סדנת C# מספר 5 – NuGet חבילות חיצוניות ומה שביניהם

## רקע כללי

לא. לא ה nuggets האלה.



NuGet הוא כלי מבית היוצר של Microsoft אשר מנגיש שימוש במספר רב של חבילות חיצוניות ומאפשר לשלבם בפרויקט בצורה נוחה ופשוטה.

ישנם המון מפתחים מוכשרים בארץ ובעולם שכותבים חבילות וספריות בעצמם. בעבר תהליך השילוב של חבילות חיצוניות היה לוקח הרבה זמן ודורש תחזוקה רבה. Microsoft סיפקה לאותם מפתחים מוכשרים פלטפורמה בשםNuGet בה יוכלו להעלות את החבילות שלהם לאינטרנט כך שמפתחים אחרים יוכלו ליהנות מהם גם כן.

במילים אחרות: NuGet זה מנהל החבילות של NET. שם ניתן ליצור ולהעלות את החבילות שלך ולהוריד ולצרף לפרויקט שלך חבילות של אחרים.

## על מה נדבר בסדנא הזאת

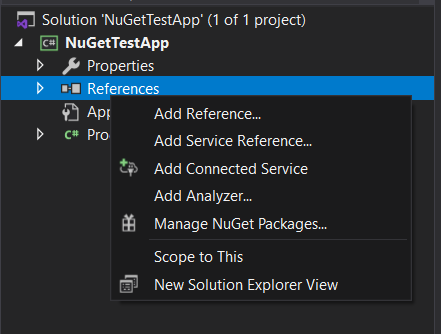
בסדנא זו נדבר על איך משתמשים בחבילות חיצוניות של NuGet, איך לשלב אותם בצורה קלה בפרויקט שלכם, ואיך אתם חוסכים לעצמכם הרבה זמן של פיתוח ע"י חיפוש נכון של ספריות מוכנות. בנוסף נציג דוגמאות שימוש בחבילות מסוימות אשר עשויות להועיל לכם בשאר הפרויקטים בקורס.

בואו נתחיל.

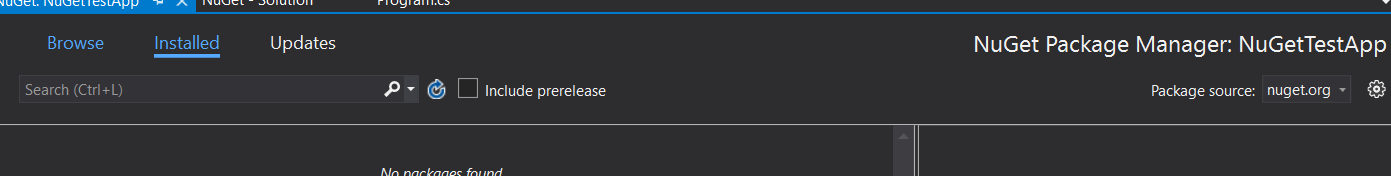
יש תיעוד מעולה ומפורט באתר הרשמי של Microsoft: [NuGet Docs](https://docs.microsoft.com/en-us/nuget/). אנא השקיעו זמן על מנת לעבור על החלקים השונים שלו, ולהתעמק בהסברים שמעניינים אתכם. תנסו להגיע למצב שלאחר שעברתם על התיעוד, אתם מבינים את המטרה של הכלי ואת אופן השימוש הבסיסי בו.

## הוספת חבילה NuGet לפרויקט

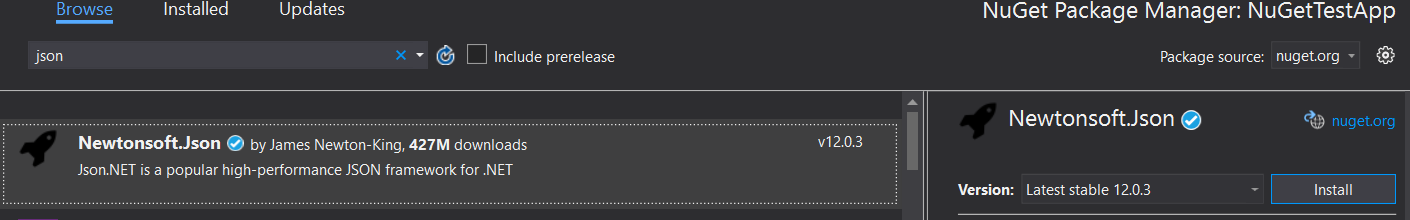
פתחו פרויקט חדש בשם לבחירתכם מסוג ConsoleApplication. בתוך ה solution explorer, לחצו לחצן ימני על References מתחת לשם של הפרויקט, ולאחר מכן לחצו על “Manage NuGet Packages”. ייפתח לכם ה GUI שאיתו תוכלו לנהל חבילות, ולהתקין חבילות חדשות



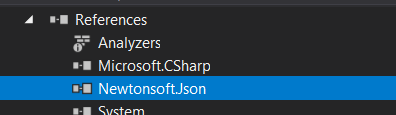
במסך הזה, אתם יכולים לראות את כל החבילות החיצוניות שכבר התקנתם, כרגע הוא ריק. עברו ל Tab שנקרא Browse ובו תוכלו לחפש חבילות ולהתקין אותם.



לאחר שבחרתם חבילה, לחצו על Install על מנת להתקין אותה, ועקבו אחר ההוראות



לאחר שהתקנתם את החבילה, בחלק שמכיל את כל ה references של הפרויקט תוכלו לראות שנוספה התלות שזה עתה התקנתם. כעת ה namespace שמכיל את החבילה החדשה התווסף לתוכנית ואתם יכולים להשתמש בו.

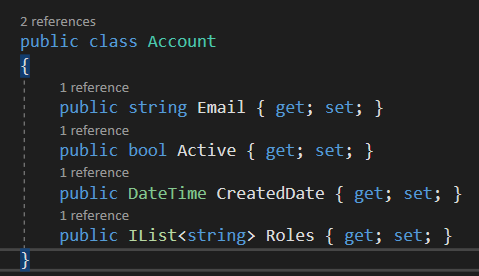


## שימוש בחבילת JSON כחבילה חיצונית

הדוגמא הקודמת הדגימה כיצד מתקינים חבילה גנרית. כעת עליכם להוסיף לפרויקט שפתחתם את החבילה שנקראת: *Newtonsoft.Json.* לרוב החבילות יש תיעוד באתר הרשמי של [NuGet](https://www.nuget.org/packages/Newtonsoft.Json/). בנוסף תיעוד על השימוש בחבילה תוכלו למצוא פה: [Newtonsoft.Json](https://www.newtonsoft.com/json). דוגמאות לשימוש: [JSON Docs](https://www.newtonsoft.com/json/help/html/Introduction.htm)

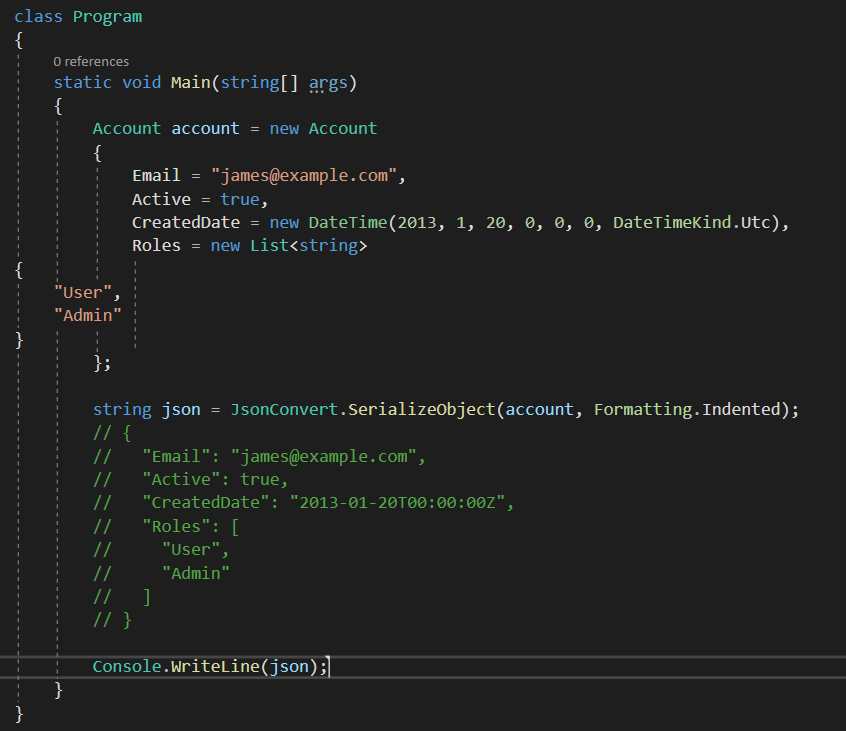
נבנה תוכנית לדוגמא אשר עושה Serialization לאובייקט Account.

נגדיר אובייקט Account שמכיל את השדות הבאים:



נבנה תוכנית Main שתשתמש בספרייה שזה עתה הוספנו, ובאובייקט Account שיצרנו. המטרה של התוכנית היא ליצור אובייקט מסוג Account, לאתחל את השדות שלו, ולאחר מכן להציג את האובייקט בצורה טקסטואלית.

Serialization is a mechanism of converting the state of an object into a byte stream



חשבו על אם הייתם נדרשים להדפיס את האובייקט account, סביר להניח שהייתם מדפיסים כל אחד מהשדות בנפרד. התוצאה לא רחוקה, מזה. הפלט נראה מעין הדפסה של מילון. כאשר המפתחות הם השמות של השדות, והערכים הם השדות עצמם. לדוגמא Email: [mymail@gmail.com](mailto:mymail@gmail.com). המפתח הוא string וכך גם הערך של השדה. אבל מה קורה עם טיפוסים מורכבים? לדוגמא Roles, הסוג של השדה הוא רשימה. אין נדפיס את זה? בדיוק, נדפיס פשוט רשימה.

הפלט של התוכנית ייראה כך:

A screen shot of a social media post

Description automatically generated

בואו נעשה את אותו דבר אבל הפוך. נשתמש ב Deserialization על מנת לקרוא אובייקט מתוך טקסט.

Deserialization is the reverse process where the byte stream is used to recreate the actual C# object in memory

ניצור אובייקט מסוג string ונרשום אליו את האובייקט Account. לאחר מכן נטען אותו לתוך משתנה מסוג Account.



הפורמט אליו אנחנו עושים serialization, או שממנו אנחנו עושים deserialization נקראת JSON. מדובר בפורמט טקסטואלי, אשר ניתן לשמור לקובץ, וכך ניתן בצורה פשוט להעביר אובייקטים בינאריים בין אנשים או בין שני אפליקציות שונות. האפליקציות חייבות להכיר את הפורמט של האובייקט ואת הדרך שבה "קידדו" אותו, על מנת שיוכל לעשות לו deserialization לאובייקט.

תוכלו למצוא עוד דוגמאות רבות, לשימוש בספרייה הזאת. חשוב שתשימו לב לפשטות ולקלות של השימוש בספריות חיצוניות, ולדרך הנכונה שניתן לשלבם בפרויקט שלכם, על מנת לחסוך שעות רבות של כתיבת קוד.

## סיכום

לסיכום, ראינו מה המוטיבציה להשתמש בספריות חיצוניות, איך ניתן לעשות זאת בצורה קלה ופשוטה. ואיך זה יכול לחסוך לנו הרבה שעות פיתוח.

בנוסף לכך ראינו דוגמא לשימוש בספרייה חיצונית פופולארית, שתעזור לכם בהמשך תרגיל הטריוויה. ישנם עוד דוגמאות קוד רבות לשימוש בספרייה, מוזמנים לעבור על הלינקים לאורך המסמך.

כל הקוד שמוצג פה, מצורף לסדנא בתיקיית Examples.

