

MailMessage



Apr 11, 2024 20:56

Supported Clients

SmartClient WebClient NGClient

Methods Summary

Array	getAttachments()	Returns an array of Attachment instances corresponding to the attachments of this message.
String	getCCAddresses()	Returns a String with all addresses present in the CC field of this message.
String	getFromAddresses()	Returns a String with all addresses present in the From field of this message.
String	getHeaders()	Returns a String with all headers of this message.
String	getHtmlMsg()	Returns a String with the HTML content of this message.
String	getPlainMsg()	Returns a String with the plain content of this message.
String	getRecipientAddresses()	Returns a String with all addresses in the To field of this message.
String	getReplyAddresses()	Returns a String with all addresses in the Reply-To field of this message.
Date	getSentDate()	Returns a Date instance corresponding to the moment when this message was sent.
String	getSubject()	Returns a String with the subject of this message.

Methods Details

getAttachments()
Returns an array of Attachment instances corresponding to the attachments of this message.

Returns

[Array](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```

var msgs = plugins.mail.receiveMail(username, password, true, 0, null, properties);
if (msgs != null)
{
    for (var i=0; i < msgs.length; i++)
    {
        var msg = msgs[i];
        var str = '';
        str += 'From: ' + msg.getFromAddresses() + '\n';
        str += 'To: ' + msg.getRecipientAddresses() + '\n';
        str += 'CC: ' + msg.getCCAddresses() + '\n';
        str += 'Reply to: ' + msg.getReplyAddresses() + '\n';
        str += 'Received on: ' + msg.getReceivedDate() + '\n';
        str += 'Sent on: ' + msg.getSentDate() + '\n\n';
        str += 'Subject: ' + msg.getSubject() + '\n\n';
        str += 'Plain message: ' + msg.getPlainMsg() + '\n\n';
        str += 'HTML message: ' + msg.getHtmlMsg() + '\n\n';
        str += 'Headers: ' + msg.getHeaders() + '\n\n';
        var attachments = msg.getAttachments();
        if (attachments != null) {
            str += 'Number of attachments: ' + attachments.length + '\n\n';
            for (var j=0; j < attachments.length; j++)
            {
                var attachment = attachments[j];
                str += 'Attachment ' + j + '\n';
                str += '    Name: ' + attachment.getName() + '\n';
                str += '    Size: ' + attachment.getData().length + '\n\n';
            }
        }
        plugins.file.writeTXTFile('msg' + i + '.txt', str);
        application.output('Message ' + i + ' retrieved.');
```

getCCAddresses()

Returns a String with all addresses present in the CC field of this message.

Returns

[String](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```

var msgs = plugins.mail.receiveMail(username, password, true, 0, null, properties);
if (msgs != null)
{
    for (var i=0; i < msgs.length; i++)
    {
        var msg = msgs[i];
        var str = '';
        str += 'From: ' + msg.getFromAddresses() + '\n';
        str += 'To: ' + msg.getRecipientAddresses() + '\n';
        str += 'CC: ' + msg.getCCAddresses() + '\n';
        str += 'Reply to: ' + msg.getReplyAddresses() + '\n';
        str += 'Received on: ' + msg.getReceivedDate() + '\n';
        str += 'Sent on: ' + msg.getSentDate() + '\n\n';
        str += 'Subject: ' + msg.getSubject() + '\n\n';
        str += 'Plain message: ' + msg.getPlainMsg() + '\n\n';
        str += 'HTML message: ' + msg.getHtmlMsg() + '\n\n';
        str += 'Headers: ' + msg.getHeaders() + '\n\n';
        var attachments = msg.getAttachments();
        if (attachments != null) {
            str += 'Number of attachments: ' + attachments.length + '\n\n';
            for (var j=0; j < attachments.length; j++)
            {
                var attachment = attachments[j];
                str += 'Attachment ' + j + '\n';
                str += '    Name: ' + attachment.getName() + '\n';
                str += '    Size: ' + attachment.getData().length + '\n\n';
            }
        }
        plugins.file.writeTXTFile('msg' + i + '.txt', str);
        application.output('Message ' + i + ' retrieved.');
```

getFromAddresses()

Returns a String with all addresses present in the From field of this message.

Returns

[String](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```

var msgs = plugins.mail.receiveMail(username, password, true, 0, null, properties);
if (msgs != null)
{
    for (var i=0; i < msgs.length; i++)
    {
        var msg = msgs[i];
        var str = '';
        str += 'From: ' + msg.getFromAddresses() + '\n';
        str += 'To: ' + msg.getRecipientAddresses() + '\n';
        str += 'CC: ' + msg.getCCAddresses() + '\n';
        str += 'Reply to: ' + msg.getReplyAddresses() + '\n';
        str += 'Received on: ' + msg.getReceivedDate() + '\n';
        str += 'Sent on: ' + msg.getSentDate() + '\n\n';
        str += 'Subject: ' + msg.getSubject() + '\n\n';
        str += 'Plain message: ' + msg.getPlainMsg() + '\n\n';
        str += 'HTML message: ' + msg.getHtmlMsg() + '\n\n';
        str += 'Headers: ' + msg.getHeaders() + '\n\n';
        var attachments = msg.getAttachments();
        if (attachments != null) {
            str += 'Number of attachments: ' + attachments.length + '\n\n';
            for (var j=0; j < attachments.length; j++)
            {
                var attachment = attachments[j];
                str += 'Attachment ' + j + '\n';
                str += '    Name: ' + attachment.getName() + '\n';
                str += '    Size: ' + attachment.getData().length + '\n\n';
            }
        }
        plugins.file.writeTXTFile('msg' + i + '.txt', str);
        application.output('Message ' + i + ' retrieved.');
```

getHeaders()

Returns a String with all headers of this message.

Returns

[String](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```

var msgs = plugins.mail.receiveMail(username, password, true, 0, null, properties);
if (msgs != null)
{
    for (var i=0; i < msgs.length; i++)
    {
        var msg = msgs[i];
        var str = '';
        str += 'From: ' + msg.getFromAddresses() + '\n';
        str += 'To: ' + msg.getRecipientAddresses() + '\n';
        str += 'CC: ' + msg.getCCAddresses() + '\n';
        str += 'Reply to: ' + msg.getReplyAddresses() + '\n';
        str += 'Received on: ' + msg.getReceivedDate() + '\n';
        str += 'Sent on: ' + msg.getSentDate() + '\n\n';
        str += 'Subject: ' + msg.getSubject() + '\n\n';
        str += 'Plain message: ' + msg.getPlainMsg() + '\n\n';
        str += 'HTML message: ' + msg.getHtmlMsg() + '\n\n';
        str += 'Headers: ' + msg.getHeaders() + '\n\n';
        var attachments = msg.getAttachments();
        if (attachments != null) {
            str += 'Number of attachments: ' + attachments.length + '\n\n';
            for (var j=0; j < attachments.length; j++)
            {
                var attachment = attachments[j];
                str += 'Attachment ' + j + '\n';
                str += '    Name: ' + attachment.getName() + '\n';
                str += '    Size: ' + attachment.getData().length + '\n\n';
            }
        }
        plugins.file.writeTXTFile('msg' + i + '.txt', str);
        application.output('Message ' + i + ' retrieved.');
```

getHtmlMsg()

Returns a String with the HTML content of this message.

Returns

[String](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```

var msgs = plugins.mail.receiveMail(username, password, true, 0, null, properties);
if (msgs != null)
{
    for (var i=0; i < msgs.length; i++)
    {
        var msg = msgs[i];
        var str = '';
        str += 'From: ' + msg.getFromAddresses() + '\n';
        str += 'To: ' + msg.getRecipientAddresses() + '\n';
        str += 'CC: ' + msg.getCCAddresses() + '\n';
        str += 'Reply to: ' + msg.getReplyAddresses() + '\n';
        str += 'Received on: ' + msg.getReceivedDate() + '\n';
        str += 'Sent on: ' + msg.getSentDate() + '\n\n';
        str += 'Subject: ' + msg.getSubject() + '\n\n';
        str += 'Plain message: ' + msg.getPlainMsg() + '\n\n';
        str += 'HTML message: ' + msg.getHtmlMsg() + '\n\n';
        str += 'Headers: ' + msg.getHeaders() + '\n\n';
        var attachments = msg.getAttachments();
        if (attachments != null) {
            str += 'Number of attachments: ' + attachments.length + '\n\n';
            for (var j=0; j < attachments.length; j++)
            {
                var attachment = attachments[j];
                str += 'Attachment ' + j + '\n';
                str += '    Name: ' + attachment.getName() + '\n';
                str += '    Size: ' + attachment.getData().length + '\n\n';
            }
        }
        plugins.file.writeTXTFile('msg' + i + '.txt', str);
        application.output('Message ' + i + ' retrieved.');
```

getPlainMsg()

Returns a String with the plain content of this message.

Returns

[String](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```

var msgs = plugins.mail.receiveMail(username, password, true, 0, null, properties);
if (msgs != null)
{
    for (var i=0; i < msgs.length; i++)
    {
        var msg = msgs[i];
        var str = '';
        str += 'From: ' + msg.getFromAddresses() + '\n';
        str += 'To: ' + msg.getRecipientAddresses() + '\n';
        str += 'CC: ' + msg.getCCAddresses() + '\n';
        str += 'Reply to: ' + msg.getReplyAddresses() + '\n';
        str += 'Received on: ' + msg.getReceivedDate() + '\n';
        str += 'Sent on: ' + msg.getSentDate() + '\n\n';
        str += 'Subject: ' + msg.getSubject() + '\n\n';
        str += 'Plain message: ' + msg.getPlainMsg() + '\n\n';
        str += 'HTML message: ' + msg.getHtmlMsg() + '\n\n';
        str += 'Headers: ' + msg.getHeaders() + '\n\n';
        var attachments = msg.getAttachments();
        if (attachments != null) {
            str += 'Number of attachments: ' + attachments.length + '\n\n';
            for (var j=0; j < attachments.length; j++)
            {
                var attachment = attachments[j];
                str += 'Attachment ' + j + '\n';
                str += '    Name: ' + attachment.getName() + '\n';
                str += '    Size: ' + attachment.getData().length + '\n\n';
            }
        }
        plugins.file.writeTXTFile('msg' + i + '.txt', str);
        application.output('Message ' + i + ' retrieved.');
```

getRecipientAddresses()

Returns a String with all addresses in the To field of this message.

Returns

[String](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```

var msgs = plugins.mail.receiveMail(username, password, true, 0, null, properties);
if (msgs != null)
{
    for (var i=0; i < msgs.length; i++)
    {
        var msg = msgs[i];
        var str = '';
        str += 'From: ' + msg.getFromAddresses() + '\n';
        str += 'To: ' + msg.getRecipientAddresses() + '\n';
        str += 'CC: ' + msg.getCCAddresses() + '\n';
        str += 'Reply to: ' + msg.getReplyAddresses() + '\n';
        str += 'Received on: ' + msg.getReceivedDate() + '\n';
        str += 'Sent on: ' + msg.getSentDate() + '\n\n';
        str += 'Subject: ' + msg.getSubject() + '\n\n';
        str += 'Plain message: ' + msg.getPlainMsg() + '\n\n';
        str += 'HTML message: ' + msg.getHtmlMsg() + '\n\n';
        str += 'Headers: ' + msg.getHeaders() + '\n\n';
        var attachments = msg.getAttachments();
        if (attachments != null) {
            str += 'Number of attachments: ' + attachments.length + '\n\n';
            for (var j=0; j < attachments.length; j++)
            {
                var attachment = attachments[j];
                str += 'Attachment ' + j + '\n';
                str += '    Name: ' + attachment.getName() + '\n';
                str += '    Size: ' + attachment.getData().length + '\n\n';
            }
        }
        plugins.file.writeTXTFile('msg' + i + '.txt', str);
        application.output('Message ' + i + ' retrieved.');
```

getReplyAddresses()

Returns a String with all addresses in the Reply-To field of this message.

Returns

[String](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```

var msgs = plugins.mail.receiveMail(username, password, true, 0, null, properties);
if (msgs != null)
{
    for (var i=0; i < msgs.length; i++)
    {
        var msg = msgs[i];
        var str = '';
        str += 'From: ' + msg.getFromAddresses() + '\n';
        str += 'To: ' + msg.getRecipientAddresses() + '\n';
        str += 'CC: ' + msg.getCCAddresses() + '\n';
        str += 'Reply to: ' + msg.getReplyAddresses() + '\n';
        str += 'Received on: ' + msg.getReceivedDate() + '\n';
        str += 'Sent on: ' + msg.getSentDate() + '\n\n';
        str += 'Subject: ' + msg.getSubject() + '\n\n';
        str += 'Plain message: ' + msg.getPlainMsg() + '\n\n';
        str += 'HTML message: ' + msg.getHtmlMsg() + '\n\n';
        str += 'Headers: ' + msg.getHeaders() + '\n\n';
        var attachments = msg.getAttachments();
        if (attachments != null) {
            str += 'Number of attachments: ' + attachments.length + '\n\n';
            for (var j=0; j < attachments.length; j++)
            {
                var attachment = attachments[j];
                str += 'Attachment ' + j + '\n';
                str += '    Name: ' + attachment.getName() + '\n';
                str += '    Size: ' + attachment.getData().length + '\n\n';
            }
        }
        plugins.file.writeTXTFile('msg' + i + '.txt', str);
        application.output('Message ' + i + ' retrieved.');
```

getSentDate()

Returns a Date instance corresponding to the moment when this message was sent.

Returns

[Date](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```

var msgs = plugins.mail.receiveMail(username, password, true, 0, null, properties);
if (msgs != null)
{
    for (var i=0; i < msgs.length; i++)
    {
        var msg = msgs[i];
        var str = '';
        str += 'From: ' + msg.getFromAddresses() + '\n';
        str += 'To: ' + msg.getRecipientAddresses() + '\n';
        str += 'CC: ' + msg.getCCAddresses() + '\n';
        str += 'Reply to: ' + msg.getReplyAddresses() + '\n';
        str += 'Received on: ' + msg.getReceivedDate() + '\n';
        str += 'Sent on: ' + msg.getSentDate() + '\n\n';
        str += 'Subject: ' + msg.getSubject() + '\n\n';
        str += 'Plain message: ' + msg.getPlainMsg() + '\n\n';
        str += 'HTML message: ' + msg.getHtmlMsg() + '\n\n';
        str += 'Headers: ' + msg.getHeaders() + '\n\n';
        var attachments = msg.getAttachments();
        if (attachments != null) {
            str += 'Number of attachments: ' + attachments.length + '\n\n';
            for (var j=0; j < attachments.length; j++)
            {
                var attachment = attachments[j];
                str += 'Attachment ' + j + '\n';
                str += '    Name: ' + attachment.getName() + '\n';
                str += '    Size: ' + attachment.getData().length + '\n\n';
            }
        }
        plugins.file.writeTXTFile('msg' + i + '.txt', str);
        application.output('Message ' + i + ' retrieved.');
```

getSubject()

Returns a String with the subject of this message.

Returns

[String](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var msgs = plugins.mail.receiveMail(username, password, true, 0, null, properties);
if (msgs != null)
{
    for (var i=0; i < msgs.length; i++)
    {
        var msg = msgs[i];
        var str = '';
        str += 'From: ' + msg.getFromAddresses() + '\n';
        str += 'To: ' + msg.getRecipientAddresses() + '\n';
        str += 'CC: ' + msg.getCCAddresses() + '\n';
        str += 'Reply to: ' + msg.getReplyAddresses() + '\n';
        str += 'Received on: ' + msg.getReceivedDate() + '\n';
        str += 'Sent on: ' + msg.getSentDate() + '\n\n';
        str += 'Subject: ' + msg.getSubject() + '\n\n';
        str += 'Plain message: ' + msg.getPlainMsg() + '\n\n';
        str += 'HTML message: ' + msg.getHtmlMsg() + '\n\n';
        str += 'Headers: ' + msg.getHeaders() + '\n\n';
        var attachments = msg.getAttachments();
        if (attachments != null) {
            str += 'Number of attachments: ' + attachments.length + '\n\n';
            for (var j=0; j < attachments.length; j++)
            {
                var attachment = attachments[j];
                str += 'Attachment ' + j + '\n';
                str += '    Name: ' + attachment.getName() + '\n';
                str += '    Size: ' + attachment.getData().length + '\n\n';
            }
        }
        plugins.file.writeTXTFile('msg' + i + '.txt', str);
        application.output('Message ' + i + ' retrieved.');
```

```
    }
}
else
{
    application.output("Failed to retrieve messages.");
}
```