

群发终端客户接口 V8.0

版本: V8.0 最后更新: 2017 年 11 月 21 日

目录

一、帐号信息相关接口.....	3
1.1 查询用户信息.....	3
1.2 查询群发服务信息.....	3
二、群发统计相关接口.....	5
2.1 取得指定日期的群发任务列表 (Web 群发专用).....	5
2.2 获取指定日期的群发统计概览.....	6
2.3 导出发送失败地址.....	6
2.4 指定日期邮件批次发送统计表信息.....	7
2.5 指定日期扣点信息.....	8
三、跟踪统计相关接口 (Web 群发专用).....	9
3.1 获取指定群发任务的跟踪统计概览.....	9
3.2 获取指定任务邮件打开统计详情.....	10
3.3 获取指定任务链接打开统计详情.....	11
3.4 导出指定群发任务跟踪邮箱信息.....	11
3.5 指定批次的邮件各链接的点击统计.....	13
四、联系人分类相关接口.....	14
4.1 取得联系人分类列表.....	14
4.2 取得联系人分类详情.....	14
4.3 添加联系人分类.....	15
4.4 修改联系人分类.....	15
4.5 删除联系人分类.....	16
4.6 导出联系人分类地址.....	17
4.7 导出退订地址.....	17
4.8 分类地址列表.....	17
4.9 新增分类地址.....	19
4.10 修改分类地址.....	24
4.11 删除分类地址.....	24
4.12 清空联系人分类.....	24
4.13 邮件订阅.....	25
4.14 邮件退订.....	25
五、邮件模板相关接口.....	26
5.1 创建群发任务.....	26

5.2 上传邮件模板.....	28
5.3 模板列表.....	28
5.4 删除模板.....	29
5.5 修改模板.....	30
5.6 上传模板附件.....	30
5.7 删除模板附件.....	31
六、群发任务相关接口.....	32
6.1 群发任务列表.....	32
6.2 创建群发任务.....	33
6.3 修改群发任务.....	34
6.4 删除群发任务.....	35
七、域名及域名下的邮箱账号.....	35
7.1 域名列表.....	35
7.2 添加域名(管理员).....	36
7.3 删除域名(管理员).....	36
7.4 指定域名下的账号列表.....	36
7.5 添加账号到指定域名.....	37
7.6 删除指定域名下的账号.....	38
7.7 修改指定域名的账号.....	38
八、其他.....	39
8.1 客户自定义群发登录页面说明.....	39
九、用户认证方式示例代码以及接口调用示例代码.....	40
9.1 Python 用户认证示例代码.....	40
9.2 C#用户认证示例代码:	41
9.3 java 用户认证示例代码:	41
9.4 php 用户认证示例代码.....	42
9.5 Ruby On Rails 示例代码:	44
9.6 JS 用户认证示例代码:	45
9.7 接口调用 php 示例代码.....	45
9.8 接口调用 java 示例代码.....	46

【特别备注】

1、老接口地址/api/可继续使用。接口调用前：请先验证用户名和密码，验证方式请参照第九章【用户认证方式示例代码以及接口调用示例代码】说明。使用新功能建议使用新接口/apinew/

2、所有接口采用 utf-8 编码格式;如果需要传入中文、韩文、西班牙语时，请务必将接口 http 的 header 头的编码格式设为 utf-8，并将参数的字符编码 gbk，ASCII、BIG5 等原编码转为 utf-8 编码传入，方可避免编码不一致乱码情况。

一、帐号信息相关接口

1.1 查询用户信息

接口功能：查询用户信息

接口地址：

```
http://www.bestedm.org/mm-ms/apinew/account.php?do=customer-info
```

输出格式：

```
<?xml version="1.0" encoding="UTF-8"?>
<customer>
  <id>用户 ID</id>
  <username>用户名称</username>
  <company>公司名称</company>
  <contact>联系人</contact>
  <phone>联系电话号码</phone>
  <mobile>移动电话号码</mobile>
  <email>联系邮件地址</email>
  <im>即时聊天工具地址</im>
  <address>通信地址</address>
</customer>
```

1.2 查询群发服务信息

接口功能：查询群发服务信息

接口地址：

```
http://www.bestedm.org/mm-ms/apinew/account.php?do=service-info
```

输出格式：

```
<?xml version="1.0" encoding="UTF-8"?>
<service>
  <id>用户 ID</id>
  <service_status>群发服务状态</service_status>
  <service_type>群发服务类型</service_type>
  <send_type>发送方式</send_type>
  <beginning_date>服务开始日期</beginning_date>
  <expiration_date>服务截止日期</expiration_date>
```

```
<qty_available>剩余有效群发量</qty_available>  
<qty_buytotal>群发量购买总计</qty_buytotal>  
</service>
```

说明:

1. **"service_status"**: 用户群发服务状态, "enabled"为允许用户使用群发服务; "disabled"为禁止用户使用群发服务。
2. **"service_type"**: 群发服务类型, "all"为按群发总量进行发送, "day"为按每日发送量进行发送。
3. **"send_type"**: 群发发送方式, "remote"为客户使用自己的服务器与平台对接发送, "local"为使用平台服务器来进行发送。

U-Mail 邮件群发

二、群发统计相关接口

2.1 取得指定日期的群发任务列表（Web 群发专用）

接口功能：取得指定日期的群发任务列表

接口地址：

<http://www.bestedm.org/mm-ms/apinew/task.php?do=list-task&date=2012-03-02>

GET 参数(必要参数)：

1. "date"：所要获取的群发任务的日期

输出格式：

```
<?xml version="1.0" encoding="UTF-8"?>
<task_list>
  <task>
    <id>任务 ID</id>
    <sn>任务序号</sn>
    <sender>发件人地址</sender>
    <sender_name>发件人名称</sender_name>
    <replyto>回复地址</replyto>
    <tpl_id>使用的模板 ID</tpl_id>
    <tpl_name>使用的模板名称</tpl_name>
    <addr_type>发送地址类型</addr_type>
    <list_id>联系人分类 ID</list_id>
    <list_name>联系人分类名称</list_name>
    <send_qty>预定发送数量(用户指定)</send_qty>
    <send_qty_remak>预定发送数量(实际发送)</send_qty_remak>
    <send_time>预定发送时间</send_time>
    <time_start>实际发送开始时间</time_start>
    <time_end>实际发送结束时间</time_end>
    <send_count>实际发送数量</send_count>
    <error_count>发送失败数量</error_count>
    <send_status>发送状态</send_status>
    <track_status>跟踪统计状态</track_status>
  </task>
</task_list>
```

说明：

1. "addr_type" 字段为"0"表示为使用用户联系人分类中的邮件地址来进行发送，为"1"表

示使用邮件订阅列表中的邮件地址来进行发送。

2. "send_qty"字段为用户设置的发送数量，如为"0"则表示使用指定联系人分类的所有地址来发送；此时"send_qty_remark"则显示添加此任务时指定联系人分类中的实际地址数量。
3. "send_status"字段各值含义如下："-1"：暂不发送；"1"：等待发送；"2"：正在发送；"3"：发送完成。
4. "track_status"字段各值含义如下："0"：不踊跃；"1"：跟踪邮件打开情况；"2"：跟踪邮件打开与链接点击情况。

2.2 获取指定日期的群发统计概览

接口功能：获取指定群发任务的跟踪统计概览

接口地址：

<http://www.bestedm.org/mm-ms/apinew/statistic.php?date=2012-03-02>

GET 参数(必要参数)：

1. "date"：所要获取的统计情况的日期

输出格式：

```
<?xml version="1.0" encoding="UTF-8"?>
<statistic>
  <date>日期</date>
  <ws_qty>web发送统计任务量</ws_qty>
  <ws_error>web发送统计失败量</ws_error>
  <ws_count>web发送统计发送量</ws_count>

  <invalid_addr>无效地址</invalid_addr>
  <format_error>格式错误</format_error>

  <send_count>预统计发送量</send_count>
  <deduc_count>预扣点</deduc_count>

  <actual_send>实际发送量</actual_send>
  <send_error>投递失败</send_error>
  <actual_deduc>实际扣点</actual_deduc>
</statistic>
```

2.3 导出发送失败地址

接口功能：导出发送失败地址

接口地址：

<http://www.bestedm.org/mm-ms/apinew/failexport.php>

GET 参数：

1. **"date" (必要参数)**：所要导出失败地址的日期
例：2012-03-02
2. **"type" (必要参数)**：导出错误地址类型
"format_error"或 9：格式错误；
"invalid_addr"或 8：无效地址；
"not_exist"或 1：邮箱不存在；
"over_quota"或 2：邮箱空间满；
"user_reject"或 3：用户拒收
3. **"task_id"**：指定的任务 id，就是获取该次任务的错误地址记录。
4. **"task_ident"**：指定的任务批次号，就是获取该次任务批次相关的任务的错误地址记录。type 参数为 1,2,3 时，task_id 或者 task_ident 其一是必要参数

输出格式：直接输出文本文件，每行一个邮件地址。

2.4 指定日期邮件批次发送统计表信息

接口功能：指定日期邮件批次发送统计表信息

接口地址：

<http://www.bestedm.org/mm-ms/apinew/stask.php?date=2012-03-02&ident=20120302-362-F9B1E5>

GET 参数：

1. **"date"**：所要获取的统计情况的日期
2. **'ident'**：具体的任务批次号

输出格式：

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
  <task>
    <task_id>任务id</task_id>
    <task_date>日期</task_date>
    <task_ident>任务批次</task_ident>
    <total>发送量</total>
    <failed>投递失败量</failed>
```

```

    <success>投递成功数</success>
    <actual>实际扣除点数</actual>
    <email_not_exist>邮箱不存在</email_not_exist>
    <over_quota>空间不足</over_quota>
    <user_reject>用户拒收</user_reject>
  </task>
  ...
</data>

```

2.5 指定日期扣点信息

接口功能：指定日期扣点信息

接口地址：

<http://www.bestedm.org/mm-ms/apinew/deduction.php?date=2012-03-02>

GET 参数(必要参数)：

1. "date"：所要获取的统计情况的日期

输出格式：

```

<?xml version="1.0" encoding="UTF-8"?>
<data>
  <list>
    <date>日期</date>
    <company>公司名</company>
    <deduction>扣点类型</deduction>
    <send_num>发送数量</send_num>
    <total>扣量总计</total>
  </list>
  ...
</data>

```


三、跟踪统计相关接口（Web 群发专用）

3.1 获取指定群发任务的跟踪统计概览

接口功能：获取指定群发任务的跟踪统计概览

接口地址：

<http://www.bestedm.org/mm-ms/apinew/track.php?do=overview>

GET 参数：

1. “id”：群发任务 send_id
2. “ident”：任务批次号

参数 id, ident 两者必须有一个，如果两个都存在，取 ident

输出格式：

<track>

<track_id>跟踪统计 id</track_id> ----- 20121207 新增

<task_id>任务 id</task_id>

<task_sn>任务批次号</task_sn>

<send_count>发送量</send_count>

<real_send_count>成功发送量</real_send_count>

<error_send_count>失败量</error_send_count>

<open_unique>唯一打开数</open_unique>

<open_total>总打开数</open_total>

<open_first>首次打开时间</open_first>

<open_last>最后打开时间</open_last>

<click_unique>唯一点击数</click_unique>

<click_total>总点击数</click_total>

<click_first>首次点击时间</click_first>

<click_last>最后点击时间</click_last>

<open_ratio>打开率</open_ratio>

<click_ratio>点击率</click_ratio>

<link_statistic>

<link>

<link_id>链接 id</link_id>

<link>链接</link>

<click_unique>唯一点击数</click_unique>

<click_total>总点击数</click_total>

<click_first>首次点击时间</click_first>

```

        <click_last>最后点击时间</click_last>
    </link>
    <link>
    ...
    </link>
    ...
</link_statistic>
</track>

```

3.2 获取指定任务邮件打开统计详情

接口功能：获取指定群发任务邮件打开统计详情

接口地址：

<http://www.bestedm.org/mm-ms/apinew/track.php?do=open-detail>

GET 参数：

1. "id"：群发任务 send_id
2. "ident"：任务批次号

参数 id, ident 两者必须有一个，如果两个都存在，取 ident

输出格式：

```

<?xml version="1.0" encoding="UTF-8"?>
<data>
    <user>
        <email_id>邮箱 id</email_id>
        <email>邮箱</email>
        <browser>浏览器</browser>
        <os>操作系统</os>
        <ip_first>首次打开 ip</ip_first>
        <ip_last>最后打开 ip</ip_last>
        <open_total>总打开数</open_total>
        <click_total>总点击数</click_total>
        <open_first>首次打开时间</open_first>
        <open_last>最后打开时间</open_last>
        <click_first>首次点击时间</click_first>
        <click_last>最后点击时间</click_last>
    </user>
    ...
</data>

```

3.3 获取指定任务链接打开统计详情

接口功能：获取指定群发任务邮件打开统计详情

接口地址：

<http://www.bestedm.org/mm-ms/apinew/track.php?do=click-detail>

GET 参数：

1. "id"：群发任务 send_id
 2. "ident"：任务批次号
 3. "link_id"：链接 id，指定任务指定链接点击统计详情
- 参数 id, ident 两者必须有一个，如果两个都存在，取 ident

输出格式：

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
  <click>
    <click_id>点击 id</click_id>
    <email>邮箱地址</email>
    <link>链接地址</link>
    <click_unique>唯一点击数</click_unique>
    <click_total>总点击数</click_total>
    <click_first>首次点击时间</click_first>
    <click_last>最后点击时间</click_last>
  </click>
  ...
</data>
```

3.4 导出指定群发任务跟踪邮箱信息

接口功能：导出指定群发任务跟踪统计邮箱信息

接口地址：

<http://www.bestedm.org/mm-ms/apinew/trackexport.php>

GET 参数：

1. track_id (必须)：群发任务跟踪 id

2. is_click: 0 或 1, 0 为导出所有邮箱, 1 为只导出有点击的邮箱
3. link_id: 链接地址 id, 导出点击过该链接的邮箱
4. email_id: 邮箱 id, 导出指定邮箱

U-Mail 邮件群发

3.5 指定批次的邮件各链接的点击统计

接口功能：指定批次的邮件各链接的点击统计

接口地址：

<http://www.bestedm.org/mm-ms/apinew/track.php?do=link-stat>

GET 参数：

1. "id"：群发任务 send_id
 2. "ident"：任务批次号
 3. "link_id"：链接 id，指定任务，指定链接点击统计
- 参数 id, ident 两者必须有一个，如果两个都存在，取 ident

输出格式：

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
  <link>
    <link_id>链接 id</link_id>
    <link>链接地址</link>
    <click_unique>唯一点击数</click_unique>
    <click_total>总点击数</click_total>
    <click_first>首次点击数</click_first>
    <click_last>最后点击数</click_last>
  </link>
  ...
</data>
```

四、联系人分类相关接口

4.1 取得联系人分类列表

接口功能：取得联系人分类列表

接口地址：

```
http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=maillist-list
```

输出格式：

```
<?xml version="1.0" encoding="UTF-8"?>
<maillist>
  <item>
    <id>联系人分类 ID</id>
    <subject>联系人分类名称</subject>
    <description>分类说明</description>
    <count>地址数量统计</count>
    <status>联系人分类状态</status>
  </item>
  ...
</maillist>
```

说明：

1. "status"：当前联系人分类的状态，"enabled"为正常使用，"disabled"为禁止使用。

4.2 取得联系人分类详情

接口功能：取得联系人分类详情

接口地址：

```
http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=maillist-detail
```

GET 参数(必要参数)：

1. "id"：联系人分类 ID

输出格式：

```
<?xml version="1.0" encoding="UTF-8"?>
<maillist>
```

```

<id>联系人分类 ID</id>
<subject>联系人分类名称</subject>
<description>联系人分类说明</description>
<count>地址数量统计</count>
<status>联系人分类状态</status>
</aillist>

```

说明：“status”字段含义同上。

4.3 添加联系人分类

接口功能：添加联系人分类

接口地址：

```

http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=maillist-add

```

POST 参数：

1. “subject”：联系人分类名称 **(必要参数)**
2. “status”：联系人分类状态，“enabled”为正常使用，“disabled”为禁止使用 **(必要参数)**
3. “description”：联系人分类说明

输出格式：

```

<?xml version="1.0" encoding="UTF-8"?>
<result>
  <status>执行结果标识符</status>
  <data>78</data>
</result>

```

执行结果标识符说明：

1. “add_maillist_error”：添加联系人分类时发生错误
2. “add_describe_error”：添加联系人分类描述信息时发生错误
3. “success”：添加联系人分类成功
4. “data”：添加联系人分类成功时 data 直接返回分类 id，失败时则返回失败描述

4.4 修改联系人分类

接口功能：修改联系人分类

接口地址:

<http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=maillist-edit>

GET 参数(必要参数):

1. "id": 联系人分类 ID

POST 参数:

1. "subject": 联系人分类主题
2. "status": 联系人分类状态, "enabled"为正常使用, "disabled"为禁止使用
3. "description": 联系人分类说明

输出格式:

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <status>执行结果标识符</status>
  <data>执行结果说明</data>
</result>
```

执行结果标识符说明:

1. "update_maillist_error": 修改联系人分类时发生错误
2. "update_describe_error": 修改联系人分类描述信息时发生错误
3. "success": 修改联系人分类成功

4.5 删除联系人分类

接口功能: 删除联系人分类

接口地址:

<http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=maillist-del>

GET 参数(必要参数):

1. "id": 联系人分类 ID

输出格式:

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <status>执行结果标识符</status>
  <data>执行结果说明</data>
</result>
```


执行结果标识符说明:

1. "delete_address_error": 删除联系人分类出错
2. "delete_describe_error": 删除联系人分类描述信息时发生错误
3. "delete_maillist_error": 删除联系人分类出错
4. "success": 删除联系人分类成功

4.6 导出联系人分类地址

接口功能: 导入联系人分类地址

接口地址:

<http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=subscription-export>

GET 参数(必要参数):

1. "id": 联系人分类 ID

输出格式: 直接输出文本文件, 每行格式为“邮件地址\t 用户名称” 邮件地址。

4.7 导出退订地址

接口功能: 导出退订地址

接口地址:

<http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=unsubscribe-export>

GET 参数(必要参数):

1. "id": 联系人分类 ID

输出格式: 直接输出文本文件, 每行格式为“退订时间\t 邮件地址” 邮件地址。

4.8 分类地址列表

接口功能: 获取联系人分类下的地址

接口地址:

<http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=ml-addr-list>

GET 参数说明: (红色必须)

1. list_id: 分类 id 必须以 get 方式传递, 如存在, 则查找该分类下的地址。否则查找全部。为 0 时查找未分类地址。
2. keyword: 在 address, fullname 两项中模糊查找
3. page: 分页数, 如果存在该参数, 则为分页查找, 否则为查找全部
4. limit: 分页查找时每页显示的数据记录数目, 默认为 50

输出格式:

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
  <total_num>地址总数目</total_num >
  <addr>
    <id>地址 id</id>
    <list_id>邮件订阅 id</list_id>
    <address>![CDATA[邮箱地址]]</address>
    <brithday>1986-08-15</brithday>
    <phone>13528775736</phone>
    <area>![CDATA[深圳市福田区科技园中路 3 号]]</area>
    <fullname>![CDATA[名称]]</fullname >
    <time>添加时间</time>
    <var1>变量 1</var1>
    <var2>变量 2</var2>
    <var3>变量 3</var3>
    <var4>变量 4</var4>
    <var5>变量 5</var5>
    <var6>变量 6</var6>
    <var7>变量 7</var7>
    <var8>变量 8</var8>
    <var9>变量 9</var9>
    <var10>变量 10</var10>
  </addr>
  <addr>
    <id>地址 id</id>
    <list_id>邮件订阅 id</list_id>
    <address>![CDATA[邮箱地址]]</address>
    <brithday>1986-08-15</brithday>
    <phone>13528775736</phone>
    <area>![CDATA[深圳市福田区科技园中路 3 号]]</area>
    <fullname>![CDATA[名称]]</fullname >
    <time>添加时间</time>
    <var1>变量 1</var1>
    <var2>变量 2</var2>
    <var3>变量 3</var3>
    <var4>变量 4</var4>
```

```

<var5>变量 5</var5>
<var6>变量 6</var6>
<var7>变量 7</var7>
<var8>变量 8</var8>
<var9>变量 9</var9>
<var10>变量 10</var10>
</addr>
...
</data>

```

4.9 新增分类地址

4.9.1、添加一个

接口功能：增加联系人分类下的地址

接口地址：

<http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=ml-addr-add>

GET 参数说明：（红色必须）

1. list_id: 分类 id
2. address: 邮箱地址
3. fullname: 用户名
4. var1, var2. ... Var10: 变量 1~10
5. return_id: 成功返回信息包含新添加的地址 id, 1 为包含, 0 为不包含。默认为 0
6. sex 格式: M (M 男 F 女)
7. birthday 格式: 0000-00-00
8. phone 格式: 13528775748
9. area 格式: 北京市丰台

说明：除 username, password, do 三个参数外, 其他的参数还可以通过 post 发送

4.9.2、批量添加

接口功能：增加联系人分类下的地址

接口地址：

<http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=ml-addr-add>

GET 参数说明：（红色必须）

1. list_id: 分类 id
2. addr_type: 地址信息数据类型, 默认为 'string'
3. separate: 每个地址信息之间的分隔符, 默认为 "\n" 换行符, 可以自定义任意字符。特别注意只在 addr_type 为 string 时生效

POST 参数说明：（红色必须）**1. ml_addr：地址信息**

说明：ml_addr 为地址信息，格式为：

文件格式说明

user1@domain.com	姓名	性别	生日	手机	地区	变量1	变量2					
user2@domain.com	姓名	性别	生日	手机	地区	变量1	变量2	变量3	变量4	变量5	...	
user3@domain.com	姓名	性别	生日	手机	地区	变量3						

说明：

邮箱地址;名称;性别;生日;手机;地区;变量 1; 变量 2; 变量 3; 变量 4; 变量 5; 变量 6; 变量 7; 变量 8;变量 9;变量 10（换行）

格式：

Coming1@bestdem.org;coming1;M;1986-08-10;13528775968;北京;var1;var2;....var10;

test@bestdem.org; test;F;1986-08-10;13528775968;area;var1;var2;....var10;

一行一个地址信息；

每行各个参数之间用“;”隔开。适合批量添加

4.9.3、从文件导入

接口功能：从文件中导入到联系人分类

接口地址：

<http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=ml-addr-add-file>

GET 参数说明：（红色必须）

1. list_id：分类 id，也可通过 post 方式传送

POST 参数说明：（红色必须）

1. file：文件信息

示例：建立一个 html 表单，上传文件

```
<form
method="post"
action="http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=ml-addr-add-file&l
ist_id=418804"
enctype="multipart/form-data">
<input type="file" name="file" />
<input type="submit" value="submit" />
</form>
```

说明：系统定时检测导入地址任务，自动过滤无效地址和重复地址。

从文件导入 Java 调用代码示例：

```
package cn.edu.ustc.file;
```

```

import java.io.DataOutputStream;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.UUID;
public class UploadFileToPhp {
    private static final int TIME_OUT = 10 * 1000000; // 超时时间
    private static final String CHARSET = "utf-8"; // 设置编码
    public static boolean uploadFile(File file) {
        String BOUNDARY = UUID.randomUUID().toString(); // 边界标识 随机生成
        String PREFIX = "--", LINE_END = "\r\n";
        String CONTENT_TYPE = "multipart/form-data"; // 内容类型
        String RequestURL =
"http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=ml-addr-add-file&list_id=1";
        System.out.println(RequestURL);
        try {
            URL url = new URL(RequestURL);
            HttpURLConnection conn = (HttpURLConnection) url.openConnection();
            conn.setReadTimeout(TIME_OUT);
            conn.setConnectTimeout(TIME_OUT);
            conn.setDoInput(true); // 允许输入流
            conn.setDoOutput(true); // 允许输出流
            conn.setUseCaches(false); //不允许使用缓存
            conn.setRequestMethod("POST"); // 请求方式
            conn.setRequestProperty("Charset", CHARSET); // 设置编码
            conn.setRequestProperty("connection", "keep-alive");
            conn.setRequestProperty("Content-Type", CONTENT_TYPE +
";boundary="+BOUNDARY);
            if (file != null) {
                /**
                 * 当文件不为空，把文件包装并且上传
                 */
                OutputStream outputStream = conn.getOutputStream();

                DataOutputStream dos = new DataOutputStream(outputStream);
                StringBuffer sb = new StringBuffer();
                sb.append(PREFIX);

```

```

sb.append(BOUNDARY);
sb.append(LINE_END);
/**
 * 这里重点注意: name 里面的值为服务器端需要 key (php 里面需要是
file), 只有这个 key 才可以得到对应的文件
 * filename 是文件的名称, 包含后缀名的 比如: abc.png, 可以看到下面
在开始与结束都写入了一些分隔符等标示.
 */
sb.append("Content-Disposition: form-data; name=\"file\";
filename=\"\"
        + file.getName() + "\"" + LINE_END);
sb.append("Content-Type: application/octet-stream; charset="
        + CHARSET + LINE_END);
sb.append(LINE_END);
dos.write(sb.toString().getBytes());
InputStream is = new FileInputStream(file);
byte[] bytes = new byte[1024];
int len = 0;
while ((len = is.read(bytes)) != -1) {
    dos.write(bytes, 0, len);
}
is.close();
dos.write(LINE_END.getBytes());
byte[] end_data = (PREFIX + BOUNDARY + PREFIX + LINE_END)
        .getBytes();
dos.write(end_data);
dos.flush();
/**
 * 获取响应码 200=成功 当响应成功, 获取响应的流
 */
int res = conn.getResponseCode();

System.out.println(res);
if (res == 200) {
    /**
     * 获取内容在此写代码
     */
    return true;
}
} catch (MalformedURLException e) {
    e.printStackTrace();
} catch (IOException e) {

```

```

        e.printStackTrace();
    }
    return false;
}

public static void main(String[] args) {
    File file = new File("/dir/address.txt");
    boolean res = uploadFile(file);
    System.out.println(res);
}
}

```

address.txt 格式说明:

邮箱地址;名称;性别;生日;手机;地区;变量 1; 变量 2; 变量 3; 变量 4; 变量 5; 变量 6;
变量 7; 变量 8;变量 9;变量 10 (换行)

格式:

Coming1@bestdem.org;coming1;M;1986-08-10;13528775968;北京;var1;var2;... var10;
test@bestdem.org; test;F;1986-08-10;13528775968;area;var1;var2;... var10;

一行一个地址信息;

每行各个参数之间用“;”隔开。适合批量添加

输出格式:

```

<?xml version="1.0" encoding="UTF-8"?>
<result>
<status>success</status>
<data><![CDATA[10]]></data>;//10 为查询是否解析成功的标识 id, 需要记录
</result>

```

Eg:

```

<?xml version="1.0" encoding="UTF-8"?>
<result>
  <status>success</status>
  <data>
    <![CDATA[210586]]>
  </data>
</result>

```

4.9.4. 从文件导入查询地址是否解析成功

接口功能: 文件导入到联系人分类后查询地址是否解析成功

接口地址:

<http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=ml-addr-add-file-query&id=1>

GET 参数说明: (红色必须)

1. id: 文件导入后返回的 id

输出格式:

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
<status>success</status>
<data><![CDATA[解析完成]]></data>;
</result>
```

Status 返回值说明:

success/not_exist/error/no_complete 解析完成/不存在 id/参数错误/正在解析

4.10 修改分类地址

接口功能: 修改联系人分类下的地址

接口地址:

<http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=ml-addr-edit>

GET 参数说明: (红色必须)

1. id: 地址 id
2. address: 邮箱地址
3. fullname: 用户名
4. var1, var2. ... Var10: 变量 1~10
6. sex 格式: M (M 男 F 女)
7. birthday 格式: 1988-04-08
8. phone 格式: 13528775748
9. area 格式: 北京市丰台

说明: 除 username, password, do 三个参数外, 其他的参数还可以通过 post 发送

4.11 删除分类地址

接口功能: 删除联系人分类下的地址

接口地址:

<http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=ml-addr-del>

GET 参数说明: (红色必须)

1. id: 地址 id

4.12 清空联系人分类

接口功能: 清空联系人分类下的地址

接口地址:

<http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=ml-addr-empty>

GET 参数说明：（红色必须）

1. **list_id**: 联系人分类 id

4.13 邮件订阅

接口功能：添加地址到订阅列表

接口地址：

```
http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=ml-sub&list_id=邮件订阅id&address=xxx@xxx.com&name=邮箱用户名称
```

GET 参数（必要参数）：

- 1, "list_id": 邮件订阅 ID
- 2, "address": 订阅邮箱
- 3, "name": 邮箱用户名称
- 4, "maillist": 批量订阅邮箱地址

说明：

name 参数如果没有提供，默认取邮箱名。如 test@bestedm.com，名称就取 test
address, name 两参数，添加一个地址时使用

maillist 批量添加时使用，可通过 get 或者 post 传递，格式为：

格式 1: 邮箱 1, 名称 1; 邮箱 2, 名称 2;

test1@bestedm.com, test1; test2@bestedm.com, test2; ...

格式 2: 一行为一个地址信息，邮箱与名称之间用 tab 键或者 “,” 分隔。

邮箱 1 名称

邮箱 2 名称

...

test1@bestedm.com test1

test2@bestedm.com test2

4.14 邮件退订

接口功能：删除邮件订阅中的地址

接口地址：

```
http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=ml-unsub&list_id=邮件订阅id&address=xxx@xxx.com
```

GET 参数(必要参数):

- 1, "list_id": 邮件订阅 ID
- 2, "address": 退订邮箱

说明:

address 可通过 get 或者 post 传递, 格式为:

格式 1: 邮箱 1, 邮箱 2,

test1@bestedm.com, test2@bestedm.com, ...

格式 2: 一行为一个地址信息

邮箱 1

邮箱 2

...

test1@bestedm.com

test2@bestedm.com

五、邮件模板相关接口

5.1 创建群发任务

接口功能: 创建群发任务

接口地址:

http://www.bestedm.org/mm-ms/apinew/task.php?do=add-task&tpl_id=16&send_domain=comingchina.com&send_account=linbihuan@comingchina.com&send_fullname=linbihuan&maillist_id=5&time=2012-07-31 12:00

GET 参数说明: (红色必须)

1. tpl_id: 模板 id
2. send_domain: 发件人域名
3. send_account: 发件人
4. send_fullname: 发件人名称
5. maillist_id: 联系人分类 id。

6. send_qty: 发送数量, 默认为 0, 全部地址
7. status: 发送状态, 默认为 1, 等待发送; -1 为暂不发送; 2 为立即发送
8. time: 发送时间, 必须。格式 2012-07-04%2016:08, 即 2012-07-04 16:08 经过 urlencode 转义后的值, 否则空格无法识别。
9. subs_link: 是否在邮件中加入退订订阅链接。默认为 1, 加入
10. track_status: 是否开启邮件跟踪。默认为 0, 不开启
11. track_domain: 自定义跟踪统计链接域名
12. send_replyto: 指定发送回复邮箱地址。例如: repaly@126.com

备注:

1. 除参数 username, password, do 之外, 其它参数均可通过 get 或者 post 方式发送
2. 如果发件人是某个域名下的所有发件人, 参数 send_domain 为域名, send_account 为 all。如: &send_domain=comingchina.com&send_account=all
3. 如果发件人是所有域名下的所有发件人, 参数 send_domain, send_account 两个都为 all。
如: &send_domain=all&send_account=all
4. 如果发件人只是某个域名下的一个发件人。Send_domain 可以不需要。如: send_account=test@comingchina.com

输出格式:**成功:**

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <status>success</status>
  <id>新创建的任务 id</id>
  <data><![CDATA[添加群发任务成功!]]></data>
</result>
```

错误:

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
  <status>错误状态</status>
  <data><![CDATA[错误描述]]></data>
</result>
```

5.2 上传邮件模板

接口功能：上传邮件模板（可上传附件）

接口地址：

<http://www.bestedm.org/mm-ms/apinew/template.php?do=add-tpl>

GET 参数说明：

1. do: 操作类型，必须

POST 参数说明：

1. name: 邮件名称，必须
2. subject: 邮件主题，必须，多个模板主题请以|隔开；如：主题一|主题二
3. content: 邮件内容，必须
4. text_content: 纯文本内容,html 邮件内容无法显示时，显示纯文本内容
5. encoding: 邮件发送编码，base64 或者 quoted-printable
6. file: 附件参数。Input 标签中 name 属性的值为 'file'，如果是上传多个文件，那么 name 属性值为 'file[]'。

说明：成功返回新建模板的 id 值

5.3 模板列表

接口功能：获取用户模板列表

接口地址：

http://www.bestedm.org/mm-ms/apinew/template.php?do=list-tpl&tpl_id=67

GET 参数说明：(红色必须)

1. 'tpl_id': 模板 id，如果存在，则只获取该 id 值的模板信息。否则获取全部模板信息

输出格式：

```
<?xml version="1.0" encoding="UTF-8"?>
<template_list>
  <template>
    <id>53</id>
    <user_id>模板 id</user_id>
    <name>
      <![CDATA[模板名称]]>
    </name>
```

```

<subject>
  <id>模板主题一 id</id>
  <name><![CDATA[模板主题 1]]></name>
</subject>
<subject>
  <id>模板主题二 id</id>
  <name><![CDATA[模板主题 2]]></name>
</subject>

<content>
<![CDATA[模板内容]]>
</content>
<content_type>模板类型</content_type>

<attachments> ----- 20121206 新增附件信息
  <attachment>
    <atta_id>附件 id</atta_id>
    <atta_name>附件文件名</atta_name>
  </attachment>
</attachments>

<created>创建时间</created>
<updated>更新时间</updated>
</template>

<template>
  ...
</template>

...
</template_list>

```

5.4 删除模板

接口功能：删除指定 id 模板

接口地址：

http://www.bestedm.org/mm-ms/apinew/template.php?do=del-tpl&tpl_id=67

GET 参数说明：

1. ‘**tpl_id**’: 模板 id

5.5 修改模板

接口功能: 修改指定模板

接口地址:

http://www.bestedm.org/mm-ms/apinew/template.php?do=edit-tpl&tpl_id=11&name=linbihuan&subject=sub&content=con

GET 参数说明:

1. ‘**tpl_id**’: 模板 id, 也可通过 post 上传, 必须
2. ‘**name**’: 模板名称, 也可通过 post 上传
3. ‘**subject**’: 模板主题, 也可通过 post 上传, 多个模板主题请以|隔开; 如主题一|主题二
4. ‘**content**’: 模板内容, 也可通过 post 上传
5. **text_content**: 纯文本内容, html 邮件内容无法显示时, 显示纯文本内容
6. **encoding**: 邮件发送编码, base64 或者 quoted-printable

备注:

1. **tpl_id** 参数必须, **name**、**subject**、**content** 三个参数必须至少有一个。
2. 四个参数均可以通过 post 或者 get 方式上传

5.6 上传模板附件

接口功能: 上传模板附件

接口地址:

http://www.bestedm.org/mm-ms/apinew/template.php?do=tpl-attachment-add&tpl_id=67

GET 参数说明:

1. ‘**tpl_id**’: 模板 id, 也可通过 post 上传, 必须

POST 参数说明:

1. ‘**file**’: 附件参数名。Input 标签中 name 属性的值为 ‘file’, 如果是多附件, 该属性值为 ‘file[]’

5.7 删除模板附件

接口功能：删除模板附件

接口地址：

http://www.bestedm.org/mm-ms/apinew/template.php?do=tpl-attachment-del&tpl_id=67&attachment_id=1

GET 参数说明：

1. 'tpl_id': 模板 id, 也可通过 post 上传, 必须
2. 'attachment_id': 附件 id, 默认为 'all', 删除模板中所有附件

六、群发任务相关接口

6.1 群发任务列表

接口功能：获取用户群发任务列表

接口地址：

<http://www.bestedm.org/mm-ms/apinew/task.php?do=task-list>

GET 参数说明：（红色必须）

1. **‘do’**：操作类型
2. **‘id’**：任务 id，如存在该参数，则只返回该任务的信息，否则返回所有任务信息
3. **‘page’**：页数，从 1 开始
4. **‘limit’**：每页显示的任务数，如果没有指定，默认为 30
5. **‘date’**：发送日期，如 ‘2013-01-05’
6. **‘send_status’**：任务发送状态。
默认为 ‘all’，表示所有群发任务。
‘1’ 为等待发送
‘2’ 为正在发送
‘3’ 为发送完成
‘-1’ 为暂不发送
‘-2’ 为暂停发送
‘-3’ 为取消发送

备注：page, limit 两个参数如果都没有指定，默认获取全部任务列表。

输出格式：

```
<?xml version="1.0" encoding="UTF-8"?>
<task_list>
  <task>
    <id>任务 id</id>
    <sn>任务号</sn>
    <acct_type>发件人类型 (all, domain, address) </acct_type>
    <acct_domain>发件人域名，发件人类型为 domain 时存在</acct_domain>
    <acct_address>发件人，发件人类型为 address 时存在</acct_address>
    <sender>发件人</sender>
    <replyto>回复地址</replyto>
    <sender_name>发件人姓名</sender_name>
```



```

<templates>
  <template>
    <tpl_id>模板 id</tpl_id>
    <tpl_name>模板名称</tpl_name>
  </template>
  <template>
    <tpl_id>模板 id</tpl_id>
    <tpl_name>模板名称</tpl_name>
  </template>
</templates>
<addr_type>地址类型</addr_type>
<list_id>联系人分类 ID</list_id>
<list_name>联系人分类名称</list_name>
<send_qty>预定发送数量(用户指定)</send_qty>
<send_qty_remak>预定发送数量(实际发送)</send_qty_remak>
<send_time>预定发送时间</send_time>
<time_start>实际发送开始时间</time_start>
<time_end>实际发送结束时间</time_end>
<send_count>实际发送数量</send_count>
<error_count>发送失败数量</error_count>
<send_status>发送状态</send_status>
<track_status>跟踪统计状态</track_status>
</task>

<task>...</task>
...
</task_list>

```

6.2 创建群发任务

接口功能：创建群发任务

接口地址：

http://www.bestedm.org/mm-ms/apinew/task.php?do=add-task&tpl_id=16&send_domain=comingchina.com&send_account=linbihuan@comingchina.com&send_fullname=linbihuan&maillist_id=5&time=2012-07-31%2012:00

GET 参数说明: (红色必须)

1. tpl_id: 模板 id 单个模板请传入 2, 该任务有多个模板则 1, 2, 3, 78 逗号隔开格式
2. send_domain: 发件人域名
3. send_account: 发件人
4. send_fullname: 发件人名称
5. maillist_id: 联系人分类 id
6. send_qty: 发送数量, 默认为 0, 全部地址
7. status: 发送状态, 默认为 1, 等待发送; -1 为暂不发送; 2 为立即发送
8. time: 发送时间, 必须。格式 2012-7-4%2016:08
9. subs_link: 是否在邮件中加入退订订阅链接。默认为 1, 加入
10. track_status: 是否开启邮件跟踪。默认为 0, 不开启
11. track_domain: 自定义跟踪统计链接域名

备注:

1. 除参数 username, password, do 之外, 其它参数均可通过 get 或者 post 方式发送
2. 如果发件人是某个域名下的所有发件人, 参数 send_domain 为域名, send_account 为 all。如: &send_domain=comingchina.com&send_account=all
3. 如果发件人是所有域名下的所有发件人, 参数 send_domain, send_account 两个都为 all。
如: &send_domain=all&send_account=all
4. 如果发件人只是某个域名下的一个发件人。Send_domain 可以不需要。如:
&send_account=test@comingchina.com

6.3 修改群发任务

接口功能: 修改群发任务

接口地址:

http://www.bestedm.org/mm-ms/apinew/task.php?do=edit-task&id=16&send_domain=comingchina.com&send_account=linbihuan@comingchina.com&send_fullname=linbihuan&maillist_id=5&time=2012-07-31%2012:00

GET 参数说明: (红色必须)

3. id: 任务 id
4. tpl_id: 模板 id 单个模板请传入 2, 该任务有多个模板则 1, 2, 3, 78 逗号隔开格式
5. send_domain: 发件人域名
6. send_account: 发件人
7. send_fullname: 发件人名称
8. maillist_id: 联系人分类 id
10. send_qty: 发送数量
11. send_status: 发送状态(-1 或 1)
12. send_time: 发送时间, 格式 2012-7-4%2016:08

- 13. subs_link: 是否在邮件中加入退订订阅链接 (0,1)。
- 14. track_status: 是否开启邮件跟踪 (0,1,2)。
- 15. track_domain: 自定义跟踪域名
- 16. send_replyto: 指定回复地址

备注:

1. 除参数 username, password, do 之外, 其它参数均可通过 get 或者 post 方式发送
2. 如果发件人是某个域名下的所有发件人, 参数 send_domain 为域名, send_account 为 all。如: &send_domain=comingchina.com&send_account=all
3. 如果发件人是所有域名下的所有发件人, 参数 send_domain, send_account 两个都为 all。
如: &send_domain=all&send_account=all
4. 如果发件人只是某个域名下的一个发件人。Send_domain 可以不需要。如: &send_account=test@comingchina.com

6.4 删除群发任务

接口功能: 删除群发任务

接口地址:

<http://www.bestedm.org/mm-ms/apinew/task.php?do=del-task&id=19124>

GET 参数说明: (红色必须)

1. id: 任务 id

七、域名及域名下的邮箱账号

7.1 域名列表

接口功能: 获取域名列表

接口地址:

<http://www.bestedm.org/mm-ms/apinew/bs.php?do=domain-list>

GET 参数说明: (红色必须)

1. customer_id: 客户 id
2. user_type: 用户类型。manager 或 customer, 默认为 customer。如果是 manager, username 跟 password 两个参数必须是管理员的用户名跟密码, 需要加 customer_id、user_type 两参

数

输出格式:

```
<?xml version="1.0" encoding="UTF-8"?>
<domain_list>
  <data>
    <customer_id>客户 id (0 为系统域名) </customer_id>
    <domain>域名</domain>
  </data>
  <data>...</data>
  ...
</domain_list>
```

7.2 添加域名(管理员)

接口功能: 添加域名

接口地址:

http://www.bestedm.org/mm-ms/apinew/bs.php?do=domain-add&customer_id=*&domain=***

GET 参数说明: (红色必须)

1. customer_id: 客户 id
2. domain: 需要添加的域名, 如 test.com

7.3 删除域名(管理员)

接口功能: 删除域名

接口地址:

http://www.bestedm.org/mm-ms/apinew/bs.php?do=domain-del&customer_id=4&domain=***

GET 参数说明: (红色必须)

1. customer_id: 客户 id
2. domain: 需要添加的域名, 如 test.com

7.4 指定域名下的账号列表

接口功能: 获取指定域名下的账号列表

接口地址:

http://www.bestedm.org/mm-ms/apinew/bs.php?do=mailbox-list&domain=***

GET 参数说明: (红色必须)

1. customer_id: 客户 id
2. domain: 需要添加的域名, 如 test.com
3. user_type: 用户类型。manager 或 customer, 默认为 customer。如果是 manager, username 跟 password 两个参数必须是管理员的用户名跟密码, 需要加 customer_id、user_type 两参数
4. page: 分页数, 如果存在该参数, 则为分页查找, 否则为查找全部
5. limit: 分页查找时每页显示的数据记录数目, 默认为 30

输出格式:

```
<?xml version="1.0" encoding="UTF-8"?>
<mailbox_list>
  <data>
    <customer_id>客户 id</customer_id>
    <domain>域名</domain>
    <name>账号</name>
    <mailbox>完整账号</mailbox>
  </data>
  <data>...</data>
  ...
</mailbox_list>
```

7.5 添加账号到指定域名

接口功能: 添加账号到指定域名

接口地址:

http://www.bestedm.org/mm-ms/apinew/bs.php?do=mailbox-add&mailbox=***&pwd=***

GET 参数说明: (红色必须)

1. **pwd:** 添加账号的密码
2. **mailbox:** [需要添加的账号, 如 test@test.com](#)
3. **customer_id:** 客户 id
4. **user_type:** 用户类型。manager 或 customer, 默认为 customer。如果是 manager, username 跟 password 两个参数必须是管理员的用户名跟密码, 需要加 customer_id、user_type 两参数

7.6 删除指定域名下的账号

接口功能: 删除指定域名下的账号

接口地址:

http://www.bestedm.org/mm-ms/apinew/bs.php?do=mailbox-del&mailbox=***

GET 参数说明: (红色必须)

1. **mailbox:** [需要删除的账号, 如 test@test.com](#)
2. **customer_id:** 客户 id
3. **user_type:** 用户类型。manager 或 customer, 默认为 customer。如果是 manager, username 跟 password 两个参数必须是管理员的用户名跟密码, 需要加 customer_id、user_type 两参数

7.7 修改指定域名的账号

接口功能: 修改指定域名下的账号

接口地址:

http://www.bestedm.org/mm-ms/apinew/bs.php?do=mailbox-edit&mailbox=***&pwd=***

GET 参数说明: (红色必须)

1. **pwd**: 修改账号的密码
2. **mailbox**: 需要修改的账号, 如 test@test.com
3. **customer_id**: 客户 id
4. **user_type**: 用户类型。manager 或 customer, 默认为 customer。如果是 manager, username 跟 password 两个参数必须是管理员的用户名跟密码, 需要加 customer_id、user_type 两参数

八、其他

8.1 客户自定义群发登录页面说明

说明:

form 表单的 action 属性为

<http://www.bestedm.org/sendstat/?module=base&action=login>

如果用户名或者密码错误返回页面也是自定义的, 在表单中另外 post 提交一个参数 errurl, 该参数值为错误返回的 url, 必须要有 http://

示例:

```
<form
action="http://www.bestedm.org/sendstat/?module=base&action=login"
method="post">
  <input type="hidden" name="errurl"
value="http://www.magvision.com/logging.html" />
  <input type="text" name="username" />
  <input type="password" name="password" />
```



```
request.add_header("Authorization", "Basic %s" % base64string)
result = urllib2.urlopen(request)
```

9.2 C#用户认证示例代码:

```
string username="username";
string password="password";
//注意这里的格式哦, 为 "username:password"
string usernamePassword = username + ":" + password;
CredentialCache mycache = new CredentialCache();
mycache.Add(new Uri(url), "Basic", new NetworkCredential(username, password));
myReq.Credentials = mycache;
myReq.Headers.Add("Authorization", "Basic " + Convert.ToBase64String(new
ASCIIEncoding().GetBytes(usernamePassword)));
WebResponse wr = myReq.GetResponse();
Stream receiveStream = wr.GetResponseStream();
StreamReader reader = new StreamReader(receiveStream, Encoding.UTF8);
string content = reader.ReadToEnd();
```

9.3 java 用户认证示例代码:

```
HttpGet request = new HttpGet(URL_SECURED_BY_BASIC_AUTHENTICATION);
String auth = DEFAULT_USER + ":" + DEFAULT_PASS;
byte[] encodedAuth =
Base64.encodeBase64(auth.getBytes(Charset.forName("US-ASCII")));
String authHeader = "Basic " + new String(encodedAuth);
request.setHeader(HttpHeaders.AUTHORIZATION, authHeader);

HttpClient client = HttpClientBuilder.create().build();
HttpResponse response = client.execute(request);

int statusCode = response.getStatusLine().getStatusCode();
assertThat(statusCode, equalTo(HttpStatus.SC_OK));
```

特别注意: 在调用 api 接口时如果需要传入中文请务必指定 http 头为 utf-8 格式, 并把所传中文转为 utf-8 格式, 如下图代码所示:

```

//验证用户
Map<String, String> requestProperties = new HashMap<String, String>();
requestProperties.put("Authorization", "Basic "+validUser());// 你的用户名密码编码
String res = "";// 响应
String boundary = "-----" + new Date().getTime(); // boundary 就是 request 头和上传文件内容的分隔符

URLConnection conn = null;
try {
    if (StringUtils.isBlank(contentType)) {
        contentType = "text/plain";
    }

    URL url = new URL(urlStr);
    conn = (URLConnection) url.openConnection();
    conn.setConnectTimeout(30000);
    conn.setReadTimeout(30000);
    conn.setDoOutput(true);
    conn.setDoInput(true);
    conn.setUseCaches(false);
    conn.setRequestMethod("POST");
    conn.setRequestProperty("Connection", "Keep-Alive");
    conn.setRequestProperty("User-Agent", "Mozilla/5.0 (Windows; U; Windows NT 6.1; zh-CN; rv:1.9.2.6)");
    conn.setRequestProperty("Content-Type", "multipart/form-data; boundary="+ boundary);
    conn.setRequestProperty("charset", "utf-8");
    if (requestProperties != null && requestProperties.size() > 0) {
        for (String key : requestProperties.keySet()) {
            conn.setRequestProperty(key, requestProperties.get(key));
        }
    }
    OutputStream out = new DataOutputStream(conn.getOutputStream());
    // 参数
    if (params != null && params.size() > 0) {
        StringBuffer strBuf = new StringBuffer();
        for (Map.Entry<String, String> entry : params.entrySet()) {
            String key = entry.getKey();
            String value = entry.getValue();
            if (StringUtils.isBlank(key) || StringUtils.isBlank(value)) {
                continue;
            }
            strBuf.append(key).append("=").append(value).append("&");
        }
        out.write(strBuf.toString());
    }
}

```

9.4 php 用户认证示例代码

方法一：//---使用 `fsocket` 的方式验证用户名和密码

```

$url = 'api 地址' ;
$fp = fsockopen("$url", 80);
fputs($fp, "GET /downloads HTTP/1.0");
fputs($fp, "Host: $url");
fputs($fp, "Authorization: Basic " . base64_encode("user:pass") . "");

```

```
fpassthru($fp);
```

推荐方法二: `///---使用 CURL 扩展的方式验证密码并调用接口`

`///---使用 CURL 扩展的方式`

```
if(extension_loaded('curl')){
    $username = "1xx2"; //用户名
    $password = "xxxxxx"; //密码
    $url =
'http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=ml-addr-lis
t&list_id=461253&page=1&limit=1';
    // $params =array();
    // $vars = http_build_query($params);
    //post 方式
    // $opts = array(
    //     CURLOPT_TIMEOUT      => 5,
    //     CURLOPT_HTTPAUTH=>CURLAUTH_BASIC,
    //     CURLOPT_USERPWD=>"$username:$password",
    //     CURLOPT_RETURNTRANSFER => 1,
    //     CURLOPT_URL          => $url,
    //     CURLOPT_POST         => 1,
    //     CURLOPT_POSTFIELDS   => $vars,
    //     CURLOPT_USERAGENT    => $_SERVER['HTTP_USER_AGENT']
    // );

    //get 方式
    $opts = array(
        CURLOPT_TIMEOUT      => 5,
        CURLOPT_HTTPAUTH=>CURLAUTH_BASIC,
        CURLOPT_USERPWD=>"$username:$password",
        CURLOPT_RETURNTRANSFER => 1,
        CURLOPT_URL          => $url,
        CURLOPT_HEADER       =>0,
        CURLOPT_USERAGENT    => $_SERVER['HTTP_USER_AGENT']
    );

    /* 初始化并执行 curl 请求 */
    $ch = curl_init();
    curl_setopt_array($ch, $opts);
    $status_code = curl_getinfo($ch, CURLINFO_HTTP_CODE); //get
status code
```

```

    $data = curl_exec($ch);
    $error = curl_error($ch);
    curl_close($ch);
}

```

9.5 Ruby On Rails 示例代码:

```

import mx.rpc.events.FaultEvent;
import mx.rpc.events.ResultEvent;
import mx.utils.Base64Encoder;
import mx.rpc.http.HTTPService;
URLRequestDefaults.authenticate = false; //设默认为 false, 否则用户校验错误时会弹出验证框

private var result:XML;
private function initApp():void
{
    var base64enc:Base64Encoder = new Base64Encoder;
    base64enc.encode("user:password"); //用户名和密码需要 Base64 编码
    var user:String = base64enc.toString();

    var http:HTTPService = new HTTPService;
    http.addEventListener(ResultEvent.RESULT, resultHandler); //监听返回事件
    http.addEventListener(FaultEvent.FAULT, faultHandler); //监听失败事件
    http.resultFormat = "e4x"; //返回格式
    http.url = "api 地址";
    http.headers = {"Authorization":"Basic " + user};
    http.send();
}

private function resultHandler(e:ResultEvent):void
{
    result = XML(e.result);
    test.dataProvider = result.status; //绑定数据
}

private function faultHandler(e:ResultEvent):void
{
    //处理失败
}

```

9.6 JS 用户认证示例代码:

```

<script type="text/javascript">
    //需要 Base64 见: http://www.cnblogs.com/pingming/p/4165063.html
    function make_base_auth(user, password) {
        var tok = user + ':' + pass;
        var hash = Base64.encode(tok);
        return "Basic " + hash;
    }

    var auth = make_basic_auth('QLeelulu', 'mypassword');
    var url = 'api url address';

    // 原始 JavaScript
    xml = new XMLHttpRequest();
    xml.setRequestHeader('Authorization', auth);
    xml.open('GET', url);

    // ExtJS
    Ext.Ajax.request({
        url : url,
        method : 'GET',
        headers : { Authorization : auth }
    });

    // jQuery
    $.ajax({
        url : url,
        method : 'GET',
        beforeSend : function(req) {
            req.setRequestHeader('Authorization', auth);
        }
    });
</script>

```

9.7 接口调用 php 示例代码

```

if(extension_loaded('curl')){
    $url = '接口地址';
    $params = array(
        'do' => 'customer-info'
    );
    $vars = http_build_query($params);
    //post 方式
    $opts = array(
        CURLOPT_TIMEOUT      => 5,
        CURLOPT_RETURNTRANSFER => 1,
        CURLOPT_URL          => $url,
        CURLOPT_POST         => 1,
        CURLOPT_POSTFIELDS    => $vars,
        CURLOPT_USERAGENT    => $_SERVER['HTTP_USER_AGENT'],
    );

    //get 方式
    $opts = array(
        CURLOPT_TIMEOUT      => 5,
        CURLOPT_RETURNTRANSFER => 1,
        CURLOPT_URL          => $url,
        CURLOPT_HEADER       => 0,
        CURLOPT_USERAGENT    => $_SERVER['HTTP_USER_AGENT']
    );

    /* 初始化并执行 curl 请求 */
    $ch = curl_init();
    curl_setopt_array($ch, $opts);
    $data = curl_exec($ch);
    $error = curl_error($ch);
    curl_close($ch);
}

```

9.8 接口调用 java 示例代码

```

public static void main(String[] args) throws Exception {
    String urlStr =
"http://www.bestedm.org/mm-ms/apinew/mloperate.php?do=ml-addr-add-file";
}

```

```

// 参数
Map < String, String > params = new HashMap <> ();
params . put ("list_id", "418315");

// 文件
Map < String, Map < String, byte[] >> fileMap = new HashMap <>
();
fileMap . put ("file", new HashMap < String, byte[] > ());
FileInputStream fis = new FileInputStream ("address.txt");
byte[] buf = new byte[fis . available()];
fis . read (buf);
fileMap . get ("file") . put ("test.txt", buf);
fis . close();

// 请求验证用户账号密码
Map < String, String > requestProperties = new HashMap <> ();
requestProperties . put ("Authorization", "Basic
c215ZWlhWx Dyuudkdkdsss f4ODg4OA=="); //你的用户名密码编码

String contentType = "multipart/form-data";

String s = fileUpload (urlStr, params, fileMap, requestProperties,
contentType);
}
/**
 * 上传文件流
 *
 * @param url
 * @param params
 * @param fileMap: name: {filename:byte[]}
 * @param requestProperties
 */
public static String fileUpload (String urlStr, Map <String, String>
params, Map <String, Map <String, byte[] >> fileMap, Map <String, String>
requestProperties, String contentType) {
    String res = ""; // 响应
    String boundary = "-----" + new
Date ().get Time (); // boundary 就是 request 头和上传文件内容的分隔符
    HttpURLConnection conn = null;
    try {
        if (StringUtils.isBlank (contentType)) {

```

```

        contentType = "text/plain";
    }
    contentType = "multipart/form-data"; // TODO
    URL url = new URL(urlStr);
    conn = (URLConnection) url.openConnection();
    conn.setConnectTimeout(CON_TIMEOUT);
    conn.setReadTimeout(READ_TIMEOUT);
    conn.setDoOutput(true);
    conn.setDoInput(true);
    conn.setUseCaches(false);
    conn.setRequestMethod("POST");
    conn.setRequestProperty("Connection", "Keep-Alive");
    conn.setRequestProperty("User-Agent", "Mozilla/5.0 (Windows;
U; Windows NT 6.1; zh-CN; rv:1.9.2.6)");
    conn.setRequestProperty("Content-Type",
"multipart/form-data; boundary=" + boundary);

    if(requestProperties != null && requestProperties.size() >
0){
        for (String key : requestProperties.keySet()) {
            conn.setRequestProperty(key,
requestProperties.get(key));
        }
    }

    OutputStream out = new
DataOutputStream(conn.getOutputStream());

    // 参数
    if (params != null && params.size() > 0) {
        StringBuffer strBuf = new StringBuffer();
        for (Map.Entry<String, String> entry : params.entrySet())
        {
            String key = entry.getKey();
            String value = entry.getValue();
            if (StringUtil.isBlank(key) ||
StringUtil.isBlank(value)) {
                continue;
            }
            strBuf.append("\r\n").append("--").append(boundary).append("\r\n"
);
            strBuf.append("Content-Disposition: form-data;
name=\"" + key + "\"\r\n\r\n");
        }
    }

```



```

        strBuf.append(value);
    }
    out.write(strBuf.toString().getBytes());
}

// 上传文件
if (fileMap != null && fileMap.size() > 0) {
    for (Map.Entry<String, Map<String, byte[]>> entry :
fileMap.entrySet()) {
        Map<String, byte[]> file = entry.getValue();
        String name = entry.getKey(); // 上传表单的 name
        String filename = file.keySet().iterator().next(); // 文
文件名
        byte[] fileByte = file.get(filename); // 文件的字节
        if (StringUtils.isBlank(name) ||
StringUtils.isBlank(filename) || fileByte == null || fileByte.length
== 0) {
            continue;
        }
        StringBuffer strBuf = new StringBuffer();
        strBuf.append("\r\n").append("--").append(boundary).append("\r\n"
);
        strBuf.append("Content-Disposition: form-data;
name=\"" + name + "\"; filename=\"" + "address.txt" + "\"\r\n");
        // strBuf.append("Content-Type: " + contentType +
"\r\n\r\n"); // 默认 txt 文件格式
        strBuf.append("Content-Type: " + "multipart/form-data"
+ "\r\n\r\n");
        out.write(strBuf.toString().getBytes());

        ByteArrayInputStream bais = new
ByteArrayInputStream(fileByte);
        int bytes = 0;
        byte[] bufferOut = new byte[1024];
        while ((bytes = bais.read(bufferOut)) != -1) {
            out.write(bufferOut, 0, bytes);
        }
        bais.close();
    }
}

out.write("\r\n--" + boundary + "--\r\n").getBytes();
out.flush();

```

```
        out.close();

        // 读取返回数据
        StringBuffer strBuf = new StringBuffer();
        BufferedReader reader = new BufferedReader(new
InputStreamReader(conn.getInputStream()));
        String line = null;
        while ((line = reader.readLine()) != null) {
            strBuf.append(line).append("\n");
        }
        res = strBuf.toString();
        reader.close();
        reader = null;
    } catch (Exception e) {
        logger.error("上传文件失败, 请求路径【" + urlStr + "】, 失败原因:
" + e);
        e.printStackTrace();
    } finally {
        if (conn != null) {
            conn.disconnect();
            conn = null;
        }
    }
    return res;
}
```