建登录服务  
/// </summary>  
/// <returns>服务接口</returns>  
ILogOnService CreateLogOnService();
```

11.3 创建序列服务

```
/// <summary>  
/// 创建序列服务  
/// </summary>  
/// <returns>服务接口</returns>  
ISequenceService CreateSequenceService();
```

11.4 创建用户服务

```
/// <summary>  
/// 创建用户服务  
/// </summary>  
/// <returns>服务接口</returns>  
IUserService CreateUserService();
```

11.5 创建日志服务

```
/// <summary>  
/// 创建日志服务
```



```
/// </summary>
/// <returns>服务接口</returns>
ILogService CreateLogService();
```

11.6 创建异常服务

```
/// <summary>
/// 创建异常服务
/// </summary>
/// <returns>服务接口</returns>
IExceptionService CreateExceptionService();
```

11.7 创建权限管理服务

```
/// <summary>
/// 创建权限管理服务
/// </summary>
/// <returns>服务接口</returns>
IPermissionItemService CreatePermissionItemService();
```

11.8 创建部门管理服务

```
/// <summary>
/// 创建部门管理服务
/// </summary>
/// <returns>服务接口</returns>
IOrganizeService CreateOrganizeService();
```

11.9 创建选项服务

```
/// <summary>
/// 创建选项服务
/// </summary>
/// <returns>服务接口</returns>
IItemsService CreateItemsService();
```



11.10 创建选项明细服务

```
/// <summary>
/// 创建选项明细服务
/// </summary>
/// <returns>服务接口</returns>
IItemDetailsService CreateItemDetailsService();
```

11.11 创建模块服务

```
/// <summary>
/// 创建模块服务
/// </summary>
/// <returns>服务接口</returns>
IModuleService CreateModuleService();
```

11.12 创建职员服务

```
/// <summary>
/// 创建职员服务
/// </summary>
/// <returns>服务接口</returns>
IStaffService CreateStaffService();
```

11.13 创建角色服务

```
/// <summary>
/// 创建角色服务
/// </summary>
/// <returns>服务接口</returns>
IRoleService CreateRoleService();
```

11.14 创建消息服务

```
/// <summary>
```



```
/// 创建消息服务
/// </summary>
/// <returns>服务接口</returns>
IMessageService CreateMessageService();
```

11.15 创建文件服务

```
/// <summary>
/// 创建文件服务
/// </summary>
/// <returns>服务接口</returns>
IFileService CreateFileService();
```

11.16 创建目录服务

```
/// <summary>
/// 创建目录服务
/// </summary>
/// <returns>服务接口</returns>
IFolderService CreateFolderService();
```

11.17 创建参数服务

```
/// <summary>
/// 创建参数服务
/// </summary>
/// <returns>服务接口</returns>
IParameterService CreateParameterService();
```

11.18 创建权限服务

```
/// <summary>
/// 创建权限服务
/// </summary>
/// <returns>服务接口</returns>
IPermissionService CreatePermissionService();
```



11.19 创建名片服务

```
/// <summary>
/// 创建名片服务
/// </summary>
/// <returns>服务接口</returns>
IBusinessCardService CreateBusinessCardService();
```

11.20 创建业务数据库服务

```
/// <summary>
/// 创建业务数据库服务
/// </summary>
/// <returns>服务接口</returns>
IDbHelperService CreateBusinessDbHelperService();
```

11.21 创建用户中心数据库服务

```
/// <summary>
/// 创建用户中心数据库服务
/// </summary>
/// <returns>服务接口</returns>
IDbHelperService CreateUserCenterDbHelperService();
```

11.22 创建用当前 workflow 服务

```
/// <summary>
/// 创建当前 workflow 服务
/// </summary>
/// <returns>服务接口</returns>
IWorkflowCurrentService CreateWorkflowCurrentService();
```

11.23 创建工作流审核管理服务

```
/// <summary>
/// 创建工作流审核步骤管理服务
/// </summary>
```




```
/// <returns>服务接口</returns>
```

```
IWorkflowActivityAdminService CreateWorkflowActivityAdminService();
```

11.23 创建工作流管理服务

```
/// <summary>
```

```
/// 创建工作流管理服务
```

```
/// </summary>
```

```
/// <returns>服务接口</returns>
```

```
IWorkflowProcessAdminService CreateWorkflowProcessAdminService();
```



十二、日志服务接口

12.1 写入日志

```
/// <summary>
/// 写入日志
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="processID">服务</param>
/// <param name="processName">服务名称</param>
/// <param name="methodName">操作名称</param>
/// <param name="description">描述</param>
[OperationContract]
string WriteLog(BaseUserInfo userInfo, string processId, string processName, string
methodName);
```

12.2 离开时的日志记录

```
/// <summary>
/// 离开时的日志记录
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="logId">日志主键</param>
[OperationContract]
void WriteExit(BaseUserInfo userInfo, string logId);
```

12.3 获取日志

```
/// <summary>
/// 获取日志
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetLogGeneral(BaseUserInfo userInfo);
```



12.3 重置访问情况

```
/// <summary>
/// 重置访问情况
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">日志主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable ResetVisitInfo(BaseUserInfo userInfo, string[] ids);
```

12.4 按日期获取日志

```
/// <summary>
/// 按日期获取日志
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="beginDate">开始时间</param>
/// <param name="endDate">结束时间</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByDate(BaseUserInfo userInfo, string beginDate, string endDate);
```

12.5 按模块获取日志

```
/// <summary>
/// 按模块获取日志
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="moduleId">模块主键</param>
/// <param name="beginDate">开始时间</param>
/// <param name="endDate">结束时间</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByModule(BaseUserInfo userInfo, string moduleId, string beginDate,
string endDate);
```



12.6 按职员获取日志

```
/// <summary>
/// 按职员获取日志
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="userId">用户主键</param>
/// <param name="beginDate">开始时间</param>
/// <param name="endDate">结束时间</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByUser(BaseUserInfo userInfo, string userId, string beginDate, string
endDate);
```

12.7 删除日志

```
/// <summary>
/// 删除日志
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int Delete(BaseUserInfo userInfo, string id);
```

12.8 批量删除日志

```
/// <summary>
/// 批量删除日志
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchDelete(BaseUserInfo userInfo, string[] ids);
```



12.9 清除日志

```
/// <summary>
/// 清除日志
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable Truncate(BaseUserInfo userInfo);
```



十三、异常接口

13.1 获取列表

```
/// <summary>
/// 获取列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDT(BaseUserInfo userInfo);
```

13.2 批量删除异常

```
/// <summary>
/// 批量删除异常
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable Delete(BaseUserInfo userInfo, string id);
```

13.3 批量删除异常

```
/// <summary>
/// 批量删除异常
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchDelete(BaseUserInfo userInfo, string[] ids);
```

13.4 清除



```
/// <summary>
/// 清除
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>影响行数</returns>
[OperationContract]
int Truncate(BaseUserInfo userInfo);
```



十四、名片管理接口

14.1 获取全部列表

```
/// <summary>
/// 获取全部列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDT(BaseUserInfo userInfo);
```

14.2 获取公开列表

```
/// <summary>
/// 获取公开列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetPublicDT(BaseUserInfo userInfo);
```

14.3 获取用户列表

```
/// <summary>
/// 获取用户列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByUser(BaseUserInfo userInfo);
```

14.4 获取某个用户名片

```
/// <summary>
/// 获取某个
/// </summary>
```




```
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable Get(BaseUserInfo userInfo, string id);
```

14.5 获取某个用户实体

```
/// <summary>
/// 获取实体
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>数据表</returns>
[OperationContract]
BaseBusinessCardEntity GetEntity(BaseUserInfo userInfo, string id);
```

14.5 添加名片

```
/// <summary>
/// 添加
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="businessCardEntity">实体</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状态信息</param>
/// <returns>主键主键</returns>
[OperationContract]
string AddEntity(BaseUserInfo userInfo, BaseBusinessCardEntity businessCardEntity, out
string statusCode, out string statusMessage);
```

14.6 添加名片

```
/// <summary>
/// 添加
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="dataTable">数据表</param>
/// <param name="statusCode">返回状态码</param>
```



```
/// <param name="statusMessage">返回状态信息</param>
/// <returns>数据表</returns>
[OperationContract]
string Add(BaseUserInfo userInfo, DataTable dataTable, out string statusCode, out string
statusMessage);
```

14.6 更新名片

```
/// <summary>
/// 更新
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="businessCardEntity">实体</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状态信息</param>
/// <returns>影响行数</returns>
[OperationContract]
int UpdateEntity(BaseUserInfo userInfo, BaseBusinessCardEntity businessCardEntity, out
string statusCode, out string statusMessage);
```

14.5 更新名片

```
/// <summary>
/// 更新
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="dataTable">数据表</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状态信息</param>
/// <returns>影响行数</returns>
[OperationContract]
int Update(BaseUserInfo userInfo, DataTable dataTable, out string statusCode, out string
statusMessage);
```

14.6 批量删除

```
/// <summary>
/// 批量删除
/// </summary>
```



```
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchDelete(BaseUserInfo userInfo, string[] ids);
```

14.6 批量保存

```
/// <summary>
/// 批量保存
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="dataTable">数据表</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchSave(BaseUserInfo userInfo, DataTable dataTable);
```



十五、序列(流水号)接口

15.1 添加序列流水号

```
/// <summary>
/// 添加序列流水号
/// </summary>
/// <param name="userInfo"></param>
/// <param name="sequenceEntity"></param>
/// <param name="statusCode"></param>
/// <param name="statusMessage"></param>
/// <returns></returns>
[OperationContract]
string Add(BaseUserInfo userInfo, BaseSequenceEntity sequenceEntity, out string
statusCode, out string statusMessage);
```

15.2 获取全部序列列表

```
/// <summary>
/// 窗体加载
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDT(BaseUserInfo userInfo);
```

15.3 获取序列

```
/// <summary>
/// 获取序列号
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="fullName">序列名称</param>
/// <returns>序列号</returns>
[OperationContract]
string GetSequence(BaseUserInfo userInfo, string fullName);
```



15.3 获取原序列号

```
/// <summary>
/// 获取原序列号
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="fullName">序列名称</param>
/// <param name="defaultSequence">默认序列</param>
/// <param name="sequenceLength">序列长度</param>
/// <param name="fillZeroPrefix">是否填充补零</param>
/// <returns>序列号</returns>
[OperationContract]
string GetOldSequence(BaseUserInfo userInfo, string fullName, int defaultSequence, int
sequenceLength, bool fillZeroPrefix);
```

15.4 获取新序列号

```
/// <summary>
/// 获取新序列号
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="fullName">序列名称</param>
/// <param name="defaultSequence">默认序列</param>
/// <param name="sequenceLength">序列长度</param>
/// <param name="fillZeroPrefix">是否填充补零</param>
/// <returns>序列号</returns>
[OperationContract]
string GetNewSequence(BaseUserInfo userInfo, string fullName, int defaultSequence, int
sequenceLength, bool fillZeroPrefix);
```

15.5 获取序列号

```
/// <summary>
/// 获取序列号
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="fullName">序列名称</param>
/// <param name="count">个数</param>
/// <returns>序列号</returns>
```



[OperationContract]

string[] GetBatchSequence(BaseUserInfo userInfo, string fullName, int count);

15.6 获取倒序序列号

```
/// <summary>
/// 获取倒序序列号
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="fullName">序列名称</param>
/// <returns>序列号</returns>
[OperationContract]
string GetReduction(BaseUserInfo userInfo, string fullName);
```

15.7 批量重置

```
/// <summary>
/// 批量重置
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>数据表</returns>
[OperationContract]
int Reset(BaseUserInfo userInfo, string[] ids);
```

15.8 批量删除日志

```
/// <summary>
/// 批量删除日志
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>数据表</returns>
[OperationContract]
int Delete(BaseUserInfo userInfo, string id);
```



15.9 批量删除权限

```
/// <summary>
/// 批量删除权限
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>数据表</returns>
[OperationContract]

int BatchDelete(BaseUserInfo userInfo, string[] ids);
```



十六、选项（字典）管理接口

16.1 获取列表

```
/// <summary>
/// 获取列表
/// </summary>
/// <param name="userInfo">用户</param>
[OperationContract]
DataTable GetDT(BaseUserInfo userInfo);
```

16.2 获取列表

```
/// <summary>
/// 获取列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
[OperationContract]
BaseItemsEntity GetEntity(BaseUserInfo userInfo, string id);
```

16.3 获取子列表

```
/// <summary>
/// 获取子列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="parentId">父节点主键</param>
[OperationContract]
DataTable GetDTByParent(BaseUserInfo userInfo, string parentId);
```

16.4 添加编码

```
/// <summary>
/// 添加编码
/// </summary>
/// <param name="userInfo">用户</param>
```




```
/// <param name="itemsEntity">实体</param>
/// <param name="statusCode">状态返回码</param>
/// <param name="statusMessage">状态返回信息</param>
[OperationContract]
string Add(BaseUserInfo userInfo, BaseItemsEntity itemsEntity, out string statusCode,
out string statusMessage);
```

16.5 创建表结构

```
/// <summary>
/// 创建表结构
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="tableName">表名</param>
/// <param name="statusCode">状态返回码</param>
/// <param name="statusMessage">状态返回信息</param>
[OperationContract]
void CreateTable(BaseUserInfo userInfo, string tableName, out string statusCode, out
string statusMessage);
```

16.6 更新编码

```
/// <summary>
/// 更新编码
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="itemsEntity">实体</param>
/// <param name="statusCode">状态返回码</param>
/// <param name="statusMessage">状态返回信息</param>
[OperationContract]
int Update(BaseUserInfo userInfo, BaseItemsEntity itemsEntity, out string statusCode,
out string statusMessage);
```

16.7 删除

```
/// <summary>
/// 删除
/// </summary>
/// <param name="userInfo">用户</param>
```



```
/// <param name="tableName">目标表</param>
/// <param name="id">主键</param>
[OperationContract]
int Delete(BaseUserInfo userInfo, string tableName, string id);
```

16.8 批量删除编码

```
/// <summary>
/// 批量删除编码
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="tableName">目标表</param>
/// <param name="ids">主键数组</param>
[OperationContract]
int BatchDelete(BaseUserInfo userInfo, string tableName, string[] ids);
```

16.9 批量移动编码

```
/// <summary>
/// 批量移动编码
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="tableName">目标表</param>
/// <param name="ids">编码主键数组</param>
/// <param name="targetId">父级主键</param>
[OperationContract]
int BatchMoveTo(BaseUserInfo userInfo, string tableName, string[] ids, string targetId);
```

16.10 批量保存编码

```
/// <summary>
/// 批量保存编码
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="dataTable">数据表</param>
/// <param name="dataTable">影响行数</param>
[OperationContract]
int BatchSave(BaseUserInfo userInfo, DataTable dataTable);
```



16.11 获取列表（详细）

```
/// <summary>
/// 获取列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="tableName">目标表</param>
[OperationContract]
DataTable GetDT(BaseUserInfo userInfo, string tableName);
```

16.12 获取子列表（详细）

```
/// <summary>
/// 获取子列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="tableName">目标表</param>
/// <param name="parentId">父节点主键</param>
[OperationContract]
DataTable GetDTByParent(BaseUserInfo userInfo, string tableName, string parentId);
```

16.13 获取下拉框数据（详细）

```
/// <summary>
/// 获取下拉框数据
/// </summary>
/// <param name="userInfo"></param>
/// <param name="code"></param>
/// <returns></returns>
[OperationContract]
DataTable GetDTByCode(BaseUserInfo userInfo, string code);
```

16.14 获取获取下拉框数据（详细）

```
/// <summary>
/// 获取批量下拉框数据
/// </summary>
/// <param name="userInfo"></param>
```



```
/// <param name="codes"></param>
/// <returns></returns>
[OperationContract]
DataSet GetDSByCodes(BaseUserInfo userInfo, string[] codes);
```

16.15 获取列表（详细）

```
/// <summary>
/// 获取列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="tableName">目标表</param>
/// <param name="id">主键</param>
[OperationContract]
BaseItemDetailsEntity GetEntity(BaseUserInfo userInfo, string tableName, string id);
```

16.16 添加编码（详细）

```
/// <summary>
/// 添加编码
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="dataTable">数据表</param>
/// <param name="statusCode">状态返回码</param>
/// <param name="statusMessage">状态返回信息</param>
[OperationContract]
string Add(BaseUserInfo userInfo, string tableName, BaseItemDetailsEntity
itemDetailsEntity, out string statusCode, out string statusMessage);
```

16.17 更新编码（详细）

```
/// <summary>
/// 更新编码
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="itemDetailsEntity">实体</param>
/// <param name="statusCode">状态返回码</param>
/// <param name="statusMessage">状态返回信息</param>
[OperationContract]
```



```
int Update(BaseUserInfo userInfo, string tableName, BaseItemDetailsEntity
itemDetailsEntity, out string statusCode, out string statusMessage);
```

16.18 删除（详细）

```
/// <summary>
/// 删除
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="tableName">目标表</param>
/// <param name="id">主键</param>
[OperationContract]
int Delete(BaseUserInfo userInfo, string tableName, string id);
```

16.19 批量删除编码（详细）

```
/// <summary>
/// 批量删除编码
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="tableName">目标表</param>
/// <param name="ids">主键数组</param>
[OperationContract]
int BatchDelete(BaseUserInfo userInfo, string tableName, string[] ids);
```

16.20 批量移动编码（详细）

```
/// <summary>
/// 批量移动编码
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="tableName">目标表</param>
/// <param name="ids">编码主键数组</param>
/// <param name="targetId">父级主键</param>
[OperationContract]
int BatchMoveTo(BaseUserInfo userInfo, string tableName, string[] ids, string targetId);
```



16.21 批量保存编码（详细）

```
/// <summary>
/// 批量保存编码
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="dataTable">数据表</param>
/// <param name="dataTable">影响行数</param>
[OperationContract]
int BatchSave(BaseUserInfo userInfo, DataTable dataTable);
```



十七、文件夹及文件接口

17.1 获取文件夹列表

```
/// <summary>
/// 获取列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDT(BaseUserInfo userInfo);
```

17.2 获取一条

```
/// <summary>
/// 获取一条
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>实体</returns>
[OperationContract]
BaseFolderEntity GetEntity(BaseUserInfo userInfo, string id);
```

17.3 按目录获取列表

```
/// <summary>
/// 按目录获取列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByParent(BaseUserInfo userInfo, string id);
```

17.4 添加文件夹

```
/// <summary>
```



```
/// 添加
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="folderEntity">实体</param>
/// <param name="statusCode">状态码</param>
/// <param name="statusMessage">状态信息</param>
/// <returns>主键</returns>
[OperationContract]
string Add(BaseUserInfo userInfo, BaseFolderEntity folderEntity, out string statusCode,
out string statusMessage);
```

17.5 添加文件夹

```
/// <summary>
/// 添加
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="parentId">父主键</param>
/// <param name="folderName">文件夹名称</param>
/// <param name="enabled">有效</param>
/// <param name="statusCode">状态码</param>
/// <param name="statusMessage">状态信息</param>
/// <returns></returns>
[OperationContract]
string AddByFolderName(BaseUserInfo userInfo, string parentId, string folderName, bool
enabled, out string statusCode, out string statusMessage);
```

17.6 更新文件夹

```
/// <summary>
/// 更新文件夹
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="folderEntity">文件夹</param>
/// <param name="statusCode">状态码</param>
/// <param name="statusMessage">状态信息</param>
/// <returns>影响行数</returns>
[OperationContract]
int Update(BaseUserInfo userInfo, BaseFolderEntity folderEntity, out string statusCode,
out string statusMessage);
```




17.7 重命名文件夹

```
/// <summary>
/// 重命名
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <param name="newName">新名称</param>
/// <param name="enabled">有效</param>
/// <param name="statusCode">状态码</param>
/// <param name="statusMessage">状态信息</param>
/// <returns>影响行数</returns>
[OperationContract]
int Rename(BaseUserInfo userInfo, string id, string newName, bool enabled, out string
statusCode, out string statusMessage);
```

17.8 查询文件夹

```
/// <summary>
/// 查询
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="search">查询</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable Search(BaseUserInfo userInfo, string searchValue);
```

17.9 删除文件夹

```
/// <summary>
/// 删除
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int Delete(BaseUserInfo userInfo, string id);
```



17.10 批量删除文件夹

```
/// <summary>
/// 批量删除
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchDelete(BaseUserInfo userInfo, string[] ids);
```

17.11 移动文件夹

```
/// <summary>
/// 移动
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="folderId">文件夹主键</param>
/// <param name="parentId">目标主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int MoveTo(BaseUserInfo userInfo, string folderId, string parentId);
```

17.12 批量移动文件夹

```
/// <summary>
/// 批量移动
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="folderIds">文件夹主键数组</param>
/// <param name="parentId">目标主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchMoveTo(BaseUserInfo userInfo, string[] folderIds, string parentId);
```

17.13 批量保存文件夹



```
/// <summary>
/// 批量保存
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="dataTable">数据表</param>
/// <returns>影响行数</returns>
[OperationContract]

int BatchSave(BaseUserInfo userInfo, DataTable dataTable);
```

17.14 文件夹是否已存在

```
/// <summary>
/// 文件是否已存在
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="folderId">文件夹主键</param>
/// <param name="fileName">文件名</param>
/// <returns>是否已存在</returns>
[OperationContract]

bool Exists(BaseUserInfo userInfo, string folderId, string fileName);
```

17.15 下载文件

```
/// <summary>
/// 下载文件
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>文件</returns>
[OperationContract]

byte[] Download(BaseUserInfo userInfo, string id);
```

17.16 上传文件

```
/// <summary>
/// 上传文件
/// </summary>
/// <param name="userInfo">用户</param>
```



```
/// <param name="folderId">文件夹主键</param>
/// <param name="fileName">文件名</param>
/// <param name="file">文件</param>
/// <param name="description">描述</param>
/// <returns>主键</returns>
[OperationContract]
string Upload(BaseUserInfo userInfo, string folderId, string fileName, byte[] file, bool
enabled);
```

17.17 获取实体

```
/// <summary>
/// 获取实体
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="id">主键</param>
/// <returns>实体</returns>
[OperationContract]
BaseFileEntity GetEntity(BaseUserInfo userInfo, string id);
```

17.18 按文件夹获取列表

```
/// <summary>
/// 按文件夹获取列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="folderId">文件夹主键</param>
/// <returns>数据集</returns>
[OperationContract]
DataTable GetDTByFolder(BaseUserInfo userInfo, string folderId);
```

17.19 按文件夹删除文件

```
/// <summary>
/// 按文件夹删除文件
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="folderId">文件夹主键</param>
/// <returns>影响的行数</returns>
```



[OperationContract]

int DeleteByFolder(BaseUserInfo userInfo, string folderId);

17.19 添加文件

```
/// <summary>
/// 添加
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="folderId">文件夹主键</param>
/// <param name="fileName">文件名</param>
/// <param name="file">文件</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状消息</param>
/// <returns>主键</returns>
[OperationContract]
string Add(BaseUserInfo userInfo, string folderId, string fileName, byte[] file, string
description, string category, bool enabled, out string statusCode, out string
statusMessage);
```

17.20 更新文件

```
/// <summary>
/// 更新
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <param name="folderId">文件夹</param>
/// <param name="fileName">文件名</param>
/// <param name="description">描述</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状消息</param>
/// <returns>影响行数</returns>
[OperationContract]
int Update(BaseUserInfo userInfo, string id, string folderId, string fileName, string
description, bool enabled, out string statusCode, out string statusMessage);
```



17.21 更新文件

```
/// <summary>
/// 更新
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <param name="fileName">文件名</param>
/// <param name="file">文件</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状态消息</param>
/// <returns>影响行数</returns>
[OperationContract]
int UpdateFile(BaseUserInfo userInfo, string id, string fileName, byte[] file, out string
statusCode, out string statusMessage);
```

17.22 重命名文件

```
/// <summary>
/// 重命名
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <param name="newName">新名称</param>
/// <param name="enabled">有效</param>
/// <param name="statusCode">状态码</param>
/// <param name="statusMessage">状态信息</param>
/// <returns>影响行数</returns>
[OperationContract]
int Rename(BaseUserInfo userInfo, string id, string newName, bool enabled, out string
statusCode, out string statusMessage);
```

17.23 查询文件

```
/// <summary>
/// 查询
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="search">查询</param>
```



```
/// <returns>数据表</returns>
[OperationContract]
DataTable Search(BaseUserInfo userInfo, string searchValue);
```

17.24 移动数据

```
/// <summary>
/// 移动数据
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <param name="folderId">文件夹主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int MoveTo(BaseUserInfo userInfo, string id, string folderId);
```

17.25 批量移动数据

```
/// <summary>
/// 批量移动数据
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <param name="folderId">文件夹主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchMoveTo(BaseUserInfo userInfo, string[] ids, string folderId);
```

17.26 删除

```
/// <summary>
/// 删除
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int Delete(BaseUserInfo userInfo, string id);
```



17.27 批量删除

```
/// <summary>
/// 批量删除
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchDelete(BaseUserInfo userInfo, string[] ids);
```

17.28 批量保存

```
/// <summary>
/// 批量保存
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="dataTable">数据表</param>
/// <returns>影响的行数</returns>
[OperationContract]
int BatchSave(BaseUserInfo userInfo, DataTable dataTable);
```




十八、即时通讯组件接口

18.1 获得内部部门（公司的组织机构）

```
/// <summary>
/// 获得内部部门（公司的组织机构）
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetInnerOrganize(BaseUserInfo userInfo);
```

18.2 发送即时消息

```
/// <summary>
/// 发送即时消息
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="receiverID">接收者主键</param>
/// <param name="content">内容</param>
/// <returns>主键</returns>
[OperationContract]
string Send(BaseUserInfo userInfo, string receiverId, string content);
```

18.3 批量发送即时消息

```
/// <summary>
/// 批量发送即时消息
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="receiverIds">接收者主键组</param>
/// <param name="organizeIds">组织机构主键组</param>
/// <param name="roleIds">角色主键组</param>
/// <param name="messageEntity">消息实体</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchSend(BaseUserInfo userInfo, string[] receiverIds, string[] organizeIds,
string[] roleIds, BaseMessageEntity messageEntity);
```



```
/// <summary>
/// 获取消息状态
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="lastChekDate">最后检查时间</param>
/// <returns>消息状态数组</returns>
[OperationContract]
string[] MessageChek(BaseUserInfo userInfo, string lastChekTime);
```

18.4 获取用户的新信息

```
/// <summary>
/// 获取用户的新信息
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTNew(BaseUserInfo userInfo);
```

18.5 获取特定用户的新信息

```
/// <summary>
/// 获取特定用户的新信息
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="receiverID">当前交互的用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable ReadFromReceiver(BaseUserInfo userInfo, string receiverId);
```

18.6 阅读短信

```
/// <summary>
/// 阅读短信
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
[OperationContract]
```



```
void Read(BaseUserInfo userInfo, string id);
```

18.7 检查在线状态

```
/// <summary>
/// 检查在线状态
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>离线人数</returns>
[OperationContract]
int CheckOnLine(BaseUserInfo userInfo);
```

18.8 获取在线用户列表

```
/// <summary>
/// 获取在线用户列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetOnLineState(BaseUserInfo userInfo);
```



十九、工作日志服务接口

19.1 获取一个工作日志列表

```
/// <summary>
/// 获取一个工作日志列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDT(BaseUserInfo userInfo, string id);
```

19.2 获取职员工作日志

```
/// <summary>
/// 获取职员工作日志
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="staffId">职员主键</param>
/// <param name="reportDate">日期</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDTByUser(BaseUserInfo userInfo, string staffId, string reportDate);
```

19.3 更新工作日志

```
/// <summary>
/// 更新工作日志
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="dataTable">数据表</param>
/// <param name="statusCode">返回的状态码</param>
/// <param name="statusMessage">返回的状态信息</param>
/// <returns>影响的行数</returns>
[OperationContract]
int Update(BaseUserInfo userInfo, DataTable dataTable, out string statusCode, out string statusMessage);
```



19.4 添加工作日志

```
/// <summary>
/// 添加
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="dataTable">数据表</param>
/// <param name="statusCode">返回状态码</param>
/// <param name="statusMessage">返回状态消息</param>
/// <returns>主键</returns>
[OperationContract]
string Add(BaseUserInfo userInfo, DataTable dataTable, out string statusCode, out string
statusMessage);
```

19.5 批量删除

```
/// <summary>
/// 批量删除
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchDelete(BaseUserInfo userInfo, string[] ids);
```

19.6 查询审核列表

```
/// <summary>
/// 查询审核列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="startDate">开始日期</param>
/// <param name="endDate">结束日期</param>
/// <param name="enabled">有效</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable SearchAuditing(BaseUserInfo userInfo, DateTime startDate, DateTime endDate,
int enabled);
```



19.7 批量设置审核状态

```
/// <summary>
/// 批量设置审核状态
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <param name="enabled">有效</param>
/// <returns>影响行数</returns>
[OperationContract]
bool BatchSetEnabled(BaseUserInfo userInfo, string[] ids, int enabled);
```

19.8 批量保存

```
/// <summary>
/// 批量保存
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="dataTable">数据表</param>
/// <param name="enabled">有效性</param>
/// <param name="startDate">开始日期</param>
/// <param name="endDate">结束日期</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable BatchSave(BaseUserInfo userInfo, DataTable dataTable, int enabled, DateTime
startDate, DateTime endDate);
```

19.9 获取项目列表

```
/// <summary>
/// 获取项目列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetProjectDT(BaseUserInfo userInfo);
```



19.10 获取项目全称

```
/// <summary>
/// 获取项目全称
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>项目全称</returns>
[OperationContract]
string GetProjectFullName(BaseUserInfo userInfo, string id);
```

19.11 检查工作日志

```
/// <summary>
/// 检查工作日志
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="paramStaffId">当前查询职员主键</param>
/// <returns>日期数组</returns>
[OperationContract]
DateTime[] CheckWorkDate(BaseUserInfo userInfo, string staffId);
```

19.12 求某天工时之和

```
/// <summary>
/// 求某天工时之和
/// </summary>
/// <param name="userInfo">当前操作员信息</param>
/// <param name="paramStaffId">职员主键</param>
/// <param name="paramDate">日期</param>
/// <returns>工时之和</returns>
[OperationContract]
Double SumManHour(BaseUserInfo userInfo, string staffId, DateTime paramDate);
```



二十、 workflow 服务接口

20.1 获取 workflow 步骤定义列表

```
/// <summary>
/// 获取 workflow 步骤定义列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="workFlowId">workflow 主键</param>
/// <returns>数据集</returns>
[OperationContract]
DataTable GetDT(BaseUserInfo userInfo, string workFlowId);
```

20.2 添加 workflow

```
/// <summary>
/// 添加 workflow
/// </summary>
/// <param name="userInfo">当前用户</param>
/// <param name="workFlowActivityEntity">workflow 定义实体</param>
/// <param name="statusCode">状态码</param>
/// <param name="statusMessage">状态信息</param>
/// <returns>主键</returns>
[OperationContract]
string Add(BaseUserInfo userInfo, BaseWorkFlowActivityEntity workFlowActivityEntity);
```

20.3 批量删除

```
/// <summary>
/// 批量删除
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">选中的数组</param>
/// <param name="flowID">流主键</param>
/// <returns>数据集</returns>
[OperationContract]
int BatchDelete(BaseUserInfo userInfo, string[] ids);
```




20.3 批量保存

```
/// <summary>
/// 批量保存
/// </summary>
/// <param name="userInfo">操作员</param>
/// <param name="dataTable">数据表</param>
/// <returns>影响行数</returns>
[OperationContract]
int BatchSave(BaseUserInfo userInfo, DataTable dataTable);
```

20.4 获取待审核主键

```
/// <summary>
/// 获取待审核主键
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="categoryId">分类主键</param>
/// <param name="objectId">主键</param>
/// <returns>主键</returns>
[OperationContract]
string GetCurrentId(BaseUserInfo userInfo, string categoryId, string objectId);
```

20.5 获取列表

```
/// <summary>
/// 获取列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetDT(BaseUserInfo userInfo);
```

20.6 获取监控列表

```
/// <summary>
/// 获取监控列表
```



```
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetMonitorDT(BaseUserInfo userInfo);
```

20.7 获取待审核列表

```
/// <summary>
/// 获取待审核列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <returns>数据表</returns>
[OperationContract]
DataTable GetWaitForAudit(BaseUserInfo userInfo);
```

20.8 获取审核历史明细

```
/// <summary>
/// 获取审核历史明细
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="categoryId">单据分类主键</param>
/// <param name="objectId">单据主键</param>
/// <returns>数据集</returns>
[OperationContract]
DataTable GetAuditDetailDT(BaseUserInfo userInfo, string categoryId, string objectId);
```

20.9 最终审核通过

```
/// <summary>
/// 最终审核通过
/// </summary>
/// <param name="userInfo">当前用户</param>
/// <param name="id">主键</param>
/// <param name="auditIdea">审核意见</param>
/// <returns>影响行数</returns>
[OperationContract]
int AuditComplete(BaseUserInfo userInfo, string id, string auditIdea);
```



20.10 撤消审批流程中的单据

```
/// <summary>
/// 撤消审批流程中的单据
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="currentWorkFlowId">当前工作流主键</param>
/// <param name="auditIdea">撤销意见</param>
/// <returns>影响行数</returns>
[OperationContract]
int AuditQuash(BaseUserInfo userInfo, string currentWorkFlowId, string auditIdea);
```

20.11 审核驳回

```
/// <summary>
/// 审核驳回
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">当前主键</param>
/// <param name="auditIdea">审核建议</param>
/// <returns>数据集</returns>
[OperationContract]
int AuditReject(BaseUserInfo userInfo, string id, string auditIdea);
```

20.12 自动工作流审核通过

```
/// <summary>
/// 自动工作流审核通过
/// </summary>
/// <param name="flowId">当前流程主键</param>
/// <param name="auditIdea">递交意见</param>
/// <returns>影响行数</returns>
[OperationContract]
int AutoAuditPass(BaseUserInfo userInfo, string flowId, string auditIdea);
```

20.13 开始审核

```
/// <summary>
```



```
/// 开始审核
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="categoryId">分类主键</param>
/// <param name="categoryFullName">分类名称</param>
/// <param name="objectIds">实体主键数组</param>
/// <param name="objectFullName">实体名称</param>
/// <param name="workFlowCode">工作流编号</param>
/// <param name="auditIdea">审核意见</param>
/// <param name="returnStatusCode">审核状态</param>
/// <returns>主键</returns>
[OperationContract]
string AutoStatr(BaseUserInfo userInfo, string categoryId, string categoryFullName,
string[] objectIds, string objectFullName, string workFlowCode, string auditIdea, out
string returnStatusCode);
```

20.14 审核驳回

```
/// <summary>
/// 审核驳回
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">当前主键</param>
/// <param name="auditIdea">审核建议</param>
/// <returns>数据集</returns>
[OperationContract]
int RoleAuditPass(BaseUserInfo userInfo, string id, string sendToRoleId, string
auditIdea);
```

20.15 按角色审核开始

```
/// <summary>
/// 按角色审核开始
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="categoryId">分类主键</param>
/// <param name="categoryFullName">分类名称</param>
/// <param name="objectIds">实体主键</param>
/// <param name="objectFullName">实体名称</param>
/// <param name="sendToRoleId">发送给角色主键</param>
```



```
/// <param name="auditIdea">审核意见</param>
/// <param name="returnStatusCode">返回状态</param>
/// <returns>主键</returns>
[OperationContract]
string RoleStatr(BaseUserInfo userInfo, string categoryId, string categoryFullName,
string[] objectIds, string objectFullName, string sendToRoleId, string auditIdea, out
string returnStatusCode);
```

20.16 审核开始

```
/// <summary>
/// 审核开始
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="flowId">流程主键</param>
/// <param name="categoryId">分类主键</param>
/// <param name="categoryFullName">分类名称</param>
/// <param name="objectId">实体主键</param>
/// <param name="objectFullName">实体名称</param>
/// <returns>主键</returns>
[OperationContract]
string StartAudit(BaseUserInfo userInfo, int flowId, string categoryId, string
categoryFullName, string objectId, string objectFullName);
```

20.17 下个流程发送给谁

```
/// <summary>
/// 下个流程发送给谁
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">当前主键</param>
/// <param name="sendToRoleId">角色主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int TransmitRole(BaseUserInfo userInfo, string id, string sendToRoleId, string
auditIdea);
```

20.18 下个流程发送给谁

```
/// <summary>
```



```
/// 下个流程发送给谁
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">当前主键</param>
/// <param name="sendToStaffID">员工主键</param>
/// <returns>数据集</returns>
[OperationContract]
int TransmitUser(BaseUserInfo userInfo, string id, string sendToUserId, string
auditIdea);
```

20.19 审核驳回

```
/// <summary>
/// 审核驳回
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">当前主键</param>
/// <param name="auditIdea">审核建议</param>
/// <returns>数据集</returns>
[OperationContract]
int UserAuditPass(BaseUserInfo userInfo, string id, string sendToUserId, string
auditIdea);
```

20.20 按用户审核启动

```
/// <summary>
/// 按用户审核启动
/// </summary>
/// <param name="userInfo">操作员信息</param>
/// <param name="categoryId">分类主键</param>
/// <param name="categoryFullName">分类名称</param>
/// <param name="objectIds">被审核的对象主键数组</param>
/// <param name="objectFullName">被审核的对象名称</param>
/// <param name="sendToRoleId">递交给谁审核</param>
/// <param name="auditIdea">递交意见</param>
/// <param name="returnStatusCode">返回状态码</param>
/// <returns>工作流主键</returns>
[OperationContract]
string UserStatr(BaseUserInfo userInfo, string categoryId, string categoryFullName,
```



```
string[] objectIds, string objectFullName, string sendToUserId, string auditIdea, out  
string returnStatusCode);
```

20.21 添加 workflow

```
/// <summary>  
/// 添加 workflow  
/// </summary>  
/// <param name="userInfo">当前用户</param>  
/// <param name="workFlowProcessEntity">workflow 定义实体</param>  
/// <param name="statusCode">状态码</param>  
/// <param name="statusMessage">状态信息</param>  
/// <returns>主键</returns>  
[OperationContract]  
string Add(BaseUserInfo userInfo, BaseWorkFlowProcessEntity workFlowProcessEntity, out  
string statusCode, out string statusMessage);
```

20.22 批量删除组织机构

```
/// <summary>  
/// 批量删除组织机构  
/// </summary>  
/// <param name="userInfo">用户</param>  
/// <param name="ids">主键数组</param>  
/// <returns>影响行数</returns>  
[OperationContract]  
int BatchDelete(BaseUserInfo userInfo, string[] ids);
```

20.23 批量保存

```
/// <summary>  
/// 批量保存  
/// </summary>  
/// <param name="userInfo">用户</param>  
/// <param name="dataTable">数据表</param>  
/// <returns>数据表</returns>  
[OperationContract]
```



```
int BatchSave(BaseUserInfo userInfo, DataTable dataTable);
```

20.24 单个删除 workflow

```
/// <summary>
/// 单个删除 workflow
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <param name="organizeId">组织主键</param>
/// <returns>影响行数</returns>
[OperationContract]
int Delete(BaseUserInfo userInfo, string id);
```

20.25 获取 workflow 列表

```
/// <summary>
/// 获取 workflow 列表
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="organizeId">组织主键</param>
/// <returns>数据集</returns>
[OperationContract]
DataTable GetDT(BaseUserInfo userInfo, string organizeId);
```

20.26 获取 workflow

```
/// <summary>
/// 获取 workflow
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="id">主键</param>
/// <returns>实体</returns>
[OperationContract]
BaseWorkflowProcessEntity GetEntity(BaseUserInfo userInfo, string id);
```




20.27 批量做删除标志

```
/// <summary>
/// 批量做删除标志
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="ids">主键数组</param>
/// <returns>影响行数</returns>
[OperationContract]
int SetDeleted(BaseUserInfo userInfo, string[] ids);
```

20.28 更新 workflow

```
/// <summary>
/// 更新 workflow
/// </summary>
/// <param name="userInfo">用户</param>
/// <param name="workFlowProcessEntity">实体</param>
/// <param name="statusCode">状态码</param>
/// <param name="statusMessage">状态信息</param>
/// <returns>影响行数</returns>
[OperationContract]
int Update(BaseUserInfo userInfo, BaseWorkFlowProcessEntity workFlowProcessEntity, out
string statusCode, out string statusMessage);
```