

# In-memory IPC

.NET

- כללי:
- יש לכתוב שני פרויקטים ב**.Net**. שיעבירו אובייקטים אחד לשני דרך זיכרון משותף
- אחד התהליכים הינו לקוח שמבצע "הזמנה" והשני הינו שרת שמאשר את ההזמנה
- חשוב:
- פרויקט אמור להיות **Solution** עם שני פרויקטים – ולא שני **Solutions** נפרדים
- בשום פנים ואופן סוג הפרויקטים **לא** יכול להיות .Net Framework.

טיפוס של "הזמנה"

```
internal record Order(  
    decimal MoneyAmount,  
    string Currency,  
    string Product,  
    int ProductAmount  
);
```

טיפוס של תשובה מהשרת

```
internal record Response(  
    bool Accepted,  
    DateTime EstimatedSupplyDate  
);
```

# דיאגרמה



לא משנה מה התהליך  
שמתחיל ראשון זה לא  
אמור לשבור את  
התוכנית

Client

Starts

Creates MemoryMappedFile

Serializes an object of type Order  
and writes to MMF as length-prefixed  
JSON string

Waits for **Server** to read the order  
and write back a response

Ends

Server

Starts

Waits for **Client** to create  
MemoryMappedFile

Waits for **Client** to place an order  
inside MemoryMappedFile

Reads and deserializes the order,  
writes back a response as length-  
prefixed JSON

Ends

חובה ליישם זאת  
באמצעות  
**EventWaitHandle**

# תיאור המשימה - לקוח

## לקוח:

- יוצר זיכרון משותף עם קיבולת 4096 בתים (4KiB) - תשתמשו ב [MemoryMappedFile.CreateNew\(String, Int64\)](#)
- יוצר אובייקט **Order** חדש - לא משנה עם איזה ערכים
- מבצע סריאליזציה של האובייקט ל **JSON-string** - [JsonSerializer.Serialize\(Tvalue\)](#)
- מחשב אורך המחזורות ובונה מערך בתים שבו ארבעה (4) בתים הראשונים - זה אורך המחזורות (**json**) ושאר הבתים הינם בתים של המחזורות. שימו לב:
- השתמשו ב [BitConverter.GetBytes\(Int32\)](#) בכדי להמיר אורך המחזורות למערך בתים
- השתמשו בקידוד אחיד בשני הצדדים גם בשרת וגם בלקוח: [Encoding.UTF8.GetBytes\(String\)](#)
- רושם את הבתים לתוך הזיכרון המשותף. השתמשו ב [MemoryMappedFile.CreateViewStream\(Int64, Int64\)](#)
- אחרי שליחת "ההזמנה" לשרת - לקוח מחכה לאיתות דרך [EventWaitHandle](#) על כך שהוא יכול להמשיך
- לאחר קבלת "אות להמשך" מהשרת לקוח קורא אובייקט **Response** מהזיכרון המשותף ומדפיס על המסך
- השתמשו ב [JsonSerializer.Deserialize<TValue>\(ReadOnlySpan<Char>, JsonSerializerOptions\)](#) כדי להמיר את תוכן הזיכרון המשותף לאובייקט

כך נראית ההדפסה של הלקוח:

```
Client send order: Order { MoneyAmount = 100, Currency = ILS, Product = Window, ProductAmount = 10 }  
Client received response: Response { Accepted = True, EstimatedSupplyDate = 18/02/2025 20:04:51 }
```

# תיאור המשימה - שרת

## שרת:

- ממתין לאות מהלקוח על כך שהוא יכול להמשיך
- פותח זיכרון משותף - השתמשו ב [MemoryMappedFile.OpenExisting\(String\)](#)
- קורא אובייקט **Order** מהזיכרון המשותף ומדפיס על המסך לאחר דסריאליזציה (**deserialization**)
- מכין "תשובה" - אובייקט מסוג **Response** - באותה דרך כמו הלקוח - **Length-Prefixed Json**
- תשובה של השרת בנויה תמיד באופן הבא:

```
{  
  "Accepted": true,  
  "EstimatedSupplyDate": `בעוד יום וחצי מרגע קבלת ההזמנה`  
}
```

- רושם תשובה לתוך הזיכרון המשותף
- מאותת ללקוח שהוא יכול להמשיך

כך נראית ההדפסה של השרת:

```
Server received order: Order { MoneyAmount = 100, Currency = ILS, Product = Window, ProductAmount = 10 }  
Server send response: Response { Accepted = True, EstimatedSupplyDate = 18/02/2025 20:04:51 }
```

יש שגיאה במצגת או עמימות כלשהיא?  
לא ברור למה התכוונתי?  
במקרה כזה נא לשלוח מכתב ל [vladimird@tcb.ac.il](mailto:vladimird@tcb.ac.il)