

sinch



Sinch E-Mail 365 – API Specification

Version 6.0 – February 2022

sinch.com



Revision History

Version	Date	Description
0.x	2015-03 -31	Baseline draft
1.0	2016-08 -05	Finalized version. Added bounce section and API request examples specific to email as channel for 1 st release. Some refinements.
1.1	2016-08 -05	Addition to bounces (SMTP error codes) table
1.2	2017-03-20	Addition of best sending time and reply to name features
1.3	2017-05-25	Addition of GET BOUNCE API
1.4	2017-07-11	Addition of Unsubscribe tag option.
1.5	2017-11-06	Addition of Header options and systemId / campaignId optional fields
1.6	2017-11-15	Addition of APP REJECTED-BLOCKED
1.6.1	2017-11-17	Addition of illustrative Complaint feedback example
1.6.2	2018-03-15	Addition of header for list-unsubscribe
1.7	2018-05-15	Deprecated Search Query, Added Limitations to Bounce Query, added Bounce Type information.



Removed references to other channels and edited some sections for clarity.

1.8	2018-07-18	Added list of HTTP Responses for the POST, Fixed example calls to get BOUNCES.
1.9	2018-09-04	Added Attachment capability and default list-unsubscribe mailto capability.
1.9.1	2018-09-18	Added more information on attachments.
1.9.2	2018-09-21	Changed example URLs
1.9.3	2018-11-02	Added ContentName as a requirement for attachments
1.9.4	2018-11-14	ContentId is optional, should not be set when attachments will use ContentName as a filename for attachments
1.9.5	2019-03-19	Added a limit for the number of attachments (20 attachments max)
1.9.6	2019-04-24	List-unsubscribe header is required for CSA certified IPs
2.0	2019-05-15	New parameter added to Get Bounce API - 'statusText' which allows grouping of the bounce results Email Subject is now a required field



2.1	2019-08-26	Removed references to un-used parameters
2.2	2019-10-11	Added note on CH_REJECTED SOFT_BOUNCES in the STATUS CODES table
2.3	2019-12-19	Table for 'channelStatusText' parameter response added
3.0	2020-04-12	URL for Ch_ERROR query
4.0	2020-06-09	Support for additional parameters to indicate processing type and email expiry time. Also support for cc and bcc fields to capture recipient email addresses
5.0	2020-08-01	Support for antivirus parameter which signals whether an email needs to go through the anti-virus scanning
6.0	2022-02-08	Change of the new unsubscribe email address for Sinch Email 365 API



Table of Tables

TABLE 1:GLOSSARY OF TERMS	11
TABLE 2: POST REQUEST OBJECT HIERARCHY	13
TABLE 3: POST REQUEST API PARAMETERS	17
TABLE 4: POST RESPONSE API PARAMETERS	18
TABLE 5: GET URL PARAMETERS	24
TABLE 6: GET BOUNCE URL PARAMETERS	28
TABLE 7: GET BOUNCE REQUEST PARAMETERS	30
TABLE 8: GET BOUNCE REPONSE OBJECT	30
TABLE 9: GET URL PARAMETERS	34
TABLE 10: GET REQUEST PARAMETERS	35
TABLE 11: GET RESPONSE OBJECT	36
TABLE 12: RESPONSE PARAMETERS FOR GET ERROR REQUEST	37
TABLE 13: LIST OF STATUS CODES	42



Table of Contents

1 Sinch E-Mail 365 API	7
2 Glossary of Terms	8
3 E-Mail 365 API Details	12
3.1 Security	12
3.2 API object reference	12
3.2.1 Post (Request)	12
3.3 Credentials	12
4 POST Request API	13
4.1 Sample Resource URL	13
4.2 Resource Information.....	13
4.3 Request API parameters.....	13
4.4 Response API parameters	17
4.5 Examples	19
4.5.1 Request (syntax only) to send EMAIL	19
4.5.2 Response to above request	20
4.5.3 Request to send EMAIL with attachment (syntax only)	21
5 GET notification Status by id	24
5.1 Sample Resource URL:	24
5.2 Resource Information.....	24
5.3 URL Parameters	24
5.4 Examples	25
5.4.1 Request to query status of notification.....	25
5.4.2 Response to status of notification.....	25
5.4.3 Request to query status notification with specific recipient	26
6 GET Bounce (specific status) API calls	28
6.1 Resource URL for bounces.....	28
6.2 Resource Information.....	28
6.3 URL Parameters	28
6.4 Request Parameters	29
6.5 Response Object.....	30
6.6 Examples	30
6.6.1 Query all bounces within a time window.....	30
6.6.2 Query CH_REJECTED with complaint information in channel status text	32
6.6.3 Query CH_REJECTED where the email request was suppressed	32
7 GET Error API calls	34
7.1 Resource URL for errors	34
7.2 Resource Information.....	34
7.3 URL Parameters	34
7.4 Request Parameters	35
7.5 Response Object.....	36
7.6 Error responses.....	36
7.7 Response Parameters	36
7.8 Examples:	38
7.8.1 Query errors within a time window.....	38
7.8.2 Sample error response to API call.....	39
7.8.3 HTTP Response values	39
8 Status Codes	41



9 Channel Status Text Responses43



1 Sinch E-Mail 365 API

Sinch E-Mail 365 delivers a set of JSON/ RESTful API's that provides the ability to optimize engagement using email. The API interfaces are easily consumable by any upstream application.



2 Glossary of Terms

Before we dwell into the API details specific to email; please see below some email service specific industry terms to get familiar with:

JSON/ RESTful API	<p>JSON API is a specification for how a client should request that resources be fetched or modified, and how a server should respond to those requests.</p> <p>JSON API is designed to minimize both the number of requests and the amount of data transmitted between clients and servers. This efficiency is achieved without compromising readability, flexibility, or discoverability.</p>
Email Service Provider	<p>An email service provider (ESP) is a company that offers email marketing or bulk email services. Sinch uses the ESP infrastructure to support the processes and governance around typical email marketing campaigns.</p> <p>Sinch uses 3rd party email infrastructure to leverage industry standard delivery and reputation management capabilities. These are behind the scenes capabilities.</p>
Domain Key Identified Mail	<p>Technically DKIM provides a method for validating a domain name identity that is associated with a message through cryptographic (public –private keys) authentication. Identification of an email’s association with a trusted domain enhances deliverability.</p> <p>DKIM values are not part of the API spec. They are defined at the email account level but set sent out as part of the outbound notification delivery to a recipient mail server.</p> <p>For a detailed understanding of using DKIM for deliverability management; please refer to the “Enhancing delivery</p>



	management for Sinch E-Mail 365 – email service using DKIM and SPF”
Sender Framework Policy	<p>Sender Policy Framework (SPF) is a simple email-validation system designed to detect email spoofing by providing a mechanism to allow receiving mail exchangers to check that incoming mail from a domain comes from a host authorized by that domain's administrators.</p> <p>SPF records are not part of the API spec.</p> <p>For a detailed understanding of using DKIM for deliverability management; please refer to the “Enhancing delivery management for Sinch E-Mail 365 –using DKIM and SPF”.</p>
Soft Bounce	<p>A bounce message, also called a Non-Delivery Report/Receipt (NDR), a (failed) Delivery Status Notification (DSN) message, a Non-Delivery Notification (NDN) or simply a bounce, is an automated electronic mail message from a mail system informing the sender of another message about a delivery problem.</p> <p>A soft bounce means that the email address was valid and the email message reached the recipient’s mail server. However, it bounced back because, for example:</p> <ul style="list-style-type: none"> • The mailbox was full (the user is over their quota) • The server was down • The message was too large for the recipient’s inbox <p>Soft bounce reports are captured and stored by Sinch E-Mail 365 service and provided as call back API services for upstream applications. Details are provided in this API spec document.</p>
Hard Bounce	<p>A hard bounce occurs when the message has been permanently rejected either because:</p> <ul style="list-style-type: none"> • The email address is invalid • The email addresses does not exist



	<p>Hard bounce reports are captured and stored by Sinch E-Mail 365 service and provided as call back API services for upstream applications. Details are provided in this API spec document</p>
<p>Sender Address</p>	<p>This is typically a generic email address (such as support@abc.customer.com) from where the marketing or transactional email is being sent.</p> <p>This value is handled using the “sender” object in the API or it can be set by default in the account profile.</p>
<p>Reply to Address</p>	<p>When an email request is received: it will typically have two values as below:</p> <ol style="list-style-type: none"> Sender ID: for e.g. info@newsletter.customer.com ;where newsletter.customer.com is the registered sender subdomain “Reply To” address: For example: name1@customer.com. This is a valid email address where a recipient’s reply is delivered (if needed). <p>The “reply to” address allows a recipient to respond back to a transactional or marketing email. When a “reply to” email is sent out, the mail transfer agent looks up DNS entries to determine the MX record of the recipient’s mail server. If an MX record is set up, then the replies are sent to the ESP Mail server from where they are forwarded to the intended recipient.</p> <p>This address is handled using the “replyTo” object in the API or it can be set by default in the account profile.</p>
<p>Custom “From” tag</p>	<p>A customer “From” tag allows a unique name to be tagged to a “sender address”. For example, if we set up the “from” tag as Mark Jones, the sender address will then show as Mark Jones <support@abc.customer.com></p> <p>The custom from tag is handled using the “senderName” object in the API.</p>



Sending sub-domain	A dedicated sending sub-domain is recommended for both transactional and marketing emails. This shall impact the default sender & reply to addresses in the notification request.
Attachments	<p>Attachments may be sent with the email notifications. The attachment itself must be in BASE64-Encoded String representation of the content. 'ContentName' is a required parameter for sending attachments. The maximum number of attachments you can send by email is 20.</p> <p>Note: <u>Approval is needed for access to sending attachments. Currently this is being done on a case by case basis</u></p>

Table 1:Glossary of Terms



3 E-Mail 365 API Details

3.1 Security

All calls will be encrypted using TLS. Access is provided using Basic Authentication. These credentials are delivered post the account provisioning process.

For details on information needed to complete the provisioning process, please ask your account rep for the provisioning form.

3.2 API object reference

3.2.1 Post (Request)

Request Object hierarchy	notification
Request Content-type	application/json
Request Authentication	Authorization: Basic
Response Content-type	application/json
Response Object	notification

3.3 Credentials

A consuming application will send a notification request to an account with credentials (typically as below) :

- Non-VPN Customer URL:
https://eu.email.sdi.sinch.com/v1/caas_email12345/notifications
- UserID: caas_email12345
- Password: xxxxxxxx
- Default Sender: name@subdom.customer.com (default sender name)
- Default Reply To: contact@customer.com



4 POST Request API

4.1 Sample Resource URL

POST

https://eu.email.sdi.sinch.com/v1/caas_email12345/notifications

4.2 Resource Information

Top Level	Next Level	Next Level
notification	contentAttachment	
	channelpreference	
	campaign	
	configuration	email

Table 2: POST request object hierarchy

Request Content-type application/json

Request Authentication Authorization: Basic

4.3 Request API parameters

Class	Field	Required	Description
notification		Required	Top level object to create/send notification.
notification	notificationId	Not Required for creation	Unique notification identifier
notification	sender	Optional	Originator of the notification



			Identified by: email: - email address
notification	recipients	Required	One or more entity that will be receiving notification. Identified by: email: - email address Maximum of 100 per API call
notification	contentText	Required if contentAttachments omitted	Character string of the notification.
notification	contentType	Required	text/plain or text/html Default text/plain
notification	contentTextEncoding	Required	BASE64 or NONE, Default NONE
notification	characterSet	Optional	Default UTF-8. ISO standard name format
notification	contentAttachments	Required if contentText is omitted	List of multi-media attachments and/or application data.
contentAttachments	contentName	Required for sending attachments	ContentName – file name of the attachment
contentAttachments	contentId	Optional	Identifier for the contentAttachment. contentId should not be set if



			ContentName value is to be used in attachment filename.
contentAttachments	attachment	Required if content is attached	BASE64 Encoded String representation of the content. Maximum of 5 MB.
contentAttachments	contentType	Required	Media types as defined by IANA
notification	channelPreferences	Optional Default: email	Channel to use for delivery.
notification	systemId	Optional	User defined identifier for a group of notifications originating from a system. Can be used as a filter in searches.
notification	campaignId	Optional	User defined identifier for a group of notifications related to a campaign. Can be used as a filter in searches.
notification	processingType	Optional	Values: 'dedicated' or 'standard'. Default empty which is processed as standard
notification	expiryHour	Optional	Values: 1-24. Default empty which means there is no expiry set.
configuration	email	Optional	EMAIL specific configuration
email	subject	Required	Email subject



email	cc	Optional	Same formal as 'recipients' field
email	bcc	Optional	Same formal as 'recipients' field
email	senderName	Optional	Display name of sender
email	replyTo	Optional	Sender reply address
email	replyToName	Optional	Sender reply name
email	alternateContentText	Optional	Character string of the notification. Used to support multipart/alternative email body
email	antivirusScan	Optional	Values: 'true or 'false. Default empty which is processed as false. Indicates whether an email should be scanned for viruses
email	alternateContentTextType	Optional	text/plain or text/html Default text/plain
email	alternateContentTextEncoding	Optional	BASE64 or NONE, Default NONE
email	deliveryUTCDate	Optional	Requested delivery date in UTC (at least 2 hours from present time and at most 48 hours in the future) in format "YYYYMMDD"
email	deliveryUTCHour	Optional (required with	Requested delivery hour in UTC (at least 2 hours from present time and at most 72 hours in the future) in format "HH"



		deliveryUTC Date)	
email	headers	Optional for non CSA certified IPs Required for CSA certified Ips	Map of "custom" x-headers. Header-name has to start with (ignore-case) "X-" with the exception of List-Unsubscribe. Otherwise it will be ignored. Some examples are: ":[{"name":"x-myheader","value":"testing"}, {"name":"List-Unsubscribe","value":"<http://www.xxx.com>,<mailto:unsubscribe@xxx.com>"}] NOTE: If the mailto set to anything with subdomain @eusmtp.email.sdi.sinch.com. Our system will handle the unsubscribe requests and create CH_REJECTED / UNSUBSCRIBE event. Customers get the Unsubscribe data through API.

Table 3: POST Request API Parameters

4.4 Response API parameters

Response Content-type application/json

Response Authentication Authorization: Basic

Response API parameters (in addition to request parameters)



NOTE Content is omitted in response

Class	Field	Required	Description
status	channelStatusCode	Optional	Specific channel specific status code
status	channelStatusText	Optional	Specific channel specific text description When available status is sent as channelStatusText: eventType-eventState-eventSubType:description + smtpReason “null” if information is not available / sent back Note: Please refer to “channel Status Responses’ table below for more details
status	statusCode	Required	Notification specific status code
status	statusText	Required	Notification specific text description
status	recipient	Required	The recipient this status is in reference to ‘ALL’ implies it’s for the entire request. It is a general status.
status	channel	Optional	Channel to which the status occurred
status	timestamp	Required	Date time oft when state was generated.

Table 4: POST Response API parameters



4.5 Examples

4.5.1 Request (syntax only) to send EMAIL

POST:

https://eu.email.sdi.sinch.com/v1/caas_email12345/notifications

Content-Type: application/json

Authorization: Basic sdfjkerwyvnnnewrlj==

```
{
  "notification":
  {
    "channelPreferences":"EMAIL",
    "recipients": ["email:name1@customer.com"],
    "contentType": "text/html",
    "contentText": "<html><a href='http://www.google.com'>hello</a></html>",
    "contentTextEncoding": "NONE",
    "sender":"pilotuser@test.sinch.com",
    "processingType":"dedicated",
    "expiryHour":"1",
    "configuration":
    {
      "email":
      {
        "senderName":"Test 365",
        "replyTo":"contact@customer.com ",

```



```

        "replyToName": "REPLY USER",
        "subject": "TestIN365",
        "cc": ["email:xxxx@gmail.com"],
        "bcc": ["email:xxxx@yahoo.com"],
        "antivirusScan": "true"

        "headers": [{"name": "x-myheader", "value": "testing"}, {"name": "List-
Unsubscribe",
"value": "<http://www.xxx.com>, <mailto:unsubscribe@xxx.com?subject=unsubscribe
>"}]
    }
}
}
}

```

4.5.2 Response to above request

```

{
  "notification": {
    "account": "caas_2_email12345",
    "notificationId": "100000009020772",
    "status": [
      {
        "statusCode": "CAAS_RECEIVED",
        "statusText": "Successful",
        "channel": "EMAIL",
        "timestamp": "2016-07-28 20:23:39.675",
        "recipient": "ALL",
        "channelStatusCode": null,
        "channelStatusText": null
      }
    ],
  },
}

```



```

"characterSet": null,
"sender": "pilotuser@test.sinch.com",
"recipients": [
  "email:testuser@customer.com"
],
"contentType": "text/html",
"contentText": null,
"contentTextEncoding": "NONE",
"channelPreferences": "EMAIL",
"configuration": {
  "email": {
    "subject": "TestIN365",
    "senderName": "Test 365 ",
    "replyTo": "contact@customer.com",
    "cc": ["email:xxxx@gmail.com"],
    "bcc": ["email:xxxx@yahoo.com"]
    "replyToName": "REPLY USER",
    "alternateContentType": null,
    "alternateContentText": null,
    "alternateContentTextEncoding": null
  }
},
"contentAttachments": null,
"campaignId": null
}
}

```

NOTE Content is omitted in response.

4.5.3 Request to send EMAIL with attachment (syntax only)

```

{
  "notification": {

```



```

"recipients": ["email:yourself@xxx.com"],
"sender": "testuser@test.sinch.com",
"contentType": "text/html",
"contentText": "<html><u>hello</u></html>",
"contentTextEncoding": "NONE",
"configuration":
{
  "email":
  {
    "senderName": "ä ü ö",
    "subject": "test test"
  }
},
"contentAttachments": [
  {"contentId": "id1", "contentType": "text/plain",
    "contentName": "testAttachment.txt",
    "attachment": "aGVsbG8gd29ybGQ="
  },
  {"contentId": "id2", "contentType": "application/pdf",
    "contentName": "testAttachment.pdf",
    "attachment": "JVBERi0xLjQKJeLjz9MKMyAwIG9iago8PAovUHJvZHVjZXIgLKFBBERi1
YQ2hhbmdlIFByaW50ZXIgLjYgXCg2LjAgYnVpbGQgMzlyLjVcKSBbV2luZG93cyAx
MCFBbnRlcnByaXNIIHg2NCBcKEJ1aWxkIDE0MzgzXCldKQovQ3JlYXRpb25EYXRI
IChEOjIwMTgwODE1MTM1MTQxLTA3JzAwJykKPj4KZW5kb2JqCjUgMCFBbnRlcnBy
DwKL1R5cGUgL0ZvbnREZXNjcmlwdG9yCi9Gb250TmFtZSAvQ291cmllck5ld1BTT
VQKL0ZsYWdzIDMyCi9Gb250QkJeCBbLTEyMiAtNjgwIDYyMyAxMDIxXQovTWlzc
2luZ1dpZHRoIDYwMAovU3RlbnUggnZkKL1N0ZW1WIDc5Ci9JdGFsaWNBbmdsZSA
wCi9DYXBIZWlnaHQgNTcxCi9YSGVpZ2h0IDQyMwovQXNjZW50IDgzMwovRGVzY
2VudCAzMzAwCi9MZWFkaW5nIDEzZW50TWF4V2lkdGggNjAwCi9BdmdXaWR0aC
A2MDAKPj4KZW5kb2JqCjYgMCFBbnRlcnBydWKL1R5cGUgL0ZvbnQKL1N1YnR5cG
UgL1RydWVUeXBICi9CYXNIRm9udCAvQ291cmllck5ld1BTTVQKL0ZpcnN0Q2hhci
A0NwovTGFzdENoYXlzMTE5Ci9FbmnVZGluZyAvV2luQW5zaUVuY29kaW5nCi9G
b250RGVzY3JpcHRvcIA1IDAgUgovV2lkdGhZIFsgNjAwIDYwMCA2MDAgNjAwIDAg

```




5 GET notification Status by id

This will pull all the events for a single notification ID. If multiple recipients are passed in request; this will pull events for all recipients for that notification ID.

NOTE Data is only available for 7 days. If the notification was sent more than 7 days back, an empty json object will be returned.

5.1 Sample Resource URL:

GET

https://eu.email.sdi.sinch.com/v1/caas_email12345/notifications/id/10000000902077

2

5.2 Resource Information

Request Authentication	Authorization: Basic
Response Content-type	application/json
Response Object	notification

5.3 URL Parameters

Class	Field	Required	Description
	accountId	Required	Customer account identifier
	notificationId	Required	notificationId returned in POST notification response
	recipient	Optional	Email address in format "email:<xxx>@<domain.com>"

Table 5: GET URL Parameters



5.4 Examples

5.4.1 Request to query status of notification

GET

https://eu.email.sdi.sinch.com/v1/caas_email12345/notifications/id/10000000902077

2

Authorization Basic sdfjkerwyvnnnewrlj==

5.4.2 Response to status of notification

```
"notification": {
  "notificationId": "100000009020772",
  "status" : [
    {
      "statusCode": "CAAS_RECEIVED",
      "statusText": "Successful",
      "channel" "email",
      "recipient": "email:testuser@sinch.com"]",
      "timestamp": "2015-03-08 14:58:30.252"
    },
    {
      "statusCode": "CAAS_SENT",
      "statusText": "Successful",
      "channel" "email",
      "recipient": "email:testuser@sinch.com"]",
      "timestamp": "2015-03-08 14:59:10.252"
```



```

    },
    {
      "statusCode": "CH_RECEIVED",
      "statusText": "Successful",
      "channel" "email",
      "recipient": "email:name1@customer.com"]",
      "timestamp": "2015-03-08 14:59:20.252"
    },
    {
      "statusCode": "CH_SENT",
      "statusText": "null",
      "channel" "email",
      "recipient": "email:name1@customer.com"]",
      "timestamp": "2015-03-08 14:59:30.252"
    },
  ]
}
}

```

5.4.3 Request to query status notification with specific recipient

GET

https://eu.email.sdi.sinch.com/v1/caas_email12345/notifications/id/100000009020772?recipient=email:testuser@sinch.com

This gives the same result as above. This would be useful if you sent to multiple recipients





6 GET Bounce (specific status) API calls

Searches for status based on insertion date in the database

Bounces are stored as SMTP error codes and delivered as call back API services.

This will by default pull all statusCode = CH_REJECTED statuses that came in during a specified time frame. These will include statusText SOFT_BOUNCE, HARD_BOUNCE, COMPLAINT or BLOCKED.

NOTE If the time range is more than one day (24-hours), a CAAS_ERROR will be returned. If the start date is more than 7 days back, a CAAS_ERROR will be returned. Both will return text that describes this.

6.1 Resource URL for bounces

GET

https://eu.email.sdi.sinch.com/v1/caas_email12345/notifications/email/bounce?statusCode=CH_REJECTED&startUTCTime=20160727115959&endUTCTime=20160805115959

6.2 Resource Information

Request Authentication Authorization: Basic

Response Content-type application/json

Response Object notificationList

6.3 URL Parameters

Class	Field	Required	Description
	accountId	Required	Customer account identifier

Table 6: GET Bounce URL Parameters



6.4 Request Parameters

Class	Field	Required	Description
	systemId	Optional	User User defined identifier for a group of notification originating from a system.
	campaignId	Optional	User defined identifier for a group of notifications related to a campaign.
	statusCode	Optional	Notification specific status code (defaults to CH_REJECTED if not supplied).
	startUTCTime	Required	DateTime in UTC timezone with format of yyyyMMddHHmmss. This is the time of the receipt of the statusCode in the query.
	endUTCTime	Required	DateTime in UTC timezone with format of yyyyMMddHHmmss. This is the time of the receipt of the statusCode in the query.
	groupIndex	Optional	Default: 1. For large result set, responses are broken up into groups of 100. groupIndex defines which group to retrieve.
	statusText	Optional	Notification specific status text. Useful values are



			HARD_BOUNCE, SOFT_BOUNCE, SUPPRESSED, COMPLAINT
--	--	--	---

Table 7: GET Bounce request parameters

6.5 Response Object

Class	Field	Required	Description
notificationList	groupIndex	Required	Default: 1. For large result set, responses are broken up into groups of 100. groupIndex defines which group to retrieve.
notificationList	groupCount	Required	Specifies the total number of groups there are in the result set.
notificationList	notifications	Optional	List of notification matching query parameter

Table 8: GET Bounce response object

6.6 Examples

6.6.1 Query all bounces within a time window

GET

https://eu.email.sdi.sinch.com/v1/caas_email12345/notifications/email/error?startUTCtime=20170819064224&endUTCtime=20170819070322&systemId=skwa1

```
{
  "notificationList": {
    "groupIndex": 0,
```



```
"groupCount": 0,
"notifications": [
  {
    "account": " caas_email12345",
    "notificationId": "100000254357015",
    "status": [
      {
        "statusCode": "CH_ERROR",
        "statusText": "SOFT_BOUNCE",
        "channel": "EMAIL",
        "timestamp": "2016-08-26 17:36:13.465",
        "recipient": "email:testuser@sinch.com",
        "channelStatusCode": "SMTP=451,DSN=",
        "channelStatusText": "mailbox was full"
      }
    ],
    "characterSet": null,
    "sender": null,
    "recipients": null,
    "contentTextType": null,
    "contentText": null,
    "contentTextEncoding": null,
    "channelPreferences": null,
    "configuration": null,
    "contentAttachments": null,
    "campaignId": "skwa1"
    "systemId": null
  }
]
}
}
```




6.6.2 Query CH_REJECTED with complaint information in channel status text

GET

https://eu.email.sdi.sinch.com/v1/caas_email12345/notifications/email/bounce?statusCode=CH_REJECTED&startUTCTime=20171010115959&endUTCTime=20171017115959&systemId=skwa1

```
{
  "statusCode": "CH_REJECTED",
  "statusText": "COMPLAINT",
  "channel": "EMAIL",
  "timestamp": "2017-10-16 21:26:03.0",
  "recipient": "email:testuser@sinch.com",
  "channelStatusCode": null,
  "channelStatusText": null
}
```

6.6.3 Query CH_REJECTED where the email request was suppressed

GET

https://eu.email.sdi.sinch.com/v1/caas_email12345/notifications/email/bounce?statusCode=CH_REJECTED&startUTCTime=20180510115959&endUTCTime=20180513115959

```
{
  "statusCode": "CH_REJECTED",
  "statusText": "SUPPRESSED",
  "channel": "EMAIL",
  "timestamp": "2017-10-16 21:26:03.0",
  "recipient": "email:testuser@sinch.com",
  "channelStatusCode": null,
}
```



```
"channelStatusText": null  
}
```



7 GET Error API calls

Searches for errors based on insertion date in the database

Errors are stored and delivered as call back API services.

This will by default pull all statusCode = CH_ERROR statuses that came in during a specified time frame.

7.1 Resource URL for errors

GET

https://eu.email.sdi.sinch.com/v1/caas_email12345/notifications/email/error?statusCode=CH_ERROR&startUTCTime=20160727115959&endUTCTime=20160805115959

7.2 Resource Information

Request Authentication	Authorization: Basic
Response Content-type	application/json
Response Object	notificationList

7.3 URL Parameters

Class	Field	Required	Description
	accountId	Required	Customer account identifier

Table 9: GET URL parameters



7.4 Request Parameters

Class	Field	Required	Description
	systemId	Optional	User User defined identifier for a group of notification originating from a system.
	campaignId	Optional	User defined identifier for a group of notifications related to a campaign.
	statusCode	Optional	Notification specific status code (defaults to CH_REJECTED if not supplied).
	startUTCTime	Required	Date/Time in UTC timezone with format of yyyyMMddHHmmss. This is the time of the receipt of the statusCode in the query.
	endUTCTime	Required	Date/Time in UTC timezone with format of yyyyMMddHHmmss. This is the time of the receipt of the statusCode in the query.
	groupIndex	Optional	Default: 1. For large result set, responses are broken up into groups of 100. groupIndex defines which group to retrieve.
	statusText	Optional	Notification specific status text.

Table 10: GET Request parameters



7.5 Response Object

Class	Field	Required	Description
notificationList	groupIndex	Required	Default: 1. For large result set, responses are broken up into groups of 100. groupIndex defines which group to retrieve.
notificationList	groupCount	Required	Specifies the total number of groups there are in the result set.
notificationList	notifications	Optional	List of notification matching query parameter

Table 11: GET response object

7.6 Error responses

If the GET request is formatted incorrectly, you will receive the following response. Be very careful with spelling and capitalization of optional key names as mistakes with them will not be noted.

7.7 Response Parameters

Class	Field	Required	Description
notificationStatus		Required	Top level error response.
notificationStatus	Status	Required	Status information



status	channelStatusCode	Optional	Specific channel specific status code
status	channelStatusText	Optional	Channel specific text description. When available status is sent as channelStatusText: eventType-eventState-eventSubType:description + smtpReason “null” if information is not available / sent back Note: Please refer to “channel Status Responses’ table below for more details
status	statusCode	Required	Notification specific status code
status	statusText	Required	Notification specific text description
status	recipient	Required	The recipient this status is in reference to ‘ALL’ implies it’s for the entire request. It’s a general status.
status	timestamp	Required	Date time of when state was generated.
status	channel	Optional	Channel to which the status occurred

Table 12: GET Error response parameters



7.8 Examples:

7.8.1 Query errors within a time window

GET

https://eu.email.sdi.sinch.com/v1/caas_email12345/notifications/email/error?startUTCtime=20170819064224&endUTCtime=20170819070322&systemId=skwa1

```
{
  "notificationList": {
    "groupIndex": 1,
    "groupCount": 2,
    "notifications": [
      {
        "account": " caas_email12345",
        "notificationId": "100000254357015",
        "status": [
          {
            "statusCode": "CH_ERROR",
            "statusText": "Read timed out",
            "channel": "EMAIL",
            "timestamp": "2016-08-26 17:36:13.465",
            "recipient": "email:testuser@sinch.com ",
            "channelStatusCode": null
            "channelStatusText": null
          }
        ],
        "characterSet": null,
        "sender": null,
        "recipients": null,
        "contentTextType": null,
        "contentText": null,
        "contentTextEncoding": null,

```



```

    "channelPreferences": null,
    "configuration": null,
    "contentAttachments": null,
    "campaignId": "skwa1"
    "systemId": null
  }
]
}
}

```

7.8.2 Sample error response to API call.

```

{
  "notificationStatus": {
    "status": {
      "statusCode": "CAAS_ERROR",
      "statusText": "Error : startUTCTime and endUTCTime are required but values are
not valid",
      "channel": null,
      "timestamp": null,
      "recipient": null,
      "channelStatusCode": null,
      "channelStatusText": null
    }
  }
}

```

7.8.3 HTTP Response values

The following table describes HTTP response possible values.



HTTP Response	Possible StatusText
HTTP 500	startUTCTime and endUTCTime are required but values are not valid. Format is yyyyMMddHHmmss, i.e. 20151225235959
	startUTCTime and endUTCTime duration must be within a 24 hours window
	startUTCTime must be within 7 days
	recipient list is greater than 100
	recipient xxx exceeded max size of 256 and must start with 'email: '
	sender xxx exceeded max size of 256
	replyTo xxx exceeded max size of 256
	systemId field exceeded max size of 10. If you're using the 'campaign' field, please change to systemId or campaignId
	campaignId field exceeded max size of 10
	Cannot generate notificationID
	Invalid Id :
	Missing Object :
	Invalid replyTo (xxx). Reply-to must be in [xxx]
	Sender replyTo (xxx). Sender must be in [xxx]
	Account is not enabled for scheduled delivery
	Invalid deliveryUTCDate (yyyyMMdd) or deliveryUTCHour (0-23) format.
	Scheduled delivery must be in the future
Scheduled delivery must be within 48 hours	
Scheduled delivery must be at least 1 hour in advance	
HTTP 401	Unauthorized
HTTP 429	LIMIT_EXCEEDED



8 Status Codes

The list of possible status code values is provided in the table below

Event Status Code	Event Type	Event Description	Currently Used
CAAS_RECEIVED	CAAS	Accepted by platform	Yes
CAAS_SENT	CAAS	Message sent from platform to downstream	Yes
CAAS_ERROR	CAAS	Error occurred while processing message by platform	Yes
CH_RECEIVED	CHANNEL	Accepted by channel	Yes
CH_REJECTED	CHANNEL	<p>Message rejected by channel (message bounced). Possible statusText includes one of the following [HARD_BOUNCE, SOFT_BOUNCE, COMPLAINT, SUPPRESSED]</p> <p>NOTE: A CH_REJECTED SOFT_BOUNCE status is where the email was unable to be delivered after a 48 hour retry period and under certain circumstances there will be no text available in the channel status text field”</p>	Yes
CH_RETRY	CHANNEL	Retrying delivery in channel	Yes



CH_SENT	CHANNEL	Message sent from channel to downstream	Yes
CH_ERROR	CHANNEL	Error occurred while processing message by channel	Yes

Table 13: List of status codes



9 Channel Status Text Responses

This is API response parameter and a status field for channel specific text description. The format for **channelStatusText** is:

channelStatusText: eventType-eventState-eventSubType:description + smtpReason

The following table describes channelStatusText possible values

Event/type	Event/state	Event/subType	Description
DROPPED	FINISHED	PREVIOUSLY_BOUNCED	Recipient address found in suppression list because recipient was previously bounced. Email not delivered. (includes unknown user)
DROPPED	FINISHED	FORBIDDEN_RECIPIENT	Forbidden local part
DROPPED	FINISHED	INVALID_ADDRESS	Recipient address syntactically incorrect. Email not delivered.
PROCESSED	INTERMEDIATE	PROCESSING_FINISHED	Message has been received, processing is finished and is ready to be delivered.
DROPPED	FINISHED	PROCESSING_FAILED	Internal processing errors. eg. missing data, invalid parameter, etc.
DELIVERED	FINISHED	OK	
DEFERRED	INTERMEDIATE	SOFT_BOUNCE	Email deferred. Further attempts pending
BOUNCED	FINISHED	HARD_BOUNCE	Email Bounced. Recipient may be added to suppression list
DROPPED	FINISHED	INVALID_DKIM_SIGNATURE	Validation of DKIM-signature in OpenDKIM verifier-milter failed and config-settings impose in this case the mail to be dropped
BOUNCED	FINISHED	SOFT_BOUNCE	Email bounced. All attempts failed. Timeout.



DROPPED	FINISHED	PROCESSING_FAILED	Internal processing errors. eg. missing data, invalid parameter, etc.
DEFERRED	INTERMEDIATE	SOFT_BOUNCE	Email deferred. Further attempts pending
BOUNCED	FINISHED	HARD_BOUNCE	Email Bounced. Recipient may be added to suppression list
BOUNCED	FINISHED	SOFT_BOUNCE	Email bounced. All attempts failed. Timeout.
DELAYED	INTERMEDIATE	RATE_LIMITED	Further processing is delayed because of the recipient ISP / Mail-Provider indicates the sender is trying to sent too many messages based on the sender reputation or ISP Quota / Rate-Limit.
DROPPED	FINISHED	SUPPRESSED	Recipient address added to suppression list
DROPPED	FINISHED	VALIDATION_ERROR	Fallback for recipient validations
FEEDBACK_LOOP	POST_DELIVERY	COMPLAINT	Recipient has sent a complaint to their ISP