Email Introduction

This documentation will help you get started using email API

Email nowadays plays an essential part in everyone’s daily business communication, and it is important to have it included in your business applications as well, either as a standalone service, or as part of our OMNI solution.

In this brief introduction, we will provide instructions on how to set up everything that will enable you to send emails over our platform in no time.

Our email service will allow you to send HTML emails, add attachments and view delivery reports to reach an ever-increasing number of your clients directly, through our API, as explained in the next steps.

Choose your domain carefully

The most important step in this initial stage is choosing the way you would like to route your emails.

Individuals and organizations often use email as their primary means of communication. However, most users recently have started complaining about their inboxes being flooded with unsolicited bulk emails or spam emails. No technique is a complete solution to the spam problem, and each has trade-offs between incorrectly rejecting legitimate email (false positives) and not rejecting all spam (false negatives). This is why the domain reputation rating system was introduced - recognizing potential threats for emails sent out from the domain, based on what was sent from it in the past.

Since the domain reputation is affected by the type of emails being sent from it, once the domain is classified as 'bad' (possibly due to a generally negative impact from sending bulk type emails) - it is very hard to restore its reputation fully.

It may be a good idea to separate the domains based on the type of messages that you are planning to send. For example: use different domains/subdomains for each type of email that you are intending to send (transactional, corporate, marketing, etc.) to keep the reputation levels separate.

Register your domain with us

For this step, you need to have access to your domain's DNS configuration. Please, contact us with your preferred domain so we can integrate it within our platform. You will receive additional instructions (DNS configuration) on how to establish a proper communication channel between your domain and our platform.

Once the configuration is properly applied, you can follow the steps outlined in the following pages to successfully send emails.

If you have any questions regarding the service integration, please contact us.

Next: Send simple email

Simple email

This method allows you to send a single email message to one destination address.

Try It

post/email/:version/send

- cURL
- HTTP
- PHP
- JavaScript
- Java
- C#

```
curl -s --user user:password \
https://107.20.199.106/email/1/send \
-F from='Jane Doe <jane.doe@somecompany.com>' \
-F to='john.smith@somedomain.com' \
-F subject='Mail subject text' \
-F text='Mail body text' \
-F bulkId='cusotmBulkId'
```

200 OK


```
{
  "bulkId":"cusotmBulkId",
  "messages": [
    {
      "to": "john.smith@somedomain.com",
      "messageCount": 1,
      "messageId": "c268350e-c85e-41d1-b5a0-a60771b134bd",
      "status": {
        "groupId": 1,
        "groupName": "PENDING",
        "id": 7,
        "name": "PENDING_ENROUTE",
```

```
        "description": "Message sent to next instance"
      }
    }
  ]
}
```

BODY PARAMS

from
string
Email address with optional sender name. Example: "Jane Doe <jane.doe@somecompany.com>"

to
string
Email address of the recipient

subject
string
Message subject

text
string
Body of the message

bulkId
string
The ID uniquely identifies the sent Email request. This filter will enable you to query delivery reports for all the messages using just one request. You will receive a bulkId in the response after sending an Email request. If you don't set your own `bulkId`, unique ID will be generated by our system and returned in the API response.

Response format

On success, response header HTTP status code will be `200 OK` and the message will be sent.

If you try to send message without authorization, you will receive an error `401 Unauthorized`.

Response

Parameter	Type	Description
bulkId	String	Either a user provided custom bulk id or an auto generated one in case bulkId was absent in the request.
messages	MessageInfo[]	Array of sent message objects, one object per every message.

MessageInfo

Parameter	Type	Description
to	String	The message destination address.
status	Status	Indicates whether the message is successfully sent, not sent, delivered, not delivered, waiting for delivery or any other possible status.
messageCount	int	The number of sent message segments.
messageId	String	The ID that uniquely identifies the message sent.

Status

Parameter	Type	Description
groupId	int	Status group ID.
groupName	String	Status group name.
id	int	Status ID.
name	String	Status name.
description	String	Human readable description of the status.
action	String	Action that should be taken to eliminate the error.

Next: Fully featured email

Fully featured email

This method allows you to send one or more email message with attachments to one or more destination addresses.

Try It

post/email/:version/send

- cURL
- HTTP
- PHP

- JavaScript
- Java
- C#

```
curl -s --user user:password \
https://107.20.199.106/email/1/send \
-F from='Jane Smith <jane.smith@somecompany.com>' \
-F to='john.smith@somedomain.com' \
-F replyTo='all.replies@somedomain.com' \
-F subject='Mail subject text' \
-F text='Mail body text' \
--form-string html='<h1>Html body</h1><p>Rich HTML message body.</p>' \
-F attachment=@files/image1.jpg \
-F bulkId='cusotmBulkId' \
-F intermediateReport='true' \
-F notifyUrl='http://www.example.com/email/advanced' \
-F notifyContentType = 'application/json' \
-F callbackData = 'DLR callback data'
```

200 OK


```
{
  "bulkId": "cusotmBulkId",
  "messages": [
    {
      "to": "jane.smith@somecompany.com",
      "messageCount": 1,
      "messageId": "c268350e-c85e-41d1-b5a0-a60771b134bd",
      "status": {
        "groupId": 1,
        "groupName": "PENDING",
        "id": 7,
        "name": "PENDING_ENROUTE",
        "description": "Message sent to next instance"
      }
    }
  ]
}
```


BODY PARAMS

from
string
Email address with optional sender name. Example: "Jane Smith <jane.smith@somecompany.com>"

to
string
Email address of the recipient

replyTo
string
Email address to which recipients of the email can reply

subject
string
Message subject

text
string
Body of the message

html
string
HTML body of the message. If html and text fields are present, text field will be ignored and html will be delivered as message body

attachment
file
File attachment

bulkId
string
The ID uniquely identifies the sent Email request. This filter will enable you to query delivery reports for all the messages using just one request. You will receive a bulkId in the response after sending an Email request. If you don't set your own `bulkId`, unique ID will be generated by our system and returned in the API response.

intermediateReport
boolean
The real-time Intermediate delivery report that will be sent on your callback server. Can be `true` or `false`.

notifyUrl
string
The URL on your callback server on which the Delivery report will be sent.

notifyContentType
string
Preferred Delivery report content type. Can be `application/json` or `application/xml`.

callbackData
string
Additional client's data that will be sent on the notifyUrl. The maximum value is 200 characters.

Response format

On success, response header HTTP status code will be `200 OK` and the message will be sent.

If you try to send message without authorization, you will receive an error `401 Unauthorized`.

Response

Parameter	Type	Description
bulkId	String	Either a user provided custom bulk id or an auto generated one in case bulkId was absent in the request.
messages	MessageInfo[]	Array of sent message objects, one object per every message.
rejectedMessages	RejectedMessageInfo[]	Array of rejected message objects, one object per every message.

MessageInfo

Parameter	Type	Description
to	String	The message destination address.
status	Status	Indicates whether the message is successfully sent, not sent, delivered, not delivered, waiting for delivery or any other possible status.
messageCount	int	The number of sent message segments.
messageId	String	The ID that uniquely identifies the message sent.

RejectedMessageInfo

Parameter	Type	Description
recipient	String	The string containing rejected email address.
reason	String	Indicates the reason for rejection of the email message.

Status

Parameter	Type	Description
groupId	int	Status group ID.
groupName	String	Status group name.
id	int	Status ID.

Parameter	Type	Description
name	String	Status name.
description	String	Human readable description of the status.
action	String	Action that should be taken to eliminate the error.

Send email to multiple recipients

You can easily send the same email to multiple recipient addresses by including more then one **to** parameters in the request

Example

Request:

- **cURL**
- HTTP
- PHP
- JavaScript
- Java

```
curl -s --user user:password \
https://107.20.199.106/email/1/send \
-F from='Jane Smith <jane.smith@somecompany.com>' \
-F to='john.smith@somedomain.com' \
-F to='tom.smith@somedomain.com' \
-F to='invalid,@example.com' \
-F subject='Mail subject text' \
-F text='Mail body text' \
--form-string html='<h1>Html body</h1><p>Rich HTML message body.</p>' \
-F attachment=@files/image1.jpg
```

Response:

- **JSON**


```
{
  "bulkId": "lrzkq6gatdkxouhrkgni",
  "messages": [
    {
      "to": "john.smith@somecompany.com",
      "messageCount": 1,
      "messageId": "9129e972-be85-49ed-b0e9-f51fbd2b27c0",
      "status": {
        "groupId": 1,
        "groupName": "PENDING",
        "id": 7,
        "name": "PENDING_ENROUTE",
        "description": "Message sent to next instance"
      }
    },
    {
      "to": "tom.smith@somecompany.com",
      "messageCount": 1,
      "messageId": "cd0c5682-42e2-481a-be95-15ed72450646",
      "status": {
        "groupId": 1,
        "groupName": "PENDING",
        "id": 7,
        "name": "PENDING_ENROUTE",
        "description": "Message sent to next instance"
      }
    }
  ],
  "rejectedMessages": [
                {
      "recipient":"invalid,@example.com",
        "reason":"Invalid email address."
    }
        ]
}
```


Send email with multiple attachments

Sending email with more than one attachment is done by adding multiple attachment parameters in the request.

Example

Request:

- **cURL**

- HTTP

- PHP

- JavaScript

- Java

```
curl -s --user user:password \
https://107.20.199.106/email/1/send \
-F from='Jane Smith <jane.smith@somecompany.com>' \
-F to='john.smith@somedomain.com' \
-F subject='Mail subject text' \
-F text='Mail body text' \
--form-string html='<h1>Html body</h1><p>Rich HTML message body.</p>' \
-F attachment=@files/image1.jpg
-F attachment=@files/image2.jpg
```

Response:

- **JSON**


```
{
  "messages": [
    {
      "to": "john.smith@somecompany.com",
      "messageCount": 1,
      "messageId": "c268350e-c85e-41d1-b5a0-a60771b134bd",
      "status": {
        "groupId": 1,
        "groupName": "PENDING",
        "id": 7,
        "name": "PENDING_ENROUTE",
        "description": "Message sent to next instance"
      }
    }
```

```
  ]
}
```

Email delivery reports

This method allows you to get one time delivery reports for sent e-mails.

Try It

get /email/:version/reports

- JSON

```
GET /email/1/reports HTTP/1.1
Host: 107.20.199.106
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Accept: application/json
```

200 OK


```
{
  "results": [
    {
      "messageId": "bbcc6960-1fcb-497c-b7ea-83ccba41492e",
      "to": "recipient@example.com",
      "sentAt": "2016-08-31T13:25:18.477+0000",
      "doneAt": "2016-08-31T13:25:50.893+0000",
      "messageCount": 1,
      "price": {
        "pricePerMessage": 0,
        "currency": "UNKNOWN"
      },
      "status": {
        "groupId": 3,
        "groupName": "DELIVERED",
        "id": 5,
        "name": "DELIVERED_TO_HANDSET",
        "description": "Message delivered to handset"
      },
      "error": {
        "groupId": 0,
        "groupName": "OK",
        "id": 0,
        "name": "NO_ERROR",
        "description": "No Error",
        "permanent": false
      },
      "channel": "EMAIL"
    }
  ]
}
```


QUERY PARAMS

bulkId
string
The ID uniquely identifies a group of Email requests. This filter will enable you to query delivery reports for all the messages with the same bulk id using just one request.

messageId
string

Message ID for which report is requested

limit
string
Maximum number of reports

Response format

On success, response header HTTP status code will be `200 OK` and delivery reports will be returned in the response body.

If you try to send a message without authorization, you will get a response with HTTP status code `401 Unauthorized`.

Parameter	Type	Description
results	SentEmailReport[]	Collection of reports, one per e-mail.

SentEmailReport

Parameter	Type	Description
messageId	String	Message ID.
to	String	Destination address.
sentAt	Date	Tells when the e-mail was sent. Has the following format: `yyyy-MM-dd'T'HH:mm:ss.SSSZ`.
doneAt	Date	Tells when the e-mail was finished processing by us(ie. delivered to destination)
messageCount	int	How many parts the message was split into. Always will be 1 for e-mail.
price	Price	Sent e-mail price.
status	Status	Indicates whether the message is successfully sent, not sent, delivered, not delivered, waiting for delivery or any other possible status.
error	Error	Indicates whether any error occurred during query execution.

Price

Parameter	Type	Description
pricePerMessage	BigDecimal	Price per one Email.
currency	String	The currency in which the price is expressed.

Status

Parameter	Type	Description
groupId	int	Status group ID
groupName	String	Status group name.
id	int	Status ID
name	String	Status name.
description	String	Human readable description of the status.
action	String	Action that should be taken to eliminate the error.

Error

Parameter	Type	Description
groupId	int	Error group ID
groupName	String	Error group name
id	int	Error ID
name	String	Error name
description	String	Human readable description of the error.
permanent	boolean	Tells if the error is permanent

Delivery report will be returned only once!

Delivery reports are returned **only once**. Additional delivery report request will return empty collection.

Additional examples

Get reports by message id

Request:

- **JSON**

- XML

- cURL

- PHP

- Ruby

- Python

- Java

- C#

- JavaScript

```
GET /email/1/reports?messageId=bbcc6960-1fcb-497c-b7ea-83ccba41492e HTTP/1.1
Host: 107.20.199.106
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Accept: application/json
```

Response:

- **JSON**

- XML

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "results": [
```

```
    {
      "messageId": "bbcc6960-1fcb-497c-b7ea-83ccba41492e",
      "to": "recipient@example.com",
      "sentAt": "2016-08-31T13:25:18.477+0000",
      "doneAt": "2016-08-31T13:25:50.893+0000",
      "messageCount": 1,
      "price": {
        "pricePerMessage": 0,
        "currency": "UNKNOWN"
      },
      "status": {
        "groupId": 3,
        "groupName": "DELIVERED",
        "id": 5,
        "name": "DELIVERED_TO_HANDSET",
        "description": "Message delivered to handset"
      },
      "error": {
        "groupId": 0,
        "groupName": "OK",
        "id": 0,
        "name": "NO_ERROR",
        "description": "No Error",
        "permanent": false
      },
      "channel": "EMAIL"
    }
        ]
}
```


Get initial two delivery reports

Request:

- **JSON**

```
```

- XML

```
```

- cURL

```
```

- PHP

- Ruby
- Python
- Java
- C#
- JavaScript

```
GET /email/1/reports?limit=2 HTTP/1.1
Host: 107.20.199.106
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Accept: application/json
```

Response:

- **JSON**
- XML

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "results": [
    {
      "messageId": "bbcc6960-1fcb-497c-b7ea-83ccba41492e",
      "to": "recipient@example.com",
      "sentAt": "2016-08-31T13:25:18.477+0000",
      "doneAt": "2016-08-31T13:25:50.893+0000",
      "messageCount": 1,
      "price": {
        "pricePerMessage": 0,
        "currency": "UNKNOWN"
      },
```

```
      "status": {
        "groupId": 3,
        "groupName": "DELIVERED",
        "id": 5,
        "name": "DELIVERED_TO_HANDSET",
        "description": "Message delivered to handset"
      },
      "error": {
        "groupId": 0,
        "groupName": "OK",
        "id": 0,
        "name": "NO_ERROR",
        "description": "No Error",
        "permanent": false
      },
      "channel": "EMAIL"
    },
    {
      "messageId": "a3ee6933-1fcb-497c-b7ea-83cdda55543f",
      "to": "recipient2@example.com",
      "sentAt": "2016-08-31T13:25:19.455+0000",
      "doneAt": "2016-08-31T13:25:51.233+0000",
      "messageCount": 1,
      "price": {
        "pricePerMessage": 0,
        "currency": "UNKNOWN"
      },
      "status": {
        "groupId": 3,
        "groupName": "DELIVERED",
        "id": 5,
        "name": "DELIVERED_TO_HANDSET",
        "description": "Message delivered to handset"
      },
      "error": {
        "groupId": 0,
        "groupName": "OK",
        "id": 0,
        "name": "NO_ERROR",
        "description": "No Error",
        "permanent": false
      },
      "channel": "EMAIL"
    }
        ]
}
```

Get reports by bulkId

Request:

- **JSON**
- XML
- cURL
- PHP
- Ruby
- Python
- Java
- C#
- JavaScript

```
GET /email/1/reports?bulkId=lrzkq6gatdkxouhrkgni HTTP/1.1
Host: 107.20.199.106
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Accept: application/json
```

Response:

- **JSON**

- XML

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "results": [
    {
      "bulkId": "lrzkq6gatdkxouhrkgni",
      "messageId": "bbcc6960-1fcb-497c-b7ea-83ccba41492e",
      "to": "recipient@example.com",
      "sentAt": "2016-08-31T13:25:18.477+0000",
      "doneAt": "2016-08-31T13:25:50.893+0000",
      "messageCount": 1,
      "price": {
        "pricePerMessage": 0,
        "currency": "UNKNOWN"
      },
      "status": {
        "groupId": 3,
        "groupName": "DELIVERED",
        "id": 5,
        "name": "DELIVERED_TO_HANDSET",
        "description": "Message delivered to handset"
      },
      "error": {
        "groupId": 0,
        "groupName": "OK",
        "id": 0,
        "name": "NO_ERROR",
        "description": "No Error",
        "permanent": false
      },
      "channel": "EMAIL"
    },
    {
      "bulkId": "lrzkq6gatdkxouhrkgni",
      "messageId": "a3ee6933-1fcb-497c-b7ea-83cdda55543f",
      "to": "recipient2@example.com",
      "sentAt": "2016-08-31T13:25:19.455+0000",
      "doneAt": "2016-08-31T13:25:51.233+0000",
      "messageCount": 1,
      "price": {
        "pricePerMessage": 0,
        "currency": "UNKNOWN"
      },
      "status": {
        "groupId": 3,
```

```
        "groupName": "DELIVERED",
        "id": 5,
        "name": "DELIVERED_TO_HANDSET",
        "description": "Message delivered to handset"
      },
      "error": {
        "groupId": 0,
        "groupName": "OK",
        "id": 0,
        "name": "NO_ERROR",
        "description": "No Error",
        "permanent": false
      },
      "channel": "EMAIL"
    }
        ]
}
```


Email messages logs

This method allows you to get logs for sent e-mail.

Try It

get /email/:version/logs

- JSON

```
GET /email/1/logs HTTP/1.1
Host: 107.20.199.106
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Accept: application/json
```

200 OK


```
{
  "results": [
    {
      "messageId": "64c98929-f160-4e2c-b156-ca88cc733547",
      "to": "recipient@example.com",
      "from": "sender@example.com",
      "text": "Test text",
      "sentAt": "2016-09-01T10:29:00.440+0000",
      "doneAt": "2016-09-01T10:29:01.130+0000",
      "price": {
        "pricePerMessage": 0.0005,
        "currency": "EUR"
      },
      "status": {
        "groupId": 3,
        "groupName": "DELIVERED",
        "id": 5,
        "name": "DELIVERED_TO_HANDSET",
        "description": "Message delivered to handset"
      },
      "channel": "EMAIL"
    }
  ]
}
```

QUERY PARAMS

from
string
Sender ID that can be alphanumeric or numeric.

to
string
The message destination address.

bulkId
string
The ID uniquely identifies a group of Email requests. This filter will enable you to query delivery reports for all the messages with the same bulk id using just one request.

messageId
string
The ID that uniquely identifies the message sent.

generalStatus
string
Sent e-mail status group. Indicates whether the message is successfully sent, not sent, delivered, not delivered, waiting for delivery or any other possible status.

sentSince
date
Lower limit on date and time of sending Email.

sentUntil
date
Upper limit on date and time of sending Email

limit
int32
Maximal number of messages in returned logs. Default value is 50.

None of the query parameters is mandatory for this request. For filtering results any combination of parameters can be used. Some examples are shown below.

Important

E-mail logs are available for the last 48 hours!

Response format

If successful, response header HTTP status code will be `200 OK` and the message logs will be returned.

If you try to send message without authorization, you will get a response with HTTP status code `401 Unauthorized`.

If you are using this method too many times in a short period, you will get status code `429 Too Many Requests`. This prevents misusing logs in cases where reports would be more appropriate. For more information about when to use logs, please see the documentation.

SMSLogsResponse

Parameter	Type	Description
results	SentEmailLog[]	Collection of logs.

SentEmailLog

Parameter	Type	Description
messageId	String	The ID that uniquely identifies the message sent.
to	String	The message destination address.
from	String	Sender e-mail address.
text	String	Text of the message that was sent.
sentAt	Date	Tells when the e-mail was sent. Has the following format: `yyyy-MM-dd'T'HH:mm:ss.SSSZ`.
doneAt	Date	Tells when the e-mail was finished processing by us (i.e. delivered to destination)
messageCount	int	How many parts the message was split into. Always will be 1 for e-mail.
price	Price	Sent e-mail price.
status	Status	Indicates whether the message is successfully sent, not sent, delivered, not delivered, waiting for delivery or any other possible status.
error	Error	Indicates whether the error occurred during the query execution.

Price

Parameter	Type	Description
pricePerMessage	BigDecimal	Price per one e-mail.
currency	String	The currency in which the price is expressed.

Status

Parameter	Type	Description
groupId	int	Status group ID
groupName	String	Status group name.
id	int	Status ID
name	String	Status name.
description	String	Human readable description of the status.
action	String	Action that should be taken to eliminate the error.

Error

Parameter	Type	Description
groupId	int	Error group ID
groupName	String	Error group name
id	int	Error ID
name	String	Error name
description	String	Human readable description of the error.
permanent	boolean	Tells if the error is permanent

Get logs with multiple messageId filter

You may retrieve message logs by supplying comma-split message IDs to the endpoint.

Request:

- **JSON**

- XML

- cURL
- PHP
- Ruby
- Python
- Java
- C#
- JavaScript

```
GET /email/1/logs?messageId=64c98929-f160-4e2c-b156-ca88cc733547,60d586a1-6448-4c5f-
860d-be3ddbea16da HTTP/1.1
Host: 107.20.199.106
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Accept: application/json
```

Response:

- **JSON**
- XML


```
{
  "results": [
    {
      "messageId": "64c98929-f160-4e2c-b156-ca88cc733547",
      "to": "recipient@example.com",
      "from": "sender@example.com",
```

```
      "text": "Test text",
      "sentAt": "2016-09-01T10:29:00.440+0000",
      "doneAt": "2016-09-01T10:29:01.130+0000",
      "price": {
        "pricePerMessage": 0.0005,
        "currency": "EUR"
      },
      "status": {
        "groupId": 3,
        "groupName": "DELIVERED",
        "id": 5,
        "name": "DELIVERED_TO_HANDSET",
        "description": "Message delivered to handset"
      },
      "channel": "EMAIL"
    },
    {
      "messageId": "60d586a1-6448-4c5f-860d-be3ddbea16da",
      "to": "recipient@example.com",
      "from": "sender@example.com",
      "text": "Test text",
      "sentAt": "2016-09-01T10:29:00.317+0000",
      "doneAt": "2016-09-01T10:29:00.807+0000",
      "price": {
        "pricePerMessage": 0.0005,
        "currency": "EUR"
      },
      "status": {
        "groupId": 3,
        "groupName": "DELIVERED",
        "id": 5,
        "name": "DELIVERED_TO_HANDSET",
        "description": "Message delivered to handset"
      },
      "channel": "EMAIL"
    }
  ]
}
```


Get logs with from, to and limit filters

`from`, `to` and `limit` accept a single parameter which will be used to filter response message logs.

Request

- **JSON**
- XML
- cURL
- PHP
- Ruby
- Python
- Java
- C#
- JavaScript

```
GET /email/1/logs?from=sender@example.com&to=recipient@example.com&limit=1
Host: 107.20.199.106
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Accept: application/json
```

Response

- **JSON**
- XML

```
{
```

```
  "results": [
    {
      "messageId": "54ddb941-2566-46e0-802a-22bea5cf94bc",
      "to": "recipient@example.com",
      "from": "sender@example.com",
      "text": "Test text",
      "sentAt": "2016-09-01T11:14:44.453+0000",
      "doneAt": "2016-09-01T11:14:45.050+0000",
      "price": {
        "pricePerMessage": 0.0005,
        "currency": "EUR"
      },
      "status": {
        "groupId": 3,
        "groupName": "DELIVERED",
        "id": 5,
        "name": "DELIVERED_TO_HANDSET",
        "description": "Message delivered to handset"
      },
      "channel": "EMAIL"
    }
  ]
}
```


Get logs with date range and general status filters

`sentSince` and `generalStatus` accept a single parameter which will be used to filter response message logs.

Request:

- **JSON**

```
```

- XML

```
```

- cURL

```
```

- PHP

```
```

- Ruby

- Python

- Java

- C#

- JavaScript

```
GET /email/1/logs?sentSince=2016-08-22T17:42:05.390%2b01:00&generalStatus=DELIVERED
HTTP/1.1
Host: 107.20.199.106
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Accept: application/json
```

Response:

- **JSON**

- XML


```
{
  "results": [
    {
      "messageId": "54ddb941-2566-46e0-802a-22bea5cf94bc",
      "to": "recipient@example.com",
      "from": "sender@example.com",
      "text": "Test text",
      "sentAt": "2016-09-01T11:14:44.453+0000",
      "doneAt": "2016-09-01T11:14:45.050+0000",
      "price": {
        "pricePerMessage": 0.0005,
        "currency": "EUR"
      },
      "status": {
        "groupId": 3,
        "groupName": "DELIVERED",
```

```
        "id": 5,
        "name": "DELIVERED_TO_HANDSET",
        "description": "Message delivered to handset"
      },
      "channel": "EMAIL"
    }
  ]
}
```

Get logs by Bulk ID

Request:

- **JSON**

- HTTP

- cURL

- PHP

- Ruby

- Python

- Java

- C#

- JavaScript

```
GET /email/1/logs?bulkId=lrzkq6gatdkxouhrkgni HTTP/1.1
Host: 107.20.199.106
Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
Accept: application/json
```

Response:

- **JSON**

- XML


```
{
  "results": [
    {
      "bulkId": "lrzkq6gatdkxouhrkgni",
      "messageId": "64c98929-f160-4e2c-b156-ca88cc733547",
      "to": "recipient@example.com",
      "from": "sender@example.com",
      "text": "Test text",
      "sentAt": "2016-09-01T10:29:00.440+0000",
      "doneAt": "2016-09-01T10:29:01.130+0000",
      "price": {
        "pricePerMessage": 0.0005,
        "currency": "EUR"
      },
      "status": {
        "groupId": 3,
        "groupName": "DELIVERED",
        "id": 5,
        "name": "DELIVERED_TO_HANDSET",
        "description": "Message delivered to handset"
      },
      "channel": "EMAIL"
    },
    {
      "bulkId": "lrzkq6gatdkxouhrkgni",
      "messageId": "60d586a1-6448-4c5f-860d-be3ddbea16da",
      "to": "recipient@example.com",
      "from": "sender@example.com",
      "text": "Test text",
      "sentAt": "2016-09-01T10:29:00.317+0000",
      "doneAt": "2016-09-01T10:29:00.807+0000",
      "price": {
        "pricePerMessage": 0.0005,
        "currency": "EUR"
      },
```

```
      "status": {
        "groupId": 3,
        "groupName": "DELIVERED",
        "id": 5,
        "name": "DELIVERED_TO_HANDSET",
        "description": "Message delivered to handset"
      },
      "channel": "EMAIL"
    }
  ]
}
```