**汇购工具类1.0**

**功能手册**

|  |  |
| --- | --- |
| 版本编号 | 0.1.0 |
| 文档编号 |  |
| 发布者 | 系统组 |
| 发布日期 | 2016-4 |

# 概述

## 编制目的

该接口文档手册编写在于描述汇购工具类1.0的功能说明。

## 内容概述

。

## 名称解释

## 参考文献

# 类说明



## 手机号码识别

**类**：hg.util. MobileUtil

**方法**： public static Result checkMobile(String mobile);

**参数说明**：

|  |  |  |  |
| --- | --- | --- | --- |
| **类别** | **参数名称** | **类型** | **说明** |
| 输入 | mobile | string | 要检查的手机号 |

**返回**: hg.util.Result的结构为

|  |  |
| --- | --- |
| **字段** | **错误描述** |
| code | 其中-1:不存在，其他值为有效手机号 |
| text | 详细文本信息 |

**示例:**

main(String[] args){

MobileUtil.checkMobile(“15305153869”);

}

**前端**：

hg-util.js

function checkMobile(mobile);

返回结构同后台。

**示例:**

<#inclide ../hg-util.js>

<script>

alert(checkMobile(“25305153869”); // 返回｛code：“-1”，“无效手机号”｝

</script>

## 加解密DES，RSA，AES

* DESUtil：

**加密：**public static byte[] encrpt（byte[] 原文，byte[] key）

**参数说明**：

|  |  |  |  |
| --- | --- | --- | --- |
| **类别** | **参数名称** | **类型** | **说明** |
| 输入 | text | byte[] | 原文 |
| 输入 | key | byte[] | 密码 |

**返回值：**加密后的密文

**解密：**public static byte[] decrypt（byte[] 密文，byte[] key）

**参数说明**：

|  |  |  |  |
| --- | --- | --- | --- |
| **类别** | **参数名称** | **类型** | **说明** |
| 输入 | text | byte[] | 密文文 |
| 输入 | key | byte[] | 密码 |

**返回值：**解密密后的原文

**示例：**

main(Sring[] args){

byte[] mm=DesUtil.encrypt（“你们”,”abc”.getBytes()）;

DesUtil.decrpty(mm, ”abc”.getBytes());

}

* **AESUtil：**

public static byte[]加密（byte[] 原文，byte[] key）

public static byte[[解密（byte[] 密文，byte[] key）

**用法以及示例同DESUtil。**

* RSAUtil：

**方法**

**产生密码对**：public static KeyPair genKey();

KeyPari:{pulicKey,privateKey}

**加密**:public static byte[] encrpt(byte[] 原文, byte[] key) return 密文

**解密**：public static byte[] decrpt(byte[] 密文,byte[] key) return 原文

**示例**

public static void main(String [] args){

//使用方要自行保存，密码对。以方便后续加解密

KeyPari kp = RSAUtil.genKey();

byte[] mm = RSAUtil.encript(“你们”,kp.getPrivateKey());

byte[] yy = RSAUtil.decrpt(mm, kp.getPublicKey());

}

## JSON数据转

**对象转json**

com.alibaba.fastjson.JSONObjectJSONObject.toJSONString

|  |  |  |  |
| --- | --- | --- | --- |
| **类别** | **参数名称** | **类型** | **说明** |
| 输入 | object | Object | 要转换的对象 |

**返回值：**

|  |  |
| --- | --- |
| **字段** | **描述** |
| json字符串 |  |

**json转对象**

com.alibaba.fastjson.JSONObject.JSONObject.parse

**示例：**

Object o = “abc”;

JSONObject.toJSONString(o);

JSONObject.parse(“{age:1,year:1999}”)

## xml操作 读写

**读/写/删某节点**

* **功能描述：**获取某个标签下的内容(不区分大小写)

**调用格式**： XmlUtil.getNodeVal(String, String)

**参数说明**：

|  |  |  |  |
| --- | --- | --- | --- |
| **类别** | **参数名称** | **类型** | **说明** |
| 输入 | str | string | 整个xml |
|  | tag | string | 标签 |

**返回值：**

|  |  |
| --- | --- |
| **字段** | **描述** |
| 节点值 |  |

* **功能描述：**删除某个标签 (不区分大小写)

**调用格式**： XmlUtil.removeNode(String, String)

**参数说明**：

|  |  |  |  |
| --- | --- | --- | --- |
| **类别** | **参数名称** | **类型** | **说明** |
| 输入 | str | string | 整个xml |
|  | tag | string | 标签 |

**返回值：无。**

* **功能描述：替换**某个标签内容 (不区分大小写)

**调用格式**： XmlUtil.removeNode(String, String)

**参数说明**：

|  |  |  |  |
| --- | --- | --- | --- |
| **类别** | **参数名称** | **类型** | **说明** |
| 输入 | str | string | 整个xml |
|  | tag | string | 标签 |
|  | newValue | string | 标签包含的新的内容 |

**返回值：无。**

**对象与xml转换**

用法以及示例同json。

## 常用日期格式转化 日期运算

yyyyMMdd,

yyyy-MM-dd

yyyy-MM-dd HH:MM:ss

yyyy-MM-dd HH:MM

yyyyMMddHHMMss

这种格式的格式化与解析

日期加：

add(日期1，单位，数量) //单位为Calendar的常量，天，小时，分等

日期差:

diff(日期1，日期2，单位) //返回 日期1到 日期2的间隔（单位为参数指定的单位）

**示例**：

main(String[] arg){

// yyyyMMdd格式的 格式化和解析，其他类同

DateUtil. yyyyMMdd.format(new Date());

DateUtil. yyyyMMdd.parse(“20190101”);

//日期运算

Date d1,d2;

d2=DateUtil.add(d1,Calendar.Hour,1);//加一小时

DateUtil.diff(d1,d2,Calender.Hour); //返回相差多少小时

}

## http client封装 https 封装

static String callWebServiceGet(String url, Map para);

static String callWebServicePost(String url, Map para);

**示例：**

public static void main(String[] fsd){

HttpUtil. callWebServiceGet(<https://baidu.com,new> Map());

}

## 二维码生成和识别

**类**：BarCodeUtil

static public OutputStream gen(String text);

static pulic String recognize(InputStream picStream) //除了二维码，含其他一些信息

**示例：**

public static void main(String[] fsd){

//生成二维码

new FileOutputStream os = new FileOutputStream(“d:/a.jpg”);

BarCodeUtil.gen(“这是一个网址<http://ab.com>”);

os.close();

//识别二维码

new FileInputStream os = new FileintputStream(“d:/a.jpg”);

String text = BarCodeUtil.recognize(os);

os.close();

}

## 图片操作类 如图片的裁 水印 合成

类PicUtil

* //上传

public static HGImage uploadImage(InputStream 图片)

* //下载

public static void downloadImage(HGImage图片,OutputStream 图片写入流)

* //加图片水印

public static void waterMark(InputStream图片, InputStream 水印读入流， OutputStream加印后的图片写入流)

* //加文字水印

public static void waterMark(InputStream图片, String 水印， OutputStream加印后的图片写入流)

* // 压缩

public static void compact (InputStream 图片, OutputStream 压缩后的写入流)

* // 剪裁

public static void cut (InputStream 图片,CutArea 裁剪区域，OutputStream裁剪后的写入流)。//

CutArea为｛topX, topY, width, height｝ 分别为剪裁区域的顶点坐标（相对于图片来说），和宽，高。

Class HGImage{

String imageID;

String imageURL;

}

**示例：**

public static void main(String[] arg){

InputStream pic = new FileInputStream(“d:/adf.jpg”);

//加图片水印

PicUtil.waterMark(pic, new FileInputStream(“d:/logo1.jgp”, new FileOutputStream(“d:/加印后图片.jpg”);

//加文本水印

PicUtil.waterMark(pic, “水印文本在此”,new FileOutputStream(“d:/加印后图片2.jpg”);

//压缩

PicUtil.compact(pic, new FileOutputStream(“d:/压缩后的.jpg”)；

//裁剪

PicUtil.cut(pic, new CutArea(0,0,10,10), new FileOutputStream(“d:/裁剪后的.jpg”);

//上传到服务器

HGImage hgimg=PicUtil.uploadImg(pic);

//下载

PicUtil.downloadImg(hgimg, new FileOutputStream(“d:/下载下的.jpg”);

}

前端js：

picUtil.js

//裁剪.

function cutPic(图片选择器,回调函数); //回调函数可以得到CutArea

**示例：**

**<**include “…/picUti.js>

<div id=img> <img src=””/></div>

<script>

cutPic(“#img”,function(area){

//选择结束后显示区域坐标

alert(area.topX, area.topY, area.width);

});

</script>