api

email API

send and receive emails through your website or application

print this page

description

Download this document

API description

email API	
summary	send and receive emails through your website or IT system
level of difficulty	3/5
estimated time	30 minutes
request method	HTTP (REST – RPC)
category	internet
price and reward	free use (with limitations) in alpha mode Go >

Use this API to create your very own email application on the Web or for integration into your IT systems. You can:

send and receive emails

manage your inbox

count and list your emails

The API is not reserved exclusively for Orange customers. It could be used by anyone.

2. which application?

send emails from a business application

combine the email API with the SMS or MMS API to stay in closer contact with your customers

All our APIs can be combined to create applications: Get inspiration from our mash-up rubic

documentation

Download this document

API Documentation

**The email API is a web service which allows you to send and receive emails.
Orange provides you with a 500 MB.**

The requests use GET or POST in HTTP or HTTPS mode.
The service exposes a REST-RPC style interface.

1. methods

a. sendMail : sends an email

The sendMail method is a simple function which allows you to send an email using HTTP request.

Have a look at the tutorials in which the sendSMS method is used

email API
sendMail
request
send email
send email

Each API call, information about the call is sent in XML format

```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
     <status>
          <status_code>200</status_code>
          <status_msg>ok</status_msg>
     </status>
     <call_info>
          <call_id>3DhG5wUla1nVtWArwKm2</call_id>
     </call_info>
     <data></data>
</response>
```

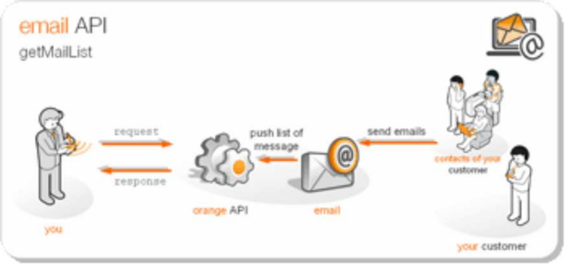
Parameters details available here >

b. getMailList : read email method

The GetMailList method provides a list of all e-mails from the inbox.

Each e-mail is assigned a number in chronological order. Remember that this is not the unique identifier to the e-mail.

Have a look at the tutorials in which the sendSMS method is used



request

```
http://mail.alpha.orange-api.net/mail/getMailList.xml?id=[api access key]
```

Parameters details available here >

response

Each API call, information about the call is sent in XML format

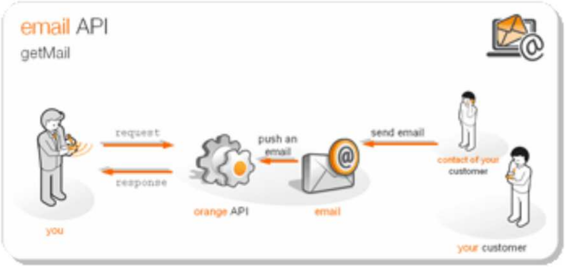
```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
     <status>
          <status_code>200</status_code>
          <status_msg>ok</status_msg>
     </status>
     <nbMsg>2</nbMsg>
     <list>
          <message>
               <number>1</number>
               <from><![CDATA[sender's email address]]></from>
               <subject><![CDATA[email subject]]></subject>
               <size>123</size>
               <date>2007-12-05T11:08:27+01:00</date>
               <new>true</new>
               <recent>false</recent>
               <unread>true</unread>
               <answered>false</answered>
               <flagged>false</flagged>
               <deleted>false</deleted>
               <draft>false</draft>
          </message>
          <message>
               <number>2</number>
               <from><![CDATA[sender's email address]]></from>
               <subject><![CDATA[email subject]]></subject>
               <size>334</size>
               <date>2007-12-05T11:08:28+01:00</date>
               <new>false</new>
               <recent>false</recent>
               <unread>false</unread>
               <answered>true</answered>
               <flagged>false</flagged>
               <deleted>false</deleted>
               <draft>false</draft>
          </message>
     </list>
     </call_info>
          <call_id>3DhG5wUla1nVtWArwKm2</call_id>
     </call_info>
     <data></data>
</response>
```

Parameters details available here >

c. getMail : read email method

The getMail method provides the content of an email

Have a look at the tutorials in which the sendSMS method is used



request

```
http://mail.alpha.orange-api.net/mail/getMail.xml?id=[API access key]&number=
[number of the email from the list]
```

Parameters details available here >

response

Each API call, information about the call is sent in XML format

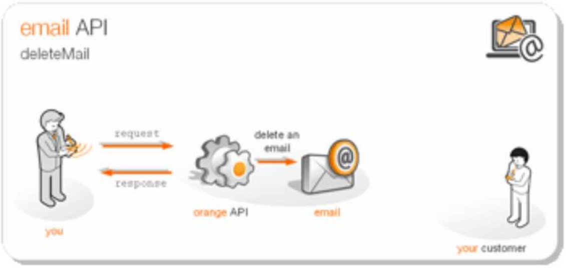
```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
     <status>
          <status_code>200</status_code>
          <status_msg>ok</status_msg>
     </status>
     <nbMsg>1</nbMsg>
     <message>
          <number>1</number>
          <toAddress><![CDATA[receiver's email address]]></toAddress>
          <from><![CDATA[sender's email address]]></from>
          <cc></cc>
          <replyTo><![CDATA[email address to respond to]]></replyTo>
          <from></from>
          <subject><![CDATA[email subject]]></subject>
          <size>123</size>
          <new>true</new>
          <recent>false</recent>
          <unread>true</unread>
          <answered>false</answered>
          <flagged>false</flagged>
          <deleted>false</deleted>
          <draft>false</draft>
          <date>2007-12-05T11:08:27+01:00</date>
          <body><![CDATA[email body]]></body>
     </message>
     <call_info>
          <call_id>3DhG5wUla1nVtWArwKm2</call_id>
     </call_info>
     <data></data>
</response>
```

Parameters details available here >

d. deleteMail : manage inbox method

This method deletes an email from your inbox

Have a look at the tutorials in which the sendSMS method is used



request

```
http://mail.alpha.orange-api.net/mail/deleteMail.xml?id=[API access
key]&number=[email number]
```

Parameters details available here >

response

Each API call, information about the call is sent in XML format

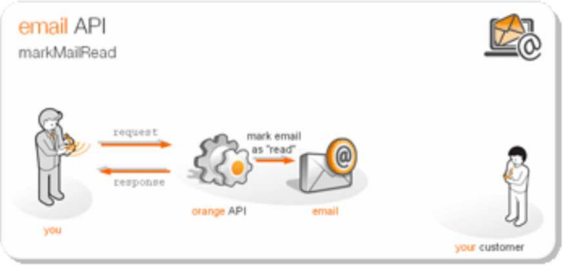
```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
     <status>
          <status_code>200</status_code>
          <status_msg>ok</status_msg>
     </status>
     <call_info>
          <call_id>3DhG5wUla1nVtWArwKm2</call_id>
     </call_info>
     <data></data>
</response>
```

Parameters details available here >

e. markMailRead : manage inbox method

This method marks an email in your inbox as "read".

Have a look at the tutorials in which the sendSMS method is used



requête

```
http://mail.alpha.orange-api.net/mail/markMailRead.xml?id=[API access
key]&number=[email number]
```

Parameters details available here >

Response

Each API call, information about the call is sent in XML format

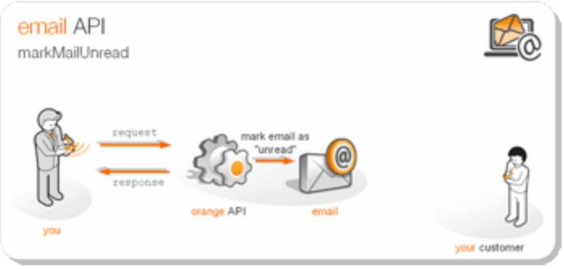
```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
     <status>
          <status_code>200</status_code>
          <status_msg>ok</status_msg>
     </status>
     <call_info>
          <call_id>3DhG5wUla1nVtWArwKm2</call_id>
     </call_info>
     <data></data>
</response>
```

Parameters details available here >

f. markMailUnread : manage inbox method

This method marks an email in your inbox as "not read".

Have a look at the tutorials in which the sendSMS method is used



request

```
http://mail.alpha.orange-api.net/mail/markMailUnread.xml?id=[API access
key]&number=[email number from the list]
```

Parameters details available here >

response

Each API call, information about the call is sent in XML format

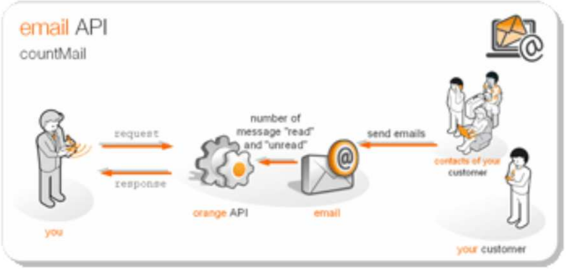
```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
     <status>
          <status_code>200</status_code>
          <status_msg>ok</status_msg>
     </status>
     <call_info>
          <call_id>3DhG5wUla1nVtWArwKm2</call_id>
     </call_info>
     <data></data>
</response>
```

Parameters details available here >

g. countMail : manage inbox method

This message returns a total of the numbers of emails and unread email in your inbox.

Have a look at the tutorials in which the sendSMS method is used



requête

```
http://mail.alpha.orange-api.net/mail/countMail.xml?id=[API access key]
```

Parameters details available here >

response

Each API call, information about the call is sent in XML format

```
<?xml version="1.0" encoding="UTF-8" ?>
<response>
     <status>
          <status_code>200</status_code>
          <status_msg>ok</status_msg>
     </status>
     <nbMsg>2</nbMsg>
     <nbUnreadMsg>1</nbUnreadMsg>
     <call_info>
          <call_id>3DhG5wUla1nVtWArwKm2</call_id>
     </call_info>
     <data></data>
</response>
```

Parameters details available here >

2. code example

send and receive emails through your website in few code lines !

1. copy/paste this code in a PHP file.
2. fill in the ACCESS KEY, FROM, TO et CONTENT parametres.
3. download the file from your web plateform.
4. execute your page.

```
<?php

// your API Access Key
$access_key = "XXXXXXX";

// URL to invoke the sendMail method of the email API
$adresse = "http://mail.beta.orange-api.net/mail/sendMail.xml";

// sender's name
$fromName = ""your name here"";
// recipient email address
$to = ""recipient@domain.com";
// subject of the email
$subject = urlencode(substr("ajoutez ici l'objet de votre email"", 0, 159));

// body of the email
$body = urlencode(substr(""ajoutez ici tout le corps du message, vous pouvez
également ajouter des retours ", 0, 159));

// execute the request > send the email
$response = file_get_contents($adresse . "?id=" . $access_key . "&from=" .
$fromName . "&to=" . $to . "&subject=" . $subject); . "&body=" . $body);

// get and display the status response message from the API call
$xml=simplexml_load_string($response);
echo($xml->status->status_code . " - " . $xml->status->status_msg);

?>
```

cost

Download this document

API pricing

How the use of the email API is charged for:

the email API is free

the number of requests per user and per day is restricted as per the table below.

restrictions

method	**limitation**
sendMail	500 requests/day
getMailList	150 requests/day
getMail	500 requests/day
deleteMail	1000 requests/day
markMailRead	1000 requests/day
markMailUnread	1000 requests/day
countMail	500 requests/day