目录

[1 监测点 3](#_Toc4314657)

[1.1 \*备案信息 3](#_Toc4314658)

[1.1.1 流程 3](#_Toc4314659)

[1.1.2 数据状态分析 3](#_Toc4314660)

[1.1.3 所需接口： 5](#_Toc4314661)

[1.1.3.1 填写备案或修改已审核备案 5](#_Toc4314662)

[1.1.3.2 修改以保存备案信息 5](#_Toc4314663)

[1.1.3.3 备案查询 5](#_Toc4314664)

[1.2 \*上报数据 6](#_Toc4314665)

[1.2.1 流程： 6](#_Toc4314666)

[1.2.1 数据状态分析 6](#_Toc4314667)

[1.2.2 所需接口： 8](#_Toc4314668)

[1.2.2.1 填写数据或上传以保存数据 8](#_Toc4314669)

[1.2.2.2 数据查询 8](#_Toc4314670)

[1.3 \*数据查询 8](#_Toc4314671)

[1.3.1 流程 8](#_Toc4314672)

[1.3.2 所需接口 9](#_Toc4314673)

[1.3.2.1 监测点用户查询数据 9](#_Toc4314674)

[1.3.2.2 各报表数据查询接口 9](#_Toc4314675)

[1.4 浏览通知 17](#_Toc4314676)

[2 市级用户 17](#_Toc4314677)

[2.1 \*上传数据审核 17](#_Toc4314678)

[2.1.1 流程 17](#_Toc4314679)

[2.1.2 所需接口： 18](#_Toc4314680)

[2.1.2.1 待审核数据查询 18](#_Toc4314681)

[2.1.2.2 等待省级审核的数据查询 18](#_Toc4314682)

[2.1.2.3 市级审核不通过 18](#_Toc4314683)

[2.1.2.4 市级审核通过 18](#_Toc4314684)

[2.1.2.5 备注 19](#_Toc4314685)

[2.2 \*数据查询 19](#_Toc4314686)

[2.2.1 流程 19](#_Toc4314687)

[2.2.2 所需接口： 19](#_Toc4314688)

[2.2.2.1 条件查询下属监测点信息 19](#_Toc4314689)

[2.2.2.2 查询下属所有监测点 20](#_Toc4314690)

[2.2.2.3 查询监测点报表数据 20](#_Toc4314691)

[2.3 \*账号信息查询 21](#_Toc4314692)

[2.3.1 流程 21](#_Toc4314693)

[2.3.2 所需接口： 21](#_Toc4314694)

[2.3.2.1 对创建的账号进行条件查询 21](#_Toc4314695)

[2.3.2.2 备注 22](#_Toc4314696)

[2.4 通知管理 22](#_Toc4314697)

[2.5 数据分析 22](#_Toc4314698)

[3 省级用户 22](#_Toc4314699)

[3.1 \*备案审核 22](#_Toc4314700)

[3.1.1 流程 22](#_Toc4314701)

[3.1.2 所需接口： 22](#_Toc4314702)

[3.1.2.1 待审核信息查询 22](#_Toc4314703)

[3.1.2.2 备案信息查询 23](#_Toc4314704)

[3.1.2.3 不通过 23](#_Toc4314705)

[3.1.2.4 通过 23](#_Toc4314706)

[3.1.2.5 备注 23](#_Toc4314707)

[3.2 \*上传数据审核 24](#_Toc4314708)

[3.2.1 流程 24](#_Toc4314709)

[3.2.2 所需接口： 24](#_Toc4314710)

[3.2.2.1 待审核数据查询 24](#_Toc4314711)

[3.2.2.2 省级审核不通过 24](#_Toc4314712)

[3.2.2.3 省级审核通过 25](#_Toc4314713)

[3.2.2.4 备注 25](#_Toc4314714)

[3.3 \*数据查询 25](#_Toc4314715)

[3.3.1 流程 25](#_Toc4314716)

[3.3.1.1 查询下属所有监测点 26](#_Toc4314717)

[3.3.2 所需接口： 26](#_Toc4314718)

[3.3.2.1 查询下属所有监测点 26](#_Toc4314719)

[3.3.2.2 查询市级用户下属监测点 27](#_Toc4314720)

[3.3.2.3 查询省级用户下属市级用户 27](#_Toc4314721)

[3.3.2.4 条件查询下属监测点信息 27](#_Toc4314722)

[3.3.2.5 查询监测点报表数据 28](#_Toc4314723)

[3.4 \*账号信息查询 28](#_Toc4314724)

[3.4.1 流程 28](#_Toc4314725)

[3.4.2 所需接口： 29](#_Toc4314726)

[3.4.2.1 对创建的账号进行条件查询 29](#_Toc4314727)

[3.4.2.2 备注 29](#_Toc4314728)

[3.5 \*上报时限 30](#_Toc4314729)

[3.5.1 流程 30](#_Toc4314730)

[3.5.2 所需接口 31](#_Toc4314731)

[3.5.2.1 新建上报时限 31](#_Toc4314732)

[3.5.2.2 修改上报时限 31](#_Toc4314733)

[3.5.2.3 ID查询 32](#_Toc4314734)

[3.5.2.4 时间点查询 32](#_Toc4314735)

[3.5.2.5 时间段查询 33](#_Toc4314736)

[3.5.2.6 查询所有时限 33](#_Toc4314737)

[3.5.2.7 备注 33](#_Toc4314738)

[3.6 \*角色管理 34](#_Toc4314739)

[3.6.1 接口描述 34](#_Toc4314740)

[3.6.1.1 权限管理模块 34](#_Toc4314741)

[3.6.1.2 角色管理模块 36](#_Toc4314742)

[3.6.1.3 用户管理模块 39](#_Toc4314743)

[3.7 通知管理 41](#_Toc4314744)

[3.8 数据分析 41](#_Toc4314745)

[3.8.1.1 报表管理 41](#_Toc4314746)

[3.8.1.2 数据汇总 42](#_Toc4314747)

[3.8.1.3 取样分析 43](#_Toc4314748)

[3.8.1.4 固定报表 44](#_Toc4314749)

[3.8.1.5 图表分析-对比分析 44](#_Toc4314750)

[3.8.1.6 图表分析-趋势分析 45](#_Toc4314751)

[3.8.1.7 数据查询与导出 46](#_Toc4314752)

# 监测点

## \*备案信息

### 流程

用户在未进行备案时可以进行备案；用户在已经备案后可以修改信息。

在进行备案时，监测点用户需要填写备案信息。用户此时可以选择保存或上传：保存就是将数据保存到数据库中，不上传，在后续用户可以自由修改保存信息；上传就是将备案信息提交到省级，由省级开始进行审核。对于保存的信息，用户可以在确认无误时，选择将其进行上传。在保存或上报后，用户都可以看到该备案信息。

当审核未通过时，用户将受到一条系统发出的关于审核未通过的通知。此时用户在备案界面可以看到其提交的备案信息，作为保存状态。（用户可以选择将该备案信息变成保存信息，继续进行编辑；或者可以选择重新进行备案信息的填写。若选择重新填写，如果在填写后并没有进行任何的保存或者上传操作，则该未通过的备案信息不会消失，在下次访问备案界面时任然会回到选择保存或重新填写的界面。）

当审核通过时，用户将受到一条系统发出的关于审核通过的通知。用户可以在备案界面备案信息。此时用户可以选择修改该备案信息。若选择修改，则用户将直接在之前审核成功的备案信息的基础上进行修改。修改不能进行保存，只能选择直接提交。在修改成功后，修改信息成为当前的有效信息，而旧备案信息则被改为过期。用户不可查看过期信息。修改信息同样需要审核。若修改未通过，则会收到一条通知，备案界面不会有变化；若修改通过，则也会收到一条通知，备案界面将修改为新的备案信息。

### 数据状态分析

**1.保存**

|  |  |
| --- | --- |
| **状态码** | 0 |
| **可见度** | 监测点用户可见，省级用户不可见 |
| **可修改程度** | 监测点用户可修改 |
| **允许存在多条** | 不允许 |
| **时间标志** | 每次创建或修改保存时，将begin\_data修改为当前时刻 |
| **是否为最终状态** | 否 |
| **状态转化过程** | 新建0，新建一个备案并保存  0到0，下载旧的保存，修改并保存  0到1，将已保存的进行上传 |

**2.上传**

|  |  |
| --- | --- |
| **状态码** | 1 |
| **可见度** | 监测点用户可见，省级用户可见 |
| **可修改程度** | 不可修改 |
| **允许存在多条** | 不允许 |
| **时间标志** | 上传时，将begin\_data修改为当前时刻 |
| **是否为最终状态** | 否 |
| **状态转化过程** | 新建1，新建一个备案并上传  0到1，将已保存的进行上传  1到2，备案审核未通过  1到3，备案审核通过 |

**3.审核通过**

|  |  |
| --- | --- |
| **状态码** | 3 |
| **可见度** | 监测点用户可见，省级用户可见 |
| **可修改程度** | 不可修改 |
| **允许存在多条** | 不允许 |
| **时间标志** | 审核通过时，将end\_data修改为当前时刻 |
| **是否为最终状态** | 是 |
| **状态转化过程** | 1到3，备案审核通过  3到4，旧备案过期 |

**4.备案过期**

|  |  |
| --- | --- |
| **状态码** | 4 |
| **可见度** | 监测点用户不可见，省级用户不可见 |
| **可修改程度** | 不可修改 |
| **允许存在多条** | 允许 |
| **时间标志** | 无 |
| **是否为最终状态** | 是 |
| **状态转化过程** | 3到4，当修改备案审核通过时，旧备案过期 |

### 所需接口：

#### 填写备案或修改已审核备案

路径

/market/record/insert

输入参数

request，HttpServletRequest类，会话管理相关（见文末备注，下同）

record对象（见文末备注，下同），包含备案及用户相关数据

stateFlag，整数，为0保存，为1上传

返回结果

result，整数，1表示操作失败，0表示操作成功

#### 修改以保存备案信息

路径

/market/record/update

输入参数

request，HttpServletRequest类，会话管理相关

record对象，包含备案及用户相关数据

stateFlag，整数，为0保存，为1上传

返回结果

result，整数，1表示操作失败，0表示操作成功

#### 备案查询

路径

/market/record/select

输入参数

request，HttpServletRequest类，会话管理相关

返回结果

查询成功则返回一个record对象，

查询无结果则返回null

## \*上报数据

### 流程：

在完成备案后，监测点账号变为激活状态，可以上报数据。人力资源市场用户会根据预先设定的模板在规定的时间范围内进行填报，填写完成后保存，保存就是将数据保存到数据库中，不上传，在后续用户可以自由修改保存信息，在确认无误后就可以进行上传，上传是指将数据提交到市级，由市级进行审核，审核状态下可以进行查看但无法进行数据修改。（下同）

当审核未通过时，监测点用户将收到一条系统发出的关于审核未通过的通知。此时用户在上传数据界面可以看到其提交的数据信息，作为保存状态。（用户可以选择将该数据变成保存数据，继续进行编辑；或者可以选择重新进行数据填写。若选择重新填写，如果在填写后并没有进行任何的保存或者上传操作，则该未通过的数据不会消失，在下次访问上传数据界面时任然会回到选择保存或重新填写的界面。）审核未通过可能发生在市级或省级，无论在何处审核未通过都会进行上述操作。在省级审核未通过时，数据将直接退回到监测点。

当数据在省级通过后，用户将受到一条系统发出的关于数据审核在省级通过的通知。

上传的数据在对监测点用户都是可见的

### 数据状态分析

**1.保存**

|  |  |
| --- | --- |
| **状态码** | 0 |
| **可见度** | 监测点用户可见，省级用户不可见 |
| **可修改程度** | 监测点用户可修改 |
| **允许存在多条** | 不允许 |
| **时间标志** | 每次创建或修改保存时，将begin\_data修改为当前时刻 |
| **是否为最终状态** | 否 |
| **状态转化过程** | 新建0，新建一个备案并保存  0到0，下载旧的保存，修改并保存  0到1，将已保存的进行上传 |

**2.市审核中**

|  |  |
| --- | --- |
| **状态码** | 1 |
| **可见度** | 监测点用户可见，市级用户可见，省级用户不可见 |
| **可修改程度** | 不可修改 |
| **允许存在多条** | 允许 |
| **时间标志** | 上传时，将begin\_data修改为当前时刻 |
| **是否为最终状态** | 否 |
| **状态转化过程** | 1到0，数据审核未通过  1到2，数据审核通过，上交到省级 |

**3.省审核中**

|  |  |
| --- | --- |
| **状态码** | 2 |
| **可见度** | 监测点用户可见，市级用户可见，省级用户可见 |
| **可修改程度** | 不可修改 |
| **允许存在多条** | 允许 |
| **时间标志** | 上传时，将begin\_data修改为当前时刻 |
| **是否为最终状态** | 否 |
| **状态转化过程** | 2到4，数据在省级审核未通过  2到3，数据在审核通过 |

**4.省级审核未通过**

|  |  |
| --- | --- |
| **状态码** | 4 |
| **可见度** | 监测点用户可见，市级用户可见，省级用户不可见 |
| **可修改程度** | 不可修改 |
| **允许存在多条** | 允许 |
| **时间标志** | 上传时，将begin\_data修改为当前时刻 |
| **是否为最终状态** | 否 |
| **状态转化过程** | 4到0，数据经市级确认后回退到对应监测点 |

**5.入库**

|  |  |
| --- | --- |
| **状态码** | 3 |
| **可见度** | 监测点用户不可见，市级用户不可见，省级用户可见 |
| **可修改程度** | 不可修改 |
| **允许存在多条** | 允许 |
| **时间标志** | 审核通过时，将end\_data修改为当前时刻 |
| **是否为最终状态** | 是 |
| **状态转化过程** | 无 |

### 所需接口：

#### 填写数据或上传以保存数据

路径

/market/data /fillOrHand

输入参数

request，HttpServletRequest类，会话管理相关

data对象，包含备案及用户相关数据

stateFlag，整数，为0保存，为1上传

返回结果

result，整数，1表示操作失败，0表示操作成功

#### 数据查询

路径

/market/data/select

输入参数

request，HttpServletRequest类，会话管理相关

返回结果

查询成功则返回一个record对象，

查询无结果则返回null

## \*数据查询

### 流程

用于可能存在的“选择某个监测点用户并查询该监测点数据”场景。

前端提供目标监测点用户id，后端在数据库中对于数据表格表（**t\_upload\_info**）进行查询，获得创建者id和前端提供的用户id相同的所有数据表格中的数据，并将它们以uploadInfo实体数组的形式返回至前端。

### 所需接口

#### 监测点用户查询数据

路径：

/market/data/select

功能：

查询操作用户上传过的数据报表。

输入参数：

appUserDetail，AppUserDetail类，会话管理相关

返回结果：

[数组]

UploadInfo对象，包含以下信息

**public class UploadInfo {//上传数据信息表，共7条数据**

**Integer tableId;//数据表id**

**Integer uploadPeriodId;//所属上传期id**

**Integer stateFlag;//状态标记**

**Date createTime;//创建时间**

**Integer creator;//创建者**

**Date reviseTime;//修改时间**

**Integer reviser;//修改者**

**}**

#### 各报表数据查询接口

##### 供求总体人数表查询

路径：

/market/data/totalNumSelect

输入参数：

tableId，整数，目标报表的id

返回结果：

totalNum对象，内容如下

**public class TotalNum {//供求总体人数表，共3条数据**

**Integer tableId;//数据表id**

**Integer needPopu;//需求人数**

**Integer jobseekPopu;//求职人数**

**}**

##### 产业需求人数表查询

路径：

/market/data/industryNumSelect

输入参数：

tableId，整数，目标报表的id

返回结果：

industryNum对象，内容如下

**public class IndustryNum {//产业需求人数表，共23条数据**

**Integer tableId;//数据表id**

**Integer industry1Need;//第一产业需求人数**

**Integer industry2Need;//第二产业需求人数**

**Integer industry3Need;//第三产业需求人数**

**Integer mineNeed;//采矿业，第二产业**

**Integer manuNeed;//制造业，第二产业**

**Integer elecGasWaterNeed;//电力、燃气及水的生产和供应业，第二产业**

**Integer archNeed;//建筑业，第二产业**

**Integer tranStorPostNeed;//交通运输、仓储和邮政业，第三产业**

**Integer infoCompSoftNeed;//信息传输、计算机服务和软件业，第三产业**

**Integer retailNeed;//批发和零售业，第三产业**

**Integer accoCaterNeed;//住宿和餐饮业，第三产业**

**Integer financeNeed;//金融业，第三产业**

**Integer estateNeed;//房地产业，第三产业**

**Integer leaseBusiServNeed;//租赁和商务服务业，第三产业**

**Integer reseTechAddrNeed;//科学研究、技术服务和地址勘查业，第三产业**

**Integer waterEnviFaciNeed;//水利、环境和公共设施管理业，第三产业**

**Integer resiServNeed;//居民服务和其他服务业，第三产业**

**Integer eduNeed;//教育，第三产业**

**Integer healSecuWelfNeed;//卫生、社会保障和社会福利业，第三产业**

**Integer cultSportEnteNeed;//文化、体育和娱乐业，第三产业**

**Integer manaOrgaNeed;//公共管理与社会组织，第三产业**

**Integer inteOrgaNeed;//国际组织，第三产业**

**}**

##### 用人单位性质需求人数表查询

路径：

/market/data/employerNumSelect

输入参数：

tableId，整数，目标报表的id

返回结果：

employerNum对象，内容如下

**public class EmployerNum {//用人单位性质需求人数表，共15条数据**

**Integer tableId;//数据表id**

**Integer staOwnNeed;//国有企业，内资企业，企业**

**Integer collNeed;//集体企业，内资企业，企业**

**Integer coopStockNeed;//股份合作企业，内资企业，企业**

**Integer jointNeed;//联营企业，内资企业，企业**

**Integer limiLiabNeed;//有限责任公司，内资企业，企业**

**Integer limitedNeed;//股份有限公司，内资企业，企业**

**Integer privNeed;//私营企业，内资企业，企业**

**Integer otherEnteNeed;//其他企业，内资企业，企业**

**Integer hmtInveNeed;//港、澳、台投资企业，企业**

**Integer foreInveEnte;//外商投资企业，企业**

**Integer indiNeed;//个体经营，企业**

**Integer instNeed;//事业**

**Integer orgaNeed;//机关**

**Integer otherNeed;//其他**

**}**

##### 职业供求人数表查询

路径：

/market/data/profNumSelect

输入参数：

tableId，整数，目标报表的id

返回结果：

profNum对象，内容如下

**public class ProfNum {//职业供求人数表，共16条数据**

**Integer tableId;//数据表id**

**Integer leaderNeed;//单位负责人需求人数**

**Integer leaderJobseek;//单位负责人求职人数**

**Integer profTechNeed;//专业技术人员需求人数**

**Integer profTechJobseek;//专业技术人员求职人数**

**Integer staffRelaNeed;//办事人员和有关人员需求人数**

**Integer staffRelaJobseek;//办事人员和有关人员求职人数**

**Integer busiServNeed;//商业和服务人员需求人数**

**Integer busiServJobseek;//商业和服务人员求职人数**

**Integer prodNeed;//农林牧渔水利生产人员需求人数**

**Integer prodJobseek;//农林牧渔水利生产人员求职人数**

**Integer operNeed;//生产运输设备操作工需求人数**

**Integer operJobseek;//生产运输设备操作工求职人数**

**Integer otherNeed;//其他需求人数**

**Integer otherJobseek;//其他求职人数**

**Integer noRequJobseek;//无要求求职人数**

**}**

##### 需求前十职业表查询

路径：

/market/data/mostNeededSelect

输入参数：

tableId，整数，目标报表的id

返回结果：

mostNeeded对象，内容如下

**public class MostNeeded {//需求前十职业表，共41条数据**

**Integer tableId;//数据表id**

**String mostProf1Name;//最需要排名第一的职业名称**

**Integer mostProf1Num;//最需要排名第一的职业代码**

**Integer mostProf1Need;//最需要排名第一的职业需求人数**

**Integer mostProf1Jobseek;//最需要排名第一的职业求职人数**

**String mostProf2Name;//最需要排名第二的职业名称**

**Integer mostProf2Num;//最需要排名第二的职业代码**

**Integer mostProf2Need;//最需要排名第二的职业需求人数**

**Integer mostProf2Jobseek;//最需要排名第二的职业求职人数**

**String mostProf3Name;//最需要排名第三的职业名称**

**Integer mostProf3Num;//最需要排名第三的职业代码**

**Integer mostProf3Need;//最需要排名第三的职业需求人数**

**Integer mostProf3Jobseek;//最需要排名第三的职业求职人数**

**String mostProf4Name;//最需要排名第四的职业名称**

**Integer mostProf4Num;//最需要排名第四的职业代码**

**Integer mostProf4Need;//最需要排名第四的职业需求人数**

**Integer mostProf4Jobseek;//最需要排名第四的职业求职人数**

**String mostProf5Name;//最需要排名第五的职业名称**

**Integer mostProf5Num;//最需要排名第五的职业代码**

**Integer mostProf5Need;//最需要排名第五的职业需求人数**

**Integer mostProf5Jobseek;//最需要排名第五的职业求职人数**

**String mostProf6Name;//最需要排名第六的职业名称**

**Integer mostProf6Num;//最需要排名第六的职业代码**

**Integer mostProf6Need;//最需要排名第六的职业需求人数**

**Integer mostProf6Jobseek;//最需要排名第六的职业求职人数**

**String mostProf7Name;//最需要排名第七的职业名称**

**Integer mostProf7Num;//最需要排名第七的职业代码**

**Integer mostProf7Need;//最需要排名第七的职业需求人数**

**Integer mostProf7Jobseek;//最需要排名第七的职业求职人数**

**String mostProf8Name;//最需要排名第八的职业名称**

**Integer mostProf8Num;//最需要排名第八的职业代码**

**Integer mostProf8Need;//最需要排名第八的职业需求人数**

**Integer mostProf8Jobseek;//最需要排名第八的职业求职人数**

**String mostProf9Name;//最需要排名第九的职业名称**

**Integer mostProf9Num;//最需要排名第九的职业代码**

**Integer mostProf9Need;//最需要排名第九的职业需求人数**

**Integer mostProf9Jobseek;//最需要排名第九的职业求职人数**

**String mostProf10Name;//最需要排名第十的职业名称**

**Integer mostProf10Num;//最需要排名第十的职业代码**

**Integer mostProf10Need;//最需要排名第十的职业需求人数**

**Integer mostProf10Jobseek;//最需要排名第十的职业求职人数**

**}**

##### 饱和前十职业表查询

路径：

/market/data/leastNeededSelect

输入参数：

tableId，整数，目标报表的id

返回结果：

leastNeeded对象，内容如下

**public class LeastNeeded {//饱和前十职业表，共41条数据**

**Integer tableId;//数据表id**

**String leastProf1Name;//最饱和排名第一的职业名称**

**Integer leastProf1Num;//最饱和排名第一的职业代码**

**Integer leastProf1Need;//最饱和排名第一的职业需求人数**

**Integer leastProf1Jobseek;//最饱和排名第一的职业求职人数**

**String leastProf2Name;//最饱和排名第二的职业名称**

**Integer leastProf2Num;//最饱和排名第二的职业代码**

**Integer leastProf2Need;//最饱和排名第二的职业需求人数**

**Integer leastProf2Jobseek;//最饱和排名第二的职业求职人数**

**String leastProf3Name;//最饱和排名第三的职业名称**

**Integer leastProf3Num;//最饱和排名第三的职业代码**

**Integer leastProf3Need;//最饱和排名第三的职业需求人数**

**Integer leastProf3Jobseek;//最饱和排名第三的职业求职人数**

**String leastProf4Name;//最饱和排名第四的职业名称**

**Integer leastProf4Num;//最饱和排名第四的职业代码**

**Integer leastProf4Need;//最饱和排名第四的职业需求人数**

**Integer leastProf4Jobseek;//最饱和排名第四的职业求职人数**

**String leastProf5Name;//最饱和排名第五的职业名称**

**Integer leastProf5Num;//最饱和排名第五的职业代码**

**Integer leastProf5Need;//最饱和排名第五的职业需求人数**

**Integer leastProf5Jobseek;//最饱和排名第五的职业求职人数**

**String leastProf6Name;//最饱和排名第六的职业名称**

**Integer leastProf6Num;//最饱和排名第六的职业代码**

**Integer leastProf6Need;//最饱和排名第六的职业需求人数**

**Integer leastProf6Jobseek;//最饱和排名第六的职业求职人数**

**String leastProf7Name;//最饱和排名第七的职业名称**

**Integer leastProf7Num;//最饱和排名第七的职业代码**

**Integer leastProf7Need;//最饱和排名第七的职业需求人数**

**Integer leastProf7Jobseek;//最饱和排名第七的职业求职人数**

**String leastProf8Name;//最饱和排名第八的职业名称**

**Integer leastProf8Num;//最饱和排名第八的职业代码**

**Integer leastProf8Need;//最饱和排名第八的职业需求人数**

**Integer leastProf8Jobseek;//最饱和排名第八的职业求职人数**

**String leastProf9Name;//最饱和排名第九的职业名称**

**Integer leastProf9Num;//最饱和排名第九的职业代码**

**Integer leastProf9Need;//最饱和排名第九的职业需求人数**

**Integer leastProf9Jobseek;//最饱和排名第九的职业求职人数**

**String leastProf10Name;//最饱和排名第十的职业名称**

**Integer leastProf10Num;//最饱和排名第十的职业代码**

**Integer leastProf10Need;//最饱和排名第十的职业需求人数**

**Integer leastProf10Jobseek;//最饱和排名第十的职业求职人数**

**}**

##### 人员类别求职人数表查询

路径：

/market/data/jobSeekerNumSelect

输入参数：

tableId，整数，目标报表的id

返回结果：

jobseekerNum对象，内容如下

**public class JobSeekerNum {//人员类别求职人数表，共11条数据**

**Integer tableId;//数据表id**

**Integer unempYouth;//新成长失业青年**

**Integer graduate;//应届高校毕业生**

**Integer empToUnemp;//就业转失业人员**

**Integer othUnemp;//其他失业人员**

**Integer emped;//在业人员**

**Integer laidOff;//下岗职工**

**Integer retiree;//退休人员**

**Integer student;//在学人员**

**Integer cityRural;//本市农村人员**

**Integer fore;//外埠人员**

**}**

##### 性别供求人数表查询

路径：

/market/data/sexNumSelect

输入参数：

tableId，整数，目标报表的id

返回结果：

sexNum对象，内容如下

**public class SexNum {//性别供求人数表，共6条数据**

**Integer tableId;//数据表id**

**Integer maleNeed;//男性需求人数**

**Integer maleJobseek;//男性求职人数**

**Integer femaleNeed;//女性需求人数**

**Integer femaleJobseek;//女性求职人数**

**Integer noRequNeed;//无要求需求人数**

**}**

##### 年龄供求人数表查询

路径：

/market/data/ageNumSelect

输入参数：

tableId，整数，目标报表的id

返回结果：

ageNum对象，内容如下

**public class AgeNum {//年龄供求人数表，共10条数据**

**Integer tableId;//数据表id**

**Integer sixteenTwentyfourNeed;//16-24岁需求人数**

**Integer sixteenTwentyfourJobseek;//16-24岁求职人数**

**Integer twentyfiveThirtyfourNeed;//25-34岁需求人数**

**Integer twentyfiveThirtyfourJobseek;//25-34岁求职人数**

**Integer thirtyfiveFortyfourNeed;//35-44岁需求人数**

**Integer thirtyfiveFortyfourJobseek;//35-44岁求职人数**

**Integer overFortyfourNeed;//45岁以上需求人数**

**Integer overFortyfourJobseek;//45岁以上求职人数**

**Integer noRequNeed;//无要求需求人数**

**}**

##### 文化程度供求人数表查询

路径：

/market/data/degreeNumSelect

输入参数：

tableId，整数，目标报表的id

返回结果：

degreeNum对象，内容如下：

**public class DegreeNum {//文化程度供求人数表，共12条数据**

**Integer tableId;//数据表id**

**Integer belowJuniHighSchoNeed;//初中及以下需求人数**

**Integer belowJuniHighSchoJobseek;//初中及以下求职人数**

**Integer highSchoNeed;//高中需求人数**

**Integer highSchoJobseek;//高中求职人数**

**Integer otherHighSchoNeed;//职高技校中专需求人数**

**Integer otherHighSchoJobseek;//职高技校中专求职人数**

**Integer juniCollNeed;//大专需求人数**

**Integer juniCollJobseek;//大专求职人数**

**Integer univNeed;//大学需求人数**

**Integer univJobseek;//大学求职人数**

**Integer noRequNeed;//无要求需求人数**

**}**

##### 技术等级供求人数表查询

路径：

/market/data/techGradeNumSelect

输入参数：

tableId，整数，目标报表的id

返回结果：

techGradeNum对象，内容如下：

**public class TechGrageNum {//技术等级供求人数表，共19条数据**

**Integer tableId;//数据表id**

**Integer profLevel5Need;//职业资格五级（初级技能）需求人数**

**Integer profLevel5Jobseek;//职业资格五级（初级技能）求职人数**

**Integer profLevel4Need;//职业资格四级（中级技能）需求人数**

**Integer profLevel4Jobseek;//职业资格四级（中级技能）求职人数**

**Integer profLevel3Need;//职业资格三级（高级技能）需求人数**

**Integer profLevel3Jobseek;//职业资格三级（高级技能）求职人数**

**Integer profLevel2Need;//职业资格二级（技师）需求人数**

**Integer profLevel2Jobseek;//职业资格二级（技师）求职人数**

**Integer profLevel1Need;//职业资格一级（高级技师）需求人数**

**Integer profLevel1Jobseek;//职业资格一级（高级技师）求职人数**

**Integer primProfNeed;//初级专业技术职务需求人数**

**Integer primProfJobseek;//初级专业技术职务求职人数**

**Integer mediProfNeed;//中级专业技术职务需求人数**

**Integer mediProfJobseek;//中级专业技术职务求职人数**

**Integer seniProfNeed;//高级专业技术职务需求人数**

**Integer seniProfJobseek;//高级专业技术职务求职人数**

**Integer noTechJobseek;//无技术等级或职务求职人数**

**Integer noRequNeed;//无要求需求人数**

**}**

## 浏览通知

# 市级用户

## \*上传数据审核

### 流程

监测点用户的数据需要提交到市级，进行审核。当有数据上传到市级时，市级将收到一条系统发出的关于有待审核数据的通知，市级用户需要对该数据进行审核后选择通过或不通过。

当选择通过时，数据会上传到省级再次进行审核，审核过程中市级可以查看数据。

当选择不通过时，数据将会退回到对应的监测点，退回后将不能查看该数据。

市级用户只能看到用户提交的未被审核的，以及通过市级审核并提交到省级等待审核的数据。对于其他状态的数据，市级是不可见的。

任何时候市级都不能修改用户上传的数据。

### 所需接口：

#### 待审核数据查询

路径

/city /data /examineQuery

输入参数

无

返回结果

[数组]

data对象，包含监测点用户上传的数据

#### 等待省级审核的数据查询

路径

/city /data /examineWaitProvinceQuery

输入参数

无

返回结果

[数组]

data对象，包含监测点用户上传的数据

#### 市级审核不通过

路径

/city /data/ reject

输入参数

request，HttpServletRequest类，会话管理相关

aimId，整数，目标监测点用户的id

返回结果

result，整数，成功为0，失败为1。

#### 市级审核通过

路径

/city /data /pass

输入参数

request，HttpServletRequest类，会话管理相关

aimId，整数，目标监测点用户的id

返回结果

result，整数，成功为0，失败为1

#### 备注

data类包含如下内容：

regionEmpName,字符串，人力资源市场名称

regionName，字符串，所属地区

stateFlag，整数，状态标志，保存0，市级审核中1，省级审核中2，省级审核未通过3，入库4

createTime，日期，创建时间

creater，整数,创建人id

以及各种人力供求数据,均为整数

request为HttpServletRequest类的对象，进行会话管理，代表当前进行操作的用户

## \*数据查询

### 流程

包含查询所有下属监测点数据和条件查询下属监测点数据两种，通过其他接口的组合来完成对应功能。

**查询所有下属监测点数据：**使用**市/省级用户查询下属所有监测点**接口获得相应的监测点用户信息，再通过**查询监测点用户数据**接口和**数据报表查询**接口获得相关数据。

**条件查询下属监测点数据：**使用**市/省级用户条件查询下属用户信息**接口获取相应的监测点用户信息，再通过**查询监测点用户数据**接口和**数据报表查询**接口获得相关数据。

### 所需接口：

#### 条件查询下属监测点信息

路径：

/province/userSearch

功能：

省/市级用户可以通过此接口条件查询下级用户信息。

输入参数：

appUserDetail，AppUserDetail类，会话管理相关

inputString，字符串，用户搜索时输入的信息

返回结果：

[数组]

userInfoDisplay对象，包含以下信息

**public class userInfoDisplay{**

**Integer userId;//用户id**

**string username；//用户名**

**string fullname；//用户全称**

**}**

#### 查询下属所有监测点

路径：

province/selectAllSub

功能：

市/省级用户可以通过这个接口查询下属所有监测点的信息。

输入参数：

appUserDetail，AppUserDetail类，会话管理相关

返回结果：

[数组]

userInfoDisplay对象，包含以下信息

**public class userInfoDisplay{**

**Integer userId;//用户id**

**string username；//用户名**

**string fullname；//用户全称**

**}**

#### 查询监测点报表数据

路径

/province/data/selectMarketData

功能：

输入一个检测点用户的id，返回该用户上传过的报表的信息。

输入参数：

appUserDetail，AppUserDetail类，会话管理相关

aimUserId，整数，目标监测点用户id

返回结果：

[数组]

UploadInfo对象，包含以下信息

**public class UploadInfo {//上传数据信息表，共7条数据**

**Integer tableId;//数据表id**

**Integer uploadPeriodId;//所属上传期id**

**Integer stateFlag;//状态标记**

**Date createTime;//创建时间**

**Integer creator;//创建者**

**Date reviseTime;//修改时间**

**Integer reviser;//修改者**

**}**

## \*账号信息查询

### 流程

在查询界面中，有两个条件输入栏分别为用户名和用户全称，对应输入后点击搜索按钮，页面中就会显示出所有满足查询条件的账号列表，选中特定的账号进行双击，页面中就会显示出该账号的详细信息。两个条件均可不填，这种条件下进行查询，市级会查询到其辖区内所有的账号。

所查询的账号必须是处于激活状态的账号。

查询时可以按条件进行查询，提供的查询字段有:

username 用户名，字母、数字和‘-’、‘\_’，禁止使用中文或？%等字符（可不填）

fullname 用户全称（可不填）

对于查询到的账号，可以查看其基本信息，其中包括：

username 用户名

fullname 用户全称

superior 上级管理部门

create\_time 创建时间

creator 创建者

只能查看，不能修改

### 所需接口：

#### 对创建的账号进行条件查询

路径

/city/account/accountQuery

输入参数

appUserDetail AppUserDetail类，包含当前用户的信息

username 字符串，所查询的账号的用户名，

fullname 字符串，所查询的账号的用户全称

返回结果

[数组]

data对象，包含所查询到的账号的一些信息

#### 备注

Data类包含如下内容：

username 字符串，用户名

fullname 字符串，用户全称

superior 整数，上级管理部门ID

create\_time 日期，创建时间

creator 整数，创建者

## 通知管理

## 数据分析

# 省级用户

## \*备案审核

### 流程

监测点用户的备案信息需要提交到省级，进行审核。省级用户需要对该备案信息选择通过或不通过：当审核通过时，则完成备案或备案信息修改；当审核未通过时，备案信息将被退回。

省级用户只能看到用户提交的备案申请信息，以及通过审核的备案信息。对于用户保存的信息、未通过审核的信息还有过期信息，省级是不可见的。

任何时候省级都不能修改用户的备案数据。

### 所需接口：

#### 待审核信息查询

路径

/province/record/examineQuery

输入参数

无

返回结果

[数组]

record对象，包含备案及用户相关数据

#### 备案信息查询

路径

/province/record/examineQuery

输入参数

State，整数，0代表根据监测点名称查询，1代表根据地区查询，2代表根据联系人姓名查询

condition，字符串，为上述查询内容

返回结果

[数组]

record对象，包含备案及用户相关数据

#### 不通过

路径

/province/record/reject

输入参数

request，HttpServletRequest类，会话管理相关

aimId，整数，目标监测点用户的id

返回结果

result，整数，成功为0，失败为1

#### 通过

路径

/province/record/pass

输入参数

request，HttpServletRequest类，会话管理相关

aimId，整数，目标监测点用户的id

返回结果

result，整数，成功为0，失败为1

#### 备注

record类包含如下内容：

regionEmpName,字符串，人力资源市场名称

regionName，字符串，所属地区

contactRegionEmpName，字符串，联系人姓名

regionEmpContactMobi，字符串，联系人手机

regionEmpContactNum，字符串，联系电话

regionEmpFax，字符串，传真

stateFlag，整数，状态标志，保存0，上传1，通过2，过期3

createTime，日期，创建时间

creater，整数创建人id

reviseTIme，日期审核时间

reviser，整数，审核人id

request为HttpServletRequest类的对象，进行会话管理，代表当前进行操作的用户

## \*上传数据审核

### 流程

通过市级审核的数据需要提交到省级，进行再次审核。当有数据从市级提交到省级时，省级会收到一条系统发出的关于有数据待审核的通知，省级用户需要对该数据进行审核并选择通过或不通过。

当选择通过时，数据会被保存入库，之后该数据将只有省级可见；

当选择不通过时，数据将直接退回到相应监测点。

省级用户只能看到市级提交的未被审核的、以及通过省级审核的数据。对于其他状态的数据，省级是不可见的。

任何时候省级都不能修改市级上传的数据。。

### 所需接口：

#### 待审核数据查询

路径

/province/data/examineQuery

输入参数

无

返回结果

[数组]

record对象，包含备案及用户相关数据

#### 省级审核不通过

路径

/province/data/reject

输入参数

request，HttpServletRequest类，会话管理相关

aimId，整数，目标监测点用户的id

返回结果

result，整数，成功为0，失败为1

#### 省级审核通过

路径

/province/data /pass

输入参数

request，HttpServletRequest类，会话管理相关

aimId，整数，目标监测点用户的id

返回结果

result，整数，成功为0，失败为1

#### 备注

data类包含如下内容：

regionEmpName,字符串，人力资源市场名称

regionName，字符串，所属地区

stateFlag，整数，状态标志，保存0，市级审核中1，省级审核中2，省级审核未通过3，入库4

createTime，日期，创建时间

creater，整数,创建人id

以及各种人力供求数据,均为整数

request为HttpServletRequest类的对象，进行会话管理，代表当前进行操作的用户

## \*数据查询

### 流程

包含查看所有下属监测点数据，查看某市下属监测点数据，按条件查询下属监测点数据三种，通过其他接口的组合来完成对应功能。

**查询所有下属监测点数据：**使用**市/省级用户查询下属所有监测点**接口获取所有监测点用户信息，再通过**查询监测点用户数据**接口和**数据报表查询**接口获得相关数据报表相关信息。

**查看某市下属监测点数据：**首先通过**查询省级用户下属市级用户**接口显示出所有的市级用户，之后通过**查询目标市级用户下属监测点**接口查询其中某个市级用户下属的监测点用户，之后通过**查询监测点用户数据**接口和**数据报表查询**接口获得相关信息。

**条件查询下属监测点数据**：使用**市/省级用户条件查询下属用户信息**接口获取相应的监测点用户信息，再通过**查询监测点用户数据**接口和**数据报表查询**接口获得相关数据。

#### 查询下属所有监测点

路径：

province/selectAllSub

功能：

市/省级用户可以通过这个接口查询下属所有监测点的信息。

输入参数：

appUserDetail，AppUserDetail类，会话管理相关

返回结果：

[数组]

userInfoDisplay对象，包含以下信息

**public class userInfoDisplay{**

**Integer userId;//用户id**

**string username；//用户名**

**string fullname；//用户全称**

**}**

### 所需接口：

#### 查询下属所有监测点

路径：

province/selectAllSub

功能：

市/省级用户可以通过这个接口查询下属所有监测点的信息。

输入参数：

appUserDetail，AppUserDetail类，会话管理相关

返回结果：

[数组]

userInfoDisplay对象，包含以下信息

**public class userInfoDisplay{**

**Integer userId;//用户id**

**string username；//用户名**

**string fullname；//用户全称**

**}**

#### 查询市级用户下属监测点

路径：

/province/selectCitySub

功能：

输入一个市级用户的id，返回该用户下属监测点用户的信息。

输入参数：

appUserDetail，AppUserDetail类，会话管理相关

aimUserId，目标市级用户id

返回结果：

[数组]

userInfoDisplay对象，包含以下信息

**public class userInfoDisplay{**

**Integer userId;//用户id**

**string username；//用户名**

**string fullname；//用户全称**

**}**

#### 查询省级用户下属市级用户

路径：

/province/selectCityUser

功能：

查询操作用户下属的市级用户。

输入参数：

appUserDetail，AppUserDetail类，会话管理相关

返回结果：

[数组]

userInfoDisplay对象，包含以下信息

**public class userInfoDisplay{**

**Integer userId;//用户id**

**string username；//用户名**

**string fullname；//用户全称**

**}**

#### 条件查询下属监测点信息

路径：

/province/userSearch

功能：

省/市级用户可以通过此接口条件查询下级用户信息。

输入参数：

appUserDetail，AppUserDetail类，会话管理相关

inputString，字符串，用户搜索时输入的信息

返回结果：

[数组]

userInfoDisplay对象，包含以下信息

**public class userInfoDisplay{**

**Integer userId;//用户id**

**string username；//用户名**

**string fullname；//用户全称**

**}**

#### 查询监测点报表数据

路径

/province/data/selectMarketData

功能：

输入一个检测点用户的id，返回该用户上传过的报表的信息。

输入参数：

appUserDetail，AppUserDetail类，会话管理相关

aimUserId，整数，目标监测点用户id

返回结果：

[数组]

UploadInfo对象，包含以下信息

**public class UploadInfo {//上传数据信息表，共7条数据**

**Integer tableId;//数据表id**

**Integer uploadPeriodId;//所属上传期id**

**Integer stateFlag;//状态标记**

**Date createTime;//创建时间**

**Integer creator;//创建者**

**Date reviseTime;//修改时间**

**Integer reviser;//修改者**

**}**

## \*账号信息查询

### 流程

在查询界面中，有两个条件输入栏分别为用户名和用户全称，对应输入后点击搜索按钮，页面中就会显示出所有满足查询条件的账号列表，选中特定的账号进行双击，页面中就会显示出该账号的详细信息。两个条件均可不填，这种条件下进行查询，省级会查询到所有的账号。

所查询的账号必须是处于激活状态的账号。

查询时可以按条件进行查询，提供的查询字段有:

username 用户名，字母、数字和‘-’、‘\_’，禁止使用中文或？%等字符（可不填）

fullname 用户全称（可不填）

对于查询到的账号，可以查看其基本信息，其中包括：

username 用户名

fullname 用户全称

superior 上级管理部门

create\_time 创建时间

creator 创建者

只能查看，不能修改

### 所需接口：

#### 对创建的账号进行条件查询

路径

/province/account/accountQuery

输入参数

appUserDetail AppUserDetail类，包含当前用户的信息

username 字符串，所查询的账号的用户名，

fullname 字符串，所查询的账号的用户全称

返回结果

[数组]

data对象，包含所查询到的账号的一些信息

#### 备注

Data类包含如下内容：

username 字符串，用户名

fullname 字符串，用户全称

superior 整数，上级管理部门ID

create\_time 日期，创建时间

creator 整数，创建者

## \*上报时限

### 流程

包括新建时限，修改时限和查询时限，其中查询时限又分出时间点查询，时间段查询，时限id查询三种。

**日期规则：**

开始日期不能晚于结束日期。

一个调查期的起止日期间隔需要大于三十天。

##### 新建上报时限：

用户输入起止日期后进行确认，后端按照以下逻辑进行操作：

如果起止日期在服务器的系统时间之前或是结束日期在开始日期前，那么会返回错误数值；

如果时间间隔小于一个月，那么会返回错误数值；

后端程序会查询已有的所有时限进行检测，如果发现输入的日期和已有日期重叠则会返回错误数值。

如果日期合法，那么后端会在上传期表中进行insert操作，其中起止日期来自用户输入，而

上传期id，创建者id，创建时间由后端自动生成，最后一次修改时间和最后一次修改者则为null，删除标记为0。

##### 修改时限:

用户在前端可以选择某个上报时限进行操作。此时前端显示的每个上报时限都有对应的id，当用户选择修改时，需要输入新的起止日期，进行确认后后端获得起止日期和目标上报时限id，按照以下逻辑操作：

如果起止日期在服务器的系统时间之前或是结束日期在开始日期前，那么会返回错误数值。

如果时间间隔小于一个月，那么会返回错误数值；

后端程序会查询已有的所有时限进行检测，如果发现输入的日期和已有日期重叠则会返回错误数值。

如果日期合法，那么后端会在上传期表中选择对应id的时限进行update操作，开始日期和结束日期字段会根据用户输入进行更新，最后修改日期和最后修改者字段会由后端自动更新。

##### 按时间点查询时限

用户可以在前端选择某个时间点进行查询，此时后端会所有时限进行查询，将上传期表中起止日期包括该时间点的时限相关信息返回，返回数据为一个包含所有时限表字段的json文件。如果无结果则返回null。

##### 按时间段查询时限

用户在前端选择起止时间，此时后端会对所有时限进行查询，将上传期表中创建日期在该时间范围内的时限信息返回，返回数据为一个包含所有时限表字段的json文件。如果无结果则返回null。

##### 按id查询时限

前端提供目标时限id。后端进行select操作，查询目标id的时限。如果获得结果则将时限相关信息返回，返回数据为一个包含所有时限表字段的json文件。如果无结果则返回null。

### 所需接口

#### 新建上报时限

**路径：**

province/investigatePeriod/insert

**功能：**

通过用户输入的起止日期新建上报时限。

**输入参数：**

appUserDetail，AppUserDetail类，会话管理相关

uploadPeriod对象，详细信息见文末备注，此处仅需要下列属性：

startDate，string，开始日期，将被转换为sql.date

endDate，string，结束日期，将被转换为sql.date

**返回结果：**

result对象，其中具体的错误代码存放于result中的data中

1. 时限不合法，返回错误代码2
2. 新建失败，返回错误代码1
3. 新建成功，返回正确代码0

#### 修改上报时限

**路径：**

province/investigatePeriod/update

**功能：**

修改一个已存在的上报时限的起止日期。

**输入参数：**

appUserDetail，AppUserDetail类，会话管理相关

uploadPeriod对象，详细信息见文末备注，此处仅需要下列属性：

uploadPeriodId，整数，需要修改的时限id

startDate，string，此处为修改后的开始日期，将被转换为sql.date

endDate，string，此处为修改后的结束日期，将被转换为sql.date

**返回结果：**

result对象，其中具体的错误代码存放于result中的data中

1. 时限不合法，返回错误代码2
2. 修改失败，返回错误代码 1
3. 修改成功，返回正确代码 0

#### ID查询

**路径：**

province/investigatePeriod/selectById

**功能：**

向前端返回id和输入匹配的上报时限

**输入参数：**

appUserDetail，AppUserDetail类，会话管理相关

uploadPeriodId，整数，目标上传时限id

**返回结果：**

查询成功时：一个result对象，其中的data属性包含一个uploadPeriod对象（详细见文末备注）

查询失败时：一个result对象，其中data属性为空，message属性中会包含错误说明

#### 时间点查询

**路径：**

province/investigatePeriod/selectByTime

**功能：**

向前端返回起止时间包含目标时间点的上报时限

**输入参数：**

appUserDetail，AppUserDetail类，会话管理相关

inputDate，string，目标时间点，仅包含日期，被转换为sql.date

**返回结果：**

查询成功时：一个result对象，其中的data属性包含一个uploadPeriod对象（详细见文末备注）

查询失败时：一个result对象，其中data属性为空，message属性中会包含错误说明

#### 时间段查询

**路径：**

province/investigatePeriod/selectByPeriod

**功能：**

向前端返回所有创建时间在输入范围内的上报时限

**输入参数：**

appUserDetail，AppUserDetail类，会话管理相关

uploadPeriod对象，详细信息见文末备注，此处仅需要下列属性：

startDate，string，开始日期，将被转换为sql.date

endDate，string，结束日期，将被转换为sql.date

**返回结果：**

[数组]

查询成功时：result对象，每个result的data属性中都包含了创建时间符合要求的uploadPeriod对象列表。

查询失败时：一个result对象，其中data属性为空，message属性中会包含错误说明。

#### 查询所有时限

路径：

province/investigatePeriod/selectAllPeriod

输入参数：

appUserDetail，AppUserDetail类，会话管理相关

返回结果：

[数组]

查询成功时：result对象，每个result的data属性中都包含了创建时间符合要求的uploadPeriod对象列表。

查询失败时：一个result对象，其中data属性为空，message属性中会包含错误说明。

#### 备注

**uploadPeriod类：**

标志着一个调查期的全部信息，包含如下属性：

{

uploadPeriodId，整数，时限id

startDate，string，开始日期，将被转换为sql.date（仅包含日期，不包括具体时间）

endDate，string，结束日期，将被转换为sql.date

creatTime，string，创建时间，将被转换为util.date（包含日期和详细时间）

creator，整数，创建用户id

reviseTime，string，最后修改时间，将被转换为util.date

reviser，整数，最后修改用户id

deleteFlag，整数，删除标志位，弃用

}

## \*角色管理

？？？？？？

估计是设置不同的权限角色，进行权限管理的

### 接口描述

#### 权限管理模块

##### 查看权限列表

路径：

/right/list

描述：

获得一个树状权限列表，注意返回结果是一棵树，深度最多2层

输入参数：

无

返回结果：

[数组项

{

id, 整数：权限ID

name，字符串：权限名称

description：字符串：权限描述

status，整数：权限状态，0表示启用，1表示禁用

[数组项

{//子权限

…(同样的格式)

}

]

}

]

使用场景：

1. 管理员的权限添加删除
2. 省/管理员创建角色使用

##### 查看权限信息

路径：

/right/info

描述：

查看某权限信息

输入参数：

rightId，整数：权限ID

返回结果：

name，字符串：权限名称

description，字符串：权限描述

status，整数：权限状态，0启用，1禁用

##### 添加权限（管理员使用）

路径：

/right/add

描述：

添加一条权限，

注：

1. 权限名称不得重复
2. 权限名称纯英文大写，用下划线隔开单词

输入参数：

name，字符串：权限名称

description，字符串：权限描述

menuName，字符串：菜单名称

menuPath，字符串：菜单路径

father，整数：父权限ID

返回结果：

无

##### 修改权限（管理员使用）

路径：

/right/modify

描述：

修改一条权限，

注：

1. 权限名称不得重复
2. 权限名称纯英文大写，用下划线隔开单词

输入参数：

id，整数：权限ID

name，字符串：权限名称

description，字符串：权限描述

menuName，字符串：菜单名称

menuPath，字符串：菜单路径

返回结果：

无

##### 禁用权限（管理员使用）

路径：

/right/disable

描述：

禁用一条权限，注：类似假删

输入参数：

rightId，整数：目标权限的Id

返回结果：

无

##### 启用权限（管理员使用）

路径：

/right/enable

描述：

启用一条被禁用的权限

输入参数：

rightId，整数：目标权限的Id

返回结果：

无

#### 角色管理模块

##### 查看角色列表

路径：

/role/list

描述：

获取已有角色列表

输入参数：

无

返回结果：

[数组]

id，整数：角色ID

name，字符串：角色名称

description，字符串：角色描述

status，整数：角色状态，0启用，1禁用

##### 查看角色信息

路径：

/role/info

描述：

查看某角色信息

输入参数：

roleId，整数：角色ID

返回结果：

name，字符串：角色名称

description，字符串：角色描述

status，整数：角色状态，0启用，1禁用

##### 查看角色权限

路径：

/role/rights

描述：

获得一个角色所拥有的全部权限，数组形式。

输入参数：

roleId，整数：角色ID

返回结果：

[数组]

整数：权限ID

##### 添加角色

路径：

/role/add

描述：

添加一个角色，注：角色名称不得重复

输入参数：

name，字符串：角色名称

description，字符串：角色描述

返回结果：

无

##### 修改角色

路径：

/role/modify

描述：

修改一个角色，注：角色名称不得重复

输入参数：

id，整数：角色ID

name，字符串：角色名称

description，字符串：角色描述

返回结果：

无

##### 禁用角色

路径：

/role/disable

描述：

禁用一个角色，注：类似假删

输入参数：

roleId，整数：目标角色的Id

返回结果：

无

##### 启用角色

路径：

/role/enable

描述：

启用一个被禁用的角色

输入参数：

roleId，整数：目标角色的Id

返回结果：

无

##### 设置权限

路径：

/role/setRights

描述：

修改一个角色的权限

注：

后端将先删除后添加。

输入参数：

roleId，整数：角色ID

rights，数组，整数：该角色拥有的所有权限的ID

返回结果：

无

#### 用户管理模块

##### 查看用户列表

路径：

/user/list

描述：

查看已有用户列表，不包含自身

输入参数：

无

返回结果：

[数组]

username，字符串：用户名

fullname，字符串：用户全称

id，整数：用户ID

status，整数：0表示用户有效，1表示无效

father，整数：父级ID

##### 禁用用户

路径：

/user/disable

描述：

禁用一个用户，注：类似假删

输入参数：

userId，整数：目标用户的Id

返回结果：

无

##### 启用用户

路径：

/user/enable

描述：

启用一个被禁用的用户

输入参数：

userId，整数：目标用户的Id

返回结果：

无

##### 添加用户

路径：

/user/addUsers

描述：

批量生成用户，注意用户名不得重复

生成的格式为prefix+index。Index范围（startCount至count+startCount）

name同理

输入参数：

count，整数：要生成的个数，默认为1

prefix，字符串：生成的用户名的前缀

name，字符串：生成的用户全称的前缀

startCount，整数：生成时的起始下标

parent，整数：父级用户ID

返回结果：

无

##### 查看用户角色

路径：

/user/role/info

描述：

查看一个用户拥有的角色

输入参数：

userId，整数：用户ID

返回结果：

[数组]

id，整数：角色ID

name，字符串：角色名称

description，字符串：角色描述

##### 修改用户角色

路径：

/user/role/change

描述：

修改一个用户的角色

输入参数：

userId，整数：用户ID

roles，数组，整数：该用户拥有的所有角色的ID

返回结果：

无

##### 查看用户菜单

路径：

/user/menu

描述：

获得当前登录用户的菜单，返回结果是一棵树

注：

admin会得到全部有效菜单

输入参数：

无

返回结果：

[数组]

menuName，字符串：菜单名称

menuPath，字符串：菜单路径

child，子菜单项，格式同上。

## 通知管理

## 数据分析

#### 报表管理

路径：

/provincial\_level/check

描述：

审核地市（区县）上报的数据

内部流程：

系统将数据库中存储的数据返回

输入参数：

无

返回结果：

[数组]

Content，字符串：数据

Date，字符串：上报时间

#### 数据汇总

路径：

/provincial\_level/summarizing

描述：

查看人力资源市场用户的汇总数据

内部流程：

系统将数据进行汇总并返回

输入参数：

无

返回结果：

[数组]

Title，字符串：用户ID

Content，字符串：用户信息

Date，字符串：收集时间

#### 取样分析

路径：

/provincial\_level/sampling

描述：

显示各市（区县）人力资源市场用户的数量和占比

内部流程：

系统将各市（区县）人力资源市场用户数量和计算出的总体占比返回

输入参数：

无

返回结果：

[数组]

Title，字符串：市（区县）名称

Number，整数：用户数量

percent, 浮点数：用户数量占总体百分比

#### 固定报表

路径：

/provincial\_level/fixed\_table

描述：

查看各调查期内各市（区县）人力资源市场用户需求人数、需求比重、环比和同化的变化情况

输入参数：

starttime 字符串： 开始日期

endtime 字符串：结束日期

返回结果：

[数组]

Title，字符串：市（区县）的名称

NeedNumber，字符串：需求人数

NeededPercent，浮点数：需求比重

Chain,浮点数：环比

year-one-year，浮点数：同比

#### 图表分析-对比分析

路径：

/provincial\_level/chart\_analysis

描述：

使用折线图、表格形式对比两个调查期的人力资源市场数据变动情况

输入参数：

Startstime 字符串：开始日期

Endtime 字符串：结束日期

Startstime2 字符串：开始日期2

Endtime2 字符串：结束日期2

返回结果：

以折线图、表格形式显示的两个调查期的人力资源市场数据对比情况

#### 图表分析-趋势分析

路径：

/provincial\_level/chart\_analysis

描述：

使用折线图、表格形式展示多个连续调查期内的需求人数、求职人数变动情况

输入参数：

Startstime 字符串：日期

Startstime2 字符串：日期2

Starttime3 字符串：日期3

…………….

Starttimen 字符串：日期n

返回结果：

使用折线图、表格形式展示的对应连续调查期内的需求人数、求职人数变动情况

#### 数据查询与导出

路径：

/provincial\_level/dataquery\_educe

描述：

对全省已创建账号进行条件查询

输入参数：

condition1 字符串：条件一（可选）

condition2 字符串：条件二（可选）

condition3 字符串：条件三（可选）

condition4 字符串：条件四（可选）

condition5 字符串：条件五（可选）

返回结果：

所有满足条件的账号