

mail

 Apr 07, 2024 09:44

Supported Clients

SmartClient WebClient NGClient

Methods Summary

Attachment	<code>createBinaryAttachment(filename, binarydata)</code>	Creates a binary attachment object.
Attachment	<code>createBinaryAttachment(filename, binarydata, mimeType)</code>	Creates a binary attachment object.
Attachment	<code>createTextAttachment(filename, textdata)</code>	Creates a text based attachment object with the default 'text/plain' mimetype
Attachment	<code>createTextAttachment(filename, textdata, mimeType)</code>	Creates a text based attachment object.
String	<code>getLastSendMailExceptionMsg()</code>	Get the exception that occurred in the last sendMail attempt (null if no exception occurred).
MailMessage	<code>getMailMessage(binaryblob/string)</code>	Helper method, returns MailMessage object from binary or 7bits string.
Array	<code>getPlainMailAddresses(addressesString)</code>	Helper method to only get the plain addresses.
Boolean	<code>isValidEmailAddress(email)</code>	Checks whether the given e-mail address is valid or not.
Array	<code>receiveMail(username, password, leaveMsgsOnServer)</code>	Receive mails from pop3 account.
Array	<code>receiveMail(username, password, leaveMsgsOnServer, receiveMode)</code>	Receive mails from pop3 account.
Array	<code>receiveMail(username, password, leaveMsgsOnServer, receiveMode, onlyReceiveMsgWithSentDate)</code>	Receive mails from pop3 account.
Array	<code>receiveMail(username, password, leaveMsgsOnServer, receiveMode, onlyReceiveMsgWithSentDate, pop3Host)</code>	Receive mails from pop3 account.
Array	<code>receiveMail(username, password, leaveMsgsOnServer, receiveMode, onlyReceiveMsgWithSentDate, properties)</code>	Receive mails from pop3 account.
Boolean	<code>sendBulkMail(to, from, subject, msgText)</code>	Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).
Boolean	<code>sendBulkMail(to, from, subject, msgText, cc)</code>	Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).
Boolean	<code>sendBulkMail(to, from, subject, msgText, cc, bcc)</code>	Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).
Boolean	<code>sendBulkMail(to, from, subject, msgText, cc, bcc, attachment)</code>	Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).
Boolean	<code>sendBulkMail(to, from, subject, msgText, cc, bcc, attachment, smtpHost)</code>	Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).
Boolean	<code>sendBulkMail(to, from, subject, msgText, cc, bcc, attachment, overrideProperties)</code>	Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).
Boolean	<code>sendBulkMail(to, from, subject, msgText, cc, bcc, attachments)</code>	Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).
Boolean	<code>sendBulkMail(to, from, subject, msgText, cc, bcc, attachments, smtpHost)</code>	Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).
Boolean	<code>sendBulkMail(to, from, subject, msgText, cc, bcc, attachments, overrideProperties)</code>	Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).
Boolean	<code>sendMail(to, from, subject, msgText)</code>	Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).
Boolean	<code>sendMail(to, from, subject, msgText, cc)</code>	Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).
Boolean	<code>sendMail(to, from, subject, msgText, cc, bcc)</code>	Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).
Boolean	<code>sendMail(to, from, subject, msgText, cc, bcc, attachment)</code>	Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).
Boolean	<code>sendMail(to, from, subject, msgText, cc, bcc, attachment, smtpHost)</code>	Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).
Boolean	<code>sendMail(to, from, subject, msgText, cc, bcc, attachment, overrideProperties)</code>	Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).
Boolean	<code>sendMail(to, from, subject, msgText, cc, bcc, attachments)</code>	Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).
Boolean	<code>sendMail(to, from, subject, msgText, cc, bcc, attachments, smtpHost)</code>	Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).
Boolean	<code>sendMail(to, from, subject, msgText, cc, bcc, attachments, overrideProperties)</code>	Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

Methods Details

createBinaryAttachment(filename, binarydata)

Creates a binary attachment object.

Parameters

[String](#) filename ;
[Array](#) binarydata ;

Returns

[Attachment](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var attachment1 = plugins.mail.createBinaryAttachment('logo1.gif',plugins.file.readFile('c:/temp/a_logo.gif'));
var attachment2 = plugins.mail.createBinaryAttachment('logo2.gif',plugins.file.readFile('c:/temp/another_logo.gif'));
var success = plugins.mail.sendMail('to_someone@example.com', 'John Cobb <from_me@example.org>', 'subject', 'msgText',null,null,new Array(attachment1,attachment2));
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send mail','OK');
}
```

createBinaryAttachment(filename, binarydata, mimeType)

Creates a binary attachment object.

Parameters

[String](#) filename ;
[Array](#) binarydata ;
[String](#) mimeType ;

Returns

[Attachment](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var attachment1 = plugins.mail.createBinaryAttachment('logo1.gif',plugins.file.readFile('c:/temp/a_logo.gif', 'image/gif'));
var attachment2 = plugins.mail.createBinaryAttachment('logo2.gif',plugins.file.readFile('c:/temp/another_logo.gif', 'image/gif'));
var success = plugins.mail.sendMail('to_someone@example.com', 'John Cobb <from_me@example.org>', 'subject', 'msgText',null,null,new Array(attachment1,attachment2));
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send mail','OK');
}
```

createTextAttachment(filename, textdata)

Creates a text based attachment object with the default 'text/plain' mimetype

Parameters

[String](#) filename ;
[String](#) textdata ;

Returns

[Attachment](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var attachment = plugins.mail.createTextAttachment('readme.html','<html>bla bla bla');
var success = plugins.mail.sendMail('to_someone@example.com', 'John Cobb <from_me@example.com>', 'subject',
'msgText',null,null,attachment);
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send mail','OK');
}
```

createTextAttachment(filename, textdata, mimeType)

Creates a text based attachment object.

Parameters

[String](#) filename ;
[String](#) textdata ;
[String](#) mimeType;

Returns

[Attachment](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var attachment = plugins.mail.createTextAttachment('readme.html','<html>bla bla bla', 'text/html');
var success = plugins.mail.sendMail('to_someone@example.com', 'John Cobb <from_me@example.com>', 'subject',
'msgText',null,null,attachment);
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send mail','OK');
}
```

getLastSendMailExceptionMsg()

Get the exception that occurred in the last sendMail attempt (null if no exception occurred).

Returns

[String](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var success = plugins.mail.sendMail('to_someone@example.com,to_someone_else@example.org', 'John Cobb
<from_me@example.com>', 'subject', 'my message',null,'unnamed@example.com');
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert',plugins.mail.getLastSendMailExceptionMsg(),'OK');
}
```

getMailMessage(binaryblob/string)

Helper method, returns MailMessage object from binary or 7bits string.

Parameters

[Object](#) binaryblob/string;

Returns

[MailMessage](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var msg = plugins.mail.getMailMessage(myBlob);
if (msg != null) //if is null error occurred!
{
    application.output(msg.getFromAddresses())
}
```

getPlainMailAddresses(addressesString)

Helper method to only get the plain addresses.

Parameters

[Object](#) addressesString ;

Returns

[Array](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
var plainArray = plugins.mail.getPlainMailAddresses('John Cobb <from_me@example.com>,Pete Cobb<from_pete@example.com>');
application.output(plainArray[0]) //will return 'from_me@example.com'
```

isValidEmailAddress(email)

Checks whether the given e-mail address is valid or not.

Parameters

[String](#) email ;

Returns

[Boolean](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```
plugins.mail.isValidEmailAddress("me@example.com");
```

receiveMail(username, password, leaveMsgsOnServer)

Receive mails from pop3 account.

Parameters

[String](#) username ;
[String](#) password ;
[Boolean](#) leaveMsgsOnServer ;

Returns

[Array](#)

Supported Clients

SmartClient, WebClient, NGClient

Sample

```

var msgs = plugins.mail.receiveMail('mylogin', 'secretpass', true);
if (msgs != null) //if is null error occurred!
{
    for (var i = 0 ; i < msgs.length ; i++)
    {
        var msg = msgs[i]
        application.output(msg.getFromAddresses())
        application.output(msg.getRecipientAddresses())
        application.output(msg.getReplyAddresses())
        application.output(msg.getSentDate())
        application.output(msg.getHeaders())
        application.output(msg.getSubject())
        application.output(msg.getHtmlMsg())
        application.output(msg.getPlainMsg())
        var attachments = msg.getAttachments()
        if (attachments != null)
        {
            for (var j = 0 ; j < attachments.length ; j++)
            {
                var attachment = attachments[j]
                application.output(attachment.getName())
                var attachmentDataByteArray = attachment.getData()
                //write attachmentDataByteArray to a file...
            }
        }
    }
}

```

receiveMail(username, password, leaveMsgsOnServer, receiveMode)

Receive mails from pop3 account.

Parameters

```

String username      ;
String password      ;
Boolean leaveMsgsOnServer;
Number receiveMode   ;

```

Returns

Array

Supported Clients

SmartClient, WebClient, NGClient

Sample

```

var receiveMode = 1;//0=FULL,1=HEADERS_ONLY,2=NO_ATTACHMENTS
var msgs = plugins.mail.receiveMail('mylogin', 'secretpass', true, 0);
if (msgs != null) //if is null error occurred!
{
    for (var i = 0 ; i < msgs.length ; i++)
    {
        var msg = msgs[i]
        application.output(msg.getFromAddresses())
        application.output(msg.getRecipientAddresses())
        application.output(msg.getReplyAddresses())
        application.output(msg.getSentDate())
        application.output(msg.getHeaders())
        application.output(msg.getSubject())
    }
}

```

receiveMail(username, password, leaveMsgsOnServer, receiveMode, onlyReceiveMsgWithSentDate)

Receive mails from pop3 account.

Parameters

```
String username          ;
String password          ;
Boolean leaveMsgsOnServer ;
Number receiveMode       ;
Date    onlyReceiveMsgWithSentDate ;
```

Returns

```
Array
```

Supported Clients

```
SmartClient,WebClient,NGClient
```

Sample

```
//it is also possible to first receive the headers and later receive a full message with particular 'sentdate'
//var receiveMode = 1;//0=FULL,1=HEADERS_ONLY,2=NO_ATTACHMENTS
var msgs = plugins.mail.receiveMail('mylogin', 'secretpass', true, 0,
theSentDateObjectFromPreviousHeaderLoading);
if (msgs != null) //if is null error occurred!
{
    for (var i = 0 ; i < msgs.length ; i++)
    {
        var msg = msgs[i]
        application.output(msg.getFromAddresses())
        application.output(msg.getRecipientAddresses())
        application.output(msg.getReplyAddresses())
        application.output(msg.getSentDate())
        application.output(msg.getHeaders())
        application.output(msg.getSubject())
    }
}
```

receiveMail(username, password, leaveMsgsOnServer, receiveMode, onlyReceiveMsgWithSentDate, pop3Host)

Receive mails from pop3 account.

Parameters

```
String username          ;
String password          ;
Boolean leaveMsgsOnServer ;
Number receiveMode       ;
Date    onlyReceiveMsgWithSentDate ;
String pop3Host          ;
```

Returns

```
Array
```

Supported Clients

```
SmartClient,WebClient,NGClient
```

Sample

```
//it is also possible to first receive the headers and later receive a full message
var receiveMode = 0;//0=FULL,1=HEADERS_ONLY,2=NO_ATTACHMENTS
var pop3Host = 'myserver.com';
var msgs = plugins.mail.receiveMail('mylogin', 'secretpass', true, receiveMode, null, pop3Host);
if (msgs != null) //if is null error occurred!
{
    for (var i = 0 ; i < msgs.length ; i++)
    {
        var msg = msgs[i]
        application.output(msg.getFromAddresses())
        application.output(msg.getRecipientAddresses())
        application.output(msg.getReplyAddresses())
        application.output(msg.getSentDate())
        application.output(msg.getHeaders())
        application.output(msg.getSubject())
        application.output(msg.getHtmlMsg())
        application.output(msg.getPlainMsg())
        var attachments = msg.getAttachments()
        if (attachments != null)
        {
            for (var j = 0 ; j < attachments.length ; j++)
            {
                var attachment = attachments[j]
                application.output(attachment.getName())
                var attachmentDataByteArray = attachment.getData()
                //write attachmentDataByteArray to a file...
            }
        }
    }
}
```

receiveMail(username, password, leaveMsgsOnServer, receiveMode, onlyReceiveMsgWithSentDate, properties)

Receive mails from pop3 account.

Parameters

String	username	;
String	password	;
Boolean	leaveMsgsOnServer	;
Number	receiveMode	;
Date	onlyReceiveMsgWithSentDate	;
Array	properties	;

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient

Sample

```

var receiveMode = 1;//0=FULL,1=HEADERS_ONLY,2=NO_ATTACHMENTS

var properties = new Array();
properties[0] = 'mail.pop3.port=995';
properties[1] = 'mail.pop3.ssl.enable=true';
properties[2] = 'mail.pop3.host=myserver.com';
properties[3] = 'mail.pop3.user=user@myserver.com';

var msgs = plugins.mail.receiveMail('mylogin', 'secretpass', true, receiveMode, null, properties);
if (msgs != null) //if is null error occurred!
{
    for (var i = 0 ; i < msgs.length ; i++)
    {
        var msg = msgs[i]
        application.output(msg.getFromAddresses())
        application.output(msg.getRecipientAddresses())
        application.output(msg.getReplyAddresses())
        application.output(msg.getSentDate())
        application.output(msg.getHeaders())
        application.output(msg.getSubject())
    }
}

```

sendBulkMail(to, from, subject, msgText)

Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

A bulk email makes it possible for one to not receive "out of office" emails back from receiver.

Parameters

String to A string containing 1 or multiple addresses seperated by a comma.
String from A string containing an address and an optional reply address, seperated by a comma.
String subject The subject of the bulk mail
String msgText The message text

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```

var msgText = 'plain msg<html>styled html msg</html>';
var success = plugins.mail.sendBulkMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>,replyTo@example.com', 'subject', msgText);
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send bulk mail','OK');
}

```

sendBulkMail(to, from, subject, msgText, cc)

Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

A bulk email makes it possible for one to not receive "out of office" emails back from receiver.

Parameters

String to A string containing 1 or multiple addresses seperated by a comma.
String from A string containing an address and an optional reply address, seperated by a comma.
String subject The subject of the bulk mail
String msgText The message text
String cc One or more addresses seperated by a comma

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var msgText = 'plain msg<html>styled html msg</html>';
var success = plugins.mail.sendBulkMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>', 'subject', msgText, 'cc1@example.com,cc2@example.com');
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send bulk mail','OK');
}
```

sendBulkMail(to, from, subject, msgText, cc, bcc)

Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

A bulk email makes it possible for one to not receive "out of office" emails back from receiver.

Parameters

String to A string containing 1 or multiple addresses seperated by a comma.
String from A string containing an address and an optional reply address, seperated by a comma.
String subject The subject of the bulk mail
String msgText The message text
String cc One or more addresses seperated by a comma
String bcc One or more addresses seperated by a comma

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var msgText = 'plain msg<html>styled html msg</html>';
var success = plugins.mail.sendBulkMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>', 'subject', msgText,null,'bcc1@example.com');
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send bulk mail','OK');
}
```

sendBulkMail(to, from, subject, msgText, cc, bcc, attachment)

Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

A bulk email makes it possible for one to not receive "out of office" emails back from receiver.

Parameters

String to A string containing 1 or multiple addresses seperated by a comma.
String from A string containing an address and an optional reply address, seperated by a comma.
String subject The subject of the bulk mail
String msgText The message text
String cc One or more addresses seperated by a comma
String bcc One or more addresses seperated by a comma
Attachment attachment A single attachment

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var attachment = plugins.mail.createBinaryAttachment('embedded',plugins.file.readFile('c:/temp/a_logo.gif'));
var msgText = 'plain msg<html>styled html msg</html>';
var success = plugins.mail.sendBulkMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>', 'subject', msgText,null,'bcc1@example.com,bcc2@example.com',attachment);
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send bulk mail','OK');
}
```

sendBulkMail(to, from, subject, msgText, cc, bcc, attachment, smtpHost)

Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

A bulk email makes it possible for one to not receive "out of office" emails back from receiver.

Parameters

String	to	A string containing 1 or multiple addresses seperated by a comma.
String	from	A string containing an address and an optional reply address, seperated by a comma.
String	subject	The subject of the bulk mail
String	msgText	The message text
String	cc	One or more addresses seperated by a comma
String	bcc	One or more addresses seperated by a comma
Attachment	attachment	A single attachment
String	smtpHost	The smtp host

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var attachment = plugins.mail.createBinaryAttachment('embedded',plugins.file.readFile('c:/temp/a_logo.gif'));
var msgText = 'plain msg<html>styled html msg</html>';
var smtpHost = 'myserver.com';
var success = plugins.mail.sendBulkMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>', 'subject', msgText,null,'unnamed@example.com',attachment,smtpHost);
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send bulk mail','OK');
}
```

sendBulkMail(to, from, subject, msgText, cc, bcc, attachment, overrideProperties)

Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

A bulk email makes it possible for one to not receive "out of office" emails back from receiver.

Parameters

String	to	A string containing 1 or multiple addresses seperated by a comma.
String	from	A string containing an address and an optional reply address, seperated by a comma.
String	subject	The subject of the bulk mail
String	msgText	The message text
String	cc	One or more addresses seperated by a comma
String	bcc	One or more addresses seperated by a comma
Attachment	attachment	A single attachment
Array	overrideProperties	An array of properties

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var attachment = plugins.mail.createBinaryAttachment('embedded',plugins.file.readFile('c:/temp/a_logo.gif'));
var msgText = 'plain msg<html>styled html msg</html>';
//it is possible to set all kind of smtp properties
var properties = new Array()
properties[0] = 'mail.smtp.host=myserver.com'
// properties specification can be found at:https://javaee.github.io/javamail/docs/api/com/sun/mail/smtp/package-summary.html
var success = plugins.mail.sendBulkMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>', 'subject', msgText,null,'unnamed@example.com',attachment,properties);
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send bulk mail','OK');
}
```

sendBulkMail(to, from, subject, msgText, cc, bcc, attachments)

Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

A bulk email makes it possible for one to not receive "out of office" emails back from receiver.

Parameters

String to	A string containing 1 or multiple addresses seperated by a comma.
String from	A string containing an address and an optional reply address, seperated by a comma.
String subject	The subject of the bulk mail
String msgText	The message text
String cc	One or more addresses seperated by a comma
String bcc	One or more addresses seperated by a comma
Array attachments	The attachments

Returns

[Boolean](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var attachment1 = plugins.mail.createBinaryAttachment('embedded',plugins.file.readFile('c:/temp/a_logo.gif'));
var attachment2 = plugins.mail.createTextAttachment('embedded','A text attachment');
var msgText = 'plain msg<html>styled html msg</html>';
var success = plugins.mail.sendBulkMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>', 'subject', msgText,null,'bcc1@example.com,bcc2@example.com',[attachment1,attachment2]);
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send bulk mail','OK');
}
```

sendBulkMail(to, from, subject, msgText, cc, bcc, attachments, smtpHost)

Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

A bulk email makes it possible for one to not receive "out of office" emails back from receiver.

Parameters

String to	A string containing 1 or multiple addresses seperated by a comma.
String from	A string containing an address and an optional reply address, seperated by a comma.
String subject	The subject of the bulk mail
String msgText	The message text
String cc	One or more addresses seperated by a comma
String bcc	One or more addresses seperated by a comma
Array attachments	The attachments
String smtpHost	The smtp host

Returns

[Boolean](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var attachment1 = plugins.mail.createBinaryAttachment('embedded',plugins.file.readFile('c:/temp/a_logo.gif'));
var attachment2 = plugins.mail.createTextAttachment('embedded','A text attachment');
var msgText = 'plain msg<html>styled html msg</html>';
var smtpHost = 'myserver.com';
var success = plugins.mail.sendBulkMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>', 'subject', msgText,null,'unnamed@example.com',[attachment1,attachment2],smtpHost);
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send bulk mail','OK');
}
```

sendBulkMail(to, from, subject, msgText, cc, bcc, attachments, overrideProperties)

Send a bulk mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

A bulk email makes it possible for one to not receive "out of office" emails back from receiver.

Parameters

String to	A string containing 1 or multiple addresses seperated by a comma.
String from	A string containing an address and an optional reply address, seperated by a comma.
String subject	The subject of the bulk mail
String msgText	The message text
String cc	One or more addresses seperated by a comma
String bcc	One or more addresses seperated by a comma
Array attachments	The attachments
Array overrideProperties	An array of properties

Returns[Boolean](#)**Supported Clients**

SmartClient,WebClient,NGClient

Sample

```
var attachment1 = plugins.mail.createBinaryAttachment('embedded',plugins.file.readFile('c:/temp/a_logo.gif'));
var attachment2 = plugins.mail.createTextAttachment('embedded','A text attachment');
var msgText = 'plain msg<html>styled html msg</html>';
//it is possible to set all kind of smtp properties
var properties = new Array()
properties[0] = 'mail.smtp.host=myserver.com'
// properties specification can be found at:https://javaee.github.io/javamail/docs/api/com/sun/mail/smtp/package-summary.html
var success = plugins.mail.sendBulkMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>', 'subject', msgText,null,'unnamed@example.com',[attachment1,attachement2],properties);
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send bulk mail','OK');
}
```

sendMail(to, from, subject, msgText)

Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

Parameters

String to	A string containing 1 or multiple addresses seperated by a comma.
String from	A string containing an address and an optional reply address, seperated by a comma.
String subject	The subject of the mail
String msgText	The message text

Returns[Boolean](#)**Supported Clients**

SmartClient,WebClient,NGClient

Sample

```
var msgText = 'plain msg<html>styled html msg</html>';
var success = plugins.mail.sendMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>,replyTo@example.com', 'subject', msgText);
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send mail','OK');
}
```

sendMail(to, from, subject, msgText, cc)

Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

Parameters

String to	A string containing 1 or multiple addresses seperated by a comma.
String from	A string containing an address and an optional reply address, seperated by a comma.
String subject	The subject of the mail
String msgText	The message text
String cc	One or more addresses seperated by a comma

Returns[Boolean](#)**Supported Clients**

SmartClient,WebClient,NGClient

Sample

```
var msgText = 'plain msg<html>styled html msg</html>';
var success = plugins.mail.sendMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>', 'subject', msgText,'cc1@example.com,cc2@example.com');
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send mail','OK');
}
```

sendMail(to, from, subject, msgText, cc, bcc)

Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

Parameters

[String](#) to A string containing 1 or multiple addresses seperated by a comma.
[String](#) from A string containing an address and an optional reply address, seperated by a comma.
[String](#) subject The subject of the mail
[String](#) msgText The message text
[String](#) cc One or more addresses seperated by a comma
[String](#) bcc One or more addresses seperated by a comma

Returns[Boolean](#)**Supported Clients**

SmartClient,WebClient,NGClient

Sample

```
var msgText = 'plain msg<html>styled html msg</html>';
var success = plugins.mail.sendMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>', 'subject', msgText,null,'bcc1@example.com');
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send mail','OK');
}
```

sendMail(to, from, subject, msgText, cc, bcc, attachment)

Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

Parameters

[String](#) to A string containing 1 or multiple addresses seperated by a comma.
[String](#) from A string containing an address and an optional reply address, seperated by a comma.
[String](#) subject The subject of the mail
[String](#) msgText The message text
[String](#) cc One or more addresses seperated by a comma
[String](#) bcc One or more addresses seperated by a comma
[Attachment](#) attachment A single attachment

Returns[Boolean](#)**Supported Clients**

SmartClient,WebClient,NGClient

Sample

```
var attachment = plugins.mail.createBinaryAttachment('embedded',plugins.file.readFile('c:/temp/a_logo.gif'));
var msgText = 'plain msg<html>styled html msg</html>';
var success = plugins.mail.sendMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>', 'subject', msgText,null,'bcc1@example.com,bcc2@example.com',attachment);
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send mail','OK');
}
```

sendMail(to, from, subject, msgText, cc, bcc, attachment, smtpHost)

Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

Parameters

String	to	A string containing 1 or multiple addresses seperated by a comma.
String	from	A string containing an address and an optional reply address, seperated by a comma.
String	subject	The subject of the mail
String	msgText	The message text
String	cc	One or more addresses seperated by a comma
String	bcc	One or more addresses seperated by a comma
Attachment	attachment	A single attachment
String	smtpHost	The smtp host

Returns

[Boolean](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var attachment = plugins.mail.createBinaryAttachment('embedded',plugins.file.readFile('c:/temp/a_logo.gif'));
var msgText = 'plain msg<html>styled html msg</html>';
var smtpHost = 'myserver.com';
var success = plugins.mail.sendMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>', 'subject', msgText,null,'unnamed@example.com',attachment,smtpHost);
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send mail','OK');
}
```

sendMail(to, from, subject, msgText, cc, bcc, attachment, overrideProperties)

Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

Parameters

String	to	A string containing 1 or multiple addresses seperated by a comma.
String	from	A string containing an address and an optional reply address, seperated by a comma.
String	subject	The subject of the mail
String	msgText	The message text
String	cc	One or more addresses seperated by a comma
String	bcc	One or more addresses seperated by a comma
Attachment	attachment	A single attachment
Array	overrideProperties	An array of properties

Returns

[Boolean](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var attachment = plugins.mail.createBinaryAttachment('embedded',plugins.file.readFile('c:/temp/a_logo.gif'));
var msgText = 'plain msg<html>styled html msg</html>';
//it is possible to set all kind of smtp properties
var properties = new Array()
properties[0] = 'mail.smtp.host=myserver.com'
// properties specification can be found at:https://javaee.github.io/javamail/docs/api/com/sun/mail/smtp/package-summary.html
var success = plugins.mail.sendMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>', 'subject', msgText,null,'unnamed@example.com',attachment,properties);
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send mail','OK');
}
```

sendMail(to, from, subject, msgText, cc, bcc, attachments)

Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

Parameters

String to	A string containing 1 or multiple addresses seperated by a comma.
String from	A string containing an address and an optional reply address, seperated by a comma.
String subject	The subject of the mail
String msgText	The message text
String cc	One or more addresses seperated by a comma
String bcc	One or more addresses seperated by a comma
Array attachments	The attachments

Returns

[Boolean](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```
var attachment1 = plugins.mail.createBinaryAttachment('embedded',plugins.file.readFile('c:/temp/a_logo.gif'));
var attachment2 = plugins.mail.createTextAttachment('embedded','A text attachment');
var msgText = 'plain msg<html>styled html msg</html>';
var success = plugins.mail.sendMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>', 'subject', msgText,null,'bcc1@example.com,bcc2@example.com',[attachment1,attachment2]);
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send mail','OK');
}
```

sendMail(to, from, subject, msgText, cc, bcc, attachments, smtpHost)

Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

Parameters

String to	A string containing 1 or multiple addresses seperated by a comma.
String from	A string containing an address and an optional reply address, seperated by a comma.
String subject	The subject of the mail
String msgText	The message text
String cc	One or more addresses seperated by a comma
String bcc	One or more addresses seperated by a comma
Array attachments	The attachments
String smtpHost	The smtp host

Returns

[Boolean](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```

var attachment1 = plugins.mail.createBinaryAttachment('embedded',plugins.file.readFile('c:/temp/a_logo.gif'));
var attachment2 = plugins.mail.createTextAttachment('embedded','A text attachment');
var msgText = 'plain msg<html>styled html msg</html>';
var smtpHost = 'myserver.com';
var success = plugins.mail.sendMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>', 'subject', msgText,null,'unnamed@example.com',[attachment1,attachment2],smtpHost);
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send mail','OK');
}

```

sendMail(to, from, subject, msgText, cc, bcc, attachments, overrideProperties)

Send a mail, if you make the msgText start with <html> the message will be sent in html (and you can use all html formatting).

Parameters

String to	A string containing 1 or multiple addresses separated by a comma.
String from	A string containing an address and an optional reply address, separated by a comma.
String subject	The subject of the mail
String msgText	The message text
String cc	One or more addresses separated by a comma
String bcc	One or more addresses separated by a comma
Array attachments	The attachments
Array overrideProperties	An array of properties

Returns

[Boolean](#)

Supported Clients

SmartClient,WebClient,NGClient

Sample

```

var attachment1 = plugins.mail.createBinaryAttachment('embedded',plugins.file.readFile('c:/temp/a_logo.gif'));
var attachment2 = plugins.mail.createTextAttachment('embedded','A text attachment');
var msgText = 'plain msg<html>styled html msg</html>';
//it is possible to set all kind of smtp properties
var properties = new Array()
properties[0] = 'mail.smtp.host=myserver.com'
// properties specification can be found at:https://javaee.github.io/javamail/docs/api/com/sun/mail/smtp/package-summary.html
var success = plugins.mail.sendMail('to_someone@example.com,to_someone_else@example.net', 'John Cobb
<from_me@example.com>', 'subject', msgText,null,'unnamed@example.com',[attachment1,attachment2],properties);
if (!success)
{
    plugins.dialogs.showWarningDialog('Alert','Failed to send mail','OK');
}

```