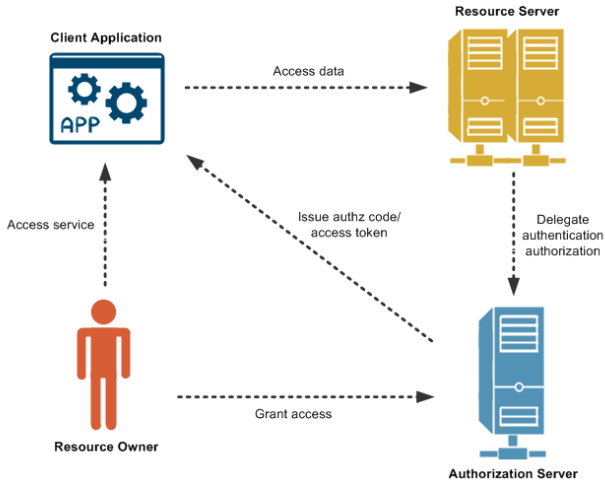
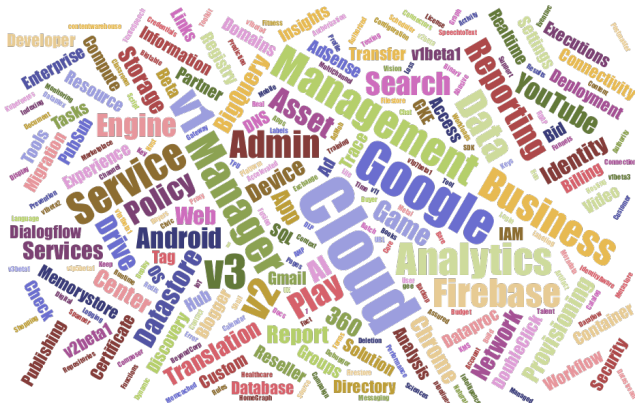


OAuth2.0



Είδη Google API



Τι κάνουν τα Google API



To API to gmail

- Πως στέλνουμε ένα e-mail από ένα Python script;
- Ο παραδοσιακός τρόπος είναι να χρησιμοποιήσουμε την βιβλιοθήκη `smtpplib`.
- Ωστόσο με τον τρόπο αυτό θα πρέπει να βάλουμε κάπου το κωδικό που έχει ο λογαριασμός στον mail server.
- Η Google (πλέον) δεν υποστηρίζει τον τρόπο αυτό (τουλάχιστον στους standard λογαριασμούς)



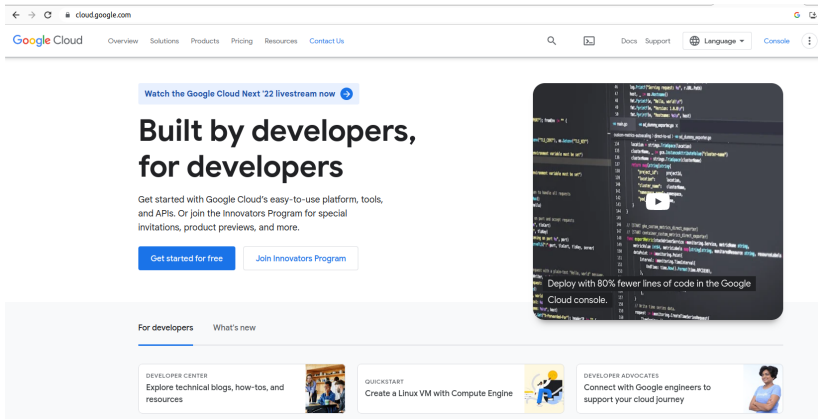
Disclaimer



Disclaimer



Ξεκινάμε από το GCP



The screenshot shows the Google Cloud website homepage. At the top, there's a navigation bar with the Google Cloud logo and links for Overview, Solutions, Products, Pricing, Resources, and Contact Us. On the right, there are search, docs, support, and language options. The main content area features a large headline "Built by developers, for developers" with a subtext "Get started with Google Cloud's easy-to-use platform, tools, and APIs. Or join the Innovators Program for special invitations, product previews, and more." Below this are two buttons: "Get started for free" and "Join Innovators Program". To the right of the headline is a video player showing a code editor with a play button overlay. Below the headline, there are two tabs: "For developers" (selected) and "What's new". Under the "For developers" tab, there are three cards: "DEVELOPER CENTER" (Explore technical blogs, how-tos, and resources), "QUICKSTART" (Create a Linux VM with Compute Engine), and "DEVELOPER ADVOCATES" (Connect with Google engineers to support your cloud journey).

cloud.google.com

Google Cloud Overview Solutions Products Pricing Resources Contact Us

Watch the Google Cloud Next '22 livestream now

Built by developers, for developers

Get started with Google Cloud's easy-to-use platform, tools, and APIs. Or join the Innovators Program for special invitations, product previews, and more.

Get started for free Join Innovators Program

Deploy with 80% fewer lines of code in the Google Cloud console.

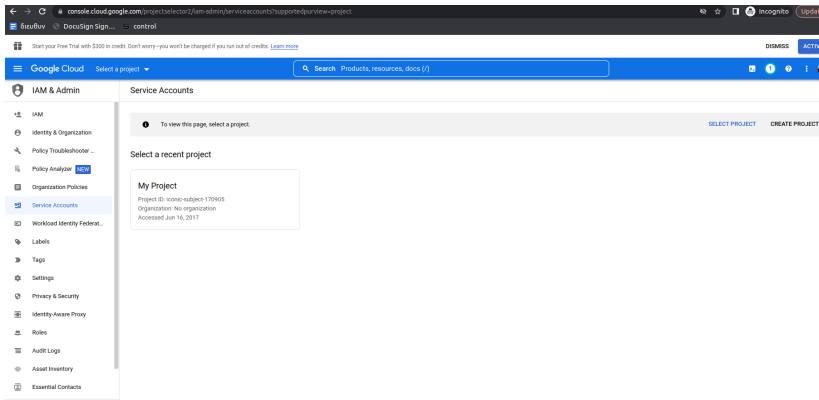
For developers What's new

DEVELOPER CENTER
Explore technical blogs, how-tos, and resources

QUICKSTART
Create a Linux VM with Compute Engine

DEVELOPER ADVOCATES
Connect with Google engineers to support your cloud journey

Δημιουργούμε έναν client



Δημιουργούμε έναν client

← → ↻ console.cloud.google.com/projectcreate?previousPage=%2Fprojectselector%2Fiam-admin%2Fserviceaccounts%3Fsupportedpurview%3Dproject&supportedpurview=project

δελφινι DocuSign Sign... control

Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#) DISMISS

Google Cloud Search Products, resources, docs (/)

New Project

⚠ You have 11 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name *
automationcourse ⓘ

Project ID: automationcourse-368707. It cannot be changed later. [EDIT](#)

Location *
No organization [BROWSE](#)

Parent organization or folder

CREATE CANCEL

Δημιουργούμε έναν client

The screenshot shows the Google Cloud IAM & Admin console. The left sidebar lists navigation options: IAM, Identity & Organization, Policy Troubleshooter, Policy Analyzer (NEW), Organization Policies, Service Accounts (selected), Workload Identity Federation, Labels, Tags, Settings, and Privacy & Security. The main content area is titled 'Service accounts' and includes buttons for '+ CREATE SERVICE ACCOUNT', 'DELETE', 'MANAGE ACCESS', 'REFRESH', and 'HELP ASSISTANT'. Below this, there is a section for 'Service accounts for project "automationcourse"' with a description and links to learn more. A table with columns 'Email', 'Status', 'Name', 'Description', 'Key ID', 'Key creation date', 'OAuth 2 Client ID', and 'Actions' is shown, but it contains no rows. On the right, a 'Recommended for' section lists related topics like 'Service accounts', 'Creating and managing accounts', 'Manage access to projects and organizations', 'Create and manage service keys', and 'Managing service account impersonation'.

Δημιουργούμε έναν client

The screenshot shows the Google Cloud IAM & Admin console in a web browser. The URL is `console.cloud.google.com/iam-admin/serviceaccounts/create?project=automationcourse-368707&supportedpurview=project`. The page title is "Create service account".

Service account details

- Service account name:** automation
- Display name for this service account:** (empty)
- Service account ID:** automation
- Email address:** automation@automationcourse-368707.iam.gserviceaccount.com
- Description:** (empty)

Grant this service account access to project (optional)

The "Description" field has a "Copy to clipboard" button next to it. The "CREATE AND CONTINUE" button is visible at the bottom of the form.

Δημιουργούμε έναν client

The screenshot shows the Google Cloud IAM & Admin console in a web browser. The address bar shows the URL: `console.cloud.google.com/iam-admin/serviceaccounts/create?project=automationcourse-368707&supportedpurview=project`. The page title is "Create service account".

The left sidebar shows the "IAM & Admin" menu with the following items: IAM, Identity & Organization, Policy Troubleshooter, Policy Analyzer (marked as NEW), Organization Policies, Service Accounts (highlighted), Workload Identity Federat..., Labels, Tags, Settings, Privacy & Security, and Identity-Aware Proxy.

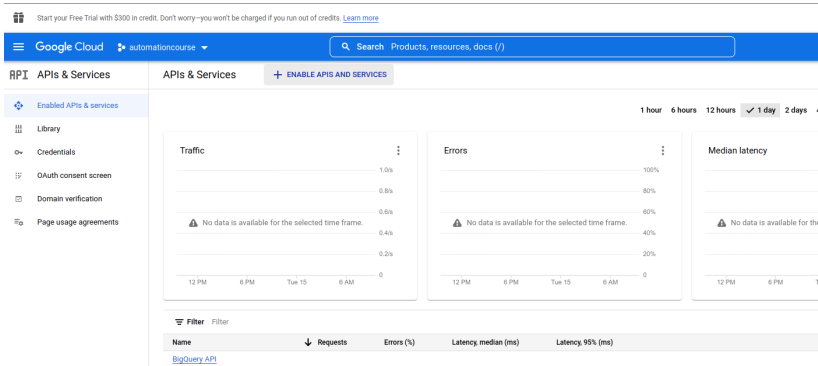
The main content area shows the "Create service account" wizard with the following steps:

- Service account details** (checked)
- Grant this service account access to project (optional)**
 - Text: "Grant this service account access to automationcourse so that it has permission to complete specific actions on the resources in your project. [Learn more](#)"
 - Form: "Select a role" dropdown menu.
 - Form: "IAM condition (optional)" with a link to "ADD IAM CONDITION".
 - Text: "+ ADD ANOTHER ROLE"
 - Button: "CONTINUE"
- Grant users access to this service account (optional)**
 - Buttons: "DONE" and "CANCEL"

Δημιουργούμε έναν client

The screenshot shows the Google Cloud IAM & Admin console. The left sidebar lists navigation options: IAM, Identity & Organization, Policy Troubleshooter, Policy Analyzer, Organization Policies, Service Accounts (selected), Workload Identity Federation, Labels, Tags, Settings, Privacy & Security, and Identity-Aware Proxy. The main content area is titled 'automation' and has tabs for DETAILS, PERMISSIONS, KEYS (active), METRICS, and LOGS. A warning message states: 'Service account keys could pose a security risk if compromised. We recommend you avoid downloading service account keys and instead use the Workload Identity Federation. You can learn more about the authenticate service accounts on Google Cloud here.' Below this, instructions are provided to 'Add a new key pair or upload a public key certificate from an existing key pair.' A table for key creation is shown with columns for 'Key creation date' and 'Key expiration date'. A dropdown menu 'ADD KEY' is open, showing options: 'Create new key' and 'Upload existing key'.

Δημιουργούμε έναν client



Ανοίγουμε το Gmail API



Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

Google Cloud automationcourse ▾

API API Library

gmail

API Library > "gmail"

Filter Type to filter

2 results

Visibility



Public

(2)

Category



Google Workspace

(2)

Other

(1)



Gmail API

Google

With the Gmail API, you can view and manage Gmail mailbox data like threads, messages, and labels.



Gmail Postmaster Tools API

Google

The Gmail Postmaster API is a RESTful API that provides programmatic access to email traffic metrics (like spam reports, delivery errors etc) otherwise available through the Gmail Postmaster Tools UI currently.

Ανοίγουμε το Gmail API



Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)



Google Cloud

automationcourse ▾



Gmail API

[Google](#)

View and manage Gmail mailbox data.

ENABLE

TRY THIS API [↗](#)

OVERVIEW

DOCUMENTATION

SUPPORT

Overview

With the Gmail API, you can view and manage Gmail mailbox data like threads, messages, and labels.

[Learn more](#) [↗](#)

Additional details

Type: [SaaS & APIs](#)

Last updated: 11/7/22

Category: [Cloud Workspace](#)

Ανοίγουμε το Gmail API

 Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of credits. [Learn more](#)

Google Cloud automationcourse

Search Products, resources, docs (/)

APIs & Services

← API/Service Details ■ DISABLE API

Enabled APIs & services

Library

Credentials

OAuth consent screen

Domain verification

Page usage agreements



Gmail API

The Gmail API lets you view and manage Gmail mailbox data like threads, messages, and labels.

By Google

Service name	Type	Status
gmail.googleapis.com	Public API	Enabled

[OVERVIEW](#) [QUICKSTARTS](#) [API REFERENCE](#)

METRICS

QUOTAS

CREDENTIALS

Select Graphs
4 Graphs

1 hour 6 hours 12 hours 1 day 2 days 4 days

Filters:

Credentials
11711546516500729551...

Methods
GetDiscovery and GetDisc...

Traffic by response code



Θωμάς Καραλάκης
Αυτοματοποιημένη Διαχείριση Συστημάτων

Χαροκόπειο Πανεπιστήμιο Αθηνών

Εγκαθιστούμε τις απαραίτητες βιβλιοθήκες

```
root@debian:~# apt-get install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
 binutils binutils-common binutils-x86-64-linux-gnu build-essential cpp
 cpp-10 dpkg-dev fakeroot fontconfig-config fonts-dejavu-core g++ g++-10 gcc
 gcc-10 javascript-common libalgorithm-diff-perl libalgorithm-diff-xs-perl
 libalgorithm-merge-perl libasan6 libatomic1 libbinutils libc-dev-bin
 libc-devtools libc6-dev libcc1-0 libcrypt-dev libctf-nobfd0 libctf0
 libdeflate0 libdpkg-perl libexpat1-dev libfakeroot libfile-fcntllock-perl
 libfontconfig1 libgcc-10-dev libgd3 libgomp1 libisl23 libitm1 libjbig0
 libjpeg62-turbo libjs-jquery libjs-sphinxdoc libjs-underscore liblsan0
 libmpc3 libmpfr6 libnsl-dev libpython3-dev libpython3.9 libpython3.9-dev
```

Εγκαθιστούμε τις απαραίτητες βιβλιοθήκες

```
Don't periodically check PyPI to determine
whether a new version of pip is available for
download. Implied with --no-index.

--no-color                Suppress colored output.
--no-python-version-warning
                           Silence deprecation warnings for upcoming
                           unsupported Pythons.
--use-feature <feature>   Enable new functionality, that may be backward
                           incompatible.
--use-deprecated <feature> Enable deprecated functionality, that will be
                           removed in the future.
root@debian:~# pip install --upgrade google-api-python-client google-auth-httpcli
b2 google-auth-oauthlib
```

Gmail Quickstart

```
2 import os.path
3 from google.auth.transport.requests import Request
4 from google.oauth2.credentials import Credentials
5 from google_auth_oauthlib.flow import InstalledAppFlow
6 from googleapiclient.discovery import build
7 from googleapiclient.errors import HttpError
8
9 # API scope
10 SCOPES = ['https://mail.google.com/']
11 creds = None
```

Listing: testgmail.py

Gmail Quickstart

```
13 # Do we already have a token?
14 if os.path.exists('token.json'):
15     # If so get credentials from token file
16     creds = Credentials.from_authorized_user_file('token.json', SCOPES)
17 # Now check whether we need to refresh the token or obtain a new one
18 if not creds or not creds.valid:
19
20     # If the token has expired, refresh it
21     if creds and creds.expired and creds.refresh_token:
22         creds.refresh(Request())
23
24     # otherwise we must login
25     else:
26         flow = InstalledAppFlow.from_client_secrets_file(
27             'credentials.json', SCOPES)
28         creds = flow.run_local_server(port=0)
29
30     # Save the token for the next run
31     with open('token.json', 'w') as token:
32         token.write(creds.to_json())
```

Listing: testgmail.py

Gmail Quickstart

```
34 try:
35     # Call the Gmail API
36     service = build('gmail', 'v1', credentials=creds)
37     results = service.users().labels().list(userId='me').execute()
38     labels = results.get('labels', [])
39
40     if not labels:
41         print('No labels found.')
42     else:
43         print('Labels:')
44         for label in labels:
45             print(label['name'])
46
47 except HttpError as error:
48     print(f'An error occurred: {error}')
```

Listing: testgmail.py

Gmail Quickstart

```
34 try:
35     # Call the Gmail API
36     service = build('gmail', 'v1', credentials=creds)
37     results = service.users().labels().list(userId='me').execute()
38     labels = results.get('labels', [])
39
40     if not labels:
41         print('No labels found.')
42     else:
43         print('Labels:')
44         for label in labels:
45             print(label['name'])
46
47 except HttpError as error:
48     print(f'An error occurred: {error}')
```

Listing: testgmail.py

Βασικές Λειτουργίες - libgmail.py

```
2 import mimetypes
3 import os.path
4 from google.auth.transport.requests import Request
5 from google.oauth2.credentials import Credentials
6 from google_auth_oauthlib.flow import InstalledAppFlow
7 from googleapiclient.discovery import build
8 from googleapiclient.errors import HttpError
9 import base64
10 from email.message import EmailMessage
11
12 SCOPES = ['https://mail.google.com/']
```

Listing: testgmail.py

Βασικές Λειτουργίες - libgmail.py

```
14 class gmailapi:
15
16     def __init__(self, token_file = 'token.json', credentials_file = 'credentials.json', from_ad = '
17         thomaskamalakis@gmail.com'):
18         self.from_ad = from_ad
19         try:
20             creds = None
21             if os.path.exists(token_file):
22                 creds = Credentials.from_authorized_user_file(token_file, SCOPES)
23
24             if not creds or not creds.valid:
25                 if creds and creds.expired and creds.refresh_token:
26                     creds.refresh(Request())
27                 else:
28                     flow = InstalledAppFlow.from_client_secrets_file(
29                         credentials_file, SCOPES)
30                     creds = flow.run_local_server(port = 0)
31                 with open(token_file, 'w') as token:
32                     token.write(creds.to_json())
33
34             self.service = build('gmail', 'v1', credentials = creds)
35
36 except HttpError as error:
37     print(f'An error occurred: {error}')
```

Listing: testgmail.py

Βασικές Λειτουργίες - libgmail.py

```
38 def build_message(self, to, subject, body):  
39     message = EmailMessage()  
40  
41     message['To'] = to  
42     message['From'] = self.from_ad  
43     message['Subject'] = subject  
44     message.set_content( body )  
45     return message
```

Listing: libmail.py

Βασικές Λειτουργίες - libgmail.py

```
47 def build_draft(self, message):
48     encoded_message = base64.urlsafe_b64encode(message.as_bytes()).decode()
49     create_message = {
50         'message': {
51             'raw': encoded_message
52         }
53     }
54     draft = self.service.users().drafts().create(userId = "me",
55                                                    body = create_message).execute()
56     return draft
```

Listing: libmail.py

Βασικές Λειτουργίες - libgmail.py

```
58 def create_draft(self, to, subject, body):
59     message = self.build_message(to, subject, body)
60     return self.build_draft(message)
61
62 def create_draft_with_attachments(self, to, subject, body, attachments = []):
63     message = self.build_message(to, subject, body)
64
65     for attachment in attachments:
66         type_subtype, _ = mimetypes.guess_type(attachment)
67         maintype, subtype = type_subtype.split('/')
68
69         with open(attachment, 'rb') as f:
70             data = f.read()
71             filename = os.path.basename( attachment )
72             message.add_attachment(data, maintype, subtype, filename = filename)
73     return self.build_draft( message )
```

Listing: libmail.py

Δημιουργία προσωρινού e-mail

```
75     def send(self, to, subject, body, attachments = []):
76         draft_dict = self.create_draft_with_attachments(to, subject, body, attachments = attachments)
77         id = draft_dict['id']
78         draft = self.service.users().drafts().send(body = {'id': id}, userId = 'me').execute()
```

Listing: libmail.py

```
1 from libgmail import gmailapi
2
3 subject = 'Test e-mail'
4 to = 'thkam@hua.gr'
5 body = \
6     """
7     This is a test email
8     using the gmail API
9     """
10
11 g = gmailapi()
12 q.create_draft(to, subject, body)
```

Listing: testdraft.py

Αποστολή e-mail

```
1 from libgmail import gmailapi
2
3 subject = 'Test e-mail'
4 to = 'thkam@hua.gr'
5 body = \
6     """
7     This is a test email
8     using the gmail API
9     """
10
11 attachment = 'pic.png'
12 g = gmailapi()
13 g.send(to, subject, body, attachments = [attachment])
```

Listing: testsend.py

Disclaimer



Calendar Quickstart

```
2 import datetime
3 import os.path
4
5 from google.auth.transport.requests import Request
6 from google.oauth2.credentials import Credentials
7 from google_auth_oauthlib.flow import InstalledAppFlow
8 from googleapiclient.discovery import build
9 from googleapiclient.errors import HttpError
10
11 SCOPES = ['https://www.googleapis.com/auth/calendar']
```

Listing: testcalendar.py

Calendar Quickstart

```

26 try:
27     service = build('calendar', 'v3', credentials=creds)
28
29     # Call the Calendar API
30     now = datetime.datetime.utcnow().isoformat() + 'Z' # 'Z' indicates UTC time
31     print('Getting the upcoming 10 events')
32     events_result = service.events().list(calendarId='primary', timeMin=now,
33                                         maxResults=10, singleEvents=True,
34                                         orderBy='startTime').execute()
35     events = events_result.get('items', [])

```

Listing: testcalendar.py

Calendar Quickstart

```

37     if not events:
38         print('No upcoming events found.')
39     else:
40
41         # Prints the start and name of the next 10 events
42         for event in events:
43             start = event['start'].get('dateTime', event['start'].get('date'))
44             print(start, event['summary'])
45
46 except HttpError as error:
47     print('An error occurred: %s' % error)

```

Listing: testcalendar.py