

# 山海易绘 EzManager 运维系统 API

## 二次开发手册



方正国际软件（北京）有限公司

2015 年 6 月

## 文档修订说明表

序号	手册版本	产品版本	修订人	审核人	修订时间	修订内容	备注
1	V1.0	EzManagerAPI V2.1.3_20090625	唐方、岳佳	刘晓平	2009-10-30	文档创建，版面格式修订	
2	V1.0	EzManagerAPI V2.2.2_20091126	唐方、刘晓平	薛红波	2009-11-26	1、增加了接口说明第 3.27 和 3.28 章节； 2、部分章节的“注意事项”删除； 3、章节编号错误修订； 4、去掉❶版本变更❷等附录内容。	
3	V1.1	EzManagerAPI V2.2.3.100112	唐方、刘晓平	薛红波	2010-1-12	1、增加了第 3.26 章节； 2、增加了"硬件环境"说明； 3 、 安 装 包 名 称 “EzManager.jar”。	
4	V1.2	EzManagerAPIV3.0.0.201007211600	王胜超	薛红波	2010-7-23	1、增加第 3.16-3.21 章节； 2、修改其他章节的实例程序。	
5	V1.3	EzManagerAPI_V3.1.1.20110328	唐方、刘晓平	薛红波	2011-3-31	1、第 1 章“概述”中增加注释说明； 2、增加第 3 章“组织机构开发接口说明”【原“用户开发接口”改为第 4 章】。 3、增加第 4.37-4.46 章节	
6	V1.4	EzManagerAPI3.1.4	蒋小明		2011-6-30	1、增加获取子结构接口说明。2、增加说明 API 下的 UserManager 类包含 user 包下 UserManager 类的所有方法。	
7	V1.5	EzManagerAPI3.1.7	赵向明	孙璐	2011-11-25		
8	V1.6	EzManagerAPIV3.1.11	别媛媛		2012-8-22	根据方正模板格式修订	
9	V1.7	API_V3.1.13.201209071603	杨金锋、别媛媛		2012-9-7	1，满足宁波对运维界面的结构增加新的 API 方法；2 根据方正模板格式修订	
10	V6.0	V6.0	赵向明		2015-6-8		

本手册内容改动及版本更新将不再另行通知。本手册的范例中使用的人名、公司名和数据如果没有特别指明，均属虚构。对于本手册、及本手册涉及的技术和产品，方正国际软件（北京）有限公司拥有其专利、商标、著作权或其它知识产权，除非得到方正国际软件（北京）有限公司的书面许可，本手册不授予这些专利、商标、著作权或其它知识产权的许可。

版权所有 C（2008-2013）

方正国际软件（北京）有限公司保留所有权利。

- 山海易绘是方正国际软件（北京）有限公司的注册商标。
- 其它标牌和产品名称是其各自公司的商标或注册商标。
- 山海易绘产品的一切有关权利属于方正国际软件（北京）有限公司所有。
- 本手册中所涉及的软件产品及其后续升级产品均由方正国际软件（北京）有限公司制作并负责全权销售。

如果您对本产品有任何建议和疑问，请与以下地址联系：

方正国际软件（北京）有限公司公共安全与地理信息事业部

地 址：北京市海淀区北四环西路 52 号方正国际大厦 18 层

邮 编：100080

传 真：（010）82179019

电 话：（010）82179081/82179484/82343654

E-mail: [gis-support@founder.com](mailto:gis-support@founder.com)

网 址: <http://www.founderinternational.com>

# 目 录

1	概述 .....	1
2	安装指南 .....	2
2.1	运行环境 .....	2
2.1.1	硬件环境 .....	2
2.1.2	软件环境 .....	2
2.2	安装步骤 .....	2
2.2.1	步骤 1 .....	2
2.2.2	步骤 2 .....	2
3	注意事项 .....	3
4	NEWAPI 包的 FUNCTIONMANAGER 接口说明 .....	4
4.1	根据系统或功能编码（CODE）查询系统功能信息 .....	4
4.1.1	功能说明 .....	4
4.1.2	语法定义 .....	4
4.1.3	示例程序 .....	4
4.2	根据角色编码（ROLECODE）查询角色拥有的系统功能信息 .....	4
4.2.1	功能说明 .....	4
4.2.2	语法定义 .....	4
4.2.3	示例程序 .....	5
4.3	根据机构编码查询机构拥有的系统功能信息 .....	5
4.3.1	功能说明 .....	5
4.3.2	语法定义 .....	5
4.3.1	示例程序 .....	5
4.4	根据用户 ID 查询用户拥有的应用系统 .....	6
4.4.1	功能说明 .....	6
4.4.2	语法定义 .....	6
4.4.3	示例程序 .....	6
4.5	根据用户 ID 查询应用系统所有的根节点 .....	6
4.5.1	功能说明 .....	6
4.5.2	语法定义 .....	7
4.5.3	示例程序 .....	7
4.6	根据系统功能编号查询该编号下的直属子节点 .....	7
4.6.1	功能说明 .....	7
4.6.2	语法定义 .....	7
4.6.3	示例程序 .....	7
4.7	查询某个角色下的某个应用系统拥有的直属功能权限 .....	8
4.7.1	功能说明 .....	8
4.7.2	语法定义 .....	8
4.7.3	示例程序 .....	8
4.8	保存应用信息 .....	9
4.8.1	功能说明 .....	9

4.8.2	语法定义 .....	9
4.8.3	示例程序 .....	9
4.9	删除应用信息 .....	9
4.9.1	功能说明 .....	9
4.9.2	语法定义 .....	9
4.9.3	功能示例 .....	10
4.10	更新应用信息 .....	10
4.10.1	功能说明 .....	10
4.10.2	语法定义 .....	10
4.10.3	功能示例 .....	10
<b>5</b>	<b>NEWAPI 包的 LAYERMANAGER 接口说明 .....</b>	<b>12</b>
5.1	根据用户 ID 和应用功能 CODE 查询该用户在某个功能上所拥有的图层权限 .....	12
5.1.1	功能说明 .....	12
5.1.2	语法定义 .....	12
5.1.3	示例程序 .....	12
5.2	根据用户 ID 和应用功能 CODE 查询该用户在某个功能上所拥有的专题权限 .....	13
5.2.1	功能说明 .....	13
5.2.2	语法定义 .....	13
5.2.3	示例程序 .....	13
<b>6</b>	<b>NEWAPI 包下的 ORGPOSMANAGER 接口明 .....</b>	<b>14</b>
6.1	依据岗位 ID 查询岗位实体对象 .....	14
6.1.1	功能说明 .....	14
6.1.2	语法定义 .....	14
6.1.3	示例程序 .....	14
6.2	根据组织机构代码查询该组织机构下的岗位列表 .....	14
6.2.1	功能说明 .....	14
6.2.2	语法定义 .....	14
6.2.3	示例程序 .....	15
6.3	根据用户 ID 查询用户所拥有的岗位列表 .....	15
6.3.1	功能说明 .....	15
6.3.2	语法定义 .....	15
6.3.3	示例程序 .....	15
6.4	查询指定所属组织机构的用户所拥有的岗位列表 .....	16
6.4.1	功能说明 .....	16
6.4.2	语法定义 .....	16
6.4.3	示例程序 .....	16
<b>7</b>	<b>NEWAPI 包下的 POSMANAGER 接口 .....</b>	<b>17</b>
7.1	根据岗位体系编号查询岗位体系对象 .....	17
7.1.1	功能说明 .....	17
7.1.2	语法定义 .....	17
7.1.3	示例程序 .....	17
7.2	查询所有的岗位体系 .....	17

7.2.1	功能说明 .....	17
7.2.2	语法定义 .....	17
7.2.3	示例程序 .....	18
8	NEWAPI 包下的 ROLEMANAGER 接口 .....	19
8.1	根据角色编码查询角色信息 .....	19
8.1.1	功能说明 .....	19
8.1.2	语法定义 .....	19
8.1.3	示例程序 .....	19
8.2	根据用户 ID 查询用户所拥有的所有角色权限 .....	19
8.2.1	功能说明 .....	19
8.2.2	语法定义 .....	19
8.2.3	示例程序 .....	20
9	组织机构开发接口说明 .....	21
9.1	查找所有组织机构接口说明 .....	21
9.1.1	功能说明 .....	21
9.1.2	语法定义 .....	21
9.1.3	示例程序 .....	21
9.2	根据 ID 查找组织机构接口说明 .....	22
9.2.1	功能说明 .....	22
9.2.2	语法定义 .....	22
9.2.3	示例程序 .....	22
9.3	根据组织机构树级别查询组织机构接口说明 .....	23
9.3.1	功能说明 .....	23
9.3.2	语法定义 .....	23
9.3.3	示例程序 .....	23
9.4	根据名称查询组织机构接口说明 .....	24
9.4.1	功能说明 .....	24
9.4.2	语法定义 .....	24
9.4.3	示例程序 .....	24
9.5	根据组织结构代码获取子结构列表说明 .....	25
9.5.1	功能说明 .....	25
9.5.2	语法定义 .....	25
9.5.3	示例程序 .....	25
9.6	查询所有的机构类型和组织机构类型 .....	26
9.6.1	功能说明 .....	26
9.6.2	语法定义 .....	26
9.6.3	示例程序 .....	27
9.7	根据类型名称查询机构类型或组织机构类型的对象 .....	28
9.7.1	功能说明 .....	28
9.7.2	语法定义 .....	28
9.7.3	示例程序 .....	28
9.8	根据用户 ID 查询出同级和下级的所有组织机构集合 .....	29
9.8.1	功能说明 .....	29

9.8.2	语法定义 .....	29
9.8.3	示例程序 .....	29
10	用户权限开发接口说明 .....	31
10.1	获取版本号接口说明 .....	错误!未定义书签。
10.1.1	功能说明 .....	31
10.1.2	语法定义 .....	31
10.2	用户登录验证接口说明 .....	31
10.2.1	功能说明 .....	31
10.2.2	语法定义 .....	31
10.2.3	示例程序 .....	32
10.3	获取用户信息接口说明 .....	33
10.3.1	功能说明 .....	33
10.3.2	语法定义 .....	33
10.3.3	示例程序 .....	33
10.3.4	注意事项 .....	34
10.4	级联查询用户信息接口说明 .....	34
10.4.1	功能说明 .....	34
10.4.2	语法定义 .....	34
10.4.3	示例程序 .....	35
10.5	获取组织机构下的用户接口 1 说明 .....	36
10.5.1	功能说明 .....	36
10.5.2	语法定义 .....	36
10.5.3	示例程序 .....	36
10.6	获取组织机构下的用户接口 2 说明 .....	37
10.6.1	功能说明 .....	37
10.6.2	语法定义 .....	37
10.6.3	示例程序 .....	38
10.7	根据类型获取组织机构接口说明 .....	39
10.7.1	功能说明 .....	39
10.7.2	语法定义 .....	39
10.7.3	示例程序 .....	39
10.8	获取用户所属组织机构接口说明 .....	41
10.8.1	功能说明 .....	41
10.8.2	语法定义 .....	41
10.8.3	示例程序 .....	41
10.9	查询所有用户接口说明 .....	42
10.9.1	功能说明 .....	42
10.9.2	语法定义 .....	43
10.9.3	示例程序 .....	43
10.10	根据用户 ID 和密码查询用户接口说明 .....	44
10.10.1	功能说明 .....	44
10.10.2	语法定义 .....	44
10.10.3	示例程序 .....	44
10.11	根据所属组织机构编号查找用户接口说明 .....	45

10.11.1	功能说明.....	45
10.11.2	语法定义.....	45
10.11.3	示例程序.....	45
10.12	根据名称查找用户接口说明 .....	46
10.12.1	功能说明.....	46
10.12.2	语法定义.....	46
10.12.3	示例程序.....	46
<b>11</b>	<b>图层信息接口说明 .....</b>	<b>48</b>
11.1	获取图层字段信息接口说明 .....	48
11.1.1	功能说明 .....	48
11.1.2	语法定义 .....	48
11.1.3	示例程序 .....	48
11.2	获取图层表字段名称接口说明 .....	49
11.2.1	功能说明 .....	49
11.2.2	语法定义 .....	49
11.2.3	示例程序 .....	50



# 1 概述

山海易绘 EzManager 运维管理系统（以下简称“EzManager”）的权限管理 API jar 包（EzManagerAPI.jar），主要用于对各个系统（如“综合查询应用”）的权限验证，其功能包括用户登录名和密码的验证、用户的功能模块权限验证、用户的数据权限验证、用户的 IP 权限验证、系统日志写入等。

注：EzManager API 包从 V3.1 版本开始修改了类库结构，新增了 api 包。新增 api 包中的类不影响之前所有的类使用，但所有的新增业务接口都将在 api 包中进行维护。

## 2 安装指南

使用运维权限管理 jar 包,首先要有一个 java 的 Web 工程,把“EzManagerAPI.jar”和“gson-1.6.jar”包拷贝到工程的“WEB-INF”目录的“lib”文件夹下,并在需要使用的 java 文件或 jsp 文件中导入引用。

### 2.1 运行环境

#### 2.1.1 硬件环境

- 1) 硬盘空间: 20GB 以上;
- 2) 内存空间: 512MB 以上;
- 3) CPU: P4 1GHz 以上。

#### 2.1.2 软件环境

- 1) 支持的软件平台: JAVA;
- 2) Jdk 版本: 1.6 以上;
- 3) 数据库: Oracle 9i 以上;
- 4) 对应的运维管理系统 V6.0.0;

### 2.2 安装步骤

#### 2.2.1 步骤 1

将“EzManagerAPI.jar”和“gson-1.6.jar”包拷贝到“java web”程的“WEB-INF”目录下的 lib 文件夹下。

#### 2.2.2 步骤 2

在 java 文件中导入“UserManager”,代码如下:

```
“Import com.easymap.management.*;”
```

在 jsp 文件中导入“UserManager”,代码如下:

```
“<%@page import="com.easymap.management.*"%>”
```

### 3 注意事项

为了实现对之前老接口的兼容，api 尽量保留了原先的 API。因为对象之间的关系有大的改变，建议除了用户和组织机构的查询，其它接口尽量用用 `com.easymap.management.newapi` 中的业务类接口去实现。在使用过程中，如果显示已过时的接口，尽量不要去使用，但这些老的接口仍然可以在 api html doc 中查看，即 api 的 html 文档，一些老的接口比如获取专题树权限的接口仍然可以使用。

另外，除了 `com.easymap.management.newapi` 中的业务类 `FunctionManager` 接口实现了增删改查接口以外，其它业务类接口均只实现了查询操作，后续会逐渐完善其它增删改的实现。

在包引入中，引入 jar 包需要引入“EzManagerAPI.jar”和“gson-1.6.jar”两个 jar 包，或者 gson 其它版本的 jar 包也可以。

## 4 newapi 包的 FunctionManager 接口说明

### 4.1 根据系统或功能编码（code）查询系统功能信息

#### 4.1.1 功能说明

根据系统或功能编码（code）查询系统功能信息

#### 4.1.2 语法定义

语法定义：“public Function getFunctionByCode(String code);”

请求参数说明：无

参数名称	描述	参数约束
code	系统功能	

返回值说明：Function 对象。

异常抛出：当有异常抛出时，查询失败。

#### 4.1.3 示例程序

```
try {  
    FunctionManager fm = new FunctionManager("http://localhost:8080/EzManager");  
    List<Function> list=fm. getFunctionByCode ("5");  
    System.out.println(list);  
    System.out.println(list.size());  
} catch (Exception e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

### 4.2 根据角色编码（rolecode）查询角色拥有的系统功能信息

#### 4.2.1 功能说明

根据角色编码（rolecode）查询角色拥有的系统功能信息

#### 4.2.2 语法定义

语法定义：“public List<Function> getFunctionsByRoleId(String roleid);”

请求参数说明：无

参数名称	描述	参数约束
roleid	角色 id	

返回值说明：Function 集合。

异常抛出：当有异常抛出时，查询失败。

### 4.2.3 示例程序

```
try {
    FunctionManager fm = new FunctionManager("http://localhost:8080/EzManager");
    List<Function> list=fm. getFunctionsByRoleId ("5");
    System.out.println(list);
    System.out.println(list.size());
} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

## 4.3 根据机构编码查询机构拥有的系统功能信息

### 4.3.1 功能说明

根据机构编码（code）查询机构拥有的系统功能信息

### 4.3.2 语法定义

语法定义：“public List<Function> getFunctionsByOrgId(String orgid);”

请求参数说明：无

参数名称	描述	参数约束
orgid	组织机构编码(id)	

返回值说明：Function 集合。

异常抛出：当有异常抛出时，查询失败。

### 4.3.1 示例程序

```
try {
    FunctionManager fm = new FunctionManager("http://localhost:8080/EzManager");
    List<Function> list=fm. getFunctionsByOrgId ("100010101");
    System.out.println(list);
    System.out.println(list.size());
}
```

```
} catch (Exception e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

## 4.4 根据用户 ID 查询用户拥有的应用系统

### 4.4.1 功能说明

根据用户 ID 查询用户拥有的应用系统

### 4.4.2 语法定义

语法定义：“public List<Function> getFunctionByUserId(String userid);”

请求参数说明：无

参数名称	描述	参数约束
userid	用户 id	

返回值说明：Function 集合。

异常抛出：当有异常抛出时，查询失败。

### 4.4.3 示例程序

```
try {  
    FunctionManager fm = new FunctionManager("http://localhost:8080/EzManager");  
    List<Function> list=fm. getFunctionByUserId ("admin");  
    System.out.println(list);  
    System.out.println(list.size());  
} catch (Exception e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

## 4.5 根据用户 ID 查询应用系统所有的根节点

### 4.5.1 功能说明

根据用户 ID 查询应用系统所有的根节点

### 4.5.2 语法定义

语法定义：“public List<Function> getRootFunction(String userid);”

请求参数说明：无

参数名称	描述	参数约束
userid	用户 id	

返回值说明：Function 集合。

异常抛出：当有异常抛出时，查询失败。

### 4.5.3 示例程序

```
try {
    FunctionManager fm = new FunctionManager("http://localhost:8080/EzManager");
    List<Function> list=fm. getRootFunction ("admin");
    System.out.println(list);
    System.out.println(list.size());
} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

## 4.6 根据系统功能编号查询该编号下的直属子节点

### 4.6.1 功能说明

根据系统功能编号查询该编号下的直属子节点（不包含自身）

### 4.6.2 语法定义

语法定义：“public List<Function> getFunctionByParentCode(String code);”

请求参数说明：无

参数名称	描述	参数约束
code	Function 的 code	

返回值说明：Function 集合。

异常抛出：当有异常抛出时，查询失败。

### 4.6.3 示例程序

```
try {
    FunctionManager fm = new FunctionManager("http://localhost:8080/EzManager");
    List<Function> list=fm. getFunctionByParentCode ("aaa");
}
```

```
        System.out.println(list);
        System.out.println(list.size());
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
```

## 4.7 查询某个角色下的某个应用系统拥有的直属功能权限

### 4.7.1 功能说明

funcode 为 null，查询某个角色拥有的应用系统权限；funcode 不为 null,查询某个角色下的某个应用系统拥有的直属功能权限

### 4.7.2 语法定义

语法定义：“public List<Function> getFunctionByRoleCodeAndFunCode(String roleCode,String funCode);”

请求参数说明：无

参数名称	描述	参数约束
rolecode	角色的 code	
funcode	Function 的 code	

返回值说明：Function 集合。

异常抛出：当有异常抛出时，查询失败。

### 4.7.3 示例程序

```
try {
    FunctionManager fm = new FunctionManager("http://localhost:8080/EzManager");
    List<Function> list=fm. getFunctionByRoleCodeAndFunCode ("rolecode","aaa");
    System.out.println(list);
    System.out.println(list.size());
} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```



## 4.8 保存应用信息

### 4.8.1 功能说明

保存应用信息

### 4.8.2 语法定义

语法定义：“public void saveFunction(Function func);”

请求参数说明：无

参数名称	描述	参数约束
Function	Function 对象	

返回值说明：无。

异常抛出：当有异常抛出时，保存失败。

### 4.8.3 示例程序

```
try {  
    FunctionManager fm = new FunctionManager("http://localhost:8080/EzManager");  
    Function func=new Function();  
    func.setCode("pp");  
    func.setName("裴");  
    func.setOwner("pxm");  
    func.setType("APP");  
    func.setAppType("NORMAL");  
    fm.saveFunction(func);  
} catch (Exception e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

## 4.9 删除应用信息

### 4.9.1 功能说明

删除应用信息

### 4.9.2 语法定义

语法定义：“public void deletFunction(String funcode);”

请求参数说明：无

参数名称	描述	参数约束
funcode	Function 的 code	

返回值说明：无。

异常抛出：当有异常抛出时，删除失败。

### 4.9.3 功能示例

```
try {
    FunctionManager fm = new FunctionManager("http://localhost:8080/EzManager");
    fm.deleteFunction("pp");
} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

## 4.10 更新应用信息

### 4.10.1 功能说明

更新应用信息

### 4.10.2 语法定义

语法定义：“public void updateFunction(Function func);”

请求参数说明：无

参数名称	描述	参数约束
func	Function 对象	

返回值说明：无。

异常抛出：当有异常抛出时，更新失败。

### 4.10.3 功能示例

```
try {
    FunctionManager fm = new FunctionManager("http://localhost:8080/EzManager");
    func.setCode("pp");
    func.setName("裴");
    func.setOwner("pxm");
    func.setType("APP");
    func.setAppType("EXTERNAL");
    fm.updateFunction(func);
}
```

```
} catch (Exception e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

## 5 newapi 包的 LayerManager 接口说明

### 5.1 根据用户 ID 和应用功能 code 查询该用户在某个功能上所拥有的图层权限

#### 5.1.1 功能说明

根据用户 ID 和应用功能 code 查询该用户在某个功能上所拥有的图层权限

#### 5.1.2 语法定义

语法定义：“public List<Layer> getOwnLayersByUseridAndFuncode(String userid,String funcode);”

请求参数说明：无

参数名称	描述	参数约束
userid	userid	
funcode	应用功能 code	

返回值说明：专题图层列表。

异常抛出：当有异常抛出时，查询失败。

#### 5.1.3 示例程序

```
try {  
    List<Layer> layers = um.getOwnLayersByUseridAndFuncode("admin","gps 功能");  
    System.out.println(layers.size());  
    for (Layer layer : layers) {  
        System.out.println(layer.getChname());  
    }  
} catch (Exception e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

## 5.2 根据用户 ID 和应用功能 code 查询该用户在某个功能上所拥有的专题权限

### 5.2.1 功能说明

根据用户 ID 和应用功能 code 查询该用户在某个功能上所拥有的专题权限

### 5.2.2 语法定义

语法定义：“`public List<Theme> getOwnThemesByUserid(String userid,String funcode);`”

请求参数说明：无

参数名称	描述	参数约束
userid	userid	
funcode	系统或功能 code	

返回值说明：专题列表。

异常抛出：当有异常抛出时，查询失败。

### 5.2.3 示例程序

```
try {  
    List<Theme> themes = um.getOwnThemesByUserid("admin","gps");  
    System.out.println(themes.size()+" "+gson.toJson(themes));  
} catch (Exception e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

## 6 newapi 包下的 OrgPosManager 接口明

### 6.1 依据岗位 ID 查询岗位实体对象

#### 6.1.1 功能说明

依据岗位 ID 查询岗位实体对象

#### 6.1.2 语法定义

语法定义：“public OrgPos getOrgPosById(int id);”

请求参数说明：无

参数名称	描述	参数约束
id	岗位 id	

返回值说明：岗位列表。

异常抛出：当有异常抛出时，查询失败。

#### 6.1.3 示例程序

```
try {  
    OrgPosManager opm=new OrgPosManager("http://172.18.70.192:8080/EzManager6");  
    OrgPos op=opm.getOrgPosById(1566);  
    System.out.println(op);  
} catch (Exception e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

### 6.2 根据组织机构代码查询该组织机构下的岗位列表

#### 6.2.1 功能说明

根据组织机构代码查询该组织机构下的岗位列表

#### 6.2.2 语法定义

语法定义：“public List<OrgPos> getOrgPosByOrgCode(String orgCode);”

请求参数说明：无

参数名称	描述	参数约束
orgCode	组织机构代码	

返回值说明：岗位列表。

异常抛出：当有异常抛出时，查询失败。

### 6.2.3 示例程序

```
try {
    OrgPosManager opm=new OrgPosManager("http://172.18.70.192:8080/EzManager6");
    List<OrgPos> opList=opm.getOrgPosByOrgCode("aaac");
    for(OrgPos op:opList){
        System.out.println(op);
    }
} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

## 6.3 根据用户 ID 查询用户所拥有的岗位列表

### 6.3.1 功能说明

根据用户 ID 查询用户所拥有的岗位列表

### 6.3.2 语法定义

语法定义：“public List<OrgPos> getOrgPosById(String userid);”

请求参数说明：无

参数名称	描述	参数约束
userid	登录名	

返回值说明：岗位列表。

异常抛出：当有异常抛出时，查询失败。

### 6.3.3 示例程序

```
try {
    OrgPosManager opm=new OrgPosManager("http://172.18.70.192:8080/EzManager6");
    List<OrgPos> opList= opm.getOrgPosById("user3");
    for(OrgPos op:opList){
        System.out.println(op);
    }
}
```

```
} catch (Exception e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

## 6.4 查询指定所属组织机构的用户所拥有的岗位列表

### 6.4.1 功能说明

查询指定所属组织机构的用户所拥有的岗位列表

### 6.4.2 语法定义

语法定义：“public List<OrgPos> getOrgPosByUserIdAndOrgCode(String userid,String orgCode);”

请求参数说明：无

参数名称	描述	参数约束
userid	登录名	
orgCode	组织机构代码	

返回值说明：岗位列表。

异常抛出：当有异常抛出时，查询失败。

### 6.4.3 示例程序

```
try {  
    OrgPosManager opm=new OrgPosManager("http://172.18.70.192:8080/EzManager6");  
    List<OrgPos> opList=opm.getOrgPosByUserIdAndOrgCode("admin", "002");  
    for(OrgPos op:opList){  
        System.out.println(op);  
    }  
} catch (Exception e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```



## 7 newapi 包下的 PosManager 接口

### 7.1 根据岗位体系编号查询岗位体系对象

#### 7.1.1 功能说明

根据岗位体系编号查询岗位体系对象

#### 7.1.2 语法定义

语法定义：“public Position getPositionByCode(String code);”

请求参数说明：无

参数名称	描述	参数约束
code	岗位体系编码	

返回值说明：岗位体系对象。

异常抛出：当有异常抛出时，查询失败。

#### 7.1.3 示例程序

```
try {  
    PosManager pm=new PosManager("http://172.18.70.192:8080/EzManager6");  
    Position pos=pm.getPositionByCode("121");  
    System.out.println(pos);  
} catch (Exception e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

### 7.2 查询所有的岗位体系

#### 7.2.1 功能说明

查询所有的岗位体系

#### 7.2.2 语法定义

语法定义：“public List<Position> getAllPosition();”

请求参数说明：无

参数名称	描述	参数约束
------	----	------

返回值说明：岗位体系对象集合。

异常抛出：当有异常抛出时，查询失败。

### 7.2.3 示例程序

```
try {  
    PosManager pm=new PosManager("http://172.18.70.192:8080/EzManager6");  
    List<Position> pos=pm. getAllPosition();  
    System.out.println(pos);  
} catch (Exception e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

## 8 newapi 包下的 RoleManager 接口

### 8.1 根据角色编码查询角色信息

#### 8.1.1 功能说明

根据角色编码查询角色信息

#### 8.1.2 语法定义

语法定义：“public Role getRoleByCode(String code);”

请求参数说明：无

参数名称	描述	参数约束
code	角色编码	

返回值说明：角色对象。

异常抛出：当有异常抛出时，查询失败。

#### 8.1.3 示例程序

```
try {  
    RoleManager rm = new RoleManager("http://localhost:8080/EzManager6");  
    Role role = um.getRoleByCode("role");  
    System.out.println(role);  
} catch (Exception e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

### 8.2 根据用户 id 查询用户所拥有的所有角色权限

#### 8.2.1 功能说明

根据用户 id 查询用户所拥有的所有角色权限

#### 8.2.2 语法定义

语法定义：“public List<Role> getOwnRolesByUserId(String userid);”

请求参数说明：无

参数名称	描述	参数约束
userid	登录名	

返回值说明：角色对象集合。

异常抛出：当有异常抛出时，查询失败。

### 8.2.3 示例程序

```
try {  
    RoleManager rm = new RoleManager("http://localhost:8080/EzManager6");  
    List<Role> roles = um.getOwnRolesByUserId("admin");  
    System.out.println(roles);  
} catch (Exception e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

## 9 组织机构开发接口说明

组织机构接口在“com.easymap.management.api”包的“OrganizationManager”类中,“Organization”实体类在 unit 包中,原 user 包中的“Organization”类已过时,被“unit”包中的类取代。

### 9.1 查找所有组织机构接口说明

#### 9.1.1 功能说明

查找到所有组织机构。

#### 9.1.2 语法定义

语法定义:“public List<Organization> queryOrganizations();”

请求参数说明:无

参数名称	描述	参数约束
------	----	------

返回值说明:组织机构对象列表。

异常抛出:当有异常抛出时,查询组织机构失败。

#### 9.1.3 示例程序

```
/**
 * OrganizationManagerTest.java
 * 作者 Ton 2011-03-11 上午 09:30:37
 */

package api;

import java.util.ArrayList;
import java.util.List;
import com.easymap.management.api.OrganizationManager;
import com.easymap.management.unit.Organization;

public class OrganizationManagerTest {

    /**
     * @param args
     */
    public static void main(String[] args) {
        OrganizationManager um = new OrganizationManager("http://192.168.10.182:8008/EzManager");
        try {
```

```
        List<Organization> orgList = om.queryOrganizations();
        System.out.println("组织机构个数: " + orgList.size());
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

## 9.2 根据 ID 查找组织机构接口说明

### 9.2.1 功能说明

根据组织机构的编号，查找组织机构。

### 9.2.2 语法定义

语法定义：“public Organization findOrganizationById(String id);”

请求参数说明：

参数名称	描述	参数约束
id	组织机构编号	

返回值说明：组织机构对象，包含组织机构的所有属性信息。

异常抛出：当有异常抛出时，查找组织机构失败。

### 9.2.3 示例程序

```
/**
 * OrganizationManagerTest.java
 * 作者 Ton 2011-03-11 上午 09:30:37
 */

package api;

import com.easymap.management.api.OrganizationManager;
import com.easymap.management.unit.Organization;

public class OrganizationManagerTest {

    /**
     * @param args
     */
    public static void main(String[] args) {
        OrganizationManager um = new OrganizationManager("http://192.168.10.182:8008/EzManager");
```

```
        try {
            Organization org = om.findOrganizationById("100000000202");
            System.out.println("组织机构名称: " + org.getName());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

## 9.3 根据组织机构树级别查询组织机构接口说明

### 9.3.1 功能说明

组织机构是一个树形结构，1 级节点就是组织机构树的根节点，此方法是根据指定的级别查找组织机构列表。

### 9.3.2 语法定义

语法定义：“public List<Organization> findOrganizationsByLevel(int level);”

请求参数说明：

参数名称	描述	参数约束
levle	组织机构树级别	

返回值说明：组织机构对象列表。

异常抛出：当有异常抛出时，组织机构查询失败。

### 9.3.3 示例程序

```
/**
 * OrganizationManagerTest.java
 * 作者 Ton 2011-03-11 上午 09:30:37
 */

package api;

import java.util.ArrayList;
import java.util.List;
import com.easymap.management.api.OrganizationManager;
import com.easymap.management.unit.Organization;
public class OrganizationManagerTest {

    /**
```

```
* @param args
*/
public static void main(String[] args) {
    OrganizationManager um = new OrganizationManager("http://192.168.10.182:8008/EzManager");
    try {
        //获取 2 级组织机构列表
        List<Organization> orgList = om.findOrganizationsByLevel(2);
        System.out.println("2 级组织机构个数: " + orgList.size());
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

## 9.4 根据名称查询组织机构接口说明

### 9.4.1 功能说明

根据名称查询组织机构，由于名称有可能出现重复，所以查询返回的是一个列表。

### 9.4.2 语法定义

语法定义：“public List<Organization> findOrganizationsByName(String name);”

请求参数说明：

参数名称	描述	参数约束
name	组织机构树级别	

返回值说明：组织机构对象列表。

异常抛出：当有异常抛出时，组织机构查询失败。

### 9.4.3 示例程序

```
/**
 * OrganizationManagerTest.java
 * 作者 Ton 2011-03-11 上午 09:30:37
 */

package api;

import java.util.ArrayList;
import java.util.List;
import com.easymap.management.api.OrganizationManager;
```



```
import com.easymap.management.unit.Organization;
public class OrganizationManagerTest {

    /**
     * @param args
     */
    public static void main(String[] args) {
        OrganizationManager um = new OrganizationManager("http://192.168.10.182:8008/EzManager");
        try {
            List<Organization> orgList = om.findOrganizationsByName("修改后的测试组织机构 1");
            System.out.println("个数: " + orgList.size());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

## 9.5 根据组织结构代码获取子结构列表说明

### 9.5.1 功能说明

根据组织结构代码获取子组织机构，如果没有子机构，返回的 list 的 size 为 0

### 9.5.2 语法定义

语法定义：“public List<Organization> findChildrenOrganizations(String zzjgdm)”

请求参数说明：

参数名称	描述	参数约束
zzjgdm	组织结构代码	

返回值说明：组织机构对象列表。

异常抛出：当有异常抛出时，组织机构查询失败。

### 9.5.3 示例程序

```
/**
 * OrganizationManagerTest.java
 * 作者 Ton 2011-03-11 上午 09:30:37
 */

package api;
```

```
import java.util.ArrayList;
import java.util.List;
import com.easymap.management.api.OrganizationManager;
import com.easymap.management.unit.Organization;
public class OrganizationManagerTest {

    /**
     * @param args
     */
    public static void main(String[] args) {
        OrganizationManager om = new OrganizationManager("http://192.168.10.182:8008/EzManager");
        try
        {
            String zzjgdm="320402590300";
            List<Organization> list = om.findChildrenOrganizations(zzjgdm);
            for (Organization organization : list)
            {
                String ssjgdm=organization.getSsjgdm();
            }
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
}
```

## 9.6 查询所有的机构类型和组织机构类型

### 9.6.1 功能说明

查询出所有的机构类型 jglx。

简单介绍一下，机构类型对象(OrgType)。

在组织机构对象(Organization)里面，属性机构类型(jglx)，代码即为 OrgType 中的 typeid,名称为 OrgType 中的 typename。其中 1 为组织机构，2 为工作组。

### 9.6.2 语法定义

语法定义：“public List<OrgType> queryOrgtypes();”

请求参数说明：

参数名称	描述	参数约束
------	----	------

无	无	无
---	---	---

返回值说明：无

异常抛出：当有异常抛出时，查询组织机构类型失败。

### 9.6.3 示例程序

```
/**
 * OrganizationManagerTest.java
 * 作者 Ton 2011-03-11 上午 09:30:37
 */

package api;

import java.util.ArrayList;
import java.util.List;
import com.easymap.management.api.OrganizationManager;
import com.easymap.management.unit.Organization;

public class OrganizationManagerTest {

    /**
     * @param args
     */
    public static void main(String[] args) {
        try {
            OrganizationManager om = new
OrganizationManager("http://192.168.10.182:8008/EzManager");
            List<OrgType> list = om.queryOrgtypes();
            for (OrgType orgType : list) {
                System.out.println(orgType.getTypeid()+" "+orgType.getTypename());
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

## 9.7 根据类型名称查询机构类型或组织机构类型的对象

### 9.7.1 功能说明

根据机构类型或组织机构类型的名称，查询出对应的机构类型或组织机构类型的对象。类型名称为“组织机构”和“工作组”。

### 9.7.2 语法定义

语法定义：“public OrgType queryOrgtypeByName(String name);”

请求参数说明

参数名称	描述	参数约束
Name	类型名称	

返回值说明：OrgType 对象

异常抛出：当有异常抛出时，查询组织机构类型失败。

### 9.7.3 示例程序

```
/**
 * OrganizationManagerTest.java
 * 作者 Ton 2011-03-11 上午 09:30:37
 */

package api;

import java.util.ArrayList;
import java.util.List;
import com.easymap.management.api.OrganizationManager;
import com.easymap.management.unit.Organization;

public class OrganizationManagerTest {

    /**
     * @param args
     */
    public static void main(String[] args) {
        try {
            OrganizationManager om = new
            OrganizationManager("http://192.168.10.182:8008/EzManager");
            OrgType ot = om.queryOrgtypeByName("消防");
            System.out.println(ot.getTypeid()+" "+ot.getTypename()+" "+ot.getParent());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
    }  
}  
  
}
```

## 9.8 根据用户 id 查询出所属组织机构和下级的所有组织机构集合

### 9.8.1 功能说明

根据用户 id 查出所属组织机构和下级所有的组织机构对象的集合

### 9.8.2 语法定义

语法定义：

“public List<Organization> queryOrganizationsCascadeByuserid(String userid);”

请求参数说明

参数名称	描述	参数约束
Userid	用户登录 id	

返回值说明：List<Organization>集合

异常抛出：当有异常抛出时，查询组织机构失败。

### 9.8.3 示例程序

```
/**  
 * OrganizationManagerTest.java  
 * 作者 Ton 2011-03-11 上午 09:30:37  
 */  
  
package api;  
import java.util.ArrayList;  
import java.util.List;  
import com.easymap.management.api.OrganizationManager;  
import com.easymap.management.unit.Organization;  
public class OrganizationManagerTest {  
  
    /**  
     * @param args  
     */  
    public static void main(String[] args) {  
        try {  
            OrganizationManager om = new  
OrganizationManager("http://192.168.10.182:8008/EzManager");
```

```
List<Organization> list = om.queryOrganizationsCascadeByuserid("wind");
for (Organization organization : list) {
    System.out.println(organization.getName());
}
System.out.println(list.size());
} catch (Exception e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}
```

## 10 用户权限开发接口说明

在以下介绍的功能接口中，请注意实例代码中导入的不同包中的实体类，这些实体类在“javadoc”中都可以查询到，并且这些实体类也是使用过程中得到需要信息的重要部分。

其中 api 包下面的接口是较新的，相对于 user 包下面的接口，所以请尽量使用 api 包下面的接口。

### 10.1 获取版本号接口说明

#### 10.1.1 功能说明

获取当前“EzManagerAPI.jar”包的版本。此方法在“com.easymap.management.api”包中的“UserManager”类中。

#### 10.1.2 语法定义

语法定义：“public String getVersion();”

返回值说明：

返回版本号。

### 10.2 用户登录验证接口说明

#### 10.2.1 功能说明

验证用户名和密码是否正确。此方法在“com.easymap.management.api”包中的“UserManager”类中。

#### 10.2.2 语法定义

语法定义：“public boolean validUser(String userId, String password);”

请求参数说明：

参数名称	描述	参数约束
userId	用户登录名	
password	用户登录密码	

返回值说明：

返回“true”或“false”，true--是验证通过，false--是验证失败。

### 10.2.3 示例程序

```
/**
 * TestManagement.java
 * 作者 Ton 2008-12-3 下午 04:30:37
 */
package com.tf;

import java.net.MalformedURLException;

import com.easymap.management.api.UserManager;

/**
 * @author Ton
 */
public class TestManagement {

    /**
     * @param args
     */
    public static void main(String[] args) {

        try {
            UserManager userManager = new UserManager("http://192.168.10.204:8008/EzManager");
            boolean flag = userManager.validUser("123", "321");
            if(flag){
                System.out.println("验证通过！");
            }else{
                System.out.println("用户名密码错误！");
            }
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }

    }

}
```



## 10.3 获取用户信息接口说明

### 10.3.1 功能说明

根据用户登录名得到用户信息。此方法在“com.easymap.management.api”包中的“UserManager”类中。

### 10.3.2 语法定义

语法定义：“public User getUser(String userId);”

请求参数说明：

参数名称	描述	参数约束
userId	用户登录名	不能为 null

返回值说明：

返回一个用户实体，包括用户的登录名、用户名、所属组织机构、Email 等信息。详细描述请看“javadoc”中“User”类的描述。

### 10.3.3 示例程序

```
/**
 * TestManagement.java
 * 作者 Ton 2008-12-3 下午 04:30:37
 */
package com.tf;
import java.net.MalformedURLException;

import com.easymap.management.user.User;
import com.easymap.management.api.UserManager;

/**
 * @author Ton
 */
public class TestManagement {

    /**
     * @param args
     */
    public static void main(String[] args) {

        try {
            UserManager userManager = new UserManager("http://192.168.10.204:8008/EzManager");
```

```
User user = userManager.getUser("123");
System.out.println("登录名: " + user.getId());
System.out.println("用户名: " + user.getName());
System.out.println("密码: " + user.getPassword());
System.out.println("备注: " + user.getDesc());
System.out.println("Email : " + user.getEmail());
System.out.println("传真: " + user.getFax());
System.out.println("所属组织机构代码: " + user.getOrgId());
System.out.println("所属组织机构名称: " + user.getOrgName());
System.out.println("电话: " + user.getPhone());

} catch (MalformedURLException e) {
    e.printStackTrace();
}

}

}
```

#### 10.3.4 注意事项

获取用户信息应该在登录验证成功以后把验证成功的登录名当做参数传递，否则会抛出“NullPointerException”。

### 10.4 级联查询用户信息接口说明

#### 10.4.1 功能说明

根据组织机构 id 查询此组织机构下所有用户信息以及其下属组织机构所有信息

根据用户登录名得到用户信息。此方法在“com.easymap.management.api”包中的“UserManager”类中。

#### 10.4.2 语法定义

语法定义：“public List<User> queryUserByCascadeOrgid(String orgid);”

请求参数说明：

参数名称	描述	参数约束
orgid	组织机构代码	

返回值说明：

返回一组用户实体，包括用户的登录名、用户名、所属组织机构、Email 等信息。详细描述请

看“javadoc”中“User”类的描述。

### 10.4.3 示例程序

```
/**
 * TestManagement.java
 * 作者 Ton 2008-12-3 下午 04:30:37
 */
package com.tf;
import java.net.MalformedURLException;

import com.easymap.management.user.User;
import com.easymap.management.api.UserManager;

/**
 * @author Ton
 */
public class TestManagement {

    /**
     * @param args
     */
    public static void main(String[] args) {

        try {
            UserManager userManager = new UserManager("http://192.168.10.204:8008/EzManager");
            List<User> lists=new ArrayList<User>();
            lists =userManager. queryUserByCascadeOrgid(orgid)
            for(int i=0;i<lists.size;i++){
                User user=lists.get(i);
                System.out.println("登录名: " + user.getId());
                System.out.println("用户名: " + user.getName());
                System.out.println("密码: " + user.getPassword());
                System.out.println("备注: " + user.getDesc());
                System.out.println("Email : " + user.getEmail());
                System.out.println("传真: " + user.getFax());
                System.out.println("所属组织机构代码: " + user.getOrgId());
                System.out.println("所属组织机构名称: " + user.getOrgName());
                System.out.println("电话: " + user.getPhone());
            }

        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
    }
}
```

```
    }  
  
    }  
  
}
```

## 10.5 获取组织机构下的用户接口 1 说明

### 10.5.1 功能说明

根据组织机构编号，获取该组织机构下的所有用户。此方法在 “com.easymap.management.api” 包中的 “UserManager” 类中。

### 10.5.2 语法定义

语法定义: “public List<User> getUserByOrgId(String orgid);”

请求参数说明:

参数名称	描述	参数约束
orgid	组织机构编号	

返回值说明:

返回指定组织机构下的用户列表，得到用户实体类。

### 10.5.3 示例程序

```
/**  
 * TestManagement.java  
 * 作者 Ton 2008-12-3 下午 04:30:37  
 */  
package com.tf;  
import java.net.MalformedURLException;  
import java.util.List;  
  
import com.easymap.management.user.User;  
import com.easymap.management.api.UserManager;  
  
/**  
 * @author Ton  
 *  
 */  
public class TestManagement {
```

```
/**
 * @param args
 */
public static void main(String[] args) {

    try {
        UserManager userManager = new UserManager("http://192.168.10.204:8008/EzManager");
        List<User> list = userManager.getUserByOrgId("321100000000");
        for(int i = 0; i < list.size(); i++){
            User user = list.get(i);
            System.out.println("*****用户" + (i+1) + "*****");
            System.out.println("登录名: " + user.getId());
            System.out.println("用户名: " + user.getName());
            System.out.println("密 码: " + user.getPassword());
            System.out.println("备 注: " + user.getDesc());
            System.out.println("Email : " + user.getEmail());
            System.out.println("传 真: " + user.getFax());
            System.out.println("所属组织机构代码: " + user.getOrgId());
            System.out.println("所属组织机构名称: " + user.getOrgName());
            System.out.println("电 话: " + user.getPhone());

        }

    } catch (MalformedURLException e) {
        e.printStackTrace();
    }

}

}
```

## 10.6 获取组织机构下的用户接口 2 说明

### 10.6.1 功能说明

根据组织机构实体类（organization）获取该组织机构下的所有用户。此方法在“com.easymap.management.api”包中的“UserManager”类中。

### 10.6.2 语法定义

语法定义：“public List<User> getUserByUnit(Organization organization);”

请求参数说明：

参数名称	描述	参数约束
organization	组织机构实体	

返回值说明:

返回指定组织机构下的用户列表，得到用户实体类。

### 10.6.3 示例程序

```
/**
 * TestManagement.java
 * 作者 Ton 2008-12-3 下午 04:30:37
 */
package com.tf;
import java.net.MalformedURLException;
import java.util.List;

import com.easymap.management.user.Organization;
import com.easymap.management.user.User;
import com.easymap.management.api.UserManager;

/**
 * @author Ton
 */
public class TestManagement {

    /**
     * @param args
     */
    public static void main(String[] args) {

        try {
            UserManager userManager = new UserManager("http://192.168.10.204:8008/EzManager");
            //首先获取一个用户所属的组织机构
            Organization org = userManager.getOrganization("123");

            List<User> list = userManager.getUserByUnit(org);
            for(int i = 0; i < list.size(); i++){
                User user = list.get(i);
                System.out.println("*****用户" + (i+1) + "*****");
                System.out.println("登录名: " + user.getId());
                System.out.println("用户名: " + user.getName());
                System.out.println("密码: " + user.getPassword());
                System.out.println("备注: " + user.getDesc());
                System.out.println("Email : " + user.getEmail());
            }
        } catch (MalformedURLException e) {
            e.printStackTrace();
        }
    }
}
```

```
        System.out.println("传 真： " + user.getFax());
        System.out.println("所属组织机构代码： " + user.getOrgId());
        System.out.println("所属组织机构名称： " + user.getOrgName());
        System.out.println("电 话： " + user.getPhone());

    }

    } catch (MalformedURLException e) {
        e.printStackTrace();
    }

}

}
```

## 10.7 根据类型获取组织机构接口说明

### 10.7.1 功能说明

根据组织机构类型编号获取该类型的组织机构。此方法在“com.easymap.management.api”包中的“UserManager”类中。

### 10.7.2 语法定义

语法定义：“public List<Organization> getOrganizationsByType(String type);”

请求参数说明：

参数名称	描述	参数约束
type	组织机构类型编号	

返回值说明：

返回指定类型的组织机构列表，得到组织机构实体类（Organization），组织机构的属性请参照“javadoc”中“Organization”类的描述。

### 10.7.3 示例程序

```
/**
 * TestManagement.java
 * 作者 Ton 2008-12-3 下午 04:30:37
 */
package com.tf;
import java.net.MalformedURLException;
import java.util.List;
```

```
import com.easymap.management.user.Organization;
import com.easymap.management.user.User;
import com.easymap.management.api.UserManager;

/**
 * @author Ton
 *
 */
public class TestManagement {

    /**
     * @param args
     */
    public static void main(String[] args) {

        try {
            UserManager userManager = new UserManager("http://192.168.10.204:8008/EzManager");

            //类型 11 代表市局
            List<Organization> list = userManager.getOrganizationsByType("11");
            for(int i = 0; i < list.size(); i++){
                Organization org = list.get(i);
                System.out.println("*****组织机构" + (i+1) + "*****");
                System.out.println("组织机构编号: " + org.getId());
                System.out.println("组织机构名称: " + org.getName());
                System.out.println("所属机构编号: " + org.getSsjgdm());
                System.out.println(".....");
            }

        } catch (MalformedURLException e) {
            e.printStackTrace();
        }

    }

}
```



## 10.8 获取用户所属组织机构接口说明

### 10.8.1 功能说明

根据用户登录名获取该用户所属的组织机构。此方法在“com.easymap.management.api”包中的“UserManager”类中。

### 10.8.2 语法定义

语法定义：“public Organization getStandarOrganization(String userid);”

请求参数说明：

参数名称	描述	参数约束
userid	用户登录名	

返回值说明：

返回用户所属的组织机构以及下级组织机构。

### 10.8.3 示例程序

```
/**
 * TestManagement.java
 * 作者 Ton 2008-12-3 下午 04:30:37
 */
package com.tf;
import java.net.MalformedURLException;
import java.util.List;

import com.easymap.management.user.Organization;
import com.easymap.management.user.User;
import com.easymap.management.api.UserManager;

/**
 * @author Ton
 */
public class TestManagement {

    /**
     * @param args
     */
    public static void main(String[] args) {

        try {
```

```
        UserManager userManager = new UserManager("http://192.168.10.204:8008/EzManager");

        Organization org = userManager.getStandarOrganization ("123");

        System.out.println("*****用户 123 所属组织机构*****");
        System.out.println("组织机构编号: "+org.getId());
        System.out.println("组织机构名称: "+org.getName());
        System.out.println("所属机构编号: "+org.getSsjgdm());
        System.out.println(".....");

        //下级组织机构
        getChild(org.getUnit());

    } catch (Exception e) {
        e.printStackTrace();
    }
}

public static void getChild(List<Organization> unit){
    for(int i = 0; i < unit.size(); i++){
        Organization org = unit.get(i);
        System.out.println("*****组织机构 "+(i+1)+"*****");
        System.out.println("组织机构编号: "+org.getId());
        System.out.println("组织机构名称: "+org.getName());
        System.out.println("所属机构编号: "+org.getSsjgdm());
        System.out.println(".....");

        //下级组织机构
        getChild(org.getUnit());
    }
}
}
```

## 10.9 查询所有用户接口说明

### 10.9.1 功能说明

查询所有用户记录。此方法在“com.easymap.management.api”包中的“UserManager”类中。

### 10.9.2 语法定义

语法定义：

“public List<User> queryUsers() throws Exception;”

请求参数说明：无

参数名称	描述	参数约束
------	----	------

返回值说明：用户对象列表

异常抛出：当有异常抛出时，查询用户失败。

### 10.9.3 示例程序

```
/**
 * UserManagerTest.java
 * 作者 Ton 2011-03-11 上午 09:30:37
 */

package api;

import java.util.ArrayList;
import java.util.List;

import com.easymap.management.api.UserManager;
import com.easymap.management.user.User;

public class UserManagerTest {

    /**
     * @param args
     */
    public static void main(String[] args) {
        UserManager um = new UserManager("http://192.168.10.182:8008/EzManager");
        try {
            List<User> uList = um.queryUsers();
            System.out.println("用户个数" + uList.size());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

## 10.10 根据用户 ID 和密码查询用户接口说明

### 10.10.1 功能说明

根据用户 ID 和密码，查询匹配用户记录。此方法在“com.easymap.management.api”包中的“UserManager”类中。

### 10.10.2 语法定义

语法定义：

“public User findUserByIdAndPassword(String userId, String password) throws Exception;”  
请求参数说明：

参数名称	描述	参数约束
userId	用户 ID	
password	密码	

返回值说明：用户对象

异常抛出：当有异常抛出时，查询用户失败。

### 10.10.3 示例程序

```
/**
 * UserManagerTest.java
 * 作者 Ton 2011-03-11 上午 09:30:37
 */

package api;

import com.easymap.management.api.UserManager;
import com.easymap.management.user.User;

public class UserManagerTest {

    /**
     * @param args
     */
    public static void main(String[] args) {
        UserManager um = new UserManager("http://192.168.10.182:8008/EzManager");
        try {
            User user = um.findUserByIdAndPassword("10000", "abcdefg");
            System.out.println("用户 ID: " + user.getId());
            System.out.println("用户名称: " + user.getName());
            System.out.println("用户所属组织机构编码: " + user.getOrgId());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
        System.out.println("用户所属组织机构名称: " + user.getOrgName());
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
```

## 10.11 根据所属组织机构编号查找用户接口说明

### 10.11.1 功能说明

根据用户的所属组织机构编号查找同机构下的所有用户用户记录。此方法在“com.easymap.management.api”包中的“UserManager”类中。

### 10.11.2 语法定义

语法定义：

“public List<User> findUsersByOrganizationId(String id) throws Exception;”

请求参数说明：

参数名称	描述	参数约束
id	组织机构编号	

返回值说明：用户对象列表

异常抛出：当有异常抛出时，查询用户失败。

### 10.11.3 示例程序

```
/**
 * UserManagerTest.java
 * 作者 Ton 2011-03-11 上午 09:30:37
 */

package api;

import java.util.ArrayList;
import java.util.List;

import com.easymap.management.api.UserManager;
import com.easymap.management.user.User;

public class UserManagerTest {
```

```
/**
 * @param args
 */
public static void main(String[] args) {
    UserManager um = new UserManager("http://192.168.10.182:8008/EzManager");
    try {
        List<User> uList = um.findUsersByOrganizationId("100000000200");
        System.out.println("用户个数" + uList.size());
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

## 10.12 根据名称查找用户接口说明

### 10.12.1 功能说明

根据姓名查找用户记录，由于可能存在重名的人所以此方法返回同名的用户列表。此方法在“com.easymap.management.api”包中的“UserManager”类中。

### 10.12.2 语法定义

语法定义：

“public List<User> findUsersByName(String name) throws Exception;”

请求参数说明：

参数名称	描述	参数约束
Name	用户名称	

返回值说明：用户对象列表

异常抛出：当有异常抛出时，查询用户失败。

### 10.12.3 示例程序

```
/**
 * UserManagerTest.java
 * 作者 Ton 2011-03-11 上午 09:30:37
 */

package api;

import java.util.ArrayList;
```

```
import java.util.List;

import com.easymap.management.api.UserManager;
import com.easymap.management.user.User;

public class UserManagerTest {

    /**
     * @param args
     */
    public static void main(String[] args) {
        UserManager um = new UserManager("http://192.168.10.182:8008/EzManager");
        try {
            List<User> uList = um.findUsersByName("admin");
            System.out.println("名称 admin 的用户个数" + uList.size());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

## 11 图层信息接口说明

### 11.1 获取图层字段信息接口说明

#### 11.1.1 功能说明

根据提供的图层表名和 GIS 数据库用户名查询出，指定图层表的所有字段信息。此方法在“com.easymap.management.api”包中的“DataManager”类中。

#### 11.1.2 语法定义

语法定义：

“public List<FieldProperty> getFieldProperty(String enname, String stdcode);”

请求参数说明：

参数名称	描述	参数约束
enname	图层表名	
stdcode	图层标准代码	

返回值说明：

返回指定图层中的所有字段，以“List<FieldProperty>”存储，通过“FieldProperty”实体类获取字段信息。

#### 11.1.3 示例程序

```
/**
 * TestManagement.java
 * 作者 Ton 2008-12-3 下午 04:30:37
 */
package com.tf;

import java.util.ArrayList;
import java.util.List;

import com.easymap.management.api.DataManager;
import com.easymap.management.data.FieldProperty;

/**
 * @author Ton
 */
public class TestManagement {
```



```
/**
 * @param args
 */
public static void main(String[] args) {

    try {
        DataManager d = new DataManager("http://192.168.10.182:8008/EzManager");

        List l = d.getFieldProperty("KC_RK_ZZRK_GL_PT", "XEZGAGIS");
        for(int i = 0; i < l.size(); i++){
            FieldProperty field = (FieldProperty)l.get(i);
            System.out.println(field.getFieldName());
        }
    } catch (Exception e) {
        // TODO 自动生成 catch 块
        e.printStackTrace();
    }
}
```

## 11.2 获取图层表字段名称接口说明

### 11.2.1 功能说明

根据提供的图层表名和 GIS 数据库用户名查询出表的字段名称。此方法在“com.easymap.management.api”包中的“DataManager”类中。

### 11.2.2 语法定义

语法定义：

“public List<String> getTableColumns(String tableName, String owner);”

请求参数说明：

参数名称	描述	参数约束
tableName	图层表名	
dbuser	GIS 数据库用户名	

返回值说明：

返回指定图层中的所有字段名称，以“List<String>”存储。

### 11.2.3 示例程序

```
/**
 * TestManagement.java
 * 作者 Ton 2008-12-3 下午 04:30:37
 */
package com.tf;

import java.util.ArrayList;
import java.util.List;

import com.easymap.management.api.DataManager;

/**
 * @author Ton
 */
public class TestManagement {

    /**
     * @param args
     */
    public static void main(String[] args) {

        try {
            DataManager d = new DataManager("http://192.168.10.182:8008/EzManager");

            List<String> l = d.getTableColumns("KC_PK_ZZPK_GL_PT", "XEZGAGIS");
            for(int i = 0; i < l.size(); i++){
                String fieldName = l.get(i);
                System.out.println(fieldName);
            }
        } catch (Exception e) {
            // TODO 自动生成 catch 块
            e.printStackTrace();
        }
    }
}
```