

## Basics

Environment  
1st gen

Function name \*  
recharge\_com

Region  
europe-central2

### Try the new Cloud Functions 2nd gen!

This next-generation of our Function-as-a-Service product comes with an advanced feature set giving you powerful infrastructure, advanced control over performance and scalability, more control around the function's runtime, more available regions, and triggers from over 90+ event sources via Cloud Audit Logs. [Learn more](#)

▼ MORE

## Trigger

⊖ HTTP

Trigger type  
HTTP

URL

https://europe-central2-weld-development.cloudfunctions.net/recharge\_com

Authentication

Allow unauthenticated invocations  
Check this if you are creating a public API or website.

Require authentication  
Manage authorized users with Cloud IAM.

Require HTTPS

SAVE CANCEL

## Runtime, build, connections and security settings

< RUNTIME BUILD CONNECTIONS SECURITY AND >

Memory allocated \*  
256 MB

Timeout \*  
60 seconds

## Autoscaling

Minimum number of instances  
0

Maximum number of instances  
3000

## Runtime service account

Runtime service account  
App Engine default service account

By default Cloud Functions uses the automatically created Default App Engine Service Account. [Learn more about service accounts.](#)

## Runtime environment variables

Name 1 * RECHARGE_API_KEY	Value 1 secretkey
Name 2 * WELD_API_KEY	Value 2 secretkey

+ ADD VARIABLE

Configuration — 2 Code

Runtime Node.js 16

Source code Inline Editor

- index.js
- package.json

Entry point \* handler

Press Alt+F1 for Accessibility Options.

```

1  const axios = require('axios');
2
3  //1 API URL
4  const baseSubscriptionsUrl = 'https://api.rechargeapps.com/subscriptions?limit=250';
5
6  //2 API Authentication
7
8  //2.1 Recharge API (set key in cloud function env variable)
9  const api_key = process.env.RECHARGE_API_KEY;
10
11 //2.2 Weld API (set key in cloud function env variable)
12 const checkAuth = (req) => {
13   const token = req.headers.authorization.split(" ")[1];
14   if (token !== process.env.WELD_API_KEY) {
15     const err = new Error("Not authorized");
16     err.status = 403;
17     throw err;
18   }
19 };
20
21 //3: HTTP call
22 exports.handler = async (req, res) => {

```