

Implementing rich email client in Python

Vitaly Glibin

Cofounder, CTO

huntflow.ru



Huntflow

VIEW

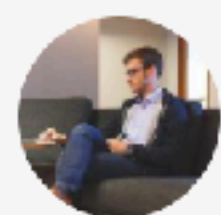
All Applicants

MY VACANCIES

Python Developer
R&D department

Sales manager
Sales department

Filters: All 3, New Lead 1, Submitted, Contacted, HR Interview 1, Client Interview, Offered 1, Offer Accepted, Hired, Trial passed, Declined



Jordan George
Consultant
X5 RETAIL GROUP



Mitchel Carol
Sales manager
X5 RETAIL GROUP

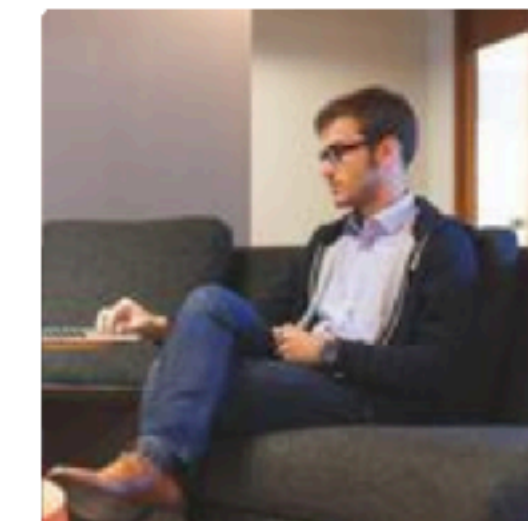


Jackson Peter
Front-end developer
Yandex

APPLY TO VACANCY SEND EDIT

Jordan George

Consultant (X5 RETAIL GROUP)



Phone +7 916 212-85-06
Email hello@huntflow.ru
Age 40 years
Labels ADD

Offered
Sales manager (Sales department)

CHANGE STATUS

Add a Comment



Offered — Offer has been sent to the personal email

ME, TODAY

Applied on Vacancy

ME, TODAY

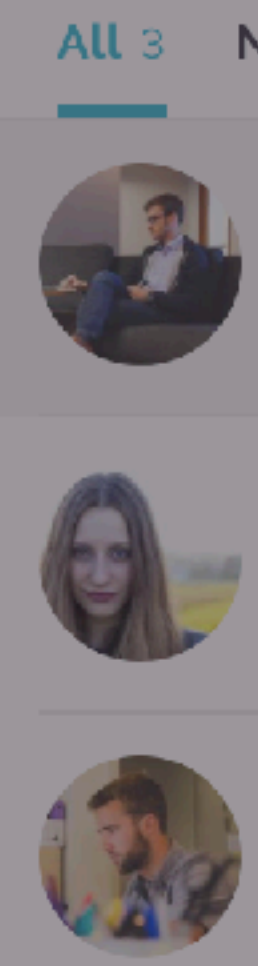
PERSONAL NOTES

HEADHUNTER

SUPPORT



- VIEW
- All Applicants
- MY VACANCIES
- Python Developer R&D department
- Sales manager Sales department



Letter: Jordan George

To

Subject TEMPLATE

Body

Hi George,

We are looking for talented Sales manager at Demo organization.

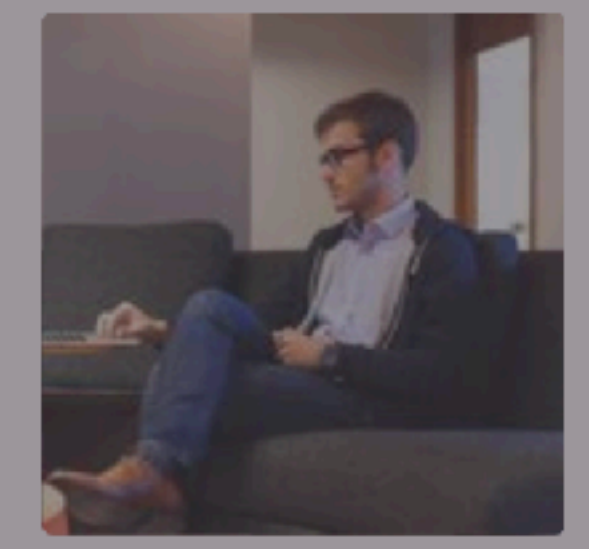
Are you interested to discuss? I will be glad to tell more.

Vitaly Glibin
 Demo organization
 +7 926 073-1778

[Attach file](#)

Send copy to

Create and edit email templates in [Settings](#)



What is an email?

RFC 561 (1973)

RFC 822 (1982)

RFC 2822 (2001)

RFC 5322 (2008)

From: glibin@huntflow.ru

Subject: Hello

Message-Id: <4F77568C-30EC-EDC52EEEDAFAF6@huntflow.ru>

Date: Mon, 16 Oct 2017 22:46:08 +0300

To: glibin.v@gmail.com

Hello, world!

MIME

multipart/alternative

multipart/mixed

multipart/related

and more...

multipart/alternative

Different view of the same content

Виталий Глибин

Test message

Кому: Виталий Глибин

Hello, **world!**

multipart/alternative

--Apple-Mail=boundary

Content-Transfer-Encoding: 7bit

Content-Type: **text/plain**; charset=us-ascii

Hello, world!

--Apple-Mail=boundary

Content-Transfer-Encoding: 7bit

Content-Type: **text/html**; charset=us-ascii

<html><body>Hello, world!</body></html>

--Apple-Mail=boundary--

multipart/alternative

Собеседование: Glibin Vitaly Nikolaevich – Python Developer

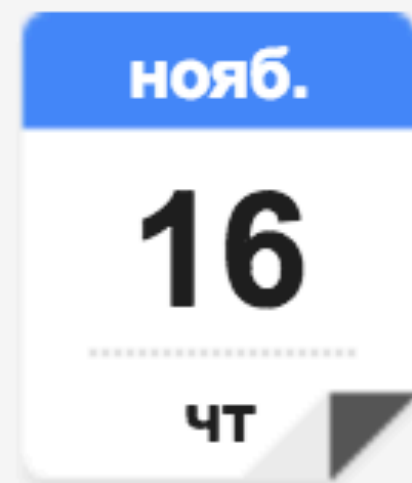


Входящие x



Виталий Глибин

кому: мне ▾



Собеседование: Glibin Vitaly Nikolaev...

[Перейти в сервис "Google Календарь"](#)

Когда чт, 16 нояб 2017 19:00 – 20:00 (MSK)

Кто glibin@huntflow.ru*

Да

Возможно

Нет

--=====5941750684243425195==

Content-Type: **text/calendar**; charset="UTF-8"; method="REQUEST"

Content-Transfer-Encoding: quoted-printable

BEGIN:VCALENDAR

BEGIN:VEVENT

UID:20171102T185654_HF_1813758_987_false

DTSTART:20171116T160000Z

DTEND:20171116T170000Z

ATTENDEE;ROLE=3DREQ-PARTICIPANT:MAILTO:glibin.v@gmail.com

ORGANIZER:MAILTO:glibin@huntflow.ru

STATUS:CONFIRMED

SUMMARY:=D0=A1=D0=BE=D0=B1=D0=B5=D1=81=D0=B5=D0=B4=D0=BE=D0=B2=D0=B0=D0=BD=


=D0=B8=D0=B5: Glibin Vitaly Nikolaevich =E2=80=93 Python Developer

END:VEVENT

END:VCALENDAR

--=====5941750684243425195==--

multipart/mixed

Виталий Глибин 

Test message


Кому: Виталий Глибин

Hello, **world!**



Glibin Vitaly
Nikolae...-3.pdf

multipart/related

Виталий Глибин 

Test message

Кому: Виталий Глибин

Welcome to PiterPy!



multipart/related

--Apple-Mail=_E48801A7-F449-42F2-A4BC-98BF9AE1B905

Content-Transfer-Encoding: 7bit

Content-Type: text/html; charset=us-ascii

<html><body>Welcome to PiterPy!

</body></html>

--Apple-Mail=_E48801A7-F449-42F2-A4BC-98BF9AE1B905

Content-Transfer-Encoding: base64

Content-Disposition: inline; filename=content_piter_py_1280_720.png

Content-Type: image/png; name="content_piter_py_1280_720.png"

Content-Id: **<501C8196-5898-4E1E-9569-2B8771579573>**

All together

- mixed
 - alternative
 - text
 - related
 - html
 - inline image
 - inline image
 - attachment
 - attachment



Mark Crispin test

IMAP protocol author

<https://tinyurl.com/crispin-torture>

Let's Python

`from email import . . .`

Let's Python

```
message = MIMEMultipart('alternative')
message['Subject'] = 'Hello'
message.attach(MIMEText('Hello, world!', 'plain', _charset='utf-8'))
message.attach(MIMEText('Hello, <b>world</b>!', 'html', _charset='utf-8'))

message.as_string()
```

Let's Python

```
message = message_from_string(msg_string)
```

```
if email_message.is_multipart():  
    for part in email_message.walk():
```

```
        ...
```

```
else:
```

```
    ...
```

Send: SMTP / smtplib

```
s = smtplib.SMTP(host, port, timeout=15)
s.starttls()
s.login(login, password)
s.sendmail(message['From'], message['To'], message.as_string())
s.quit()
```

Receive: POP3 / poplib

```
pop = poplib.POP3(host, port)
```

```
pop.user(login)
```

```
pop.pass_(password)
```

```
pop.list()
```

```
pop.retr(msg_num) # bad: would be marked as read
```

```
pop.top(msg_num, bytes_after_header) # better
```

Receive: POP3 / poplib

- Read full inbox only
- One folder
- No way to store sent messages and keep them

Receive: IMAP / imaplib

```
imap = imaplib.IMAP4(host, port)
```

```
imap.login(host, port)
```

```
imap.select('INBOX', read_only=True)
```

```
result, data = imap.search(None, '(SINCE {})'.format(ls.strftime('%d-%b-%Y')))
```

```
if result == 'NO':
```

```
    result, data = imap.search(None, 'ALL')
```

```
imap.fetch(num, '(BODY.PEEK[HEADER.FIELDS (SUBJECT)])')
```

```
imap.fetch(message['uid'], '(RFC822)')
```

Sent messages: IMAP / imaplib

```
result, dirs = imap.list()
sent = None
for folder in dirs:
    if 'sent' in folder.lower():
        if "|" in folder:
            sent = folder.split("|").pop().strip()
        elif "/" in folder:
            sent = folder.split("/").pop().strip()
if sent:
    imap.append(sent, '\\SEEN', imaplib.Time2Internaldate(time.time()),
message.as_string())
```

How to connect?

A simple dialog box titled "Add email". It contains two input fields: "Email" with the text "name@example.com" and "Password" which is empty. At the bottom, there are two buttons: "ADD" and "CANCEL".

```
name, domain = email.split('@', 1)
```

```
out_prefix = ['smtp.', 'mail.']
```

```
in_prefix = ['imap.', 'mail.']
```

A more complex dialog box titled "Add email". It includes the following fields and options:

- "Email": test@example.com
- "Password": masked with dots
- "Type of incoming email": IMAP (dropdown menu)
- "Incoming email server": imap.example.com
- "Port": empty
- Secured connection
- "Outgoing email server": smtp.example.com
- "Port": empty
- Secured connection

At the bottom, there is a red error message: "Unable to automatically detect server address". Below the error message are two buttons: "ADD" and "CANCEL".

Anything else?

GMAIL API

- HTTP REST API
- No need to store a password (OAuth2)
- We can load contacts
- Limits are hard, very accurate

Google

Hi Vitaly



glibin.v@gmail.com

Хантфлоу wants to



View your email messages and settings



Manage drafts and send emails



Send email on your behalf



View your contacts

GMAIL API

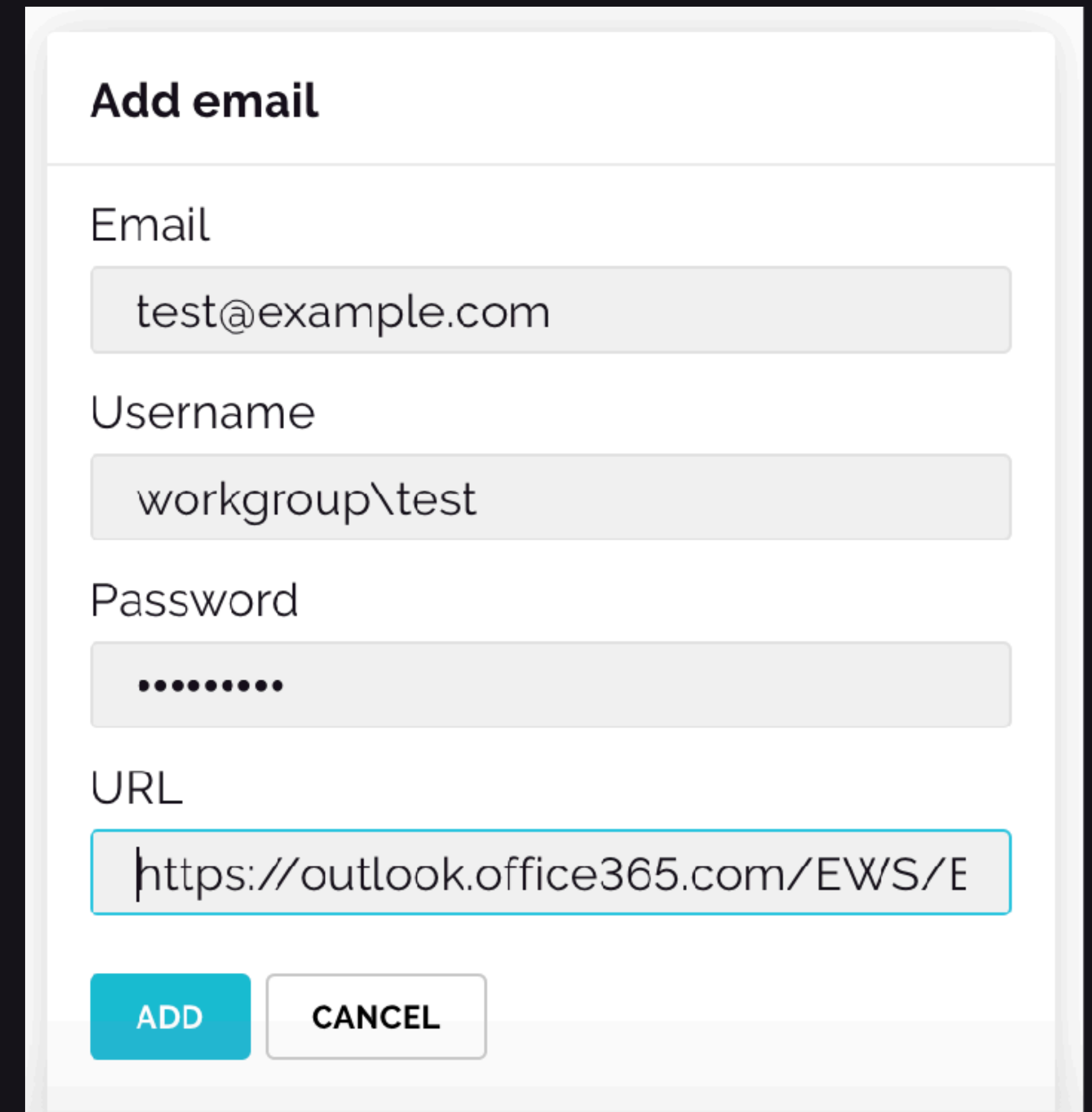
POST /upload/gmail/v1/users/me/messages/send

data=**message.as_string()**

headers={'Content-Type': 'message/rfc822'}

Exchange Web Services

- HTTP XML API
- Two types of authentication (NTLM / Basic)
- One account for email, calendar and contacts



Add email

Email
test@example.com

Username
workgroup\test

Password
.....

URL
https://outlook.office365.com/EWS/E

ADD **CANCEL**

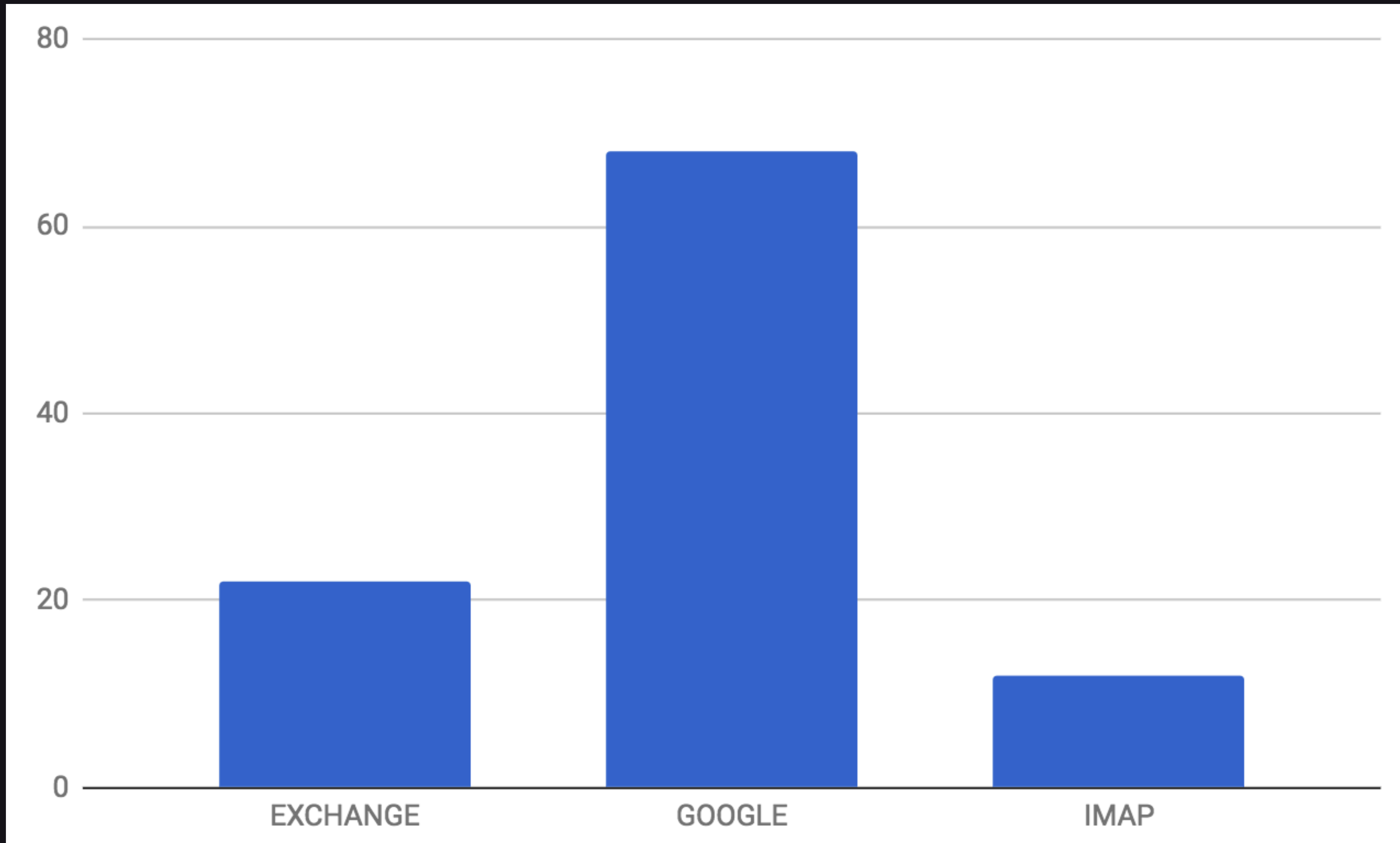
Exchange Web Services

<https://github.com/linkedin/pyexchange>

Last commit in 2015 :-)

Took a fork, fixed everything and

<https://github.com/glibin/pyexchange>



Threads

→ Headers: **In-Reply-To** and **References** (mostly works)

In-Reply-To: <1324591288@github.com>

References: <732@github.com> <1324591288@github.com>

Synchronization

- Store sync time and check every 2-3 minutes
- Store **Message-Id** header to prevent full refetch
- Check **From/To**, search emails in DB and add to applicant history

TODO

Send and receive messages in HTML

→ Markup

→ Security (bleach)

→ Anti tracking

Thanks!

Vitaly Glibin

glibin@huntflow.ru

@glibin

