**מ פ ו ת – Google maps**

מפות ומיקומים הוא אחד הנושאים היפים והמעניינים "להתעסק" איתם.

מפות זה לא רק Waze או ניווט מנקודה לנקודה ברכב.

Google Maps מאפשר 4 אפשרויות ניווט: רכב, אופניים, רגלי או תחבורה ציבורית.

בנוסף יש הרבה אפליקציות מבוססות מיקום.

לדוגמא: אני נמצע בנקודה מסוימת ורוצה לפגוש חברים הנמצאים כרגע בקירבתי.  
דוגמא נוספת: עוברים בקרבת חנות מסוימת שמעוניינת למשוך לקוחות העוברים כרגע ליד החנות.

**עבודה עם מפות במערכת Android**

עבודה עם מפות ב-Android באמצעות Xamarin אפשרית ב-2 אופנים:

1. Maps API.

2. Xamarin.Essentials.

Maps API – עבודה ישירה מול Google מפות. דורש הגדרת חשבון ב-Google Maps וכרוך בתשלום.

Xamarin.Essentials הינה מעטפת לעבודה אחידה עם מפות (ועוד) באמצעות מערכת הפעלה Android או מערכת הפעלה iOS.  
עבודה עם Essentials ב-Android איננה דורשת הגדרת חשבון ב- Google מפות ונושא התשלום לא ברור (בכל מקרה, אין שום התחייבות בהתחלה).

**המחלקה GeoLocation**

המחלקה GeoLocation מספקת ממשקי API קבלת מיקום המכשיר במונחי מיקום (קו אורך, קו רוחב).

**מיקום**

מיקום על פני כדור הארץ הוא נקודת מפגש בין קו אורך – longitude וקו רוחב – latitude.

באמצעות התקן (טלפון, טבלט, מחשב) ניתן אתר את המיקום שלו באמצעות 2 מערכות: 1. Gps 2. מערכת סלולאר וגם שילוב שלהם.

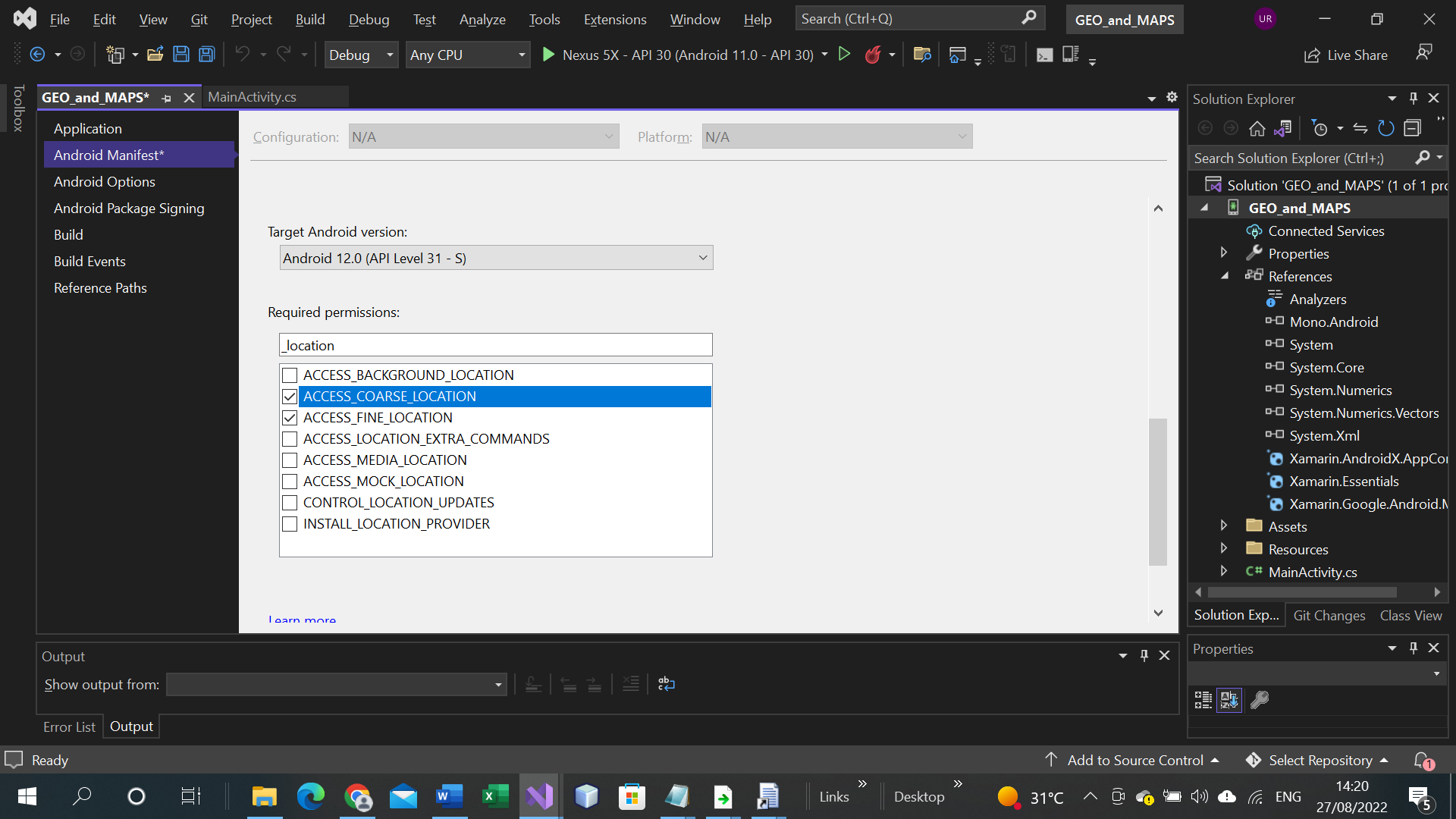
Gps – ניתן למצוא מקום בדיוק גבוהה (מטרים ספורים).  
מערכת סלולאר – ניתן למצוא מיקום בדיוק בינוני עד נמוך (כ-3 קמ"ר).

**כיצד מתחילים**

יוצרים פרויקט חדש.

בקובץ **Manifest** יש להגדיר איזה סוג דיוק אנחנו מבקשים:  
ACCESS\_COARSE\_LOCATION – דיוק מקורב.

ACCESS\_FINe\_LOCATION – דיוק גבוהה.  
  
R-Click על הפרויקט -> Properties



במידה ונרצה שהגישה למיקום תהיה גם ב-Bacground (מעקב קבוע אחר ה-Location בעבודה ברקע, למשל ב-Service) יש להוסיף לקובץ AssembliyInfo.cs את הפקודה הבאה:

]assembly: sesPermission(Manifest.Permission.AccessBackgroundLocation)[

לחילופין ניתן להגדיר זאת בקובץ Manifest ידנית ב-XML כך:

<uses-permission android:name="android.permission.ACCESS\_BACKGROUND\_LOCATION" />

**תזכורת !!!** גישה למיקום הינה פעולה העלולה לפגוע בפרטיות הלקוח לכן יש לבקש הרשאות בזמן ריצה בנוסף להגדירה בקובץ Manifest..

במטודה **OnCreate** של MainActitivty נוסיף את הפקודה:

Xamarin.Essentials.Platform.Init(this, savedInstanceState);

**עבודה עם Geolocation**

בכל Activity בו נרצע לעבוד עם Geolocation יש להגדיר:

using Xamarin.Essentials;

**קבלת מיקום אחרון**

ניתן לקבל את המיקום האחרון שנקלט במערכת.

try

}

Location location =

await Geolocation.GetLastKnownLocationAsync();

if (location != null)

{

double longitude = location.Longitude;

double latitude = location.Latitude;

}

else

{

}

}

catch (FeatureNotSupportedException fnsEx)

{

// Handle not supported on device exception

}

catch (FeatureNotEnabledException fneEx)

{

// Handle not enabled on device exception

}

catch (PermissionException pEx)

{

// Handle permission exception

}

catch (Exception ex)

{

// Unable to get location

}

ראה [כאן](https://docs.microsoft.com/en-us/dotnet/api/xamarin.essentials.location?view=xamarin-essentials) אילו נתונים ניתן לקבל מהאובייקט Location מעבר לקו אורך ורוחב.

**שים/י לב !!!** - Location יכול להיות null אם לא נשמר מקום אחרון.

**קבלת מיקום נוכחי**

קבלת מיקום נוכחי הינה פעולה התלויה בכמה פרמטים.

פרמטר ראשון הוא **מידת הדיוק** הרצויה.  
רמת הדיוק קובעת באיזו דרך לנסות לקבל את המיקום הנוכחי וכן הזמן שייקח לקבל את המיקום הנוכחי.

אם רמת הדיוק המבוקשת **נמוכה**, ניתן לקבל את המיקום הנוכחי באמצעות הרשת הסלולארי או באמצעות WiFi.  
אם האיתור מתבצע באמצעות הרשת הסלולארית יתכן כי זמן קבלת התשובה יהיה ארוך יותר מאשר איתור באמצעות WiFi.

אם רמת הדיוק המבוקשת **גבוהה** (הכי טובה אפשרית) יש לבצע את האיתור באמצעות Gps. איתור בדרך זו עלולה לארוך זמן רב.

כפי שרואים, איתור המקום הנוכחי יארך זמן מסויים.  
ניתן להמתין עד לקבלת המיקום הנוכחי וכן ניתן להחליט שהאיתור יעשה במשך זמן מסויים. אם לא התקבל מיקום בפרק הזמן המבוקש נקבל null.

**רמת דיוק – GeolocationAccuracy**

|  |  |  |
| --- | --- | --- |
| **רמת דיוק** | **הסבר** | **טווח טעות (במטרים)** |
| Best | רמת דיוק הטובה ביותר | 10 |
| High | רמת דיוק גבוהה | 10-100 |
| Medium | רמת דיוק בינוית | 30-500 |
| Low | רמת דיוק נמוכה | 300-3000 |
| Lowest | רמת דיוק נמוכה ביותר | 1000-5000 |
| Default | זהה לרמת דיוק בינונית | 30-500 |

**זמן המתנה לתוצאה**

זמן ההמתנה לתוצאה – אוביקט מהטיפוס TimeSpan.

**קביעת סוג הבקשה** (רמת דיוק, זמן המתנה לתוצאה(

GeolocationRequest request =

new GeolocationRequest(GeolocationAccuracy.Medium,

TimeSpan.FromSeconds(10)

);

**קבלת המיקום הנוכחי**

Location location = await Geolocation.GetLocationAsync(request);

**חישוב המרחק בין 2 נקודות על המפה**

באמצעות המחלקה Geolocation ניתן לקבל את המרחק בין 2 נקודות על המפה.  
המרחק המתקבל הוא המרחק הקצר ביותר בין 2 הנקודות תוך התחשבות בקמירות כדור הארץ. למעשה זה אורך הקשת בין 2 נקודות המרחב.

**הגדרת נקוגה המפה**

Location location = new Location(קו רוחב, קו אורך)

**חשוב לדעת !**

**קו רוחב** – ערכים חיוביים מציינים מקומות מצפון לקו המשווה.

**קו אורך**  - ערכים חיוביים מציינים מקומות ממזרח לקו אורך 0 – העיירה גריניץ' באנגליה.

**נגדיר 2 נקודות**

Location metula = new Location(35.5782, 33.2772);

Location eilat = new Location(29.56000, 34.95111);

**חישוב המרחק**

double distance = Location.CalculateDistance(נקודת מוצא,

נקודת יעד,

יחידת מידה);

double kilometers = Location.CalculateDistance(

eilat,

metula,

DistanceUnits.Kilometers

);

**המחלקה Geocoding**

המחלקה Geocoding מספקת ממשקי API לקידוד גיאוגרפי של כתובת

קואורדינטות מיקום – Location ולהפוך קואורדינטות GeoCode -< Location לכתובת.

**קבלת מיקום עפ"י כתוובת**

try

{

var locations = await

Geocoding.GetLocationsAsync(etAddress.Text);

var location = locations?.FirstOrDefault();

if (location != null)

{

etLat.Text = location.Latitude.ToString();

etLong.Text = location.Longitude.ToString();

}

}

catch (FeatureNotSupportedException fnsEx)

{

// Feature not supported on device

}

catch (Exception ex)

{

// Handle exception that may have occurred in geocoding

}

**קבלת כתובת עפ"י מיקום**

try

{

var placemarks =

await Geocoding.GetPlacemarksAsync(

Convert.ToDouble(etLat.Text),

Convert.ToDouble(etLong.Text));

var placemark = placemarks?.FirstOrDefault();

if (placemark != null)

{

var geocodeAddress = placemark.Thoroughfare + " " +

placemark.SubThoroughfare + ", " +

placemark.Locality + ", " +

placemark.CountryName;

Toast.MakeText(this, geocodeAddress,

ToastLength.Long).Show();

}

}

catch (FeatureNotSupportedException fnsEx)

{

// Feature not supported on device

}

catch (Exception ex)

{

// Handle exception that may have occurred in geocoding

}

**המחלקה Map**

באמצעות המחלקה Map נוכל להציג מיקום על המפה, נוכל להציע אפשרויות ניווט ועוד.

המטודה שבאמצעותה נציג את המפה היר: OpenAsync.

**הצגת מיקום על המפה**

על מנת להציג מיקום על המפה יש ליצור מיקום (Location) ולהפעיל את המטודה OpenAsync עם המיקום.

var location = new Location(Convert.ToDouble(etLat.Text),

Convert.ToDouble(etLong.Text));

try

{

await Map.OpenAsync(location);

}

catch (Exception ex)

{

// No map application available to open

}

* נוכל להוסיף שם למיקום ע"י אובייקט מהטיפוס MapLaunchOptions

var location = new Location(Convert.ToDouble(etLat.Text),

Convert.ToDouble(etLong.Text));

var options = new MapLaunchOptions { Name = "Home sweet home" };

try

{

await Map.OpenAsync(location, options);

}

catch (Exception ex)

}

// No map application available to open

}

**ניווט**

על מנת לנווט ליעד מסוים יש לקבוע את היעד אליו רוצים לנווט (Location) ולפתוח את המפה עם אופן הניוון הרצוי.

אופני ניווט:

* רכב
* רגל
* אופניים
* תחבורה ציבורית

var location = new Location(Convert.ToDouble(etLat.Text),

Convert.ToDouble(etLong.Text));

var options = new MapLaunchOptions

{ **NavigationMode = NavigationMode.Driving** };

try

{

await Map.OpenAsync(location, options);

}

catch (Exception ex)

}

// No map application available to open

}