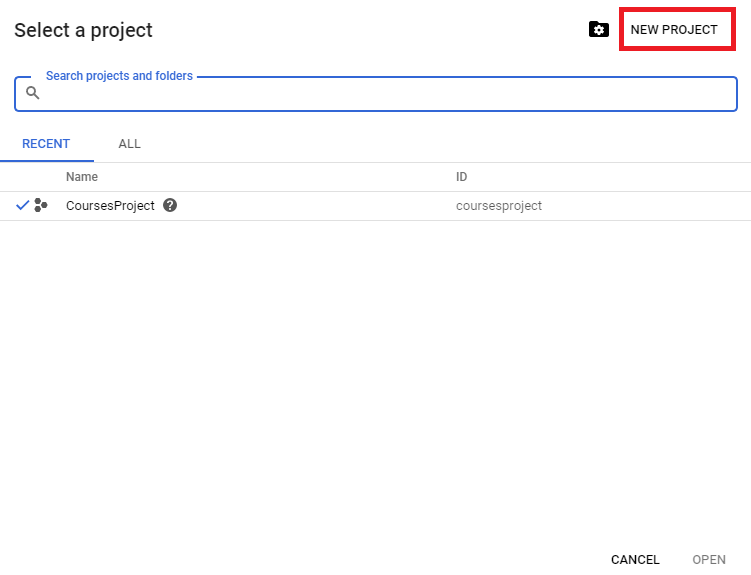
Skripta za prebacivanje Excel-To-Google

Da bi prebacili već postojeće zaposlenike iz Excela u Google sheet, potrebno je prvo otvoriti novi Google projekat, koji će nam dozvoliti pristup našim sheetovima unutar drive-a.  
Da bi otvorili novi google projekat potrebno je da se pomoću Google accounta koji sadrži naše sheetove, logujemo na Google developer konzolu: [https://console.developers.google.com](https://console.developers.google.com/).

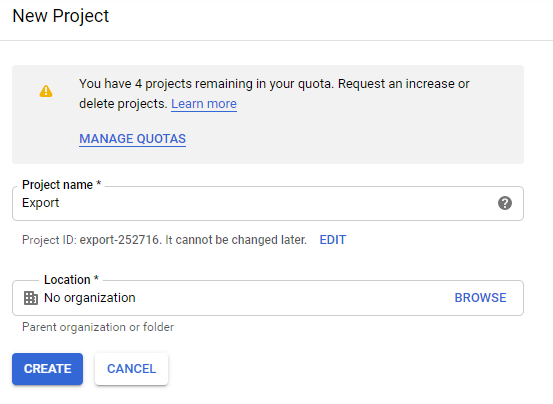
Nakon logina se nalazimo u Google konzoli.  
U gornjem lijevom ćošku se nalazi lista projekata registrovanih na naš Google account. Naš cilj je otvoriti i započeti novi projekat.



Kliknom na projekte ćemo dobiti prozor sa listom svih projekata (ili nijednim projektom), te se u gornjem desnom ćošku pop up prozora nalazi dugme „new project“.

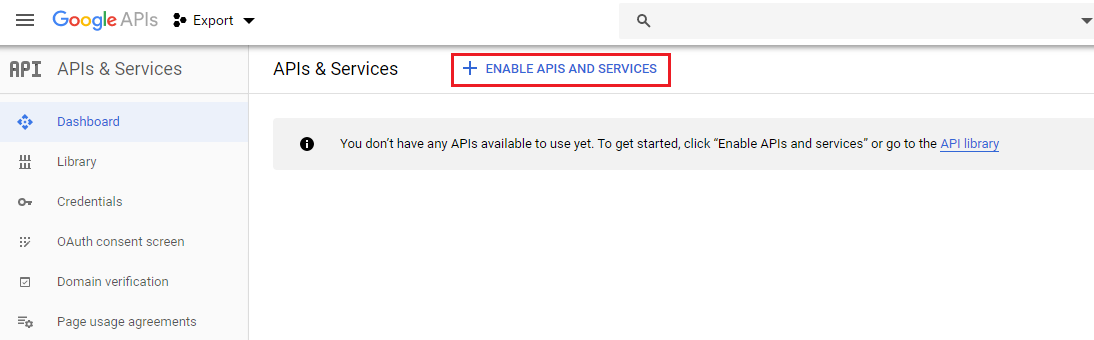


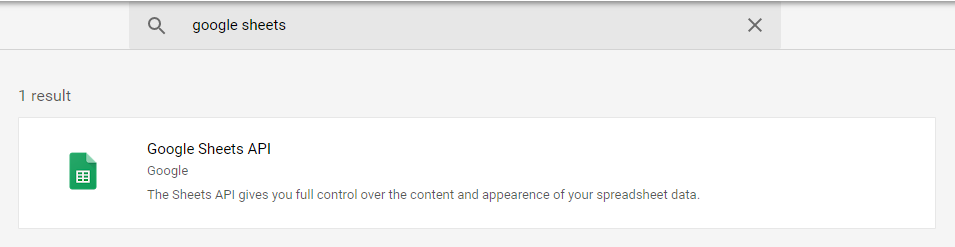
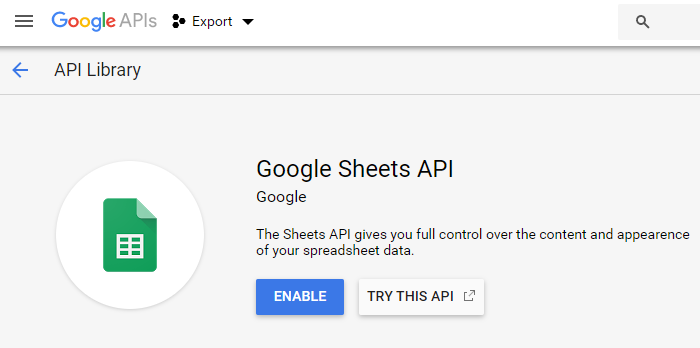
Klikom na new project ćemo dobiti mogućnost da imenujemo novi projekat, te da ga kreiramo. Za ovaj manual projekat će se zvati „Export“.

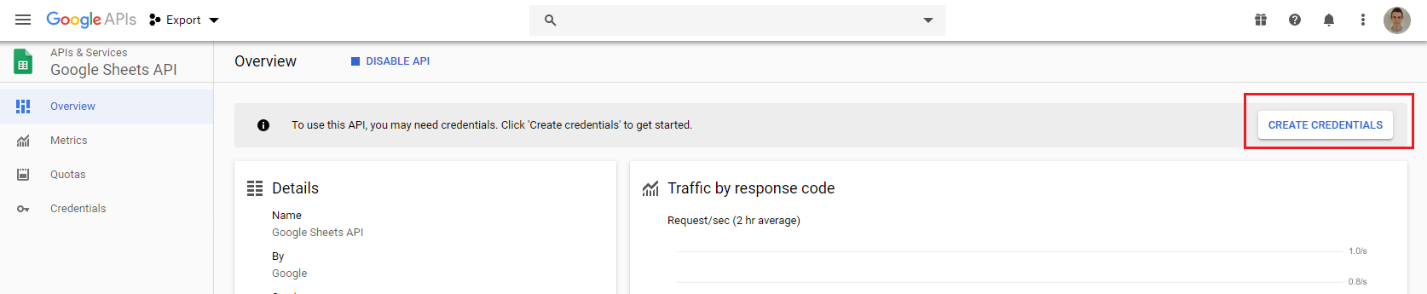


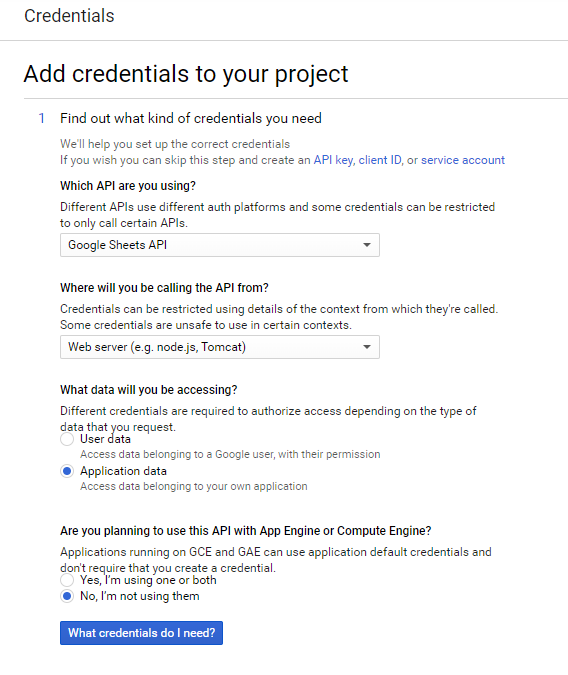
Klikom na „Create“, će se kreirati naš Export projekat (može trajati do minutu).  
Zatim, je potrebno vratiti se na listu projekata u gornjem lijevom ćošku te selectovati naš Export projekat.



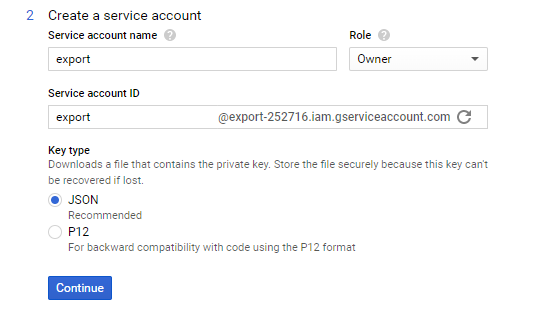
Sljedeće je potrebno da uključimo API servise koji će nam trebati. Za ovu skriptu nam je potreban samo Google Sheets API. Na dashboard-u projekta, sa gornje strane se nalazi dugme „Enable APIs and Services“.

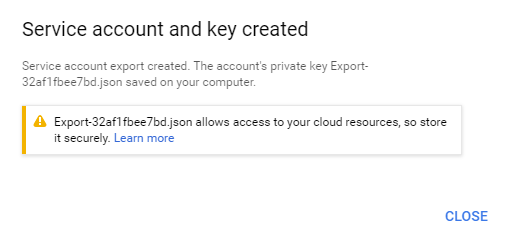
Klik na dugme će nas odvesti u API biblioteku, gdje ćemo searchati „Google Sheets“. Klikom na Google Sheets API će nam se otvoriti API, gdje ćemo kliknuti dugme „Enable“. Sa time smo uključili Google Sheets API za naš projekat.

Nakon što kliknemo enable, stranica će nas odvesti na overview API-a, gdje ćemo morati napraviti credentials, kako bi mogli pristupiti našim dokumentima sa vana. Dugme „Create Credentials“ se nalazi u gornjem desnom ćošku taba overview za Google Sheets API.

Sljedeće će nas dočekati prozor za kreiranje credentials-a. Proces se sastoji od popunjavanje dvije padajuće liste i jednog radio dugmeta.  
Prva padajuća lista nas pita za koji API želimo credentials, tu ćemo odabrati **Google Sheets API** (to bi svakako trebao biti jedini API ponuđen u ovoj listi).  
Druga lista nas pita odakle ćemo vršiti pozive ovom API-u, na ša ćemo odabrati opciju **Web server (e.g. node js, Tomcat)**.  
I na kraju, radio button nas pita koji tip podataka ćemo dohvaćati. Ovdje ćemo čekirati **Application data**.  
Klikom na aplication data će se otvoriti još jedan radio button, koji nas pita da li ćemo koristiti API na jednoj od ponuđenih mašina, te ćemo ovdje kliknuti **No, I'm not using them**.  
Ove dvije liste i radio buttoni su veoma bitni i moraju biti popunjeni na ovaj način.  
Po završetku ćemo kliknuti dugme **„What credentials do I need?“**

Otvoriti će nam se prozor za kreiranje e-mail adrese, kao i setup za role i format podataka. Ova e-mail adresa će nam biti od koristi malo kasnije. Za service account name ćemo u ovom primjeru unijeti „export“, za rolu ćemo kliknuti na padajuću listu, te na prvi tab (project) odabrati „owner“, te ćemo za format čekirati „JSON“.  
Dalje ćemo kliknuti na continue.

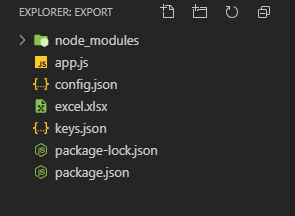


Nakon klika na continue, će se desiti par stvari. Dobiti ćemo potvrdu da su se kreirali svi ključevi, te će se automatski downloadovat-i JSON file.



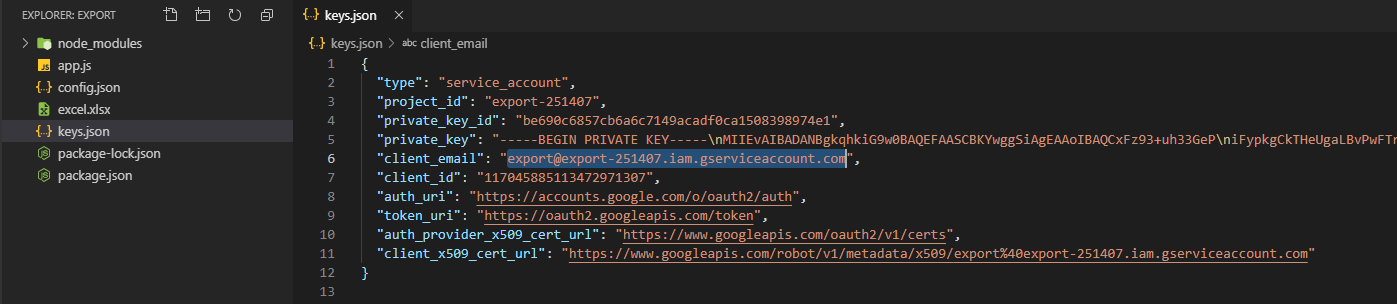
JSON file koji se downloadovao je veoma bitan. Njega ćemo pohraniti na računaru i preimenovati u **keys.json**.

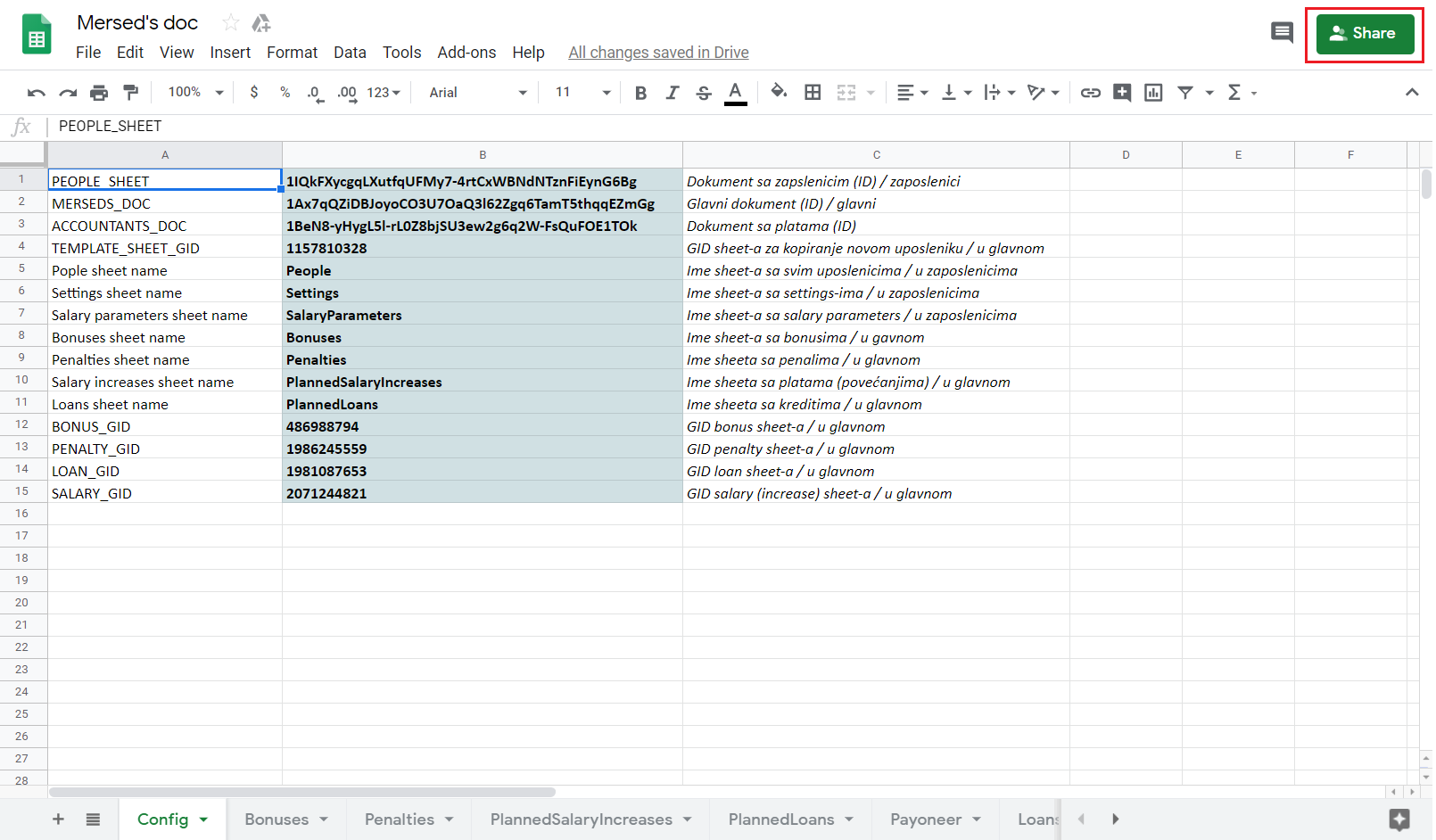
Sa time smo završili setup na Google konzoli i preko keys.json dokumenta imamo pristup svim našim sheetovima tog Google accounta.  
Sada ćemo unzip skriptu koja nosi ime „Export“, otvoriti Visual Studio Code, te otvoriti Export folder   
(File->Open folder->Export) u editoru. Unutar Export projekta će se nalaziti sljedeći sadržaj:



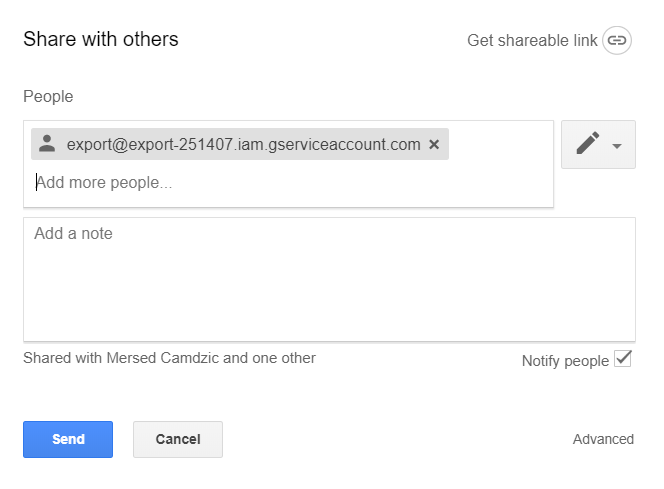
Odmah ćemo obrisati keys.json file, i zamjeniti ga sa novim keys file-om, kojeg smo upravo kreirali i pohranili, kako bi mogli pristupiti našim sheetovima.  
Keys.json file sadrži bitne informacije koje su skripti potrebne da se konektuje sa određenim spreadsheetom.  
Sljedeće je potrebno da odredimo destinaciju, gdje želimo kopirati sve zaposlenike iz Excela. Bitno je da se unutar destinacijskog spreadsheeta nalazi smanjeni template sheet, jer će se on svaki put kopirati za jendog zapolesnika, te će se u njega prebaciti svi podaci tog zaposlenika iz Excela. Template sheet je smanjen, odsječene su sve mrtve ćelije pored i ispod do 2021-e godine, kako ne bi došlo do overflow-a. Za ovaj manual će se opet koristiti primjer Mersed's doc iz Initial data unutar ProjectJupiter, ali ako se koristi neki drugi spreadhseet, potrebno je u njega ubaciti smanjeni template sheet, čiji se primjer nalazi u Mersed's i Main doc.

Nakon što smo odabrali destinacijski spreadsheet i u njega prebacili template sheet, otvoriti ćemo **config.json** u editoru, unutar projekta.  
Tu se nalaze dvije varijable: ID destinacijskog sheeta, te GID template sheeta unutar destinacijskog sheeta. Kopirat ćemo ID Merseds's doc (destinacije) i GID template sheeta i prebaciti u config.json

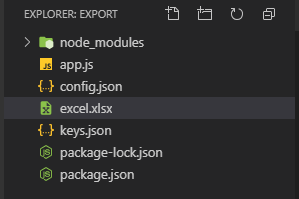
Sada smo rekli našoj skripti gdje želimo kopirati podatke i gdje se nalazi template sheet za kopiranje, za svakog uposlenika.  
Potreban je uraditi još samo jedan korak sa Google strane setupa. Naša skripta trenutno nema pravo da dohvaća i mijenja naš sheetove, prvo to moramo odobriti. Da bi to odobrili moramo sharea-ti naš destinacisjki spreadsheet Mersed's doc sa e-mail adresom na koju je registrovan naš projekat.  
Tu e-mail adresu ćemo naću unutar keys.json file-a, kojeg smo kreirali i zamjenili unutar editora.  
Klikom na keys.json file, ćemo vidjeti mnogo ključeva i tokena, ali je nama bitna **client\_email** json varijabla.  
Oznaciti ćemo i kopirati tu e-mail adresu.

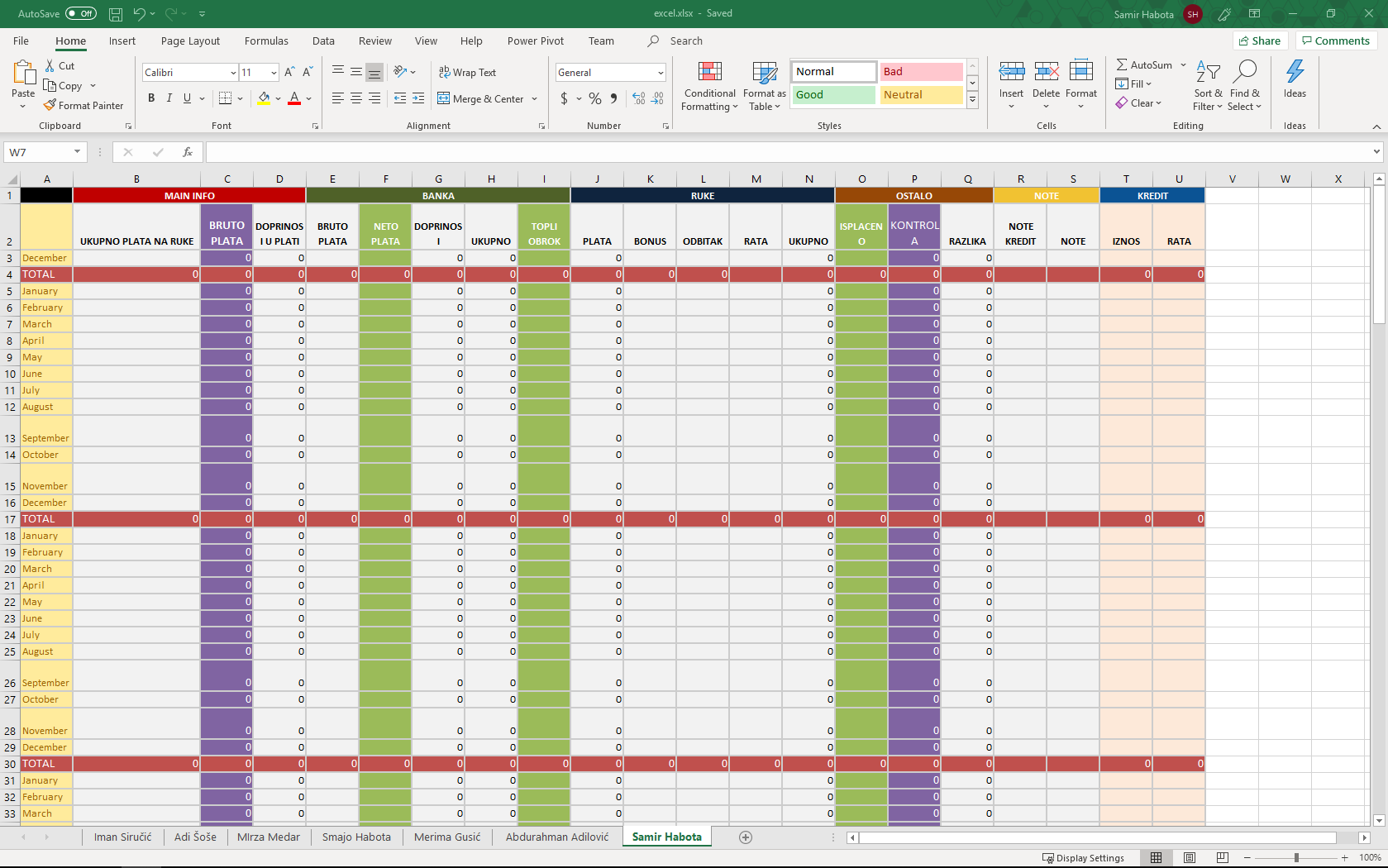
Zatim ćemo otići na naš destinacijski sheet Mersed's Doc, te u gornjem desnom ćošku kliknuti dugme **Share**.

Tu ćemo paste-ati e-mail adresu koju smo kopirali iz keys.json, te kliknuti send. Sa time smo čitav spreadsheet podijelili sa našom skriptom i našim Google projektom, te imam pravo da manipulišemo podacima. E-mail adresa nije javna, služi samo za spajanje našeg projekta sa sheetovima, preko server NodeJs poziva.



Nakon destinacije, presotalo nam je da podesimo izvor odnosno Excel dokument.  
U projektu će se nalaziti primjer Excel dokumenta, kako bi se moglo lakše sve provjeriti. U Excel file-u je potrebno obrisati sve sem tabova zaposlenika (dakle samo tabovi imena i prezimena zaposlenika sa tabelom podataka se trebaju nalaziti u excelu), te je sam file potrebo preimenovati u excel(.xlsx), te zamjeniti dummy excel file koji se nalazi unutar projekta, sa pravim.



Podaci unutar Excela se moraju poklapati sa podacima u dummy excelu, dakle da prvi zapis počinje od decembra 2010 godine na trećem redu, te da se sve kolone poklapaju. Imena kolona nisu bitna, dobiti će nova imena prilikom kopiranja, bitno je sve od trećeg reda pa na dolje, da je po kolonama na istom mjestu.  
Dakle ćelija A3 da bude popunjena sa decembrom 2010-e, te da kolone idu do U kolone. Ispod ide total red, pa januar za 2011 itd, kao primjer svega što smo do sada radili. Redovi jedan i dva, zagljavlja nisu bitna sadržajem ali se moraju nalaziti tu.

Sa ovim je skripta spremna za pokrenuti, ali ima par ograničenja.

Nakon pokretanja se dešavaju sljedeće stvari. U destinacijskom spreadsheetu se kopira template sheet zajedno sa praznom tabelom. Uzima se prvi tab iz Excela za jendu osobu te se novi Google sheet preimenuje u ime i prezime zaposlenika iz Excela. Zatim se svi podaci u Excel tabeli kopiraju u Google sheet tabelu. Nakon toga se prelazi na sljedećeg uposlenika. Skripta dok traje će konstantno javljati kada je napravila sheet za određenu osobu i kada je popunila taj sheet.   
  
Google Sheets API može podnjeti 30000 istovremenih poziva. To znači da se moraju praviti razmaci između uposlenika, jer jedan uposlenik prabacuje velik broj ćelija.  
Također, prilikom preimenovanja Google sheetova u ime i prezime uposlenika, API sam od sebe prvo prolazi kroz sve ostale sheetove, kako bi vidio da li već postoji sheet sa istim imenom, te ako postoji zaustavlja API poziv.  
Zbog toga se za istoimene i prezimene zaposlenike treba dodati unikatni karakter (tipa samir habota 1, samir habota 2) u Excelu, prije nego što se skripta pokrene.  
Zbog toga što se prvo moraju preći imena svih predhodnih sheetova da bi se ubacio novi, na dodavanju 200-og sheeta se mora preći 199 sheetova, što traje malo duže.  
Iz tog razloga se mora obezbijediti veći vremenski razmak između prebacivanja jednog uposlenika, inače će se krenuti prebacivati predhodni uposlenik, dok se trenutni još nije stigao preimenovati, jer nije bilo dovoljno vremena da se prođu svi dotadašnji sheetovi.  
Ovo ponašanje je ponašanje samog API-a i ne može se zaobići.  
Zbog toga skripta pokreće proces prebacivanja uposlenika svako 15 sekundi, što je u testiranjima bilo dovoljno vremena za prebacivanje do 250+ uposlenika.  
  
U slučaju da se skripta prekine, što nije vjerovatno, razlog je nedovoljno vremena pred kraj prebacivanja, između uposlenika, kada se mora preći preko 150 imena prije ubacivanja sljedećeg

U slučaju da dođe do prekida, rješenje je sljedeće:

Povećati vrijeme između ubacivanja uposlenika, što se može uraditi unutar **delay()** funkcije u **app.js**, na liniji 33. Trenutni razmak je 15 sekundi (15000 milisekundi), postaviti ga na recimo 20 (20000), te pokrenuti skriptu ponovno, ali prije toga provjeriti do kojeg uposlenika je skripta uspješno prebacila podatke, te prebačene uposlenike obrisati iz Excel file-a. Tako će se skripta nastaviti gdje je stala sa preostalim uposlenicima i više vremena između prebacivanja.



Na 15 sekundi razmaka je za 250 uposlenika bilo potrebno oko 30 minuta trajanja skripte.  
U najgorem slučaju je moguće da ostanu par uposlenika koji se mogu ručno prebaciti, mada opet, nije vjerovatno.

Naravno, ako se želi izbjeći bilo kakav rizik, a ima se dovoljno vremena za čekanje, što je lična preporuka, vrijeme razmaka se može odmah staviti na sigurnih 30 sekundi, ali će skripta onda trajati znatno duže.

Kod skripte se nalazi u **app.js**. Za pokretanje je potrebno otvoriti terminal prozor (View->Terminal ili ctrl/comand + backtick), koji će po dafaultu biti otvoren unutar Export foldera.   
Zatim je potrebno ukucati komandu **node app.js**.  
Ako prilikom izvršenja skripte bilo šta krene po zlu, skripta se iz editora može zaustaviti kucajući u terminal pokrenute skripte komadu **ctrl/command + c**.

