

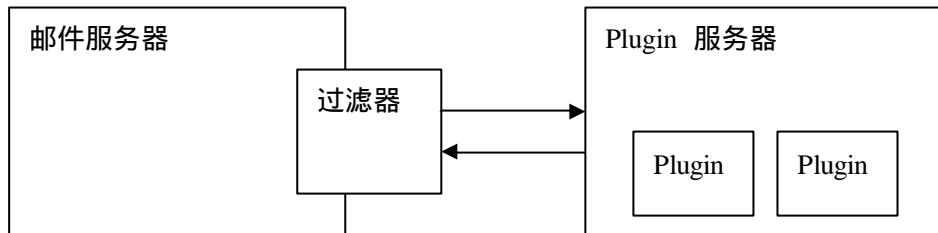
金笛电子邮件系统插件 Plugin 开发方法

一. 概述

客户可通过邮件系统 plugin 功能为系统插入额外的邮件处理功能。如：提取邮件内容，保存邮件内容到数据库等等。

邮件系统提供基于 java 的 plugin 开发接口，所有的客户化 plugin 存在于 plugin 服务器中。

Plugin 服务器与邮件服务器的关系如下：



客户可通过定义过滤规则，当邮件满足一定条件时，触发对 plugin 的调用，此时邮件服务器会通过 TCP/IP 协议发送 plugin 调用请求到 plugin 服务器，plugin 服务器根据提交的请求调用合适的 plugin 组件。

二. Plugin 开发步骤

1. 编写实现接口 `turbomail.api.Action` 的类，在方法 `doAction` 中加入自己的代码。如：
`TestAction`。

其中 `doAction` 的参数如下：

`String[] args` 为邮件服务器传入的参数组。
`args[0]` 调用的 plugin 类名全称。
`args[1]` 邮件路径。

方法返回值：

0 -- 成功执行
1 -- 返回失败，邮件系统不再处理该邮件

以下为一个 plugin 的简单例子：

```
package turbomail.api;

import java.io.FileOutputStream;
import turbomail.mime.Message;
import turbomail.web.MailMain;
import turbomail.web.ServerConf;
```

```

import turbomail.mime.HeaderPair;

public class TestMailAction implements Action{

/*
 *  (non-Javadoc)
 * @see turbomail.api.Action#doAction(java.lang.String[])
 * args[0] ---类名
 * args[1] --- 邮件目录
 * args[2] --- 相关参数字串
 *
 * @return
 * 0 -- 成功执行
 * 1 -- 返回失败， 邮件系统不再处理该邮件
 *
 */

public int doAction(String[] args) {
    try {
        //建立系统环境
        MailMain.s_config = new ServerConf();

        try {
            //新建邮件类，注意邮件类型必须为 Message.MSG_SPOOLQUEUE
            Message tempMsg = new Message(Message.MSG_SPOOLQUEUE);
            //打开邮件
            tempMsg.OpenProfile(strTempMsgFile);
            //取发件人
            HeaderPair pHP = tempMsg.m_MimePart.m_MimeHeader.FindHeaderPair("from");
            if(pHP!=null)
                System.out.println( "subject:" + pHP.m_pValue);

            //取收件人
            pHP = tempMsg.m_MimePart.m_MimeHeader.FindHeaderPair("to");
            if(pHP!=null)
                System.out.println("to:" + pHP.m_pValue);

            //邮件主题
            pHP = tempMsg.m_MimePart.m_MimeHeader.FindHeaderPair("subject");
            if(pHP!=null)
                System.out.println("subject:" + pHP.m_pValue);

            //设置附件临时目录
            String href_prefix = "c:\\\\temp";
        }
    }
}

```

```

//取邮件显示内容, sbEncode 为邮件内容 charset 编码
StringBuffer sbEncode = new StringBuffer();
String strContent = tempMsg.GetShowMsg(href_prefix, 0, true,
sbEncode);
System.out.println("content:" + strContent);

//关闭邮件
tempMsg.Close();
fos.close();
} catch (Exception mimex) {
mimex.printStackTrace();
}
} catch (Exception e) {
e.printStackTrace();
}
return 0;
}
}

```

2. 把实现类打包成.jar，并在 plugin 服务器的启动脚本（windows 系统为 startserver.bat，unix 系统为 startserver.sh）中指明.jar 文件路径。如实现类的包名 test.jar，同时 test.jar 放在 plugin 安装目录下，则需要把启动脚本内容更改如下：

```

java -classpath %PI_HOME%\pluginserver.jar;%PI_HOME%\turbomail.jar;%PI_HOME%\jsdk22.jar ;
%PI_HOME%\test.jar turbomail.pluginserver.MainServer
这样就可把 plugin 安装在 plugin 服务器上。

```

3. 配置过滤规则。只有全局过滤规则才支持 plugin 调用。在全局过滤规则 》 系统接收规则，增加新的触发 plugin 过滤规则。



在“执行动作“项，选择”Plugin (java 处理类)“，填入Plugin 实现类全类名。
 截图的过滤规则意思是，但系统接收到所有邮件，将会调用 Plugin (turbomail.api.TestAction)。

三 . Plugin 服务器安装

Plugin 服务器软件是一个压缩文件包，解压缩后即可使用。

请注意，Plugin 服务器，必须安装在可读写邮件服务器 temp 目录的机器上，否则 Plugin 读取不到要处理的邮件信息。

启动脚本，windows 启动脚本是 startserver.bat， Unix 启动脚本是 startserver.sh。

配置文件为 conf/sys.conf，以下为配置文件样本：

```
#服务器配置
#最大连接数 socket
max_sockets = 200
#最大连接处理数
max_connections = 100
#Plugin 服务端口号
port = 8989

#网络 超时秒
net_timeout = 200

login_username=admin
login_password=admin
```

在 log 目录可查看各种服务器日志信息。

四 . Plugin 服务参数

进入邮件系统管理界面，进入系统设置 》 Plugin 服务参数。

系统设置<==	
Plugin 服务器IP	127.0.0.1
Plugin 服务器端口	8989 (小于或等于0，表示使用默认端口8989)
连接超时	30 秒(小于或等于0，表示使用默认值30)
<input type="button" value="保存"/> <input type="button" value="取消"/>	

其中：

Plugin 服务器 IP ，为 plugin 服务器运行的 IP 地址。

Plugin 服务器端口，此参数根据 Plugin 服务器 的 conf/sys.conf 所配置参数来设置，
默认为 8989。

五 . 测试方法

可用系统默认实现的 turbomail.api.TestAction 测试 plugin 相关服务是否配置正常。当正常调用此 plugin ，会往 pluginserver\log\stdout.log 输出一行以下内容：

call plugin TextAction