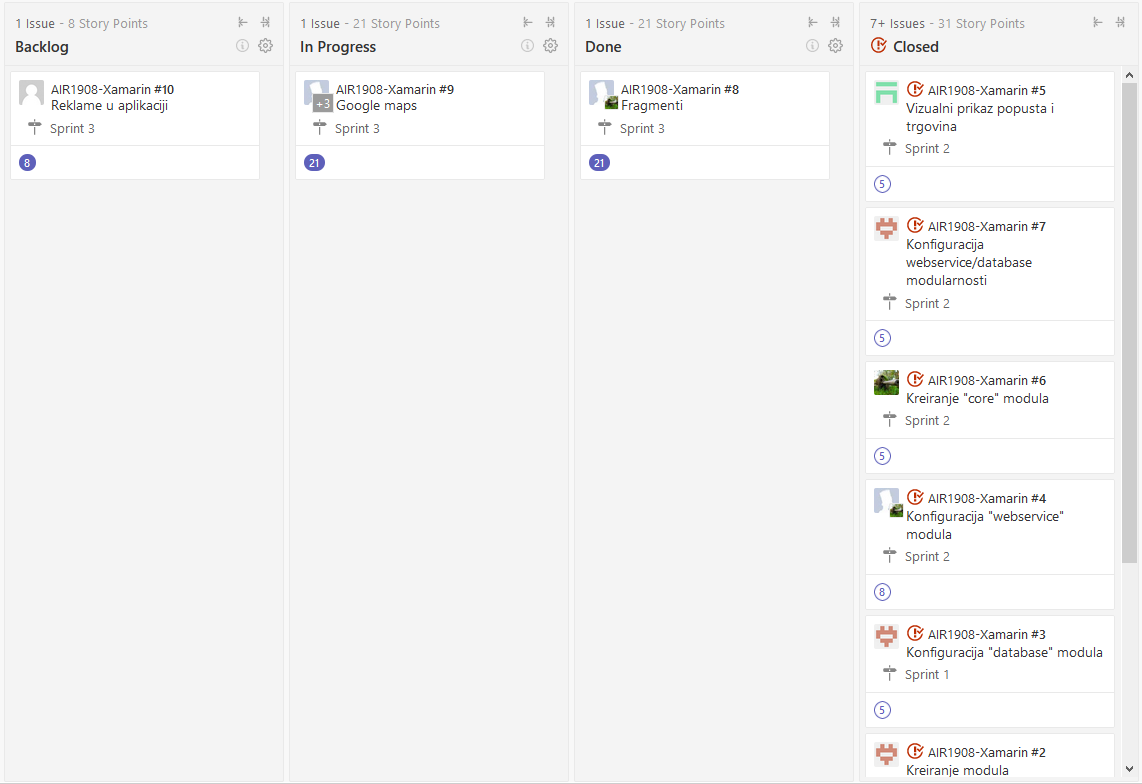
***Sažetak***

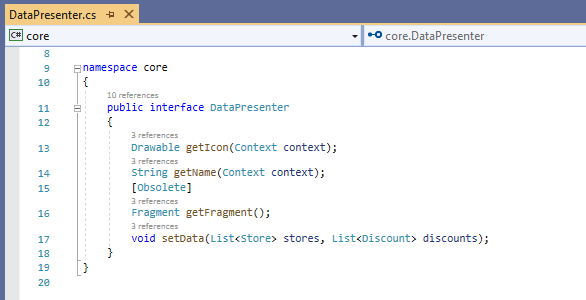
U prošloj vježbi bavili smo se fragmentima. Prikazali smo dućane i popuste u jednom fragmentu, a detalje o popustima u drugom fragmentu. Dodali smo i navigaciju (glavni izbornik aplikacije). U ovoj ćemo vježbi pripremiti sve što nam je potrebno za prikaz dućana na Google mapi. To će nam biti potrebno za modularni prikaz popusta (odabir ćemo raditi preko navigacije – ili lista ili mapa). Fragmenti će nam sada biti moduli koje ćemo u sljedećoj vježbi prikazivati.

Zadnja verzija aplikacije: Commit 1d6954d [<https://github.com/AIR-FOI-HR/AIR1908-Xamarin>]

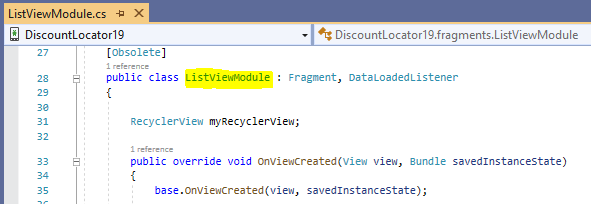
***Što radimo?***



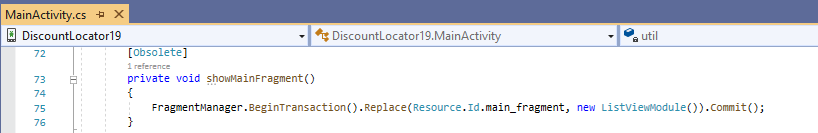
Prvo ćemo kreirati DataPresenter interface u core modulu. Desni klik na core modul -> Add -> New Item… -> On ima sljedeće metode: getIcon, getName, getFragment i setData.



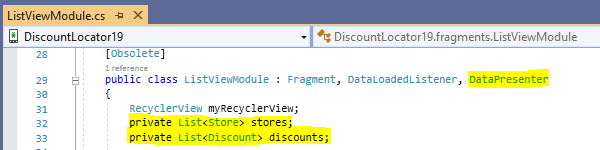
Sada ćemo kreirati novu mapu modules u koju ćemo prebaciti ListViewFragment klasu iz prošle vježbe. Desni klik na DiscountLocator19 -> Add -> New Folder -> modules. Iz fragments mape prebacimo ListViewFragment u modules mapu i preimenujemo klasu u ListViewModule.

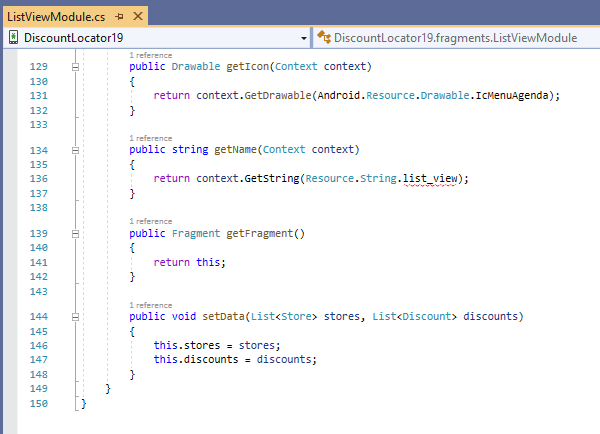


U MainActivity klasi moramo promijeniti naziv fragmenta u showMainFragment metodi.

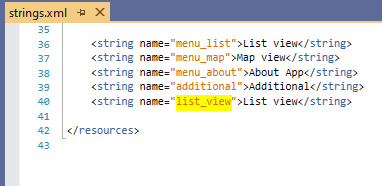


Klasa ListViewModule nasljeđivat će DataPresenter interface i implementirat će njegove metode. Dodajemo i listu dućana i listu popusta koja će nam trebati u setData metodi DataPresenter interface-a.

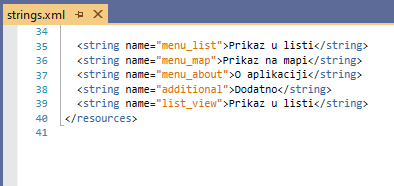




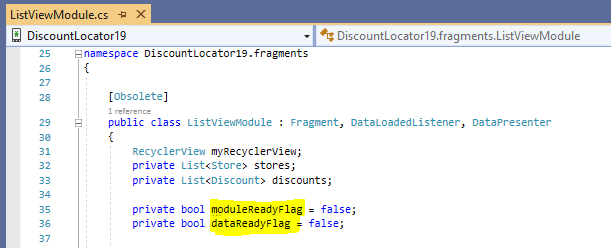
Moramo dodati novi naziv u strings.xml. DiscountLocator19 -> Resources -> values -> strings.xml:



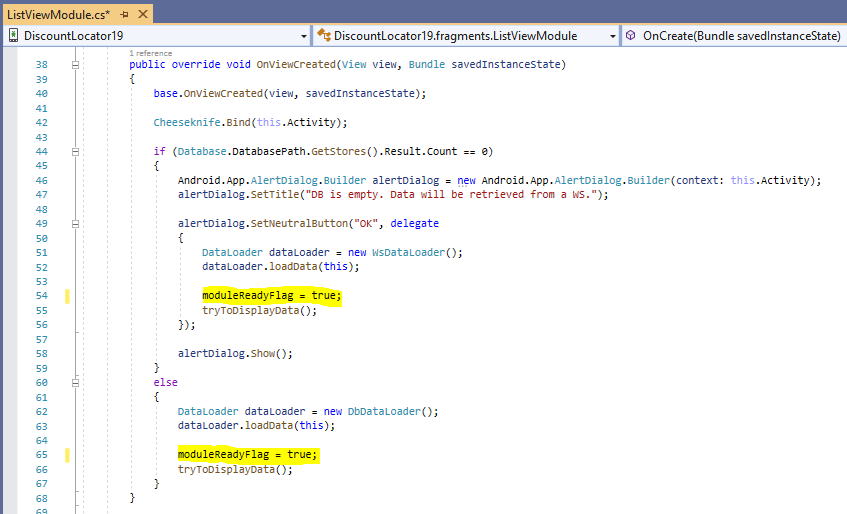
DiscountLocator19 -> Resources -> values-hr -> strings.xml:

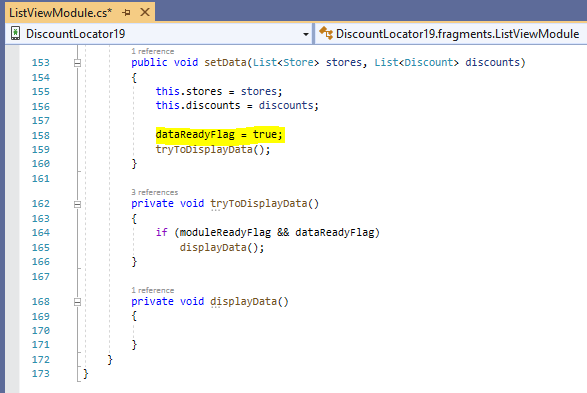


Sad ćemo dodati zastavice moduleReadyFlag i dataReadyFlag kako bi ListViewModule znao kad je modul spreman i kad su podaci spremni.

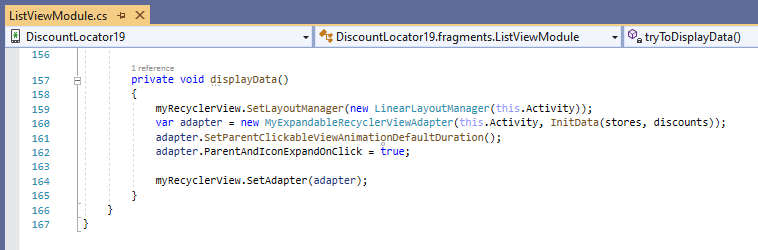


Zastavicu moduleReadyFlag postavit ćemo kad se pozove dataLoader (WsDataLoader ili DbDataLoader). Nakon što postavimo zastavicu moduleReadyFlag, pozivamo metodu tryToDisplayData koja pokušava prikazati podatke. Zastavicu dataReadyFlag postavit ćemo u setData metodi nakon što se dohvate podaci. Nakon toga opet pozivamo tryToDisplayData metodu koju u kojoj ćemo ispitati zastavice. Ako su obje postavljene, pozivamo metodu displayData.

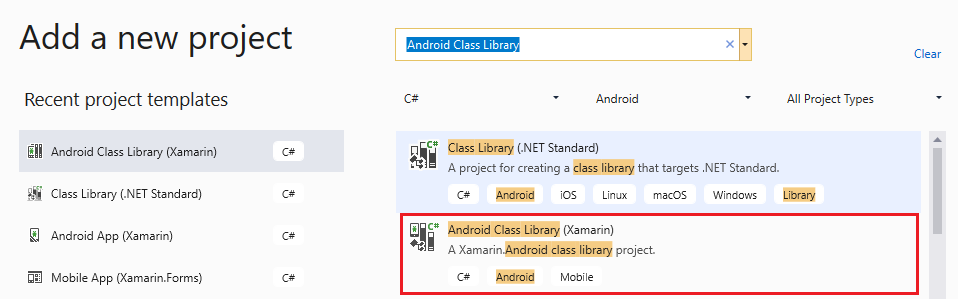


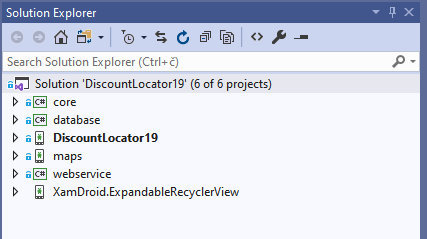


Sada ćemo izbrisati nasljeđivanje DataLoadedListener interface-a u ListViewModule klasi. Također brišemo i onDataLoaded metodu tog interface-a. Implementirat ćemo displayData metodu koja će raditi on što je radila onDataLoaded metoda.

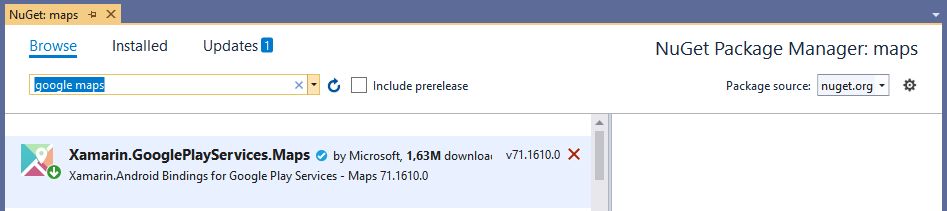


Sada ćemo dodati novi modul koji ćemo nazvati maps. On će biti zadužen za prikaz Google mape koju ćemo kasnije dodati. Desni klik na Solution -> Add -> New Project… -> Android Class Library (Xamarin) -> maps.

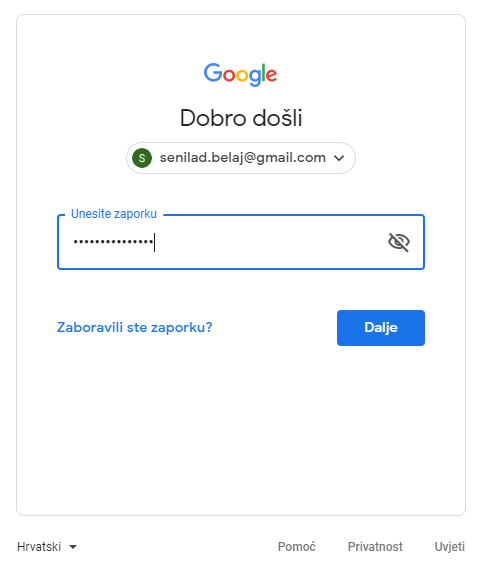




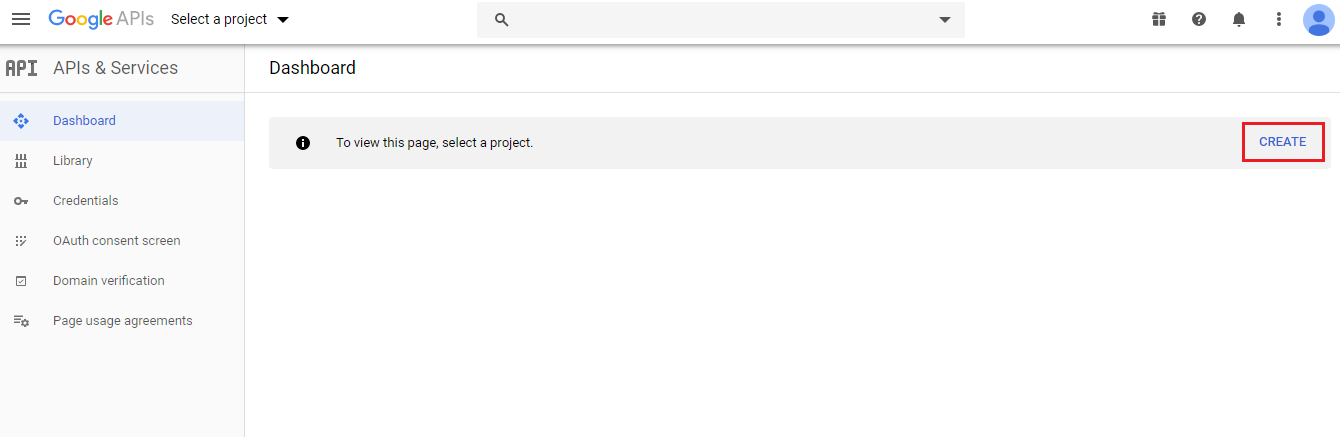
Da bismo radili s Googleovim mapama, moramo uključiti Google Play Services paket. Desni klik na maps modul -> Manage NuGet Packages… -> u tražilicu upišemo 'google maps'. Instaliramo prvi ponuđeni paket.



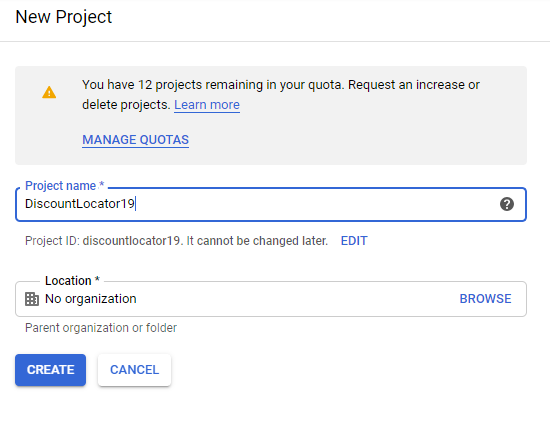
Sada moramo generirati API ključ za korištenje Google mape. Moramo otići na Googleovu web stranicu: Google Developers Console (<https://console.developers.google.com>). Prijavimo se sa svojim Google računom.



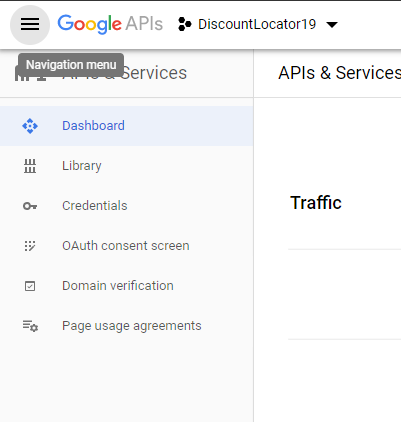
Dolazimo do početne stranice na kojoj ćemo kreirati svoj novi projekt.



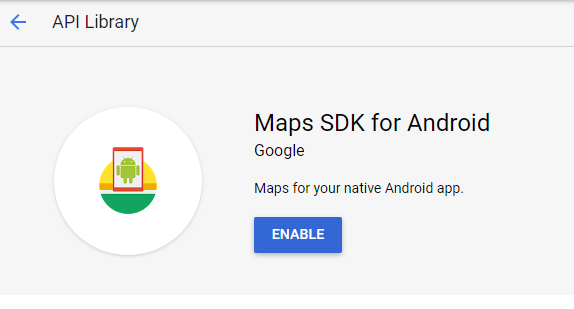
Kliknemo na 'Create' i otvorit će nam se sljedeći prozor:



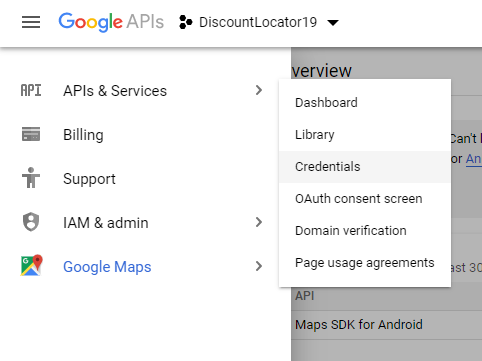
Unesemo naziv projekta i stisnemo gumb 'Create'. Sad kad smo kreirali novi projekt, moramo omogućiti Maps SDK biblioteku za Android. U izborniku lijevo odaberemo 'Library'.



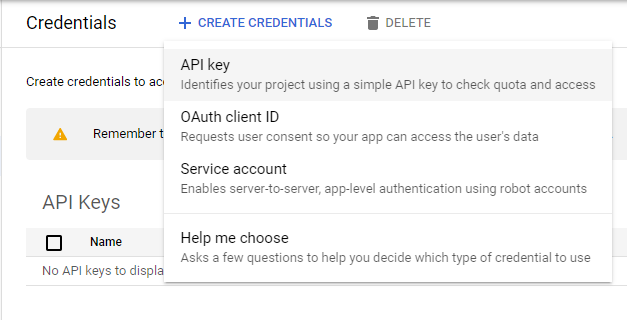
U tražilicu unesemo 'Maps SDK for Android'. Odaberemo sljedeću biblioteku i kliknemo 'Enable'.

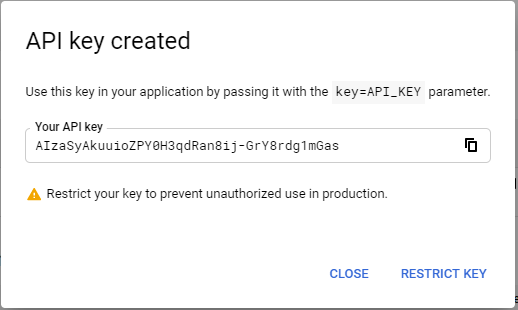


Sada kliknemo na 'APIs & Services' stavku u izborniku -> Credentials.



Odaberemo 'Create Credentials' -> API key.

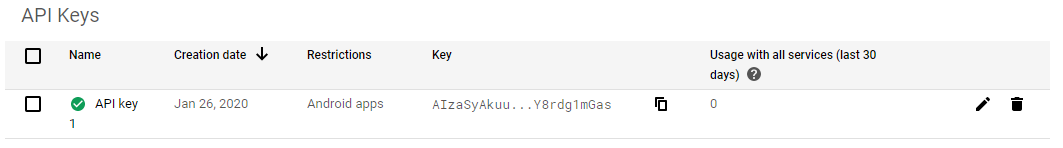




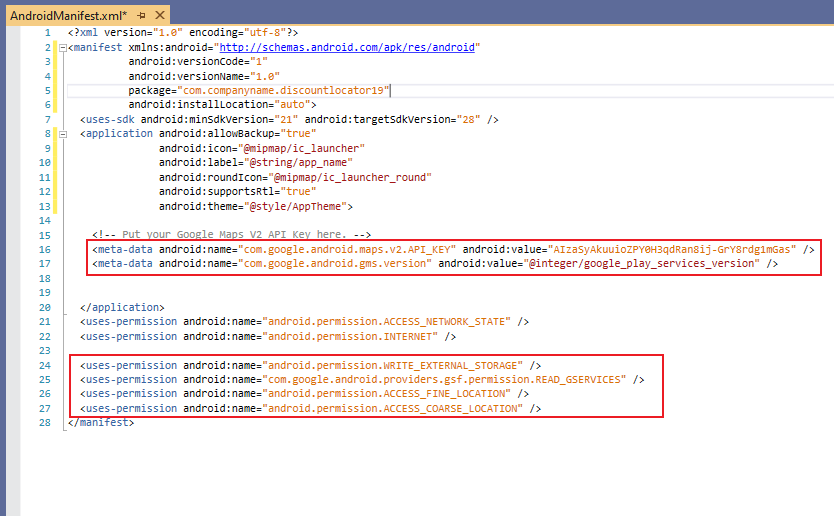
Još je moguće napraviti restrikcije kako bismo svoj ključ zaštitili od neautoriziranog korištenja. Koraci su detaljno opisani na sljedećoj web stranici:

<https://docs.microsoft.com/en-us/xamarin/android/platform/maps-and-location/maps/obtaining-a-google-maps-api-key?tabs=windows>

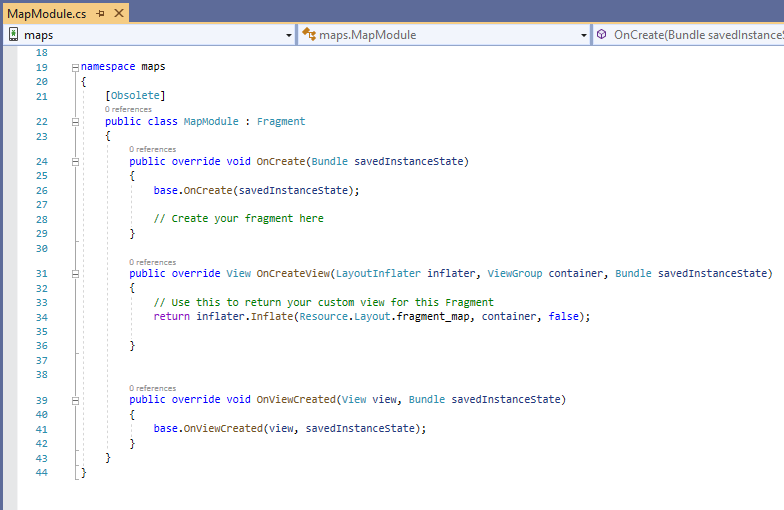
Naš novokreirani API ključ bit će prikazan na ekranu.



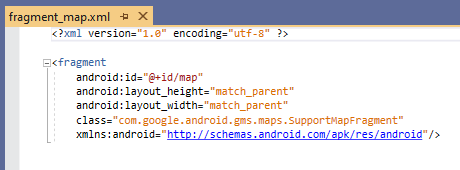
Nakon što smo kreirali API ključ, moramo ga staviti u našu aplikaciju. U AndroidManifest.xml dodat ćemo naš ključ i ostale postavke. DiscountLocator19 -> Properties -> AndroidManifest.xml.



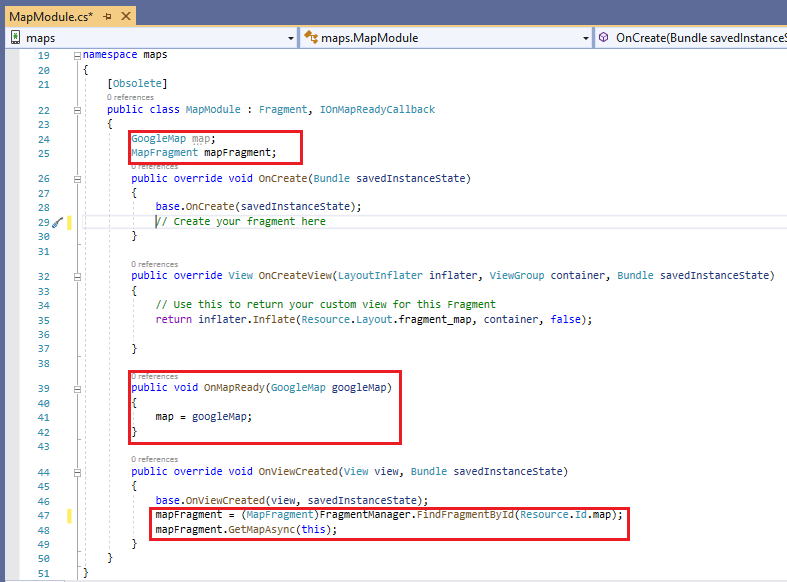
Sada ćemo kreirati novi modul MapModule koji će nam služiti kao fragment za prikaz Google mape. Desni klik na maps modul -> Add -> New Item -> Fragment -> MapModule.



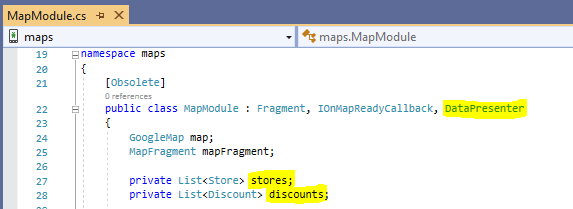
Moramo kreirati i pripadajuću xml datoteku. Modul maps -> Resources -> desni klik na mapu values -> Add -> New Item -> XML File -> fragment\_map.xml.

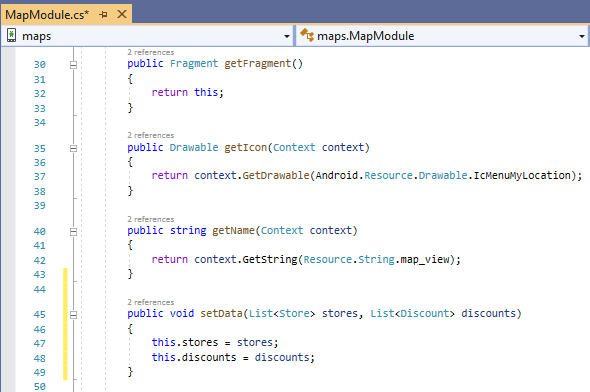


Naš MapModule fragment mora nasljeđivati IOnMapReadyCallback interface. Dodajemo dvije varijable, jednu tipa GoogleMap koja se nalazi u Android.Gms.Maps biblioteci i jednu tipa MapFragment koja se nalazi u istoj biblioteci. U metodi OnViewCreated povezujemo naš fragment s gore definiranim prikazom u xml-u.



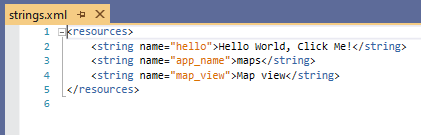
Isto kao i ListViewModule, MapModule mora nasljeđivati DataPresenter interface i implementirati sve njegove metode. Također ćemo kreirati dvije liste za dućane i popuste koje ćemo puniti kad podaci budu spremni.



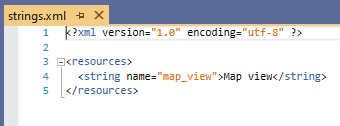


Dodajemo naslove u strings.xml datotekama za hrvatski i engleski jezik.

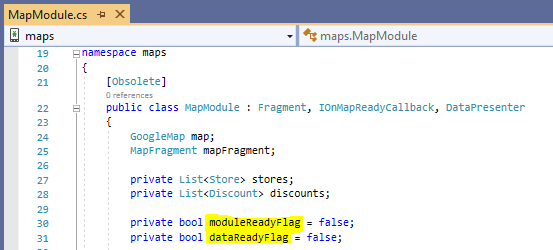
Maps -> Resources -> values -> strings.xml.



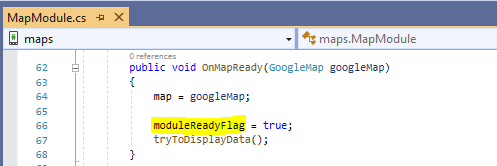
Maps -> desni klik na Resources -> Add -> New Folder -> values-hr. Desni klik na values-hr -> Add -> New File -> XML File -> strings.xml.



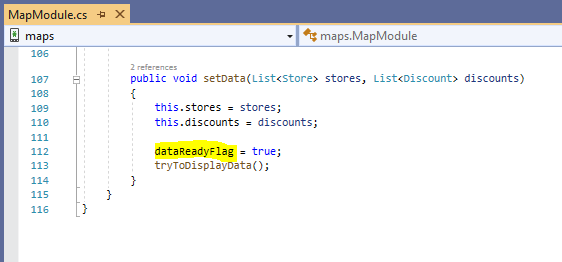
Sada je potrebno dodati zastavice u MapModule fragment kako bi on znao kad su podaci dostupni.

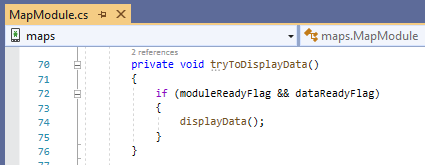


Zastavicu moduleReadyFlag postavit ćemo u OnMapReady metodi. Odmah ćemo pozvati tryToDisplayData metodu koja će pokušati prikazati podatke.

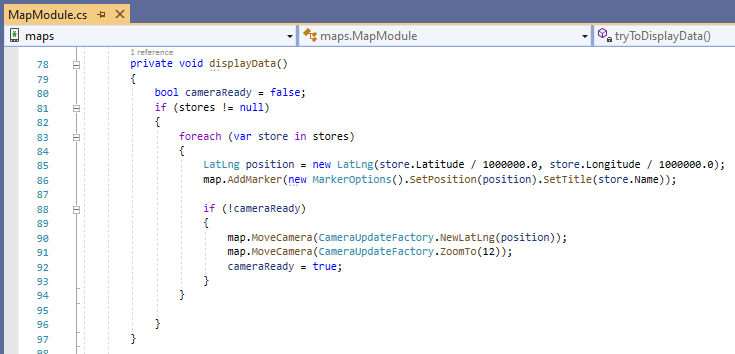


Zastavicu dataReadyFlag postavit ćemo u setData metodi kad nam dođu podaci. I odavde pozivamo tryToDisplayData metodu.





Ako su obje zastavice postavljene, poziva se metoda displayData koja prikazuje podatke na našem fragmentu. Za svaki dućan u listi dućana kreira se novi objekt LatLng klase. On prima latitudu i longitudu koju dohvaćamo preko funkcija objekta store. Zatim se dodaje marker na poziciji koju smo maloprije kreirali. Onda se kamera pozicionira na marker na Google mapi.



Sada su pripremljena oba fragmenta ListViewModule i MapModule za prikaz podataka. ListViewModule zadužen je za prikaz dućana i popusta u listi (expandable recycler view), a MapModule zadužen je za prikaz dućana na Google karti. U sljedećoj vježbi riješit ćemo logiku koja će odlučivati koji se fragment poziva na temelju korisnikova odabira u navigaciji.