

# 金笛邮件系统远程 java API 接口说明

## 一. 设置接口访问用户名和密码

配置位置：系统管理——》参数设置——》一般参数  
例子如下图：

API访问用户名	<input type="text" value="admin"/>
API访问密码	<input type="password" value="*****"/>



为了保证系统的安全性，每次调用接口都需要提供正确的用户名和密码。

## 二. API 调用环境类

类名：jdmail.api.remote.Env

说明：Env 类提供整个 API 与服务器的通讯和配置管理。

方法：

1. `public static Env getEnv(String url, String username, String password);`

说明：通过该方法可获取一个类实例。

参数说明：

- 1) url: webmail 服务器根 url
- 2) username: API 访问用户名
- 3) password: API 访问密码

返回值：

Env 对象：成功；Env 对象

Null：失败

## 三. 帐号操作类

类名：jdmail.api.remote.user.R\_UserAccount

说明：UserAccount 类提供所有用户管理相关功能，包括新增，删除，修改用户，和获取用户相关信息。

属性：

1. `public UserAccount m_useraccount`

m\_useraccount 提供用户帐号的详细信息。

方法：

1. `public static int createUserAccount(Env env, R_UserAccount ua);`

说明：用于建立新用户。

参数说明：

- 1) env : Env 对象
- 2) ua: R\_UserAccount 对象，R\_UserAccount 对象通过 m\_useraccount 属性传递 用户帐号详细信息。

返回值：

- 0: 成功
- 1: 超出分配空间
- 2: 用户已存在
- 3: 失败
- 4: 超出分配用户数

例子:

参考附件的: `public static int createUserAccount(Env env);`

2. `public static int changeUserPassword(Env env, R_UserAccount ua);`

说明: 用于更改用户密码。

参数说明:

- 1) `env`: Env 对象
- 2) `ua`: R\_UserAccount 对象, R\_UserAccount 对象通过 `m_useraccount` 属性传递 用户帐号详细信息。

返回值:

- 0: 成功
- 1: 失败

例子:

参考附件的: `public static int changeUserPassword (Env env);`

3. `public static int deleteUserAccount(Env env, R_UserAccount ua);`

说明: 用于删除用户。

参数说明:

- 1) `env`: Env 对象
- 2) `ua`: R\_UserAccount 对象, R\_UserAccount 对象通过 `m_useraccount` 属性传递 用户帐号详细信息。

返回值:

- 0: 成功
- 1: 失败

例子:

参考附件的: `public static int deleteUserAccount (Env env);`

4. `public static int auth(Env env, String username,String domain,String password);`

说明: 用于验证用户。

参数说明:

- 1) `env`: Env 对象
- 2) `username`: 用户名。
- 3) `domain`: 域名
- 4) `password`: 密码

返回值:

0: 成功

1: 失败

例子:

参考附件的: `public static int testAuth (Env env);`

#### 5. `public static int getNewMsgNum(Env env, R_UserAccount ua);`

说明: 用于删除用户。

参数说明:

1) `env`: `Env` 对象

2) `ua`: `R_UserAccount` 对象, `R_UserAccount` 对象通过 `m_useraccount` 属性传递 用户帐号详细信息。

返回值:

$\geq 0$ : 新邮件数

-1: 失败

例子:

参考附件的: `public static int getNewMsgNum (Env env);`

## 四. 文件夹操作类

类名: `jdmail.api.remote.message.R_FolderAdmin`

说明: `R_FolderAdmin` 类提供与文件夹相关的操作功能。

方法:

#### 1. `public static int getFolderList(Env env, String username,String domain ,ArrayList alRet);`

说明: 用于返回指定用户的文件夹列表。

参数说明:

1) `env`: `Env` 对象

2) `username`: 用户名

3) `domain`: 域名

4) `alRet`: 用于保存返回文件夹列表的 `java.util.ArrayList` 对象。

返回值:

0: 成功

-1: 失败

-2: 该用户不存在

例子:

参考附件的: `public static int testGetFolderList(Env env);`

## 五. 邮件操作类

类名: `jdmail.api.remote.message.R_MessageAdmin`

说明: R\_MessageAdmin 类提供与用户邮件相关的操作功能。

方法:

1. `public static int getMessageCount(Env env, String username,String domain, String foldername);`

说明: 用于获取用户的指定文件夹的邮件数。

参数说明:

- 1) env : Env 对象
- 2) username: 用户名
- 3) domain: 域名
- 4) foldername: 文件夹编号。

返回值:

>=0: 邮件数

-1: 失败

例子:

参考附件: `public static int testGetMessageCount(Env env);`

2. `public static int getMessageIDList(Env env, String username, String domain, String foldername, int iFrom, int iCount, ArrayList alRet)`

说明: 用于获取用户的指定文件夹的邮件编号列表。

参数说明:

- 1) env : Env 对象
- 2) username: 用户名
- 3) domain: 域名
- 4) foldername: 文件夹编号。
- 5) iFrom: 起始邮件序号, 注意: 本方法是于倒数计算邮件, 所以要获取所有邮件, 应该是 iFrom 值与 `getMessageCount` 返回值相同。
- 6) iCount: 指定要获取的邮件数
- 7) alRet: 用于保存返回文件夹列表的 `java.util.ArrayList` 对象。

返回值:

>=0: 邮件数

-1: 失败

例子:

参考附件: `public static int testGetMessageIDList (Env env);`

3. `public static byte[] getMessage(Env env, String username, String domain, String foldername, String messageID)`

说明: 用于获取用户的指定邮件原文, 然后可用 `package jdkmail.mime` 内相关类对邮件内容进行分析。

参数说明:

- 1) env : Env 对象
- 2) username: 用户名
- 3) domain: 域名
- 4) foldername: 文件夹编号。
- 5) messageID: 邮件编号, 该邮件编号可通过 getMessageIDList 获取。

返回值:

byte[] 对象: 邮件原文

null: 失败

例子:

参考附件: `public static int testGetMessage(Env env);`

4. `public static int sendMessage(Env env, String username, String domain, String to, String subject, String body);`

说明: 用于发送邮件。

参数说明:

- 1) env : Env 对象
- 2) username: 用户名
- 3) domain: 域名
- 4) to: 收件人
- 5) subject: 主题
- 6) body: 邮件正文

返回值:

0: 成功

-1: 失败

例子:

参考附件: `public static int sendMail (Env env);`

### 附件 (TestApi.java):

```
package jdmail.api.remote;

import java.util.*;
import jdmail.api.remote.user.*;
import jdmail.api.remote.message.*;
import jdmail.mime.*;
import jdmail.util.*;

import java.io.*;

/**
 * testcase for api
 *
 * @author
 * @version 1.0.0
 */
public class TestApi {
    public static void main(String[] args) {
        // 建立调用环境
        Env env = Env.getEnv("http://localhost:8899/tmweb/mailmain",
"admin",
        "admin");

        // int iRet = sendMail(env);
        int iRet = createUserAccount(env);
        iRet = changeUserPassword(env);
        iRet = deleteUserAccount(env);
        iRet = 0;
    }

    /**
     * 测试新建用户
     *
     * @param iIndex
     * @return
     */
    public static int createUserAccount(Env env) {

        R_UserAccount rua = new R_UserAccount();

        rua.m_useraccount = new UserAccount();
        UserAccount ua = rua.m_useraccount;
    }
}
```

```

    ua.username = "ab";
    ua.setPassword("ab");
    ua.usertype = "U";
    ua.m_domain = "a.com";
    ua.m_UserProfile = new UserProfile();

    ua.m_UserProfile.first_name = "ab";

    ua.m_UserProfile.last_name = "";
    ua.m_UserProfile.organiztion = "";
    ua.m_UserProfile.department = "";
    ua.m_UserProfile.address = "";
    ua.m_UserProfile.city = "";
    ua.m_UserProfile.postalcode = "";
    ua.m_UserProfile.telephone = "";
    ua.m_UserProfile.state_province = "";
    ua.m_UserProfile.country = "";
    ua.m_UserProfile.items = 50;

    ua.enable = "true";
    ua.enable_smtp = "true";
    ua.enable_pop3 = "true";
    ua.enable_imap4 = "true";
    ua.enable_webaccess = "true";
    ua.enable_localdomain = "false";

    ua.max_mailbox_size = -1;
    ua.max_mailbox_msgs = -1;

    int iRet = R_UserAccount.createUserAccount(env, rua);
    return iRet;
}

/**
 * 测试删除用户
 *
 * @return
 */
public static int deleteUserAccount(Env env) {
    R_UserAccount rua = new R_UserAccount();
    // rua.m_useraccount = new
    rua.m_useraccount = new UserAccount();
    UserAccount ua = rua.m_useraccount;

```

```

    ua.username = "ab";

    ua.m_domain = "a.com";
    int iRet = R_UserAccount.deleteUserAccount(env, rua);
    return iRet;
}

/**
 * 测试更改密码
 *
 * @return
 */
public static int changeUserPassword(Env env) {
    R_UserAccount rua = new R_UserAccount();
    rua.m_useraccount = new UserAccount();
    UserAccount ua = rua.m_useraccount;

    ua.username = "ab";
    ua.m_domain = "a.com";
    // 新密码
    ua.setPassword("123");
    int iRet = R_UserAccount.changeUserPassword(env, rua);
    return iRet;
}

/**
 * 测试获取用户的新邮件数
 *
 * @return
 */
public static int getNewMsgNum(Env env) {
    // 建立R_UserAccount api对象
    R_UserAccount rua = new R_UserAccount();
    // 指定要取最新邮件数的用户
    rua.m_useraccount = new UserAccount();
    UserAccount ua = rua.m_useraccount;

    ua.username = "a1";
    ua.m_domain = "a.com";
    // 调用api取最新邮件数的用户
    int iRet = R_UserAccount.getNewMsgNum(env, rua);
    return iRet;
}

```



```

}

/**
 * 测试用户验证
 *
 * @return
 */
public static int testAuth(Env env) {
    // 参数说明:
    // env , 用户名, 域, 密码
    int iRet = R_UserAccount.auth(env, "a1", "a.com", "a");

    return iRet;
}

/**
 * 测试获取文件夹列表
 *
 * @return
 */
public static int testGetFolderList(Env env) {
    ArrayList alRet = new ArrayList();

    int iRet = R_FolderAdmin.getFolderList(env, "a1", "a.com",
alRet);

    if (iRet == 0) { // 如果执行成功
        Folder folder = null;
        for (int i = 0; i < alRet.size(); i++) {
            folder = (Folder) alRet.get(i);
            System.out.println("folderid:" + folder.FolderName);
            System.out.println("foldername:" + folder.FolderDesc);
        }
    }

    return iRet;
}

public static int testGetMessageCount(Env env) {

    // 获取a1@a.com 的收件箱中的总邮件数
    int iCount = R_MessageAdmin.getMessageCount(env, "a1", "a.com",
"new");
    System.out.println("total msgs:" + iCount);
}

```

```

        int iRet = 0;
        return iRet;
    }

    public static int testGetMessageIDList(Env env) {
        int iCount = R_MessageAdmin.getMessageCount(env, "al", "a.com",
"new");

        ArrayList alRet = new ArrayList();

        int iRet = R_MessageAdmin.getMessageIDList(env, "al", "a.com",
"new",
            iCount - 1, iCount, alRet);

        if (iRet > 0) {
            for (int i = 0; i < iRet; i++) {
                System.out.println("msgid:" + (String) alRet.get(i));
            }
        }

        return iRet;
    }

    public static int testGetMessage(Env env) {
        StringBuffer sbFile = new StringBuffer();
        // System.get
        int iCount = R_MessageAdmin.getMessageCount(env, "al", "a.com",
"new");

        ArrayList alRet = new ArrayList();

        int iRet = R_MessageAdmin.getMessageIDList(env, "al", "a.com",
"new",
            iCount - 1, iCount, alRet);

        byte[] bs = null;
        if (iRet > 0) {
            for (int i = 0; i < iRet; i++) {
                System.out.println("msgid:" + (String) alRet.get(i));
                bs = R_MessageAdmin.getMessage(env, "al", "a.com", "new",
                    (String) alRet.get(i));

                if (bs != null) {

```

```

        try {
            System.out.println("message source:" + new
String(bs));

            String strTempMsgFile = "e:\\temp\\"
                + (String) alRet.get(i);
            RandomAccessFile raf = new RandomAccessFile(
                strTempMsgFile, "rw");
            raf.write(bs);
            raf.close();
            Message tempMsg = new Message(Message.MSG_MAIL);

            tempMsg.OpenProfile(strTempMsgFile, false);

            System.out.println("邮件内容:");
            HeaderPair hp = tempMsg.m_MimePart.m_MimeHeader
                .FindHeaderPair("subject");

            if (hp != null) {
                String subject =
MailCoder.ext_decode(hp.m_bsValue);

                System.out.println("subject:" + subject);
            }

            tempMsg.Close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    }

    return iRet;
}

/**
 * 测试发送邮件
 *
 * @return
 */
public static int sendMail(Env env) {
    // 参数说明:
    // env , 发件人域, 发件人用户名, 收件人邮件地址, 标题, 内容

```

```
        int iRet = R_MessageAdmin.sendMessage(env, "a2", "a.com",  
"a1@a.com",  
        "test中文标题 subject", "test中文邮件体xxx body");  
  
        return 0;  
    }  
}
```