

Αυτοματοποιημένη Διαχείριση Συστημάτων
Διάλεξη 8η

Θωμάς Καμαλάκης

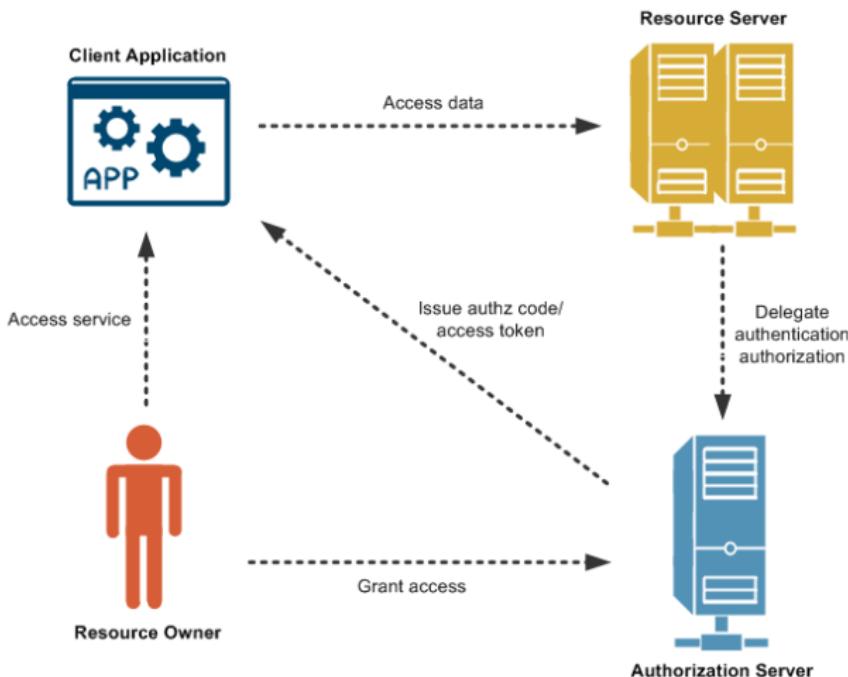
Χαροκόπειο Πανεπιστήμιο Αθηνών

Οκτώβριος 2022

OAuth2.0

- “Open Authorization”
 - Επιτρέπει την πρόσβαση σε πόρους από υπηρεσίες εκ μέρους ενός χρήστη από μία εφαρμογή.
 - Ο χρήστης θα πρέπει να δώσει την συγκατάθεση του.
 - Τότε η εφαρμογή μπορεί να έχει οριοθετημένη πρόσβαση στους πόρους χωρίς να γνωρίζει τους κωδικούς πρόσβασης του χρήστη.

OAuth2.0



OAuth2.0

- Resource server: εκεί που υπάρχουν οι πόροι τους οποίους θέλουμε να χρησιμοποιήσουμε.
 - Resource owner: αυτός που έχει την εξουσιοδότηση να δώσει πρόσβαση στους πόρους.
 - Client application: η εφαρμογή που θέλει να έχει πρόσβαση στους πόρους
 - Authorization server: αυτός που εκδίδει κουπόνια που επιτρέπουν την πρόσβαση αφού δώσει την συγκατάθεση του ο owner.
 - Scope: καθορίζει ακριβώς τι μπορεί να κάνει η εφαρμογή πάνω στους πόρους που διαχειρίζεται.

OAuth2.0: τα υπέρ

- Χρησιμοποιεί SSL για την κωδικοποίηση της επικοινωνίας
 - Δίνει περιορισμένη πρόσβαση (Scope)
 - Συνδυάζεται με λύσεις single-sign-on
 - Υλοποιείται με https οπότε μπορεί να εφαρμοστεί σε mobile, desktop, web εφαρμογές.
 - Αρκετό documentation
 - Αρκετές έτοιμες βιβλιοθήκες
 - Ήδη χρησιμοποιείται από τους “γίγαντες” (Google, κτλ)
 - Δεν εκθέτει τους κωδικούς του χρήστη

Eίδη Google API



Tι 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 smtplib.
 - 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)

Gmail Is Disabling Less Secure Apps: What To Do Next

Last updated on Jun 16, 2022 by Claire Broadley

Google has announced that it's disabling the Less Secure Apps feature on some Google accounts from May 30th, 2022.

If you're using Gmail SMTP details with our Other SMTP mailer, you may have difficulty sending emails when this feature is disabled.

We know that many of our customers rely on Gmail to send emails from

Πως στέλνουμε e-mail με το Google API

- Θα πρέπει να ξεκινήσουμε ένα νέο project στο Google Cloud Platform (GCP)
 - Ανοίγουμε εκεί το Gmail API
 - Φτιάχνουμε τα διαπιστευτήρια για την εφαρμογή μας (το script μας δηλαδή)
 - Ενσωματώνουμε στο script μας τη ροή του Oauth2.0
 - Την πρώτη φορά που τρέχει το script μας ζητάει την συγκατάθεση μας να χρησιμοποιήσει το email μας.
 - Η εφαρμογή μας λαμβάνει ένα token το οποίο μπορεί να ανανεώνει και πλέον μπορεί να στένλει e-mail εκ μέρους μας.

Disclaimer

- Προσοχή!
 - Με τα παρακάτω παραδείγματα θα δώσουμε στον κώδικα μας πλήρη έλεγχο πάνω στο @gmail.com λογαριασμό μας.
 - Όπως είμαστε προσεκτικοί με τα κλειδιά μας έτσι είμαστε και με τα κουπόνια μας.
 - Π.χ. δεν τα ανεβάζουμε σε repos, δεν τα αφήνουμε σε server που δεν ελέγχουμε πλήρως κτλ.
 - Καλύτερα να χρησιμοποιήσετε ένα δοκιμαστικό λογαριασμό αν το περιεχόμενο του γραμματοκιβωτίου είναι σημαντικό για εσάς.

Disclaimer



Disclaimer



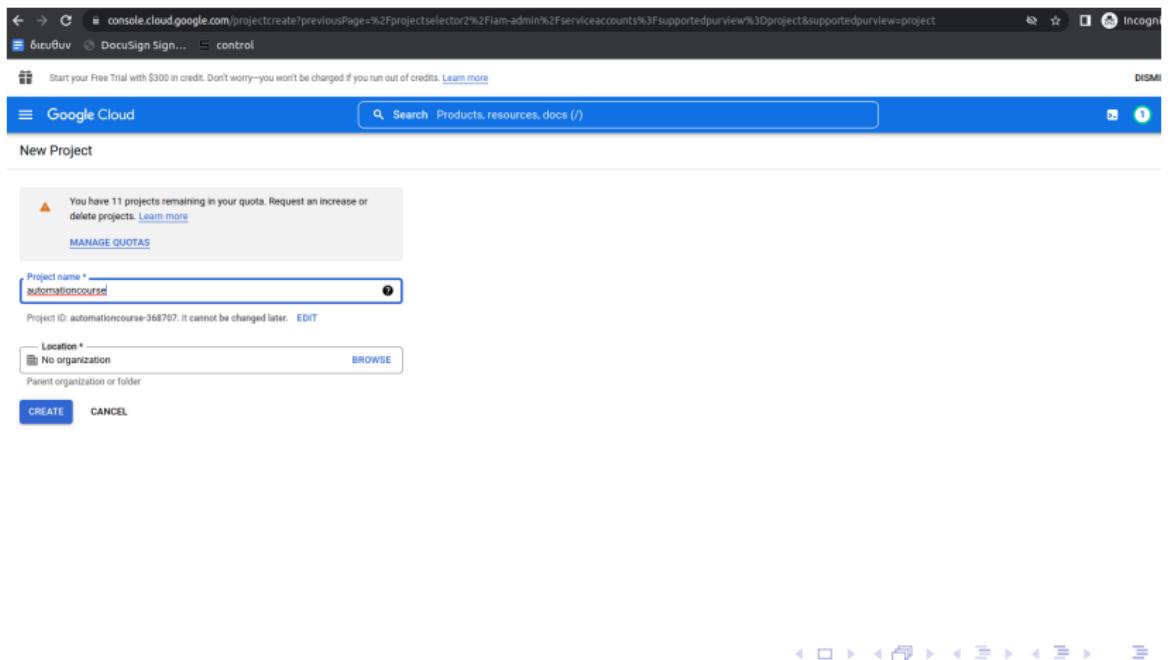
Ξεκινάμε από το GCP

The screenshot shows the Google Cloud homepage. At the top, there's a navigation bar with links for Overview, Solutions, Products, Pricing, Resources, Contact Us, Docs, Support, Language (set to English), and Console. A search bar is also present. Below the navigation, a blue banner invites users to "Watch the Google Cloud Next '22 livestream now". The main headline reads "Built by developers, for developers". Below it, a sub-headline says "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." There are two buttons: "Get started for free" and "Join Innovators Program". On the right side, there's a large image of a developer's monitor displaying a code editor with Java code and a video player showing a developer at a keyboard. Below the monitor, text says "Deploy with 80% fewer lines of code in the Google Cloud console." At the bottom, there are three sections: "For developers" (with a link to the Developer Center), "What's new", and "DEVELOPER CENTER" which links to technical blogs and how-tos. Other sections include "QUICKSTART" (Create a Linux VM with Compute Engine), "DEVELOPER ADVOCATES" (Connect with Google engineers), and a portrait of a woman.

Δημιουργούμε έναν client

The screenshot shows the Google Cloud IAM & Admin Service Accounts interface. The left sidebar lists various IAM components: IAM, Identity & Organization, Policy Troubleshooter, Policy Analyzer (NEW), Organization Policies, Service Accounts (selected), Workload Identity Federation, Labels, Tags, Settings, Privacy & Security, Identity-Aware Proxy, Roles, Audit Logs, Asset Inventory, and Essential Contacts. The main content area is titled "Service Accounts" and displays a message: "To view this page, select a project." Below this, there's a section titled "Select a recent project" with a box labeled "My Project" containing the text: "Project ID: icono-subject-170905", "Organization: No organization", and "Accessed Jun 16, 2017". At the top right, there are "SELECT PROJECT" and "CREATE PROJECT" buttons. The top navigation bar includes links for console.cloud.google.com, browser history, DocuSign Sign..., control, and a free trial offer.

Δημιουργούμε έναν client



Δημιουργούμε έναν client

The screenshot shows the Google Cloud IAM & Admin Service Accounts page. The left sidebar lists various IAM components: IAM, Identity & Organization, Policy Troubleshooter, Policy Analyzer (NEW), Organization Policies, Service Accounts (selected), Workload Identity Federat..., Labels, Tags, Settings, and Privacy & Security. The main content area is titled "Service accounts for project 'automationcourse'". It contains a brief description of what service accounts are and how they can be used to secure them. A table is present with columns: Filter, Email, Status, Name (sorted), Description, Key ID, Key creation date, OAuth 2 Client ID, and Actions. The message "No rows to display" is shown. At the top right, there are "DISMISS" and "ACTIVATE" buttons, along with a "HELP ASSISTANT" icon. A "RECOMMENDED FOR" sidebar on the right lists links to "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 Service Accounts page. The left sidebar lists various IAM components: IAM, Identity & Organization, Policy Troubleshooter, Policy Analyzer (NEW), Organization Policies, Service Accounts (selected), Workload Identity Federat..., Labels, Tags, Settings, and Privacy & Security. The main content area is titled "Create service account". It has two sections: "Service account details" and "Grant this service account access to project (optional)". In the "Service account details" section, the "Service account name" is set to "automation", and the "Display name for this service account" is also "automation". The "Service account ID" is "automation", and the "Email address" is "automation@automationcourse-368707.iam.gserviceaccount.com". A "Copy to clipboard" button is available for the account ID. Below these fields is a "Count description" input field with placeholder text "Describe what this service account will do". At the bottom of this section is a "CREATE AND CONTINUE" button. The "Grant this service account access to project (optional)" section is partially visible at the bottom.

Δημιουργούμε έναν client

The screenshot shows the Google Cloud IAM & Admin interface for creating a new service account. The left sidebar lists various IAM-related options like IAM, Identity & Organization, Policy Troubleshooter, Policy Analyzer, Organization Policies, Service Accounts (which is selected), Workload Identity Federation, Labels, Tags, Settings, Privacy & Security, and Identity-Aware Proxy. The main content area has a title 'Create service account' and a sub-section 'Service account details'. Step 2, 'Grant this service account access to project (optional)', is currently active. It includes a note about granting access to the 'automationcourse' project, a dropdown menu for selecting a role ('Select a role'), an 'IAM condition (optional)' section with a '+' button to add conditions, and a 'CONTINUE' button. Step 3, 'Grant users access to this service account (optional)', is shown below with 'DONE' and 'CANCEL' buttons.

Δημιουργούμε έναν client

The screenshot shows the Google Cloud IAM & Admin interface for managing service account keys. The left sidebar lists various management categories like IAM, Identity & Organization, Policy Troubleshooter, Policy Analyzer, Organization Policies, Service Accounts (which is selected), Workload Identity Federation, Labels, Tags, Settings, Privacy & Security, and Identity-Aware Proxy. The main content area has a breadcrumb navigation bar with 'automation' and a search bar. Below this, tabs for DETAILS, PERMISSIONS, KEYS (which is active and highlighted in blue), METRICS, and LOGS are present. A section titled 'Keys' contains a warning message about the security risk of downloading service account keys and recommends using Workload Identity Federation. It also provides options to 'Add a new key pair or upload a public key certificate from an existing key pair' and 'Block service account key creation using organization policies'. At the bottom, there are buttons for 'ADD KEY' (with dropdown options 'Create new key' and 'Upload existing key'), 'Key creation date', and 'Key expiration date'.

Δημιουργούμε έναν client

The screenshot shows the Google Cloud APIs & Services dashboard. On the left sidebar, there are sections for Enabled APIs & services, Library, Credentials, OAuth consent screen, Domain verification, and Page usage agreements. The main area displays two charts: 'Traffic' and 'Errors'. The 'Traffic' chart shows data for the selected time frame (1 day) with no data available. The 'Errors' chart also shows data for the same time frame with no data available. Below the charts, a table lists the BigQuery API with its details: Requests (1), Errors (%) (0), Latency, median (ms) (0), and Latency, 95% (ms) (0). A 'Filter' button is located at the bottom left of the table.

Name	Requests	Errors (%)	Latency, median (ms)	Latency, 95% (ms)
BigQuery API	1	0	0	0

Avoiding the 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

API Library > "gmail"

X

Filter Type to filter

Visibility

Public (2)

Category

Google Workspace (2)

Other (1)

2 results

 **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.

Avoiding the 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](#)

Avoiding the Gmail API

The screenshot shows the Google Cloud API Service Details page for the Gmail API. The left sidebar has 'Enabled APIs & services' selected. The main content area shows the Gmail API details, including its service name (gmail.googleapis.com), type (Public API), and status (Enabled). Below this, there are tabs for METRICS, QUOTAS, and CREDENTIALS. A dropdown menu for 'Select Graphs' shows '4 Graphs'. At the bottom, there are filters for 'Credentials' (set to 11711546516500729551...) and 'Methods' (set to GetDiscovery and GetDisc...). The top navigation bar includes a search bar and links for Products, resources, docs (/).

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
  libmnc3 libmngfr6 libns1-dev libnvthon3-dev libnvthon3.9 libnvthon3.9-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.
Suppress colored output.

--no-color                                     Silence deprecation warnings for upcoming
                                                unsupported Pythons.

--no-python-version-warning                   Enable new functionality, that may be backward
                                                incompatible.

--use-feature <feature>                      Enable deprecated functionality, that will be
                                                removed in the future.

--use-deprecated <feature>                   root@debian:~# pip install --upgrade google-api-python-client google-auth-httplib2
                                                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 = 'thomaskamalakis@gmail.com'):
17         self.from_ad = from_ad
18         try:
19             creds = None
20             if os.path.exists(token_file):
21                 creds = Credentials.from_authorized_user_file(token_file, SCOPES)
22
23             if not creds or not creds.valid:
24                 if creds and creds.expired and creds.refresh_token:
25                     creds.refresh(Request())
26                 else:
27                     flow = InstalledAppFlow.from_client_secrets_file(
28                         credentials_file, SCOPES)
29                     creds = flow.run_local_server(port = 0)
30                     with open(token_file, 'w') as token:
31                         token.write(creds.to_json())
32
33             self.service = build('gmail', 'v1', credentials = creds)
34
35         except HttpError as error:
36             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
74         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 g.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

Calendar API

- ① Με τον ίδιο τρόπο ενεργοποιούμε το Calendar API στο GCP
 - ② Φτιάνουμε ένα project
 - ③ Φτιάχνουμε τα διαπιστευτήρια
 - ④ Ενσωματώνουμε στη ροή του Oauth2.0
 - ⑤ Δίνουμε συγκατάθεση
 - ⑥ Η εφαρμογή μας λαμβάνει ένα token που μπορεί να χρησιμοποιήσει το calendar μας.

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

```
13 creds = None
14 if os.path.exists('cal_token.json'):
15     creds = Credentials.from_authorized_user_file('cal_token.json', SCOPES)
16 if not creds or not creds.valid:
17     if creds and creds.expired and creds.refresh_token:
18         creds.refresh(Request())
19     else:
20         flow = InstalledAppFlow.from_client_secrets_file(
21             'calendar_creds.json', SCOPES)
22         creds = flow.run_local_server(port=0)
23     with open('cal_token.json', 'w') as token:
24         token.write(creds.to_json())
```

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](#)