# JAVA实用笔记大全

### 一mabits map方式返回值的内容

|  |
| --- |
| Xml层  <select id=*"selectSampleOperType"* resultType=*"map"*>  select \* from lis\_SampleOperType  </select> |
| Dao层  List<Map<String,String>> selectSampleOperType(); |
| Server层  List<Map<String,String>> selectSampleOperType(); |
| Serveripml层  @Override  **public** List<Map<String, String>> selectSampleOperType() {    **return** lisReqmainMapper.selectSampleOperType();  } |
| Controller层  //selectComDict  @ResponseBody  @RequestMapping(value="selectSampleOperType")  **public** Object selectSampleOperType() {  List<Map<String, String>> list = LisReqmainService.selectSampleOperType();  Map<String, Object> map = **new** HashMap<>();  map.put("code", 0);  map.put("msg", "");  map.put("data",list);  **return** map;  } |
| jsp层  ajax\_GET("/applyfor/selectComHospital",{},  **function**(data){  $.each(data.data,**function**(index,item){  $('#userSongJian').append(**new** Option(item.hosName,item.hosCode));//往下拉菜单里添加元素  })  form.render("select");//菜单渲染 把内容加载进去//成功调用后返回数据 \*/  },  **function**(s){//返回的错误信息  }) |
|  |

### 二 pdf打印获取其默认打印机

|  |
| --- |
| import javax.print.DocFlavor;  import javax.print.attribute.HashPrintRequestAttributeSet;  import javax.print.PrintService;  import javax.print.PrintServiceLookup;  import javax.print.attribute.PrintRequestAttributeSet;  PrintRequestAttributeSet pras = new HashPrintRequestAttributeSet();  DocFlavor flavor = DocFlavor.BYTE\_ARRAY.PNG;  //可用的打印机列表(字符串数组)  PrintService printService[] = PrintServiceLookup.lookupPrintServices(flavor, pras);  for(int i=0;i<printService.length;i++){          Printers[i]=printService[i].getName();  }  //当前默认打印机  PrintService PS = PrintServiceLookup.lookupDefaultPrintService(); PS.getName(); |

### 三 pdf 合并在同一个目录然后前端调用打印功能

|  |
| --- |
|  |

### 四 java获取日期 当前的 前一天的 前一个月的 前一年的

|  |
| --- |
| SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");  Calendar c = Calendar.getInstance();  c.setTime(new Date());  Date end = c.getTime();  String dqrq= format.format(end);//当前日期    c.add(Calendar.DATE, -1);  Date start = c.getTime();  String qyt= format.format(start);//前一天    c.add(Calendar.MONTH, -1);  Date start = c.getTime();  String startDay = format.format(start);//前一月    c.add(Calendar.YEAR, -1);  Date start = c.getTime();  String startDay = format.format(start);//前一年 |

### 五 对文件进行操作

#### 1.0java 删除文件极其子目录

|  |
| --- |
| @ResponseBody  @RequestMapping("deleteFile")  **public** **void** deleteFile(@RequestParam(value="url",required=**true**)String url){  //删除D:/1 路径下的新建文件夹2 不为空的情况下 ==》deleteFile 先删除文件夹里的所有文件 再删除文件夹  System.***out***.println("===================================================================");  System.***out***.println(url);  System.***out***.println("执行删除命令！");  System.***out***.println("===================================================================");  File file3 = **new** File(url);  *deleteFile*(file3);  }  /\*  \* 如果是文件 ==》直接删除  \* 如果是目录 ==》必须先删除里面每一层目录里的所有文件，最后才能删除外层的目录  \* 原因：不为空的话 删不了  \*/  **public** **static** **void** deleteFile(File file) {  **if**(file.exists()) {//判断路径是否存在  **if**(file.isFile()){//boolean isFile():测试此抽象路径名表示的文件是否是一个标准文件。  file.delete();  }**else**{//不是文件，对于文件夹的操作  //保存 路径D:/1/新建文件夹2 下的所有的文件和文件夹到listFiles数组中  File[] listFiles = file.listFiles();//listFiles方法：返回file路径下所有文件和文件夹的绝对路径  **for** (File file2 : listFiles) {  /\*  \* 递归作用：由外到内先一层一层删除里面的文件 再从最内层 反过来删除文件夹  \* 注意：此时的文件夹在上一步的操作之后，里面的文件内容已全部删除  \* 所以每一层的文件夹都是空的 ==》最后就可以直接删除了  \*/  *deleteFile*(file2);  }  }  file.delete();  }**else** {  System.***out***.println("该file路径不存在！！");  }  } |

#### 2.0java判断文件是否存在

|  |
| --- |
| /\*\*  \* 判断文件夹是否存在  \* **@param** file  \*/  **public** **static** **boolean** checkDirExists(File file) {  **if** (file.exists()) {  **if** (file.isDirectory()) {  System.***out***.println("目录存在");  } **else** {  System.***out***.println("同名文件存在, 不能创建");  }  **return** **true**;  } **else** {  System.***out***.println("目录不存在，创建目录");  **return** **false**;  }      } |