# mailsac-examples Documentation

Forking Software LLC

Apr 27, 2019

# General

Welcome to the official documentation for Mailsac, the email service built for developers. If you are new to this documentation we recommend taking a look at the *introduction page* to get an overview of what the documentation has to offer.

The table of contents is the sidebar and will let you easily access the documentation for your topic of interest. You can also use the search function in the top left corner.

# CHAPTER 1

# Introduction

With Mailsac, it's super easy to interact with email via REST API, webhooks and websockets. You can reserve and release email addresses, check messages, download attachments, and route mail.

**Tip:** All API endpoints can be found in the API Documentation

## 1.1 Curl Example

You can use curl to view emails sent to user1@mailsac.com

```
$ curl -s -X GET https://mailsac.com/api/addresses/user1%40mailsac.com/messages | jq
↪".[0]"
```

Information about the most recent email is returned as JSON

```
{
  "_id": "BotvTxaona7gLID1Adtpfj8Fnfi7HSSv-0",
  "from": [
    {
      "address": "microsoftstore@e.microsoft.com",
      "name": "Microsoft Store"
    }
  ],
  "to": [
    {
      "address": "user1@mailsac.com",
      "name": ""
    }
  ],
  "cc": null,
```

(continues on next page)

(continued from previous page)


```
  "bcc": null,
  "subject": "Ahoy, Sea of Thieves for PC is here",
  "savedBy": null,
  "originalInbox": "user1@mailsac.com",
  "inbox": "user1@mailsac.com",
  "domain": "mailsac.com",
  "received": "2018-03-29T18:28:07.732Z",
  "size": 23420,
  "attachments": null,
  "ip": "65.55.234.211",
  "via": "144.202.71.79",
  "folder": "inbox",
  "labels": [],
  "read": null,
  "rtls": true,
  "links": [
    "href=https://e.microsoft.com/Key-3567701.C.CQZpy.C.K0.-.CMHlNS",
    "http://msstorepromoemail.blob.core.windows.net/windows-store-edits/Nav_MSFT_Logo.
↪jpg",
  ],
  "spam": 1.3370381090039505e-09
}
```


**Tip:** This may look for more information than you need. But it provides a great example of all the hard work mailsac has done to make parsing of email easier.

CHAPTER 2

# FAQ

## 2.1 Where are my email attachments?

You must download the message or parse it yourself.

## 2.2 Why would I use Mailsac?

Any time you need a temporary email address, just make one up@mailsac.com.

If you need to test your email system, send it to mailsac.com - even for a custom domain.

Other uses:

- if you need an email address, but do not want to receive spam in your personal inbox
- use it for sign ups on web sites that force you to login
- give it out to strangers
- use it to collaborate for projects
- send mail to Mailsac for testing purposes
- use it when you (legally) want to receive email without disclosing your identity
- it is perfect if you want an email address or multiple addresses and do not want to sign up for them
- comment on blogs without creating an account
- as a developer, to test that your application is sending email

## 2.3 Why weren't my messages received?

There are many reasons messages may not be received by Mailsac.

1. The sender blocks traffic to disposable email providers like Mailsac. This is common with email signups or email verifications. People running websites do not want a bunch of spam accounts.
2. Large message size. Mailsac only supports messages up to about 2 MB.
3. Throttling. Non-paying customers can be throttled for sending too much. We are happy to lift this for paying customers. Usually, it is easiest for us to lift it on your Verified Domain.

In all cases, if you contact support, we can help you work around these problems. Either by hosting your own instance of Mailsac, contacting us for help, or moving you to a separate inbound or outbound IP.

## 2.4 Can I use Mailsac for testing purposes?

Absolutely!

We love to hear from developers that use Mailsac - contact us anytime.

---

**Tip:** support@team.mailsac.com

---

If you expect to send more than a few messages per minute, you might get throttled. Contact us about reducing throttling, or setup a Verified Domain, or buy an API key.

## 2.5 How long is email saved?

Email messages are saved for somewhere between three days and 1 week or more, sometimes less. No guarantees!

If you are logged in and star a message, it will not be recycled until you unstar it. Or if you have made it private, messages will be kept up to your storage limit.

## 2.6 Can other people see messages that I starred?

Nope. Only you can see them when you are logged in.

## 2.7 How do I reply to my Mailsac emails?

You must create an account to reply to emails. You'll get a few to start out, then can buy more as needed.

CHAPTER 3

# Contact and Support

## 3.1 Feature Requests

To discuss new ideas and features, visit the feature discussion forum.

## 3.2 General Help

Browse our documentation and forums.

## 3.3 Mail Not Received

Read about mail not received on the *Why weren't my messages received?*.

## 3.4 Paying Customers - Email and API Support

For service setup issues, billing questions, sales contact **support@team.mailsac.com**.

Indicate your account **username** and the email address (if different from the email you are sending from).

For **API support** we encourage you to post to the Developers and API forum so your solution can be shared with other developers.

## 3.5 Support Hours

We answer support requests during business hours, USA West Coast Pacific Time (PDT/PST)

CHAPTER 4

# Privacy Policy

Revised October 2018

## 4.1 1. Preamble

1a. This Privacy Policy discloses the privacy practices for Mailsac (mailsac.com, associated sites, and backend or frontend internet services that support the mailsac.com website).

1b. This Privacy Policy applies to information collected by this web site. It will educate the user of the following:

- 1b-i. What personally identifiable information is collected from you through the web site, how it is used and with whom it may be shared.
- 1b-ii. What choices are available to you regarding the use of your data.
- 1b-iii. The security procedures in place to protect the misuse of your information.
- 1b-iv. How you can correct any inaccuracies in the information.

## 4.2 2. Email

This site provides public email, and public or private email testing services. Email inboxes that are not purchased (aka private) are public and have no privacy.

Consider this warning carefully when you opt to receive email on this website and give out addresses that will have email delivered to this website. Email is PUBLICLY VIEWABLE CONTENT unless otherwise explicitly indicated. On private email, we reserve the right to inspect or share information contained in those emails, but it REASONABLY EXPECTED that mail termed "private" will not be made viewable to the public at large. It MAY BE MADE PUBLIC to paying customers or other third parties.

## 4.3 3. Information Collection, Use, and Sharing

3a. We are the sole owners of the information collected on this site. We may choose to sell or rent this information to third parties, manually or via paid or free API, or by exporting archives of information to third parties.

3b. We will use your information to respond to you, regarding the reason you contacted us. We may share your information with third parties outside of our organization, for the purpose of providing better and more relevant services to you.

3c. Unless you ask us not to, we may contact you via email in the future to tell you about specials, new products or services, or changes to this privacy policy.

3d. Ads and analytics tracking services on (such as, but not limited to, Google Analytics) may be exposed to inbox contents, even on private or paid accounts.

## 4.4 4. Your Access To and Control Over Information

You may opt out of any future contacts from us at any time.

You can do the following at any time by contacting us via the email given on our website: * See what data we have about you, if any. * Change/correct any data we have about you. * Have us delete any data we have about you. * Express any concern you have about our use of your data.

## 4.5 5. Registration

5a. In order to gain FULL USE of Mailsac, user registration IS REQUIRED.

5b. Partial use of Mailsac is possible without user registration, in which case information such as, but not limited to, IP address, geographic information, or cookies may be used to identify the user or throttle their usage of the website.

5c. During registration a user is required to give certain information. This information may be used to contact you about the products/services. It may be shared with 3rd parties.

## 4.6 6. Orders

We request information from you on our order and checkout form via third party.

To buy services from us, you may need provide contact information and financial information, including to third parties (payment processors and other business partners). This information is used for billing purposes and to provide you service. It is stored with the payment processor and not on this site. If we have trouble processing an order, we'll use this information to contact you. We may share this information with third parties to ensure payment and provide targeted and improved services.

## 4.7 7. Cookies

We use cookies on this site. A cookie is a piece of data stored on a site visitor's device to help us improve your access to our site and identify repeat visitors to our site. For instance, when we use a cookie to identify you, you would not have to log in a password more than once, thereby saving time while on our site.

Cookies can also enable us to track and target the interests of our users to enhance the experience on our site.

Some of our business partners may use cookies on our site (for example, advertisers). However, we have no access to or control over these cookies.

## 4.8 8. Sharing

We share aggregated demographic information with our partners and advertisers. We do our best to not link to any personal information that can identify any individual person, but this cannot be guaranteed. We do our best to limit sharing email contents or usage with our financial providers, but that information is necessary often (such as but not limited to fraud investigation and collateral background info).

We use Stripe for payment processing to bill users for services and may provide information about you to them. They may retain, share, store or use personally identifiable information.

We are not responsible for third party usage of your information. Refer to their privacy policies and terms.

## 4.9 9. Links and Content

9a. This website contains links to other sites. This website displays email information received by unknown parties over the internet.

Be aware that we are not responsible for the content or privacy practices of such other sites or of email received.

9b. We encourage our users to be aware when they leave our site and to read the privacy statements of any other site that collects personally identifiable information.

Users agree to exercise extreme caution when visiting links and downloading attachments in emails received by this website.

9c. Outgoing mail sent through this service may be interceptable or publicly viewable and shared with third parties. We may use a third party to store or deliver email and in such cases it may be subject to the terms and policies of those respective services. This site is not responsible for anything resulting from the presence of you data in their handling.

9d. Incoming and outgoing mail will be parsed and metadata, or the entire message, will be stored temporarily or indefinitely by Mailsac. Some of this information may be available for free or for purchase by Mailsac partners or customers.

## 4.10 10. Surveys and Contests

From time-to-time our site requests information via surveys or contests.

Participation in these surveys or contests is completely voluntary and you may choose whether or not to participate and therefore disclose this information. Survey information will be used for purposes of monitoring or improving the use and satisfaction of this site, and to share with third parties.

## 4.11 11. Security

11a. We take precautions to protect your information. When you submit sensitive information via the website, your information is protected both online and offline. Our goal is to use encryption while information is in transit and at rest, but there are exceptions and data may be transmitted or stored without encryption.

11b. Stripe is a third party which securely processes financial data. The privacy policy is found on their website, https://www.stripe.com. Mailsac is NOT P.C.I. COMPLIANT and relies on Stripe for handling credit card information.

11c. We protect your information offline. Only workers who need the information to perform a specific job (for example, billing, troubleshooting team or customer service) are granted access to personally identifiable information, or if requested by law enforcement, or in the event of a customer security issue we may give limited information to help the customer troubleshoot, which may include IP addresses, inbound email addresses, domains, and user account or username information (but not password hashes) for other customers or users of the service who appear to have performed acts intended to cause harm to another customer. Workers may or may not be employees of Mailsac.

11d. The computers/servers in which we store personally identifiable information are kept in a secure environment. Mailsac relies on third party virtual server providers. Below is a non-exhaustive list of server providers which may be used by Mailsac. For a current list, contact support@team.mailsac.com.

Hosting provider workers may have privilieged access to Mailsac servers.

- https://aws.amazon.com
- https://cloud.google.com
- https://www.chicagovps.net
- https://www.ramnode.com
- https://www.vultr.com

11e. From time to time we may run general analytics to determine usage statistics.

11f. We may track analytics on 1) inbound email, 2) outbound email, and 3) API or website usage - as a feature for customers or third parties to consume.

## 4.12 12. Updates

This Privacy Policy may change from time to time and all updates will be posted on this website. The user is expected to check back here for updates, and we reserve the right to NOT NOTIFY the user via email when this Privacy Policy changes.

If you feel that we are not abiding by this privacy policy, you should contact us immediately via support@team.mailsac.com.

## 4.13 13. Altering this Policy

13a. If you have other privacy concerns, or are interested in an alternative privacy arrangement (such as an enterprise), please contact support@team.mailsac.com to purchase a standalone deployment or other kind of contract. This Privacy Policy may be superseded by an alternative agreement acquired by working with Mailsac Support and obtaining a signed agreement.

13b. WE RECOMMEND USERS REGULARLY REVIEW THIS PRIVACY POLICY SINCE IT MAY CHANGE WITHOUT NOTICE.

CHAPTER 5

# Terms of Service and Acceptable Use Policy

Revised October 2018

## 5.1 1. Preamble

By using Mailsac (also “mailsac”, “we”, “us”, “this/the service”, “this/the website”, “this/the site”), you (also “user”, “the user”, “end user”, “customer”) accept these terms and the Privacy Policies (separate agreement) and agree to ABIDE BY THEM AT ALL TIMES WITH NO EXCEPTION, and with no right to a refund in the event of terms violation.

You accept that any of these policies and procedures will be available on the website but that THESE TERMS MAY CHANGE AT ANY TIME WITHOUT NOTICE - you have been warned.

Terms and policies apply to paid and unpaid features of Mailsac.

## 5.2 2. Warnings and Acceptable Use

- Illegal is defined as against the law in the United States, the State of California, and the applicable jurisdictions of the end user (you).
- This website can never be used to break the law.
- You will not send unsolicited spam to or from this service and you will comply with CAN-SPAM as well as applicable international, national, and local spam laws in your jurisdiction and the State of California, United States.
- This website and its overseers may cooperate with legal enforcement authorities to uphold law enforcement, without necessarily notifying the user of such coorporation.
- This website can never be used for malicious activity, terrorist activity, planning illegal activities, discussing illegal activities, gambling, pornography, or condoning illegal activities.

- This website can never be used to spread spam, untruth, gambling, pornography of any kind whether legal or illegal.
- Apparent terrorism or threatening behavior, harmful messages or usage to conduct intolerant or hateful or discriminatory actions, whether carried out, intended, or simulated, proven or unproven, will be blocked and removed without notice, and potentially handed over to the proper authorities.
- This website RESERVES THE RIGHT TO BLOCK OR DELETE MAIL WITHOUT ANY NOTICE OR REASON, INCLUDING DATA RETAINED BY PAID USERS. The user agrees to take action outside Mailsac services to engage in a backup strategy, when persistence is desired or required. ALL MAILSAC DATA, INCLUDING EMAIL CONTENTS AND EMAIL ADDRESSES, MUST BE CONSIDERED EPHEMERAL BY THE USER.
- You agree to be a good citizen of this service and will not engage in any activity which degrades the quality, experience or integrity of it, and accept that if we determine you are degrading the service for others, your usage may be limited in a way which may be indesirable to you.
- This service reserves the right to delete or block any email for any reason without notice and without stating any reason.
- Unless otherwise indicated, by using the service you acknowledge that all of your incoming emails are public, not private, and unsecured, and not under copyright or ownership.
- This service makes no guarantees about how long email will be retained, neither minimum nor maximum, or that email will be accepted or received.
- This service is offered for use with NO WARRANTIES and MAKES NO PROMISES THAT IT WILL FUNCTION AS INTENDED OR UNINTENDED. There is no "service level agreement" nor any guarantee of service uptime. We provide uptime system metrics, such as https://status.mailsac.com, on a purely information basis. Nor is status or uptime information required to be available.

## 5.3 3. Commercial and Non-Commercial Use

3a. Personal (where personal is non-business and non-commercial) use of this website is free, though it may require the user to register for an account and provide information such as a legal name, username, email address, phone number, physical address, and password. We will store the password using a one-way hash which cannot be retrieved by Mailsac.

3b. Using this service for any business operations, even once, necessitates the purchase of a subscription-based license. Incorporated or non-incorporated business use is prohibited without a paid recurring active commercial subscription.

3c. An exception the previous clause (3c) is made for subscriptions and services containing the text "trial". However, "trial" usage will require the business to abide by these terms, the acceptable use guidelines, and the Privacy Policy (separate agreement).

## 5.4 4. Email

4a. Public and Private Email

This site provides public email services and mixed public-private email testing. Email inboxes that are not purchased (aka private) are public and have no privacy whatsoever.

Private addresses will not be accessible publicly and access to messages will be provided only to those with: - a validated username and password, which grants a temporary access "session" which may be stored in a cookie - a validated API key (smtp key, api access key, or other key)

If you believe your username or password has been compromised, you must contact the support email address immediately. This website does not accept responsibility for damages done by compromised accounts. Users acknowledge that overseers and associates (internal or external parties) may have access to inbox contents for the purpose of ongoing business operations, but that access and insight will be limited based on direct need (billing help, troubleshooting, and development comprise the standard reasons for such access, but others may apply). We may aggregate information included in emails, inbound and oubound metadata, email attachements, or email body information, for sale or sharing with third parties or customers.

The Privacy Policy outlines additional detail around sharing of public and private email. When in doubt, THE USER ASSUMES CONTENTS OF THEIR DATA ON MAILSAC.COM MAY BE SHARED WITH OTHER PARTIES. Mailsac is not a service for conducting sensitive or private matters.

4b. Receipt and Message Accuracy Guarantees

Messages receipt is not guaranteed. While this service does not intent for messages to be altered, there may be arbitrary changes whether during transit or storage which are made without knowledge of the user. The user's account username and IP address may be attached to inbound or outbound email messages.

We cannot guarantee accuracy of any email content. Furthermore this service does not know the quality, truthfulness, correctness, or origination of messages received. Users acknowledge this and will not hold the website responsible for the content of messages. Users acknowledge that outside parties often send viruses and intentionally misleading content (spam) in email messages, and Mailsac is unable to accurately determine whether messages are spam. Spam ratings and spam scores are for informational purposes only - the user acknowledges these may indicate non-spam is spam, and spam is non-spam.

4c. Email Security

The user acknowledges:

- SMTP and its variants are insecure by design
- SMTP is a fail-prone protocol
- SMTP is a "legacy" technology which lacks modern security best practices
- the SMTP protocol is not guaranteed to be transmitted under encryption
- when encrypted in transit, email may use broken or insufficient encryption algorithms
- stored emails may not be encrypted at rest
- third parties (including but not limited to routers, ISPs, intermediary providers, email providers including Mailsac) may decrypt and store email, even though it was not the message's final destination

4d. Outbound Mail

Outgoing mail may be purchased at cost and may be delivered/relayed by a third party OR Mailsac. The user and sender of this mail is responsible for the content they send. Delivery cannot be guaranteed. Messages which fail to be not be delivered will not be refunded. Contact Mailsac support if delivery appears to be failing and we may be able to assist with troubleshooting, but again we make no guarantees about outbound message delivery.

Outgoing mail may be marked as spam by receivers or intermediary deliverers. Mailsac is not responsible for email delivered by Mailsac being marked as spam, and WILL NOT refund such mail which is marked as spam or undelivered or unviewed.

Sending spam email is a serious offense.

IN THE EVENT THAT LEGAL EXPENSES OR FINANCIAL JUDGEMENTS ARE INCURRED FOR THE SENDING OF EMAIL MESSAGES BY A MAILSAC USER, THE USER AGREES TO ACCEPT FINANCIAL RESPOSIBILITY FOR THE PAYMENT OF THOSE FEES OR JUDGEMENTS. WHETHER SENT INTENTIONALLY OR UNINTENTIONALLY BY THE MAILSAC ACCOUNT OWNER, THE USER ACCEPTS FULL RESPONSIBILITY FOR THE MESSAGES SENT. The user is responsible for mail sent from their account even during a period that

the account was compromised. It is the responsibility of the account holder (user) to secure their account and contact support via when it appears to have been compromised.

## 5.5 5. Throttling

Inbound and outbound email traffic is throttled due to various dynamic factors, including but not limited to: IP addresses, email addresses, domains, frequency of inbound or outbound mail. Banning traffic will happens permanently or temporarily. Entire domains and companies may be banned. For an up-to-date list of banned traffic, contact Mailsac support.

We reserve the right to throttle API access, even for paid customers. Throttling is necessary to ensure quality of service for all other customers.

## 5.6 6. Downtime

This service follows a best-effort policy to maintain uptime. No service level agreement exists, for any customer, paying or non-paying, personal or commercial.

Customers wanting service uptime guarantees should contact sales.

## 5.7 7. Accuracy and Compatibility

7a. Mailsac provides an API service, with free and paid tiers. This service is not responsible for adverse effects in any situation, and offers no compatibility guarantee. We reserve the right to change the API without notice. No guranatees are made on the following:

- API downtime
- API changes
- API incompatibility
- API support for a particular programming language
- API help in any language other than English

7b. We make our best effort to produce bug free software and accurate documentation. However the user acknowledges that the service may perform sub-optimally and in a way that is unexpected, and in such cases Mailsac is not responsible. We make no guarantees that the service will function as described or intended, but we hope it does so. There is no guarantee that documentation about the service, including the API or email routing or email privacy, will be accurate.

## 5.8 8. Data and Retention

8a. Deleted data (including messages/email) are removed from the servers and the databases using standard deletion practices. However we cannot guarantee that backups may not exist or that business partners (such as hosting companies) will delete this data. Services which crawl publicly available content (most emails on this website are public) may retain that information indefinitely and this website is not responsible. This website makes no guarantees about retention practices but intends to have deleted content be deleted. Standard application logs and failed attempts to pass data between internal services may be retained for troubleshooting.

Saved or “starred” messages are intended to be retained up to the indicated limits on an account.

The user agrees that Mailsac is never responsible for lost data. It is the user's responsibility to backup or retain any data they may wish to keep.

If data retention or secure deltion is a concern it is recommend that the user should not use this service, and find another service upon which to rely.

8b. Usage metrics are tracked. Usage includes but is not limited to:

- IP addresses of site users and API users and SMTP connections
- IP addresses or user accounts of derivable data from service activities

Analytics on received SMTP messages are recorded. We may track which users or IP addresses view which email addresses and messages. All analytics data may be shared with third parties including being packaged and resold, or made available via API to paying or non-paying customers. This data can be used internally to study spam, or learn about users, to provide more useful services, and to sell as a product for customers to derive value.

## 5.9 9. Refunds

Refunds are not provided. All purchases are final. Refunds may be provided at the discretion of Mailsac and its overseers. As a general rule, refunds are never given for any reason. In the event the service is down or ceases to operate, any unused services and remaining service purchases are not refunded. We make exceptions on a case-by-case basis with no guarantee as to the methods for determination. ALL SALES ARE FINAL WHETHER SERVICES HAVE BEEN RENDERED OR NOT.

## 5.10 10. Updates

These Terms may change from time to time and all updates will be posted on this website. Users of the site WILL NOT BE NOTIFIED OF CHANGES. WE HIGHLY RECOMMEND USERS REGULARLY CHECK THIS WEBSITE FOR CHANGES TO THE TERMS OF SERVICE, ACCEPTABLE USE POLICY, AND PRIVACY POLICY.

If the lack of notification of terms changes presents a concern, contact support for a different arrangement.

## 5.11 11. Privileges

This service retains the right to revoke or deny access to anyone at anytime, with or without stated reason. Likewise any user may cease using the service and request that data be removed, in accordance with the data and retention policies outlined herein and in the jurisdictions indicated herin. As indicated above, refunds are not given.

## 5.12 12. Other Agreements

If a clause of this agreement is found to be invalid or violated, the rest of this agreement still stands.

This agreement represents the entire agreement between the user and Mailsac, which includes the Privacy Policy. The two parties may supersede parts of this agreement through writing signed by legal representatives of both parties. Clauses of this agreement not addressed in any superseding agreement will still stand.

CHAPTER 6

# Services Introduction

To get the most out of Mailsac, an account and API key are required. Getting a paid plan will entitle you to support. Once you are signed up with Mailsac you can start exploring the services provided:

- *Message Storage*
- *Email Attachments*
- *Email Hosting*
- *Catch-All*
- *Receive Email via JSON Webhook*
- *Mail Forwarding*
- *Sending Mail*
- *Alternate Address Redirect*
- *Receive Pushed Email via WebSocket*

CHAPTER 7

# Message Storage

When you have an account and make an email address private, message storage kicks in.

Message storage comes into play in two ways. It governs:

- The number of messages allowed across your owned email addresses
- The number of starred messages

## 7.1 When are Messages Recycled?

The **oldest, unstarred** messages are removed.

### 7.1.1 Example

If your storage limit is 100 messages, and you star 25 messages, and you own 3 email addresses, you'll be allowed 75 messages across those 3 email addresses.

Once you have 100 messages across all your starred and private email addresses, the oldest unstarred ones will be removed.

### 7.1.2 Getting More Storage

Message storage can be purchased on a per-message basis in blocks listed on the pricing page.

## 7.2 Starred (saved) Messages

Mailsac will delete messages on regular intervals. One way around this is to create an account and star messages so they will be saved.

Use the star symbol

When you star a message, it counts towards your overall storage. But once it's starred, you can be sure it will not get recycled, so you can refer back to it later.

Your user dashboard has links to all inboxes where you starred messages.

You can star messages on any address, including those you own or other public addresses.

CHAPTER 8

# Email Attachments

By clicking the Download button on any message, you can open in locally in an email client and view the attachements.

For security and to ensure the Mailsac website is not marked as a distributor of spam, attachments are not accessible from the web user interface.

Attachments can be parsed out of the raw message using the API.

Attachments on outgoing messages are only supported through the API.

CHAPTER 9

# Email Hosting

Mailsac will provide free email hosting for you domain. These email addresses will be public. If you require a domain to be private or to setup forwarding of all emails received to single address you can sign up for those services.

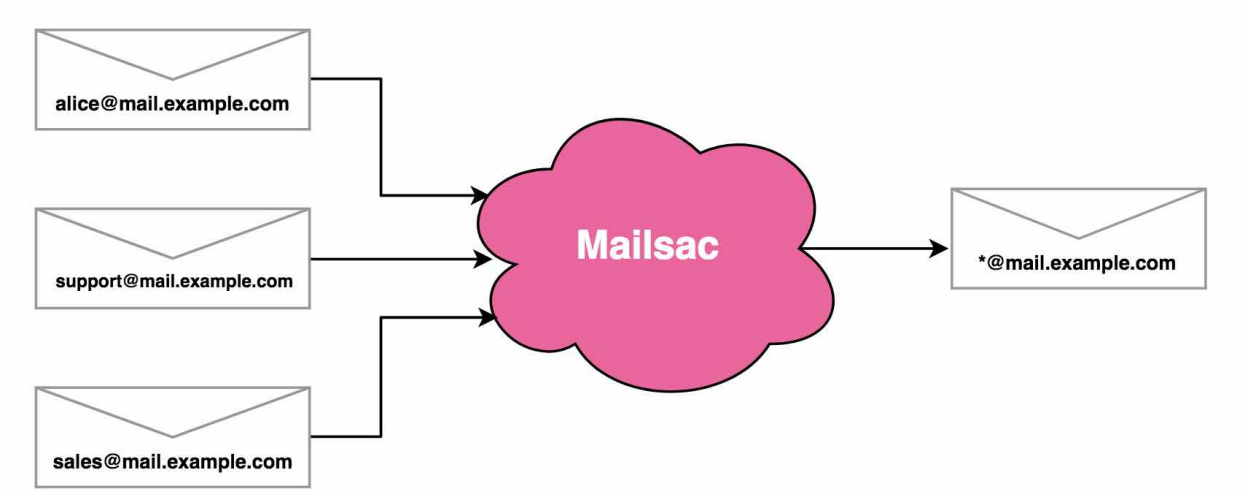
## 9.1 Configure Email Hosting

In your domain's DNS configuration, you will need to create or modify the MX records.

| Priority | Host | Value |
|---|---|---|
| 1 | @ | in.mailsac.com |
| 5 | @ | alt.mailsac.com |

CHAPTER 10

# Catch-All Email Address Setup




You can redirect all email on the domain to a single "Catch-All" address, no matter which inbox it was sent to.

This is useful if you are doing testing, or want to give our multiple email addresses that are all checked by one person.

## 10.1 Steps

In the steps below, replace yourdomain.com with your real domain. It can be a subdomain (mail.yourdomain.com).

1. *Create DNS MX records* pointing email for yourdomain.com to the inbound Mailsac servers.
2. Verify your domain. This may take a few minutes – it's simple, but you must add two DNS TXT records.
3. Go to My Custom Domains / Manage / Forwarding Tab.
4. Make sure Catch-All is Enabled.

5. You will now be able to setup a forwarding address (click Forwarding button next to the email address), or just check mail at the * inbox, like *@yourdomain.com

5. You could also use POP3 to pull the emails into GMail, Apple Mail, Thunderbird, or any other email client that supports POP3.

CHAPTER 11

# Receive Email via JSON Webhook

Private addresses (or Catch-All addresses) can be redirected to a URL of your choice, also called a webhook.

To enable webhooks for an email address, select Edit next to the address in the dashboard, then enter a URL under the 'Forward to Webhook' section, and click Save.


```
{"text": "this is a message",
"headers": {
     "dkim-signature": "redacted",
     "received": "2017-05-01T05:22:03.000Z",
     "content-type": "text/plain",
     "from": "mastadon@mailsac.com",
     "to": "gemini@mailsac.com",
     "subject": "",
     "message-id": "",
     "list-unsubscribe": "",
     "content-transfer-encoding": "7bit",
     "date": "Wed, 1 May 2017 05:22:03 +0000",
     "mime-version": "1.0"
},
"subject": "twin",
"messageId": "MS-5412430010104145004240335343@mailsac.com",
"priority": "normal",
"from": [{
     "address": "mastadon@mailsac.com",
     "name": ""
}],
"to": [{
     "address": "gemini@mailsac.com",
     "name": ""
}],
"date": "2017-05-01T05:22:03.000Z",
"receivedDate": "2017-05-01T05:22:03.000Z",
"originalInbox": "gemini@mailsac.com",
"inbox": "gemini@mailsac.com",
"domain": "mailsac.com",
```


(continues on next page)

(continued from previous page)


```
"encryptedInbox": "inbox-db54b7f5f09041165aaaaae5ca9557c806cf3d95f8",
"raw": "DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=mailsac.com;\r\n␣
↪q=dns/txt; s=mailsacrelay;\r\n bh=redacted;\r\n h=from:subject:to:mime-
↪version:content-type:content-transfer-encoding:list-unsubscribe;\r\n␣
↪b=redacted\r\nReceived: from localhost (127.0.0.1) by mailer with SMTP; Wed May␣
↪1\r\n 2017 01:22:03 GMT-0400 (EDT)\r\nContent-Type: text/plain\r\nFrom:␣
↪mastadon@mailsac.com\r\nTo: gemini@mailsac.com\r\nSubject: twin\r\nMessage-ID:\r\n␣
↪\r\nList-Unsubscribe: \r\nContent-Transfer-Encoding: 7bit\r\nDate: Wed, 1 May 2017␣
↪05:22:03 +0000\r\nMIME-Version: 1.0\r\n\r\this is a message",
"received": "2017-05-01T05:22:04.851Z",
"_id": "IhnBcuaaiC",
"ip": "127.0.0.1"
}
```


Webhook contents are very similar to web socket emails.

Read more about JSON Webhooks at the API Reference

CHAPTER 12

# How to Setup Email Forwarding

After purchasing a private address, you can forward mail to another address. (Forwarding is not available for public disposable email addresses.)

## 12.1 Forward to Other Email Address

Any email address you own may be forwarded to another email address you own. This allows you to consolidate many email addresses into a single inbox.

For example, if you own 1@mailsac.com, 2@mailsac.com, 3@mailsac.com, and main@mailsac.com, you could setup the following scheme:

- 1@mailsac.com forwards to main@mailsac.com
- 2@mailsac.com forwards to main@mailsac.com
- 3@mailsac.com forwards to main@mailsac.com
- main@mailsac.com is checked by POP3 in GMail

## 12.2 Catch-All Domain Forwarding Addresses

A *Catch-All email address* can receive all the mail for your custom domain that is hosted on Mailsac, and optionally forward it to another address. Catch-All inboxes are just an asterisk and the domain, like *@example.com.

## 12.3 Alternate Forwarding

By writing a little *code*, you can forward emails to a *Webhook* or a Web Socket.

CHAPTER 13

# Sending Mail

Sending mail, both replies and new messages, is available only if you have an account and are logged in. It takes a few seconds to create an account. Sending or replying requires mail credits.

## 13.1 Sending From mailsac.com or Custom Domain

You can send from any unowned address @mailsac.com, or send from your account's owned domains.

If somebody owns an email address, you will not be able to send mail as that address. Thus, to prevent other people from sending messages as your favorite address, you can make that address private.

## 13.2 Sending From a Custom Domain

First, verify ownership of your domain to send from your custom domain.

## 13.3 Sent Messages Are Not Saved

Outgoing messages are not saved. They may be visible or cached temporarily by our outgoing mail services, and logged in debugging messages on Mailsac servers, but not explicitly archived by Mailsac at this time.

CHAPTER 14

# Alternate Address Redirect

One of the downsides of public disposable email is that when you give out the address, and people know it is disposable, they might know they can view your mail!

But there's a way around that. You can still keep your inbox private.

## 14.1 The 'redirect' or 'alternate' Address

Underneath each email address, on the main inbox page, is a special redirect address.

The alternate email address begins with"**inbox**-" and has a long string of characters after it.

Any mail sent to the alternate/redirect address will be delivered to the actual address on the page.

**Redirect emails work for custom domains** as well. The alternate address will always display the domain as "@mailsac.com" – which will work – **but you can also change the domain to your custom domain (or any domain** that receives mail at Mailsac).

For example, if the alternate address for your email, dude@example.com is inbox-55443445kk4Knxns@mailsac.com, you can use that one, or replace the domain with your own. It would be inbox-55443445kk4Knxns@example.com.

Of course, Mailsac offers private inboxes, too. If you create an account, you'll be able to see the options for getting private mail. Then you can give out your actual email address without worrying whether people know it is public.

CHAPTER 15

# API Guide Prerequisites

There are only two programs you need to get started:

- curl
- jq

If you are using Linux or OS X curl is likely already installed. The jq download page provides installation instructions for most operating systems.

Listing 1: For operating systems using yum

```
$sudo yum install curl jq -y
```

Listing 2: For operating system using apt

```
$sudo apt-get install jq curl -y
```

**Tip:** Windows users can use chocolatey to install jq and curl

# CHAPTER 16

## Check Mail

Now that curl and jq are installed, we can start interacting with Mailsac API. The Mailsac API Reference includes all supported endpoints. The API Reference is great starting place, if you a familiar with REST APIs, or for reference after completing this step-by-step introduction.

In this example, we are going to check an arbitrary email address for mail, read that email and respond to the email.

We will list inbox email messages for *user1@mailsac.com*. To list the available messages we will use the List Inbox Email Messages endpoint.

---

**Tip:** API documentation is generalized. Modifications are needed to translate an API endpoint into a usable URL. The base URI of all Mailsac API requests will be https://mailsac.com.

---

This endpoint can be accessed with `GET /api/addresses/:email/messages`. You will substitute *:email* with *user1@mailsac.com* giving us `GET /api/addresses/user1@mailsac.com/messages`. Curl does not encode URLs. The @ character needs to be URL encoded as *%40*. `GET /api/addresses/user1%40mailsac.com/messages`. The base URI of the Mailsac API is Mailsac.com, which translates to `https://mailsac.com/api/addresses/user1%40mailsac.com/messages`

---

**Tip:** You can validate the url is properly formatted by accessing it in your web browser. Go ahead and try it with our example or try it with a different email address.

---

Curl requires us to add a few extra parameters. *-X GET* instructs curl to us a HTTP GET request. *-s* suppresses a progress bar. In the command below we pipe the contents of curl into JQ for JSON formatting. JQ requires a filter to function. We are using the simplest filter *"."* which matches all JSON. *user1@mailsac* is a popular address and receives lots of email. JQ will only show the first JSON object with the filter *".[0]"*

Listing 1: **Retrieve first email in a an inbox.**

```
$ curl -s -X GET https://mailsac.com/api/addresses/user1%40mailsac.com/messages | jq
↪".[0]"
```

Listing 2: **Inbox message**


```
{
  "_id": "BotvTxaona7gLID1Adtpfj8Fnfi7HSSv-0",
  "from": [
    {
      "address": "microsoftstore@e.microsoft.com",
      "name": "Microsoft Store"
    }
  ],
  "to": [
    {
      "address": "user1@mailsac.com",
      "name": ""
    }
  ],
  "cc": null,
  "bcc": null,
  "subject": "Ahoy, Sea of Thieves for PC is here",
  "savedBy": null,
  "originalInbox": "user1@mailsac.com",
  "inbox": "user1@mailsac.com",
  "domain": "mailsac.com",
  "received": "2018-03-29T18:28:07.732Z",
  "size": 23420,
  "attachments": null,
  "ip": "65.55.234.211",
  "via": "144.202.71.79",
  "folder": "inbox",
  "labels": [],
  "read": null,
  "rtls": true,
  "links": [
    "href=https://e.microsoft.com/Key-3567701.C.CQZpy.C.K0.-.CMHlNS",
    "http://msstorepromoemail.blob.core.windows.net/windows-store-edits/Nav_MSFT_Logo.
↪jpg",
  ],
  "spam": 1.3370381090039505e-09
}
```


As you can see, the JSON contains information about the email message including to address, from address, subject, time stamp, attachments and much more. Make note of the *_id* field, you will use it to view the contents of the email.

# CHAPTER 17

# Read Mail

To read a plain text email message you use the text endpoint. This endpoint can be accessed by `GET /api/text/:email/:messageId`. You will substitute in the email address the email was sent to and the message ID for that email. Using the values from the previous section will yield, `GET /api/text/user1%40mailsac.com:BotvTxaona7gLID1Adtpfj8Fnfi7HSSv-0`. The email message can be retrieved using curl.

Listing 1: **Retrieve text of message**

```
curl -s -X GET https://mailsac.com/api/text/user1%40mailsac.com/
↪Jn1wa9AwLigQwIbwUGyMMollJkeWSeUd-0
```

Listing 2: **Plain text message**

```
Enjoy odoo partnership program and success pack sales combo! *
Summer is around the corner! Odoo wishes  to save your time and effort from␣
↪complicated paperwork for more outdoor activities!    Therefore, we are offering an␣
↪exclusive discount for Partnership in MAY.
Partnership program pricing at USD 2950
*  and save USD 1000* on your joining!
*  *The discount offer is only applicable for the purchase with a success pack.


Schedule a demo with
Odoo Partnership Expert* BOOK NOW [2]
Our Partnership
Program*  READ MORE [3]
* Our Worldwide Partners*  READ MORE [4]
* Our Customers success story*  READ MORE [5]
None [6]  None [7]  None [8]  None [9]
Unsubscribe [10] | Contact [11]
2018 All Rights Reserved



[1] https://www.odoo.com/r/erT/m/12928017
```

(continues on next page)

(continued from previous page)

```
[2] https://www.odoo.com/r/X6Q/m/12928017
[3] https://www.odoo.com/r/so8/m/12928017
[4] https://www.odoo.com/r/fya/m/12928017
[5] https://www.odoo.com/r/jlI/m/12928017
[6] https://www.odoo.com/r/dlt/m/12928017
[7] https://www.odoo.com/r/25S/m/12928017
[8] https://www.odoo.com/r/Vcu/m/12928017
[9] https://www.odoo.com/r/nKf/m/12928017
[10] https://www.odoo.com/unsubscribe_from_list
[11] https://www.odoo.com/r/rox/m/12928017
```

# CHAPTER 18

## Sending Mail

**Important:** Sending messages requires the purchase of outgoing message credits.

Sending an email uses the outgoing-messages endpoint. This endpoint is accessed with `POST /api/outgoing-messages`. This API uses a POST method, unlike our previous two examples, which use GET. Several pieces of information are required to send an email.

- Mailsac API Key
- To address
- From address
- Text

Since emails without subjects frequently get marked as spam, we are also going to include a subject in our email. Our email message will be transmitted in JSON, therefore we have to set the content type to :`Content-Type: application/json`. Our message data will be a comma separated key value array.

Listing 1: **Send an email**

```
curl --header "Content-Type: application/json" --request POST \
--data '{"_mailsacKey":
↪"eoj1mn7x5y61w0egs25j6xrvs61wrrld0oh43rj583cgdps10tokp2ceux9s6ri8eoj1mn7x5y6", \
"to": "recipient@gmail.com", "subject": "This is a test", "from": "sender@mailsac.com
↪", \
"text": "This is a test"}' https://mailsac.com/api/outgoing-messages
```

CHAPTER 19

# API Example Overview

API interaction is at the core of Mailsac. The examples in this section will provide you with some boiler plate usages of Mailsac. But we have users who email Mailsac with new and innovative ways to use the Mailsac API.

**Tip:** All API endpoints can be found in the API Documentation

CHAPTER 20

# Delivery Confirmation Example

Consider a scenario, where you have an automated process that sends emails. You may wish to test that this automated process is sending emails and they are being received. Mailsac provides an API so you can check that these emails are being received.

## 20.1 Prerequisites

- Python3
- Mailsac API Key

This python example sends emails to user1@mailsac.com through user10@mailsac.com using an SMTP server. Your mail server may or may not use authentication, in this example we are using authentication with TLS.

```
""" This script is an example of sending an email via smtp
and confirming its receipt using the mailsac api"""

import time
import sys
import smtplib
import email.utils
import requests
from email.mime.text import MIMEText

"""
Checks if a message from a given address in the a specific mailsac
inbox. If it is it returns when the message was recived, if not it returns
a message stating the message was not received"""
def check_received(receive_address, send_address, base_url, headers):
    api_url = '{0}/addresses/{1}/messages'.format(base_url, receive_address)
    response = requests.get(api_url, headers=headers)
    for message in response.json():
        if message['from'][0]['address'] == send_address:
            return message['received']
```

(continues on next page)

(continued from previous page)

```
        return '{0} has not received an email from {1}'.format(receive_address, send_
↪address)


SMTP_USERNAME = 'mysmtp_user'
SMTP_PASSWORD = 'mysmtp_password'
SMTP_SERVER = 'mysmtp_server.company.com'
SMTP_PORT = 587
FROM_ADDRESS = 'myuser@company.com'
FROM_NAME = 'MyTest User'
SUBJECT = "Testing email to mailsac"

API_TOKEN = 'MY_API_TOKEN_FROM_MAILSAC'
BASE_URL = 'https://mailsac.com/api/'

BODY_TEXT = ("Mailsac SMTP Validate Emila Send\r\n"
             "This email was sent using the SMTP to test receipt of an email."
            )

for x in range(1, 10):
    try:
        to_address = 'user{}@mailsac.com'.format(x)
        msg = MIMEText(BODY_TEXT)
        msg['Subject'] = SUBJECT
        msg['From'] = email.utils.formataddr((FROM_NAME, FROM_ADDRESS))
        msg['To'] = to_address

        conn = smtplib.SMTP(SMTP_SERVER, SMTP_PORT)
        conn.set_debuglevel(True)
        conn.ehlo()
        conn.starttls()
        conn.ehlo()
        conn.login(SMTP_USERNAME, SMTP_PASSWORD)
        try:
            conn.sendmail(FROM_ADDRESS, to_address, msg.as_string())
            conn.close()
        except Exception as e:
            print("Error: ", e)
        else:
            print("Email Sent!")
    except Exception as ex:
        print(ex)

time.sleep(30)

for x in range(1, 10):
    print(check_received('user{}@mailsac.com'.format(x), send_address=FROM_ADDRESS,
                         base_url=BASE_URL, headers={'Mailsac-Key': API_TOKEN}))
```

CHAPTER 21

# Download All Inbox Attachments

This is a step by step coding guide for downloading all of the attachments in a Mailsac email inbox.

## 21.1 Prerequisites

- Node.js
- Mailsac API Key

## 21.2 Setup

Choose a folder to work in, and navigate to the folder in your terminal.

Configure the npm package.json:

```
npm init -y
npm install --save request request-promise-native
```

Create a file:

```
touch download-inbox-attachments.js
```

Now open the .js file and put the following code in. Change the string value of MAILSAC_API_KEY from 'YOUR API KEY GOES HERE' to your actual API key.

```
const fs = require('fs')
const request = require('request');
const requestp = require('request-promise-native')

const MAILSAC_API_KEY = 'YOUR API KEY GOES HERE';
const address = process.argv[2];
```

(continues on next page)

(continued from previous page)

```
requestp(`https://mailsac.com/api/addresses/${address}/messages`, {
        json: true,
        headers: {
            'Mailsac-Key': MAILSAC_API_KEY
        },
    })
    .then((messages) => {
        console.log(`fetched ${messages.length} messages for ${address}`);

        messages.forEach((message) => {
            if (!message.attachments) {
                console.log(`${message.subject.substring(0, 10)}... ${message._id} -␣
↪has no attachments`);
                return
            }
            if (message.attachments) {
                console.log(`${message.subject.substring(0, 10)}... ${message._id} -␣
↪has ${message.attachments.length} attachments`);
                message.attachments.forEach((checksum) => {
                    const file = fs.createWriteStream(`${checksum}.eml`);
                    request(`https://mailsac.com/api/addresses/${address}/messages/$
↪{message._id}/attachments/${checksum}`)
                        .pipe(file);
                });
            }
        });
    })
    .catch((err) => {
        console.error('Something broke!', err.message, err.stack);
    });
```

The script can be run in the terminal like this:

```
node download-inbox-attachments.js address@mailsac.com
```

This Node script does a few things:

1. Takes takes the email address 'address@mailsac.com' as an input argument. You can change this to any email address.
2. It fetches the metadata for all messages in the inbox address@mailsac.com.
3. It loops through the list of messages and prints part of the subject, the unique message identifier, and how many attachments it has.
4. When messages have attachments, the attachment is downloaded into the current folder, using it's MD5 checksum as an identifier.

That's it! Read more about message attachment APIs in the docs.

CHAPTER 22

# Verifying Email Addresses

Mailsac has two API endpoints for confirming the validity of email addresses. A single address or bulk addresses can be checked. This allows verification of an address or group of addresses without sending a mail. It helps avoid bounces.

Email validation is a useful tool for keeping invalid signups from invading your mailing list. If you run software-as-a-service, it can be used to keep users from signing up with anonymous or disposable email addresses.

Unlike our competitors, Mailsac has few limitations or costs for verifying email. You can verify single email addresses, or up to 50 at a time (more coming soon - contact support). We don't charge extra for these types of API requests.

## 22.1 Prerequisites

- Node.js
- Mailsac API Key

## 22.2 Setup

Choose a folder to work in, and navigate to the folder in your terminal.

Configure the npm package.json:

```
npm init -y
npm install --save request request-promise-native
```

## 22.3 Single Address Validation

Here is an example for validating a single email address using Node.js:

```
const requestp = require('request-promise-native')

const MAILSAC_API_KEY = 'YOUR API KEY GOES HERE';
const address = process.argv[2];

requestp(`https://mailsac.com/api/validations/addresses/${address}`, {
        json: true,
        headers: {
            'Mailsac-Key': MAILSAC_API_KEY
        },
    })
    .then((results) => {
        console.log(results[0]);
    })
    .catch((err) => {
        console.error('Something broke!', err.message, err.stack);
    });
```

Running it will produce output like this:

```
$ node validate-single-email.js greg@spam.netpirates.net
{ email: 'greg@spam.netpirates.net',
  validFormat: true,
  local: 'greg',
  domain: 'spam.netpirates.net',
  isDisposable: true,
  disposableDomains: [ 'mailinator.com' ],
  aliases: [ 'mailinator.com' ] }
```

## 22.4 Bulk Address Validation

The bulk address validation endpoint can be used to validate up to 50 email addresses at a time. Here is an example script

```
const requestp = require('request-promise-native')

const MAILSAC_API_KEY = 'YOUR API KEY GOES HERE';
const addresses = process.argv.slice(2);

requestp.post(`https://mailsac.com/api/validations/addresses`, {
        json: true,
        body: {
            emails: addresses
        },
        headers: {
            'Mailsac-Key': MAILSAC_API_KEY
        },
    })
    .then((results) => {
        console.log(results);
    })
    .catch((err) => {
        console.error('Something broke!', err.message, err.stack);
    });
```

Running it will produce the following output:

```
$ node validate-multiple-emails.js greg@spam.netpirates.net jim@mailsac.com
[ { email: 'greg@spam.netpirates.net',
    validFormat: true,
    local: 'greg',
    domain: 'spam.netpirates.net',
    isDisposable: true,
    disposableDomains: [ 'mailinator.com' ],
    aliases: [ 'mailinator.com' ] },
  { email: 'jim@mailsac.com',
    validFormat: true,
    local: 'jim',
    domain: 'mailsac.com',
    isDisposable: true,
    disposableDomains: [ 'mailsac.com', 'sanstr.com', 'totalvista.com' ],
    aliases: [ 'mailsac.com', 'sanstr.com', 'totalvista.com', '45.76.28.175' ] } ]
```

CHAPTER 23

# WebSocket Overview

A Websocket provides a unique way of monitoring the incoming email of a particular address. With a single persistent connection all content from an email address can be forward over the Websocket. This allows your application to be notified of an incoming emails as soon as it arrives. This helps reduce the number of API calls and reduces latency from when the email arrives and your application responds to it.

Want to see it in action? The WebSocket Test Page allows you to show how it works with no programming involved.

**Tip:** You will need an API key and a *private address*

# CHAPTER 24

# Configure Private Address for WebSocket

To use the Websocket feature a private email is required. Private addresses can be purchased individually as part of a API Developer Subscription.

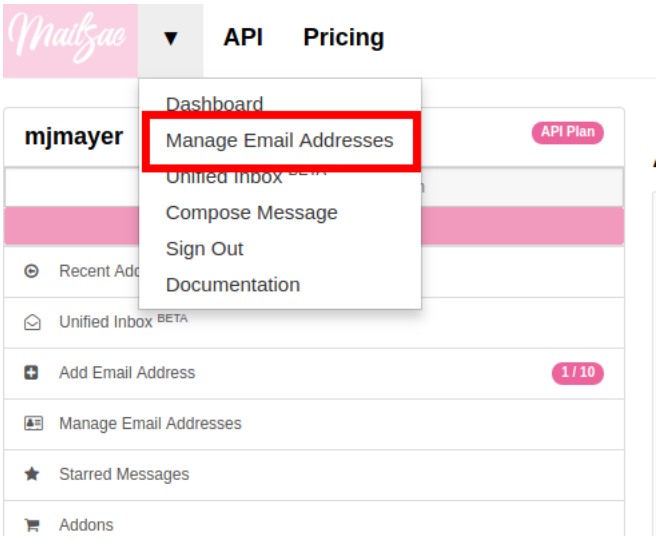
## 24.1 Define Private Address via API

The API is the easiest way to reserve a private address. A simple HTTP POST will do.

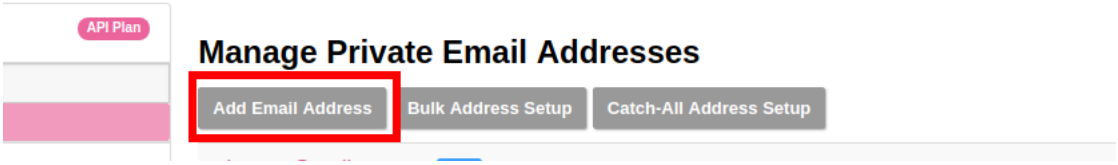
```
curl -s --request POST  https://mailsac.com/api/addresses/user1%40mailsac.com?_
↪mailsacKey=API_KEY
```

## 24.2 Define Private Address in the Dashboard

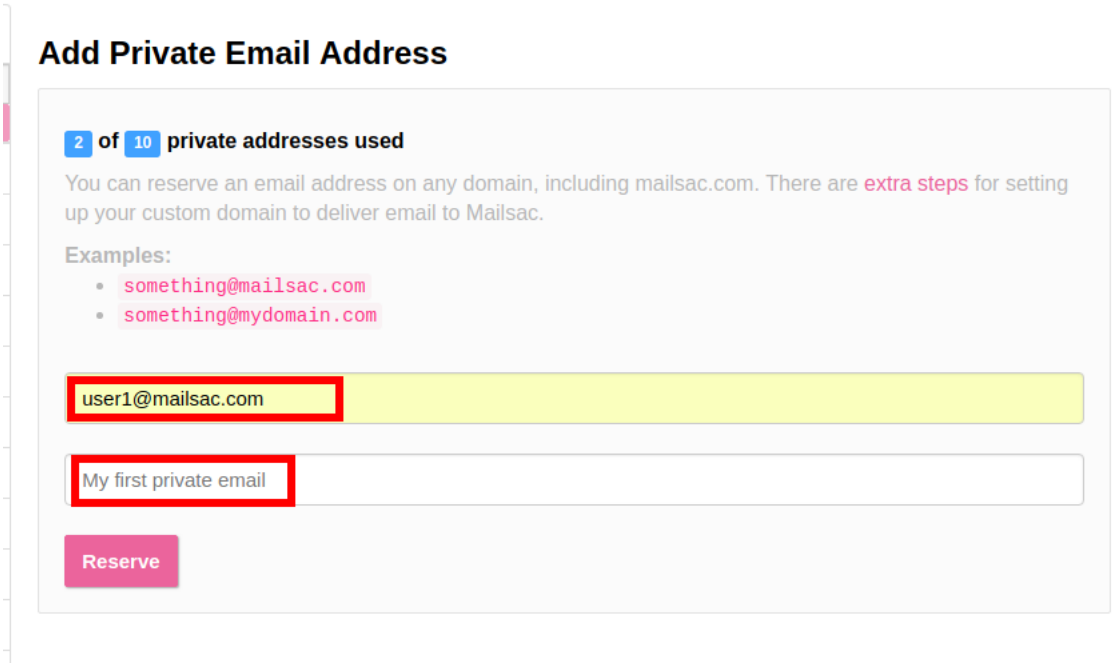
1. Sign in to your Mailsac account.
2. Select Manage Email Addresses from the dropdown.



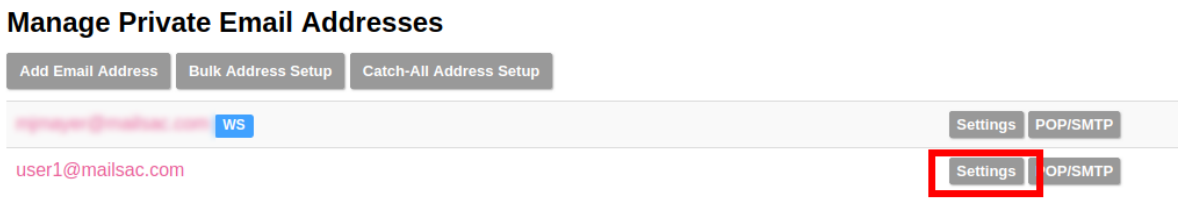

3. Select Add Email Address from Manage Email Addresses



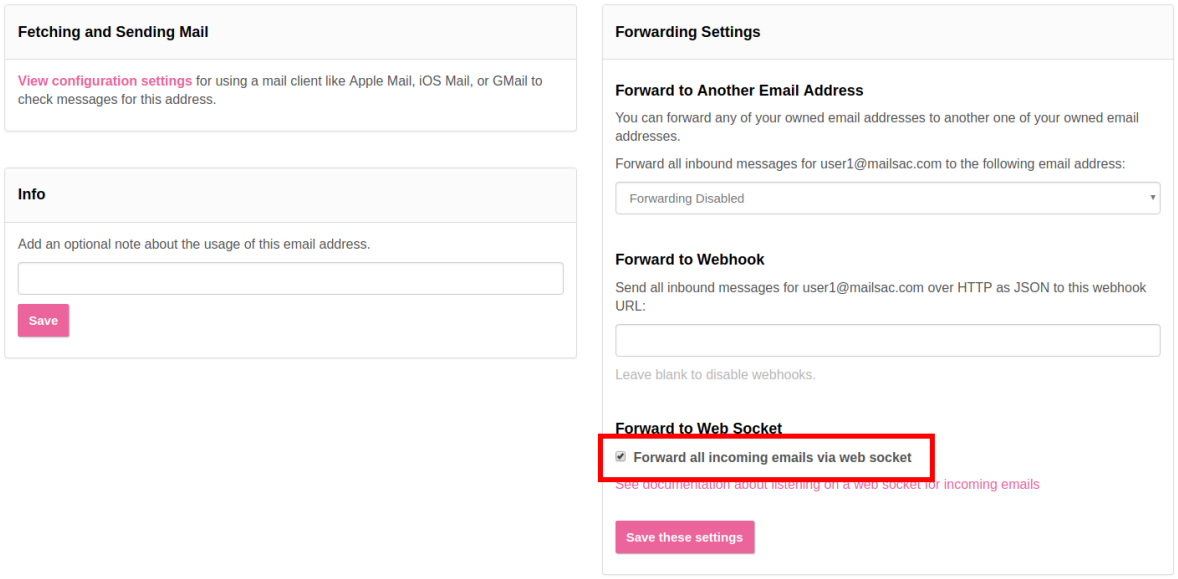

4. Fill in the desired email address and optionally add a note.




5. Select Manage Email Addresses and choose settings next to the email address you want to configure for Websocket.




6. Check the box to “Forward all incoming emails via web socket”



CHAPTER 25

# Receive Mail Using a WebSocket

Receiving mail from a Websocket allows for interacting with incoming email in near real time.

Web sockets are a powerful tool allowing you to end-to-end test your application's email delivery systems, or respond to incoming mail in sophisticated ways - without having to setup a mail server or mess around with SMTP code.

## 25.1 Prerequsites

- Mailsac API Key
- Node.js and npm
- *Private email address with websocket configured*

## 25.2 Setup

Listing 1: **Create directory for example code**

```
$ mkdir websocket-example
$ cd websocket-example
```

Listing 2: **Create package.json file with the following contents**

```
{
  "name": "mailsac-node-websocket-example",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
```

(continues on next page)

(continued from previous page)

```
  "keywords": [],
  "author": "",
  "license": "MIT",
  "dependencies": {
    "ws": "^2.2.3"
  }
```

Listing 3: **Create example.js file with the following contents**

```
const WebSocket = require('ws');
const log = console.log; // eslint-disable-line

// Mailsac uses secure web sockets. This is the web socket API base endpoint.
const BASE_URL = 'wss://sock.mailsac.com/incoming-messages';

// In this example, we pull the username and API key from environment variables.
// You could also hardcode the credentials, or use a package like node-config for␣
↪managing them.
const username = process.env.MAILSAC_USER;
const apiKey = process.env.MAILSAC_KEY;
// List the addresses you want to receive messages for.
// You MUST have web socket forwarding turned on for the addresses!
const listenAddresses = process.env.ADDRESSES;

const urlParams = '?_id=' + username + '&key=' +apiKey+ '&addresses=' +␣
↪listenAddresses;

log('attempting to open web socket to', BASE_URL + urlParams);
const ws = new WebSocket(BASE_URL + urlParams);

ws.on('open', function () {
  log('web socket opened');
});

ws.on('error', function (err) {
  log('connection error', err);
});

ws.on('message', function (data) {
  log(data);
});
```

Listing 4: **Install required node packages**

```
npm install
```

Listing 5: **Set environmental variables**

```
export MAILSAC_USER='your mailsac username / _id';
export MAILSAC_KEY='your mailsac api key';
export ADDRESSES='myaddress@mailsac.com,some-address@example.com'
```

## 25.3 Launch WebSocket Example

Listing 6: **Launch the node program**

```
node example.js
```

Listing 7: **Expected output**

```
attempting to open web socket to wss://sock.mailsac.com/incoming-messages?_
↪id=username&key=apikey&addresses=user1@mailsac.com
web socket opened
{"status":200,"msg":"Listening","addresses":["user1@mailsac.com"]}

{"status":200,"msg":"ok"}
```

Now, when an email messages are delivered to user1@mailsac.com, they will also be sent to your web socket. Try sending a message - it will be parsed into JSON and dumped to your console.