

## Code Overview

```
Projects > daily_commute_text.py > ...
1 import datetime
2 import googlemaps
3 from twilio.rest import Client as TwilioClient
```

Lines 1-3 import the modules datetime, googlemaps, and Client (as TwilioClient) from twilio.rest

If you need to install googlemaps and twilio.rest modules, run the following command in your terminal:

```
`pip install googlemaps
`pip install twilio
```

```
4
5 # A function that uses the Google Maps/Directions API to calculate the duration between start_addr and end_addr
6 def get_commute_duration():
7     google_api = "API key" # Replace with your Google Maps API key
8
9     # Initialize and creates a Google maps client object and passes the API key
10    gmaps = googlemaps.Client(key=google_api)
11
12    # Defines the starting and destination addresses
13    start_addr = "starting address"
14    end_addr = "destination address"
15
16    # runs the Google Map directions function and stores the result
17    directions_result = gmaps.directions(start_addr, end_addr, mode = "driving")
18
19    # Processes directions_results and stores the information related to trip duration
20    first_leg = directions_result[0]['legs'][0]
21    duration = first_leg['duration']['text']
22
23    return duration
```

Lines 5 - 23 defines the get\_commute\_duration() function.

- Line 7 assigns our required Google Maps API key to a variable named google\_api
- Line 10 initializes and creates a Google Maps client object named gmaps and passes our API key
- Line 13 - 14 defines our starting/destination addresses and stores them into variables start\_addr and end\_addr
- Line 17 runs the directions function and passes in start\_addr, end\_addr
  - The third argument is mode = "driving" to specify that the trip details should be calculated from a driving context
  - The information result is stored into a variable called directions\_results
- Line 20-21 processes the resulting data and identifies/stores information related to the trip duration into a variable called duration
- Line 23 makes the function return duration

```

26
27 # A function that uses the Twilio API to create a message and send to a recipient's phone number
28 def send_text_message(message):
29     # Replace with your Twilio account SID, token, and phone number (format: "1#####")
30     twilio_account_sid = "account sid"
31     twilio_account_token = "account token"
32     twilio_phone_num = "twilio phone number"
33
34     # Defines the recipient's phone number (format: "1#####")
35     recipient_phone_num = "recipient phone number"
36
37     # Intializes and creates a Twilio client object and passes Twilio credentials
38     twilio_client = TwilioClient(twilio_account_sid, twilio_account_token)
39
40     # creates a Twilio message and defines it's to, from, and body data
41     twilio_client.messages.create(
42         to = recipient_phone_num,
43         from_ = twilio_phone_num,
44         body = message
45     )
46

```

Line 28 - 45 defines the `send_text_message(message)` function

- Line 30 - 32 creates variables called `twilio_account_sid`, `twilio_account_token`, and `twilio_phone_num` to store the required Twilio API information
- Line 35 defines our recipient's phone number and stores it in a variable called `recipient_phone_num`
- Line 38 initializes and creates a Twilio client (called `twilio_client`), and passes in `twilio_account_sid` and `twilio_account_token`
- Line 41 - 45 runs the client object's `messages.create` function and passes in `recipient_phone_num`, `twilio_phone_num`, and `message`, which correspond to the `to`, `from`, and `body` arguments

```

49
50 duration = get_commute_duration() # Assigns the commute duration time between start/end addresses defined in the function
51 current_time = datetime.datetime.now() # Assigns the current time
52

```

Line 50 calls the `get_commute_duration()` function and stores the returned data into a variable called `duration`

- The returned duration format when calling `get_commute duration` is: “# hours # mins”

Line 51 calls the `datetime.datetime.now()` function, which generates the current date and time (based on your local OS) and stores into a variable called `current_time`

```

51 current_time = datetime.datetime.now() # Assigns the current time
52
53 # Splits up duration into individual indexes so that it can be processed
54 duration_parts = duration.split()
55

```

Line 54 runs the split() function on duration, which splits the string data into separate string chunks, which we can use to process and identify the literal hours/minutes information (trip duration)

```

55
56 #initialize values for hours, minutes to 0
57 hours, minutes = 0, 0
58

```

Line 57 initializes the values for variables hours, minutes to 0. These variables will eventually store the int value of the literal string data pertaining to trip duration information

```

59 # Processes duration_parts by identifying the string value of hour and min, and converting it to a int value to be stored in hours, minutes
60 for i in range(len(duration_parts)):
61     if duration_parts[i] == "hour":
62         hours = int(duration_parts[i - 1])
63     elif duration_parts[i] == "mins":
64         minutes = int(duration_parts[i - 1])
65

```

Line 60 - 64 process duration\_parts

- The for if/else statement checks each index to see if matches “hours” or “mins”. If a match is found, the prior index should contain a number string representing hours or minutes of the trip duration
- A int function is then called at the matching index position - 1, which turns the number literal string into an int

```

65
66 # Creates a timedelta object representing the duration of commute time.
67 # commute_duration differs from duration because timedelta objects can be used to perform math operations with datetime objects
68 # whereas duration is the duration represented as a string
69 commute_duration = datetime.timedelta(hours=hours, minutes=minutes)
70

```

Line 69 uses the datetime.timedelta method and passes in the int values of hours and minutes to create a timedelta object. This timedelta object is stored into variable called commute\_duration

- Timedelta objects contain time information (hours, minutes, etc)
- Mathematical operations can be performed between timedelta objects

```

70
71 # Calculates the arrival time by adding current_time (datetime object) with commute_duration (timedelta object)
72 # Formats in HH:MM AM/PM format
73 arrival_time = (current_time + commute_duration).strftime('%I:%M %p')
74

```

Line 73 adds the time information from `current_time` and `commute_duration` together, which effectively is the arrival time.

- The resulting information is formatted to HH:MM AM/PM format by calling the `.strftime()` method and passing in `'%I:%M %p'`

```

74
75 # Creates the text body message to be stored in variable message, which will be used as an argument in send_text_message function
76 message = (
77     f"\n\nGood morning!\n\n"
78     f"This is your daily morning commute forecast. \n\n"
79     f"The estimated commute time is: {duration}.\n"
80     f"If you leave now, your estimated arrival time is: {arrival_time}."
81 )

```

Line 76 creates a string message and stores it in variable named `message`

- The string contains data from variables `duration` and `arrival_time`

```

83
84 # Calls the send_text_message function
85 send_text_message(message)
86

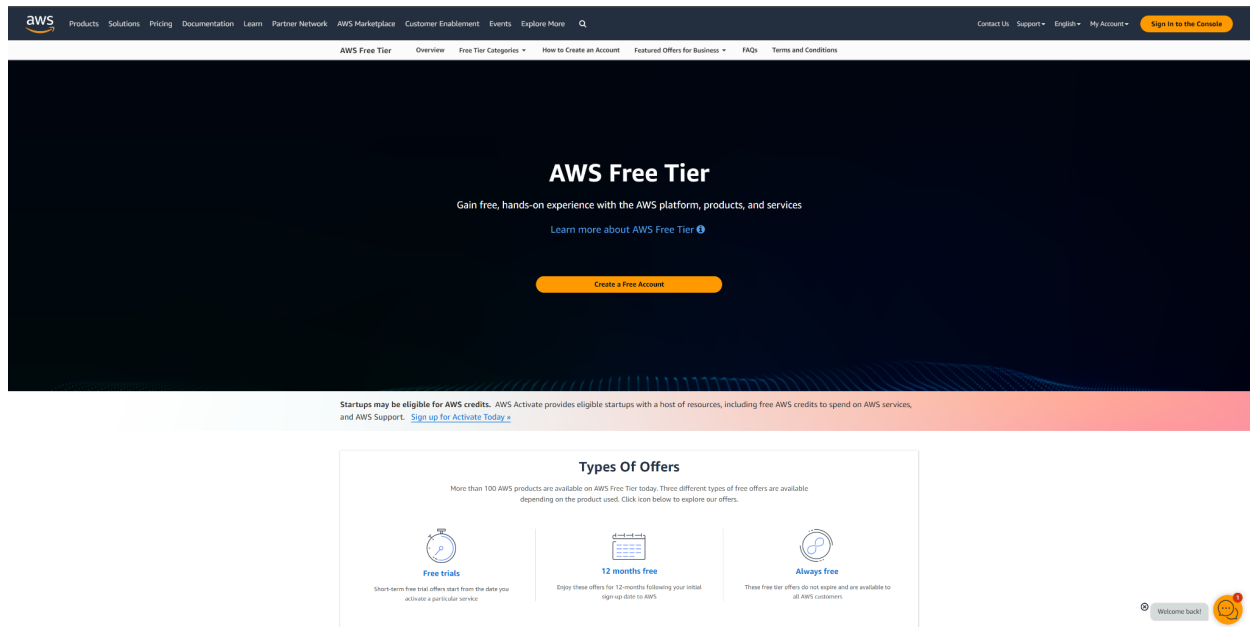
```

Line 85 calls the `send_text_message` and passes in the variable `message` as an argument

# AWS Deployment/Cron Job Automation

## Setting Up and Launching EC2 Instance

Sign into AWS. If needed, register for a free tier account.



The screenshot shows the AWS Free Tier landing page. At the top is the AWS navigation bar with links for Products, Solutions, Pricing, Documentation, Learn, Partner Network, AWS Marketplace, Customer Enablement, Events, and Explore More. Below this is a secondary navigation bar for the Free Tier, including Overview, Free Tier Categories, How to Create an Account, Featured Offers for Business, FAQs, and Terms and Conditions. The main content area has a dark blue background with the text "AWS Free Tier" and "Gain free, hands-on experience with the AWS platform, products, and services". A prominent orange button labeled "Create a Free Account" is centered. Below this, a small text block mentions startup eligibility for AWS credits. The bottom section, titled "Types Of Offers", explains that more than 100 AWS products are available for free and lists three offer types: "Free trials" (short-term), "12 months free" (for new sign-ups), and "Always free" (for all customers). A "Welcome back!" notification with a smiley face icon is visible in the bottom right corner.

**AWS Free Tier**

Gain free, hands-on experience with the AWS platform, products, and services

[Learn more about AWS Free Tier](#)

[Create a Free Account](#)

Startups may be eligible for AWS credits. AWS Activate provides eligible startups with a host of resources, including free AWS credits to spend on AWS services, and AWS Support. [Sign up for Activate Today](#)

### Types Of Offers

More than 100 AWS products are available on AWS Free Tier today. Three different types of free offers are available depending on the product used. Click icon below to explore our offers.

- Free trials**  
Short-term free trial offers start from the date you activate a particular service.
- 12 months free**  
Enjoy these offers for 12-months following your initial sign-up date to AWS.
- Always free**  
These free tier offers do not expire and are available to all AWS customers.

Welcome back!

From your main dashboard, go to Services > Compute > EC2

The screenshot displays the AWS Management Console interface. At the top, the navigation bar includes the AWS logo, a 'Services' dropdown menu, a search bar with the placeholder text 'Search', and a keyboard shortcut '[Alt+S]'. The left-hand navigation pane is titled 'All services' and lists various AWS categories with icons. The 'Compute' category is highlighted in orange. The main content area is titled 'Compute' with a close button (X) in the top right corner. It lists several compute services with their respective descriptions:

- AWS App Runner**: Build and run production web applications at scale
- Batch**: Fully managed batch processing at any scale
- EC2**: Virtual Servers in the Cloud
- EC2 Image Builder**: A managed service to automate build, customize and deploy OS images
- Elastic Beanstalk**: Run and Manage Web Apps
- Lambda**: Run code without thinking about servers
- Lightsail** (with an external link icon): Launch and Manage Virtual Private Servers
- AWS Outposts**: Run AWS Services On Premises
- Serverless Application Repository**: Assemble, deploy, and share serverless applications within teams or publicly
- AWS SimSpace Weaver**: Build and run large-scale spatial simulations

The left navigation pane lists the following services under 'All services': Analytics, Application Integration, AWS Cost Management, Blockchain, Business Applications, **Compute**, Containers, Customer Enablement, Database, Developer Tools, End User Computing, Front-end Web & Mobile, Game Development, Internet of Things, Machine Learning, Management & Governance, Media Services, Migration & Transfer, Networking & Content Delivery, Quantum Technologies, Robotics, Satellite, Security, Identity, & Compliance, and Storage.

Click on Launch Instance

Services

Search

Alt+S

New EC2 Experience

Tell us what you think

EC2 Dashboard

EC2 Global View

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

Resources

Info

EC2 Global view

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Filter resources by tag(s)

Instances (running)	1	Auto Scaling Groups	0	Dedicated Hosts	0	Elastic IPs	0
Instances	1	Key pairs	1	Load balancers	0	Placement groups	0
Security groups	2	Snapshots	0	Volumes	1		

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Launch Instance

Migrate a server

Note: Your instances will launch in the US East (N. Virginia) Region

Scheduled events

US East (N. Virginia)

No scheduled events

Migrate a server

Use AWS Application Migration Service to simplify and expedite migration from physical, virtual, and cloud infrastructure to AWS.

Get started with AWS Application Migration Service

Service health

AWS Health Dashboard

Region

US East (N. Virginia)

Zones

Zone name	Zone ID
us-east-1a	use1-az4
us-east-1b	use1-az6
us-east-1c	use1-az1
us-east-1d	use1-az2
us-east-1e	use1-az3
us-east-1f	use1-az5

Enable additional Zones

Fill out the instance information with the image being Ubuntu. Be sure to create an SSH pair if you want to connect with a client (i.e. Putty, etc)

**Launch an instance** [info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags** [info](#)

Name  [Add additional tags](#)

**Application and OS Images (Amazon Machine Image)** [info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

**Recents** **Quick Start**

**Amazon Linux** **macOS** **Ubuntu** **Windows** **Red Hat** **SUSE Li** [Browse more AMIs](#)

**Amazon Machine Image (AMI)**

**Amazon Linux 2023 AMI** [Free tier eligible](#)

ami-03af6eaae9938c858c (64-bit (x86)) / ami-03f6c2c562b3d7f15 (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs

**Description**

Amazon Linux 2023 AMI 2023.2.20230920.1 x86\_64 HVM kernel-6.1

**Architecture** **AMI ID**

64-bit (x86) ami-03af6eaae9938c858c [Verified provider](#)

**Instance type** [info](#)

Instance type

**Summary**

Number of instances [info](#)

1

**Software Image (AMI)**

Amazon Linux 2023 AMI 2023.2.2...[read more](#)  
ami-03af6eaae9938c858c

**Virtual server type (instance type)**

t2.micro

**Firewall (security group)**

New security group

**Storage (volumes)**

1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GiB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Launch instance](#) [Review commands](#)

Once you create the instance, you will see the created instance on the Instances page. It may take a few minutes for the Instance State to change to Running

**Instances (1)** [info](#)

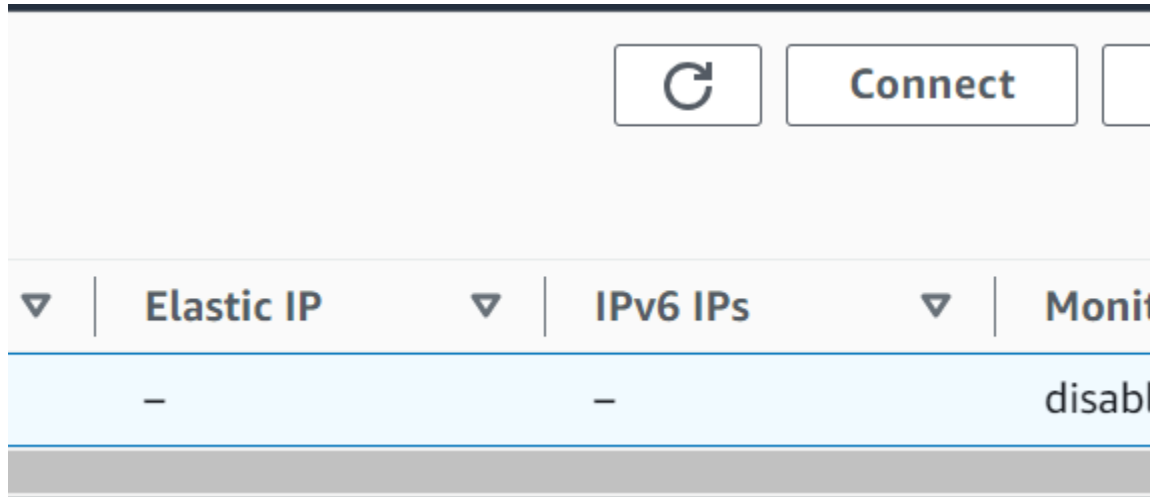
**Instance state = running** [Clear filters](#)

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
<input type="checkbox"/>	instance-01	[REDACTED]	Running	t2.micro	2/2 checks passed	No alarms	us-east-1d	[REDACTED]	[REDACTED]	-

**Select an instance**



Check your instance, and click on Connect. If you wish to connect via your own SSH client instead, copy and paste the instance's IP address and upload the SSH Key file accordingly



## Ubuntu Terminal

Once you are logged into your instance's terminal, you should complete any Ubuntu updates and install python3. Run the following commands in your terminal:

Update your package list:

```
`sudo apt update
```

Upgrade packages

```
`sudo apt upgrade
```

If not already installed, install Python3

```
`sudo apt install python3
```

If needed, reboot your instance manually or with command:

```
`sudo reboot
```

You will need to get a copy of your script file onto your local instance. You can achieve this by uploading your script file to your github repo and then running the following terminal command:

```
`wget https://github.com/<username>/<reponame>/<scriptfilename>.py
```

Ensure that you do not upload a script file with your API key information in it. You will get your API key deactivated and will need to rotate keys if you do so

Ensure that script file is executable by running this command:

```
`sudo chmod +x <scriptfilename>.py
```

You will need to add your API credentials. You can do this by opening your script file using the nano command and writing it in:

```
`nano <scriptfilename>.py
```

Additionally while in nano, add the shebang/hashbang as the first line of the script file. A shebang/hashbang is needed to specify the interpreter needed to run the script, This allows the script file to be ran without explicitly calling python3 in terminal commands

```
`#!/usr/bin/python3
```

Your script file may have Windows-style line endings (CRLF), which will not allow your script to run through the shebang/hashbang. To remediate this, convert the script file to Unix-style line endings (LF):

Install dos2unix by running terminal command:

```
`sudo apt-get install dos2unix
```

Convert the script with terminal command:

```
`dos2unix <scriptfilename>.py
```

Double-check your local instance timezone by running the terminal command:

```
`date
```

If the timezone is not correct, run this terminal command to print a list of timezones:

```
`timedatectl list-timezones
```

Then run this terminal command to make the change to the correct timezone based on prior command:

```
`sudo timedatectl set-timezone <timezone>
```

## Scheduling Cron Job

To set up a cron job, run terminal command:

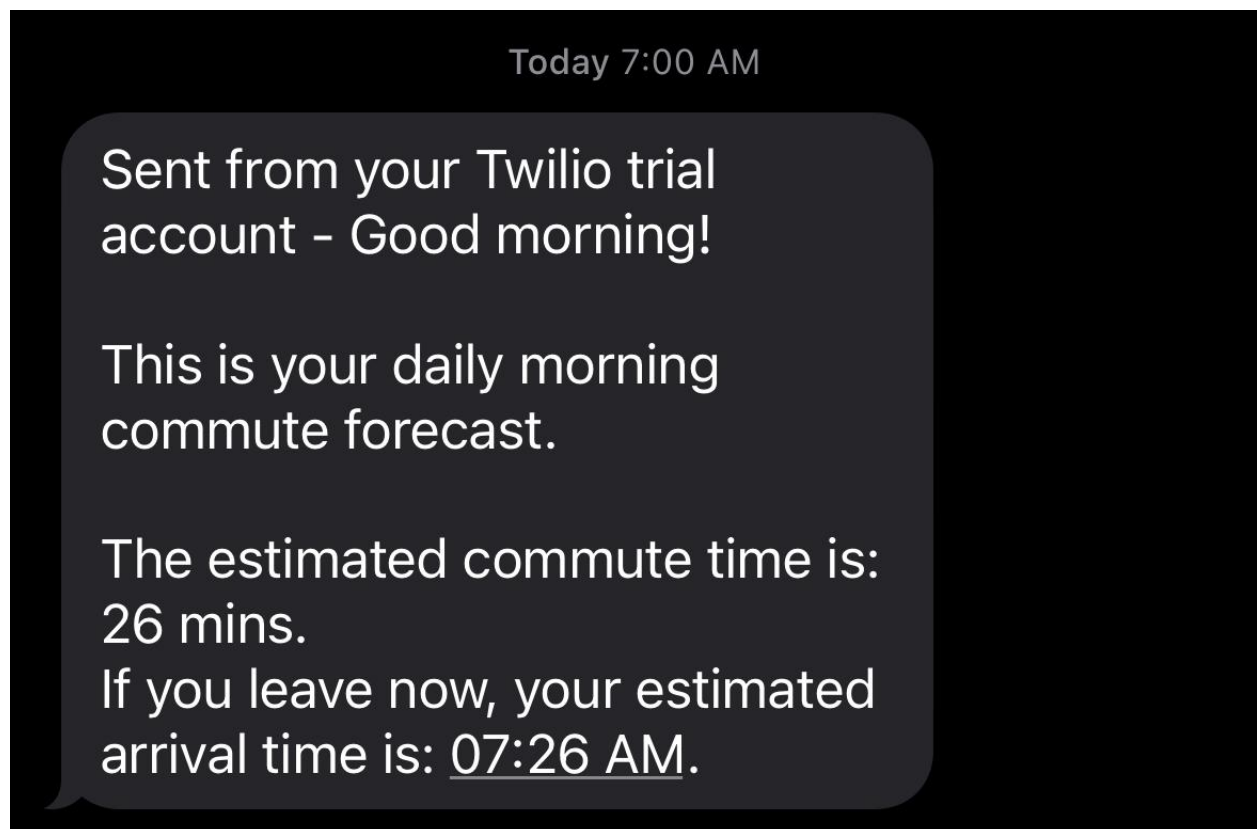
```
`crontab -e`
```

Please see reference material at <https://www.hostinger.com/tutorials/cron-job> regarding syntax and guidance

Once the cron job is scheduled, the script file will be executed at the specified schedule

## Result

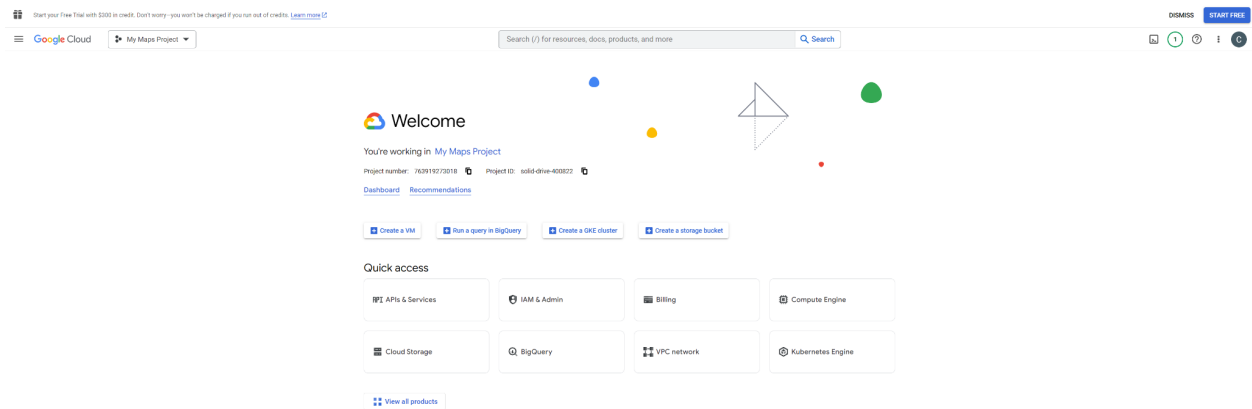
Scheduled text message:





# How to set up Google Maps API key

Go to <https://console.google.com>. You can use your gmail account or create a new one.



Search for 'Directions 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



My Maps Project

## Search

ALL

DOCUMENTATION & TUTORIALS

RESOURCES

MARKETPLACE & APIS

### Filter by

- ☐ Product or Page
- ☐ Documentation or tutorial
- ☐ Marketplace and APIs
- ☐ Organization
- ☐ Folder
- ☐ Project
- ☐ Resources

### Resource filters

Project, folder, or org  
My Maps Project

Resource type  
Any

### Search results

Showing 15 of 15 results for "directions api".



#### Directions API

Directions between multiple locations.

Type: Marketplace Product  
Producer: Google Enterprise API



#### Maps JavaScript API

Maps for your website

Type: Marketplace Product  
Producer: Google



#### MapsIndoors - A dynamic mapping platform

MapsIndoors is a Google Maps based mapping platform built for large indoor and/or private spaces

Type: Marketplace Product

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

My Maps Project

DISMISS

START FREE

Product details



#### Directions API

Google Enterprise API

Directions between multiple locations.

ENABLE

OVERVIEW

DOCUMENTATION

SUPPORT

RELATED PRODUCTS

#### Overview

Access driving, cycling, walking and public transportation routing with the Directions API using an HTTP request. Waypoints offer the ability to alter a route through a specific location. Specify origins, destinations and waypoints either as text strings (e.g. "Chicago, IL" or "Darwin, NT, Australia") or as latitude/longitude coordinates.

#### Additional details

Type: Marketplace Product

Last product update: 9/28/22

Category: Google Enterprise APIs, Maps

Service name: directions-backend.googleapis.com

#### Tutorials and documentation

[Documentation](#)

[Pricing](#)

#### Support

[Learn more](#)

#### Terms of Service

By using this product you agree to the terms and conditions of the following license: [Google Maps Platform](#).

#### Related Products

Customers who use this product also use the following products



Geolocation API

Google Enterprise API



Maps SDK for Android

Google



Places API

Google Enterprise API



Maps JavaScript API

Google



Distance Matrix API

Google Enterprise API



Geocoding API

Google Enterprise API



If you've never utilized Google Developer tools, you'll get taken to a page to confirm terms and conditions and verify billing information to activate your free trial.

The Google Maps Platform offers \$200 monthly credit at no charge for Google Maps APIs, but you should review and familiarize yourself with Google's billings and costs to avoid any unexpected charges.

You may see a charge to your credit card (for billing verification) and it should fall off your statement shortly afterwards.

Try Google Maps Platform

### Step 1 of 2 Account Information

  [SWITCH ACCOUNT](#)

Country

United States

Terms of Service

☐ I have read and agree to the [Google Cloud Platform Terms of Service](#), [Supplemental Free Trial Terms of Service](#), and the [terms of service of any applicable services and APIs](#).

Required to continue

[CONTINUE](#)

**Verify your card to get started**

Your card is used to verify you're not a robot. Don't worry, it won't be charged until you manually upgrade to a paid account.

**No charge to try Maps APIs**

Get \$200 monthly credit at no charge for Google Maps APIs. Also get an extra \$300 credit for any Cloud product for 90 days.



**Start building right away**

Launch a pre-packaged solution in minutes or create one yourself using advanced code samples and comprehensive documentation.

Try Google Maps Platform

### Step 2 of 2 Payment Information Verification


Your payment information helps us reduce fraud and abuse. If using a credit or debit card, you won't be charged until you manually activate your account.

 Account type 

Individual

Only business accounts can have multiple users. You cannot change the account type after signing up. In some countries, this selection affects your tax options. If you choose individual as your account type, you agree that use of your account is for your trade, business, craft, or profession. [Learn more](#)

Payment method

 Add credit or debit card

Card number

MM / YY CVC

#

Card number is required

Cardholder name

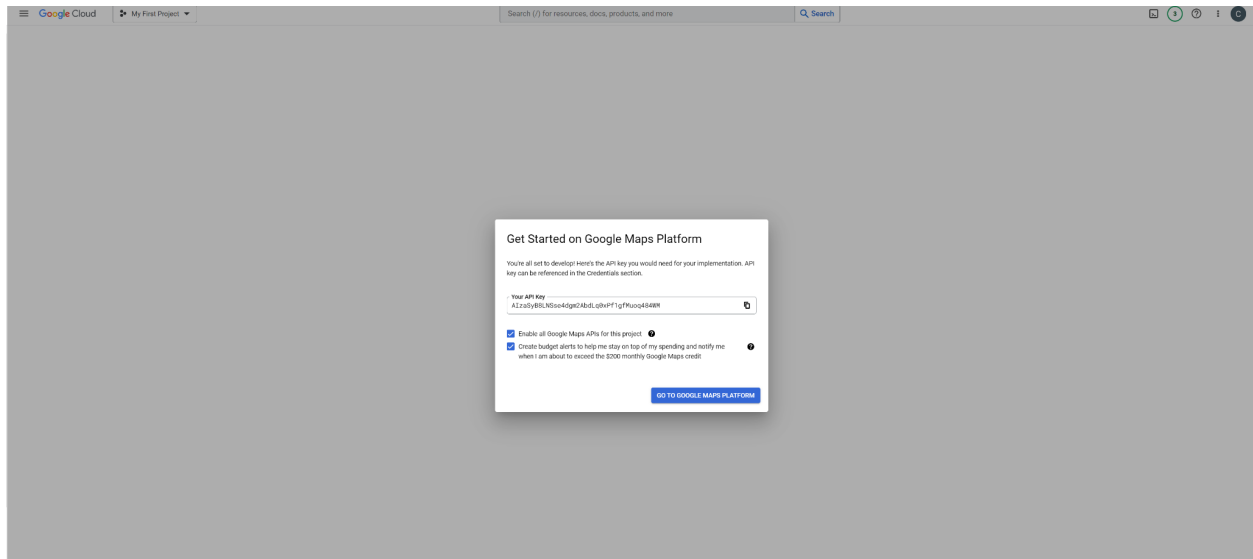
Cardholder name is required

Billing address

[START MY FREE TRIAL](#)

Privacy policy | [Terms of Service](#)

After registering and completing some survey questions, you'll get shown a screen that will have your API key information



API Credential Page:

Navigation Menu > APIs & Services > Credentials





Start your Free Trial with \$300 in credit. Don't worry—you won't be charged if you run out of



Google Cloud



My Maps Project ▼



Cloud overview



Products & solutions



#### PINNED

API

APIs & Services



Billing



IAM & Admin



Marketplace



Compute Engine



Kubernetes Engine



Cloud Storage



BigQuery



VPC network



Cloud Run



SQL



Security



Google Maps Plat...



#### RECOMMENDATIONS

Enabled APIs & services

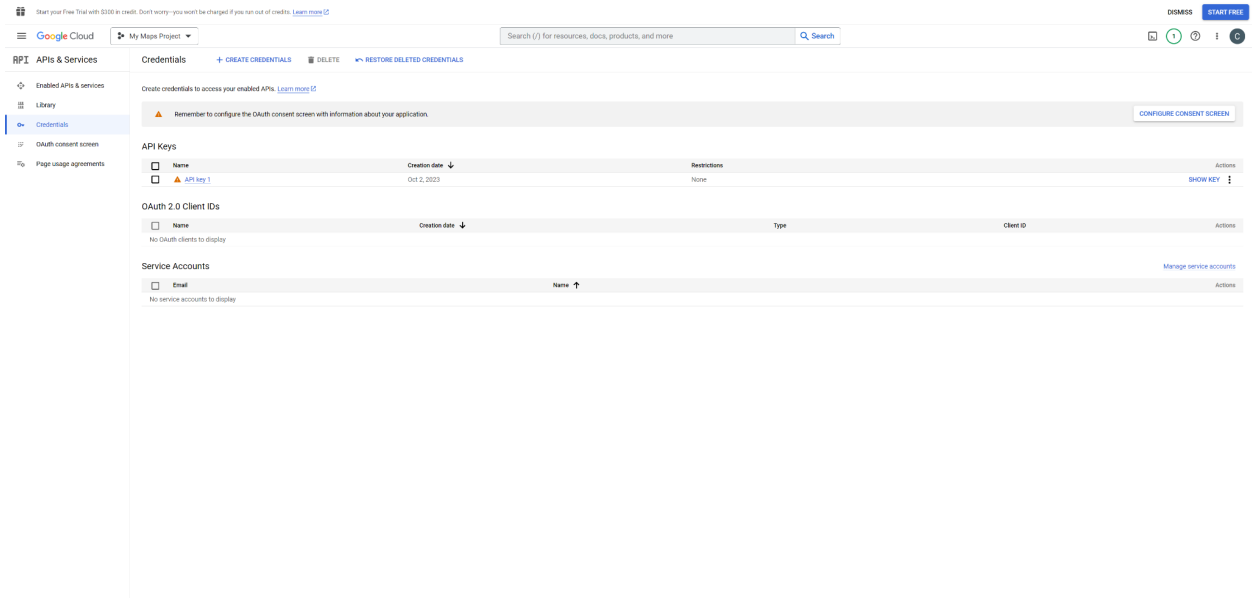
Library

Credentials

OAuth consent screen

Page usage agreements

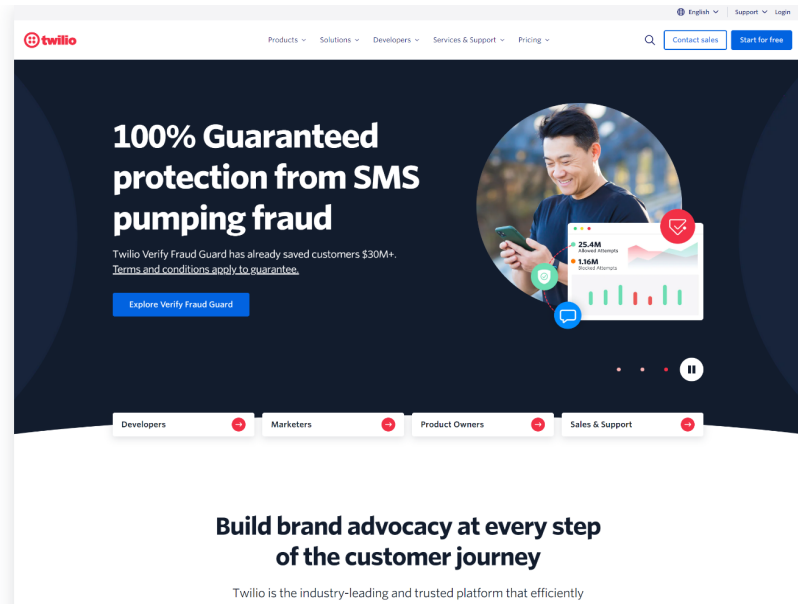
From the Credentials page, you can view or create a new API key if needed



The screenshot shows the Google Cloud console's 'Credentials' page. The left sidebar contains navigation links: 'APIs & Services', 'Enabled APIs & services', 'Library', 'Credentials' (selected), 'OAuth consent screen', and 'Page usage agreements'. The main content area is titled 'Credentials' and includes links for '+ CREATE CREDENTIALS', 'DELETE', and 'RESTORE DELETED CREDENTIALS'. Below this, there's a section for 'API Keys' with a table containing one entry: 'API key 1' created on 'Oct 2, 2023' with 'None' restrictions. A 'SHOW KEY' link is present. Below the API Keys section is the 'OAuth 2.0 Client IDs' section, which is currently empty. At the bottom is the 'Service Accounts' section, also empty. A 'CONFIGURE CONSENT SCREEN' button is visible in the top right of the main content area.

## How to set up Twilio API key

Visit <https://twilio.com>, click Start for Free, and complete registration



Already have an account? [Log in](#)

Start building with Twilio for free.  
No credit card required.

WITH TWILIO YOU CAN BUILD:

- ✓ SMS marketing
- ✓ Omnichannel contact center
- ✓ Call tracking
- ✓ Web chat
- ✓ Push notifications
- ✓ Alerts and notifications
- ✓ Phone verification

First Name \*

Last Name \*

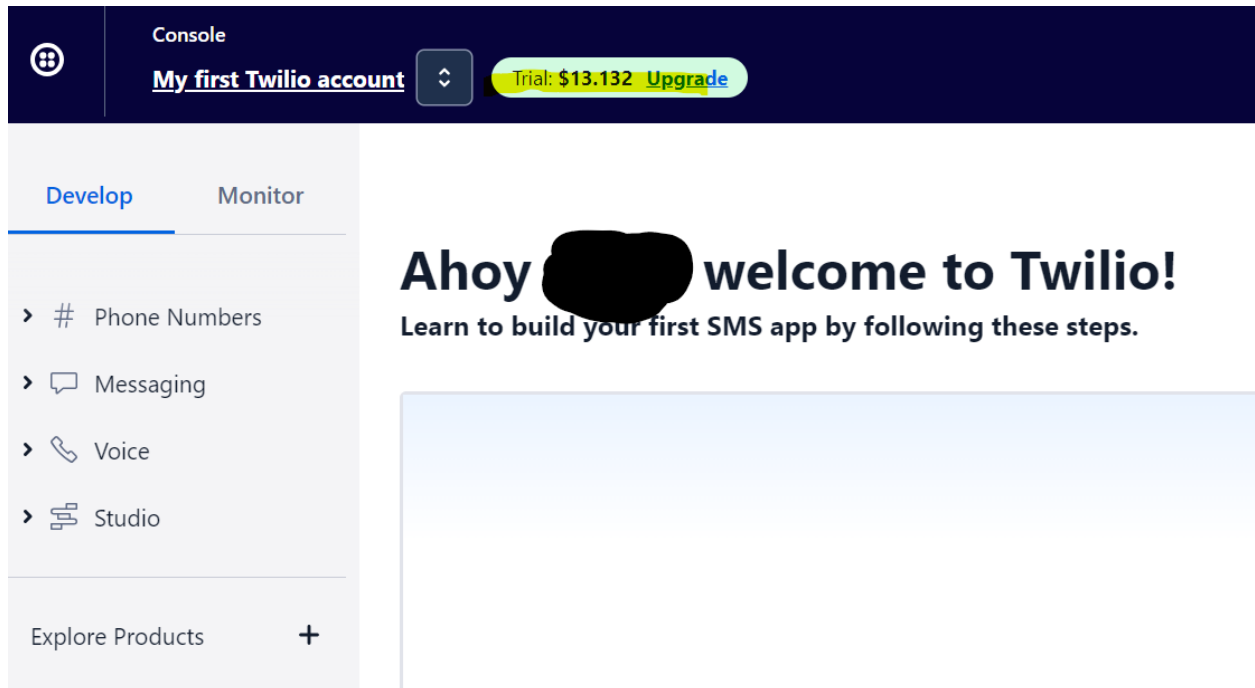
Email \*

Password (16+ Characters) \*

☐ I accept the Twilio Terms of Service and have read the Twilio Privacy Notice. (I am a person or small enterprise or a not-for-profit organization in the USA or UK. I agree to the [Customer Electronic Communications Code \(CECC\)](#) Notice.)

[Start your free trial](#)

Be aware that you will be given a trial credit of \$15.00 for your API testing. You can view your balance in the top left of your console homepage



Follow the prompts to set up API-related information, such as your Account SID, Auth Token, and your Twilio phone number

