课程目标:

1. App信息页面查询条件的加载

>>加载平台

>>加载一级、二级、三级分类

实现二级联动

1. 分析App信息显示的查询语句

>>六表连查

1. 理解无限级分类存储
2. 加载app信息显示页面(所属平台的加载)

分析:

1. 编写持久化层

略

1. 编写业务层

接口:

|  |
| --- |
| */\*\* \* 通过类型名称加载相关字典 \** ***@param typename*** *类型名称 \** ***@return*** *返回某类型的相关字典信息 \*/* **public** List<DataDictionary> getDictionary(String typename); |

实现:

|  |
| --- |
| *//返回异步数据* @RequestMapping(**"/getPt"**) @ResponseBody **public** List<DataDictionary> getPt(){  *//return dictionaryService.getDictionary("所属平台");* **return dictionaryService**.getDictionary(SystemConstant.***APP\_PT***); } |

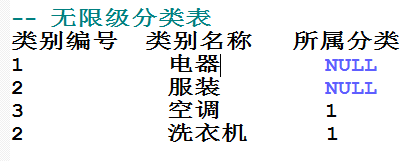
1. 编写控制器层

|  |
| --- |
| *//定义字典类（常量）*  **public class** SystemConstant {  *//字典 就是固定的常量值* **public static final** String ***APP\_STATE***=**"APP状态"**;  **public static final** String ***APP\_PT***=**"所属平台"**; }  *//返回异步数据* @RequestMapping(**"/getAppPtData"**) @ResponseBody **public** List<DataDictionary> **getAppPtData**(){  *//return dictionaryService.getDictionary("所属平台");* **return dictionaryService**.getDictionary(SystemConstant.***APP\_PT***); } |

1. 编写前端展示页面

|  |
| --- |
| *//2.发送异步请求获app的平台信息* $.get(**"${**pageContext.request.contextPath**}/getAppPtData"**,**null**,**function**(data){  *//循环数据生成option添加下拉列表* **for**(**var** i=0;i<data.**length**;i++){  **var** option=$(**"<option value='"**+data[i].valueid+**"'>"**+data[i].valuename+**"</option>"**);  $(**"#queryFlatformId"**).append(option);  } },**"json"**); |

1. 加载app信息显示页面(分类加载)
2. 掌握知识:无限级分类存储问题.



1. 分析分类的加载

-- 查询一级分类

SELECT \* FROM `app\_category` WHERE parentId IS NULL;

-- 查询某分类下的子类别

SELECT \* FROM `app\_category` WHERE parentId = 类别编号;

1. 实现类别的加载
   1. 编写持久化层

略

* 1. 编写业务层(查询某类别类下的子类)

接口：

|  |
| --- |
| */\*\*  \* 查询分类下的子类  \** ***@param id*** *类别编号  \* 如果id传null表示查询一级分类、如果不为空，则查询当前类型下的子类  \** ***@return*** *当前类别下的子类  \*/* List<AppCategory> getCatagoryByParent (Long id); |

实现:

|  |
| --- |
| @Override **public** List<AppCategory> getCatagoryByParent(Long id) {  AppCategoryExample appCategoryExample=**new** AppCategoryExample();  AppCategoryExample.Criteria criteria =appCategoryExample.createCriteria();  *//添加条件* **if**(id == **null**){  criteria.andParentidIsNull();  }**else** criteria.andParentidEqualTo(id);  **return appCategoryMapper**.selectByExample(appCategoryExample); } |

* 1. 编写控制器

|  |
| --- |
| @Controller **public class** CatagroyController {   @Autowired(required = **false**)  **private** CatagoryService **catagoryService**;    *//传递父类型编号，不传查询一级，传递类别则查询当前类别子类* @RequestMapping(**"/getCatagoryByPid"**)  @ResponseBody  **public** List<AppCategory> getCatagoryByPid(Long id){  **return catagoryService**.getCatagoryByParent(id);   } } |

* 1. 编写前端代码:

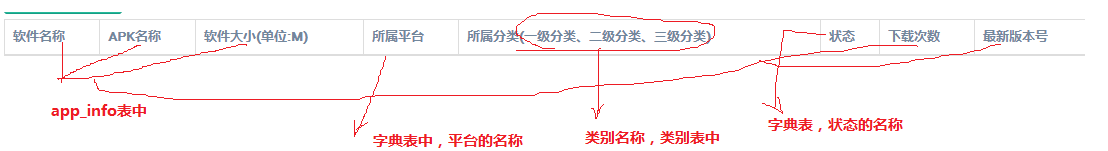
|  |
| --- |
| *//3.加载一级分类 //3.发送异步请求获取一级分类* $.post(**"${**pageContext.request.contextPath**}/getCatagoryByPid"**,**null**,**function**(data){  *//console.log(data);  //循环数据添加option标签* **for**(**var** i=0;i<data.**length**;i++){  **var** node=**"<option value='"**+data[i].**id**+**"'>"**+data[i].categoryname+**"</option>"**;  *//将node添加到select标签中 添加子节点* $(**"#queryCategoryLevel1"**).append(node);  } },**"json"**); |

>>>加载二级分类、三级分类

**利用下拉列表的onchange事件实现二级联动效果**

|  |
| --- |
| *//4.当选择一级分类时，触发onchange事件加载二级分类:(二级联动效果)* $(**"#queryCategoryLevel1"**).change(**function**(){ *//绑定事件  //1.获取一级分类值* **var** parentId=$(**"#queryCategoryLevel1"**).val();  *//2.发送异步请求通过分类的id查询子类  //参数的格式:{"请求参数名称":参数值,"请求参数名称":参数值...}* $(**"#queryCategoryLevel2>option:gt(0)"**).remove();*//清空原有项* $(**"#queryCategoryLevel3>option:gt(0)"**).remove();*//清空原有项* **if**(parentId!=**""**)  {  $.get(**"${**pageContext.request.contextPath**}/getCategory"**,{**"parentid"**:parentId},**function**(data){  *//循环数据生成option添加下拉列表* **for**(**var** i=0;i<data.**length**;i++){  **var** option=$(**"<option value='"**+data[i].**id**+**"'>"**+data[i].categoryname+**"</option>"**);  $(**"#queryCategoryLevel2"**).append(option);  }  },**"json"**);  } });    *//5.当选择二级分类时，触发onchange事件加载三级分类:(二级联动效果)* $(**"#queryCategoryLevel2"**).change(**function**(){ *//绑定事件  //1.获取二级分类值* **var** parentId=$(**"#queryCategoryLevel2"**).val();  *//2.发送异步请求通过分类的id查询子类  //参数的格式:{"请求参数名称":参数值,"请求参数名称":参数值...}* $(**"#queryCategoryLevel3>option:gt(0)"**).remove();*//清空原有项* **if**(parentId!=**""**)  {  $.get(**"${**pageContext.request.contextPath**}/getCategory"**,{**"parentid"**:parentId},**function**(data){  *//循环数据生成option添加下拉列表* **for**(**var** i=0;i<data.**length**;i++){  **var** option=$(**"<option value='"**+data[i].**id**+**"'>"**+data[i].categoryname+**"</option>"**);  $(**"#queryCategoryLevel3"**).append(option);  }  },**"json"**);  }  }); |

1. 分析app信息查询显示语句



查询解决方法:

-- 第一种: 连接查询

-- 查询app信息及关随着平台名称、状态名秒、一级分类，二级分类，三级分类

|  |
| --- |
| SELECT app\_info.id,  app\_info.softwarename,  app\_info.apkname,  app\_info.softwaresize,  app\_info.downloads,  app\_info.versionid,  ptinfo.valuename AS ptname,  stateinfo.valuename AS sname,  c1.categoryName AS cname1,  c2.categoryName AS cname2,  c3.categoryName AS cname3  FROM app\_info  LEFT JOIN (SELECT valueid,valuename FROM `data\_dictionary` WHERE typename='所属平台') AS ptinfo  ON app\_info.`flatformId`=ptinfo.valueid  LEFT JOIN (SELECT valueid,valuename FROM `data\_dictionary` WHERE typename='app状态') AS stateinfo  ON app\_info.`status`=stateinfo.valueid  LEFT JOIN app\_category AS c1  ON app\_info.`categoryLevel1`=c1.id  LEFT JOIN app\_category AS c2  ON app\_info.`categoryLevel2`=c2.id  LEFT JOIN app\_category AS c3  ON app\_info.`categoryLevel3`=c3.id |

-- 第二种 ：子查询

-- 查询app信息及关随着平台名称、状态名秒、一级分类，二级分类，三级分类

|  |
| --- |
| **SELECT app\_info.id,**  **app\_info.softwarename,**  **app\_info.apkname,**  **app\_info.softwaresize,**  **app\_info.downloads,**  **app\_info.versionid,**  **(SELECT valuename FROM `data\_dictionary` WHERE typename='所属平台' AND valueid=flatformId) AS ptname,**  **(SELECT valuename FROM `data\_dictionary` WHERE typename='app状态' AND valueid=STATUS) AS sname,**  **(SELECT categoryname FROM app\_category WHERE id=categoryLevel1) AS cname1,**  **(SELECT categoryname FROM app\_category WHERE id=categoryLevel2) AS cname2,**  **(SELECT categoryname FROM app\_category WHERE id=categoryLevel3) AS cname3**  **FROM app\_info** |