

# .NET MICROSERVICES

6. forduló



A kategória támogatója: DXC Technology

## Ismertető a feladatlaphoz

Kezdj neki minél hamarabb, mert a feladatot a forduló záró időpontjáig lehet beküldeni, nem addig lehet elkezdni!

Sok sikert!



## 1. feladat 2 pont

Melyik .NET típus felelős az alapértelmezett értékek megadásáért, amikor nem kapunk értéket egy konstruktor paraméterként?

### Válasz

- ☐ DefaultValueAttribute
- ☐ NullableType
- ☐ DefaultType
- ☐ DefaultAttributeValue
- ☐ A fentiek közül egyik sem.

## 2. feladat 1 pont

Melyik .NET típus használatos kizárólag egy időpont tárolására, anélkül, hogy figyelembe venné a hozzá tartozó dátumot?

### Válasz

- ☐ DateTime
- ☐ TimeOnly
- ☐ InstantTime
- ☐ TimeStamp
- ☐ A fentiek közül egyik sem.

## 3. feladat 0 pont

Válaszd ki a helyes állítást a .NET Core microservice-ekre és Domain Driven Design-re vonatkozóan!

### Válasz

- ☐ A domain réteg tartalmazza az alkalmazás összes infrastruktúra vonatkozásait, például message bus vagy adatbázis hozzáférések integrációját.
- ☐ Egy aggregát a DDD-ben olyan entitások kollekciója, amelyek függetlenek és elkülöníthetők egymástól.
- ☐ A domain event-eket a DDD-ben elsősorban a felhasználói felügyeleti interakciók és prezentációs logika kezelését használjuk.
- ☐ A Value Object-ek a DDD-ben immutable-ök és összehasonlításuknál referencia szerinti összehasonlítást kell alkalmazni, nem tartalmuk szerintit.
- ☐ .NET Core tartalmaz predefined DDD struktúrákat, melyeket célszerű is alkalmazni.
- ☐ A fentiek közül egyik sem.

## 4. feladat 3 pont

Épp egy microservice-t fejlesztesz a gépeden, majd teszteled a tesztelésre szánt kubernetes klaszteren. Tesztelés során azt tapasztalod, hogy DateTime.Now esetén nem stimmel az időzóna. Jelöld meg a helyes állításokat!

## Válaszok

- ☐ Az időzóna fordítási időben eldől.
- ☐ A konténer a kubernetes pod rendszer idejét használja
- ☐ A kubernetes node egy harmadik időzóna beállítást is használhat
- ☐ Az időzóna operációs rendszer szintű beállítás

## 5. feladat 0 pont

Válaszd ki a helyes REST API-ra vonatkozó állításokat.

## Válaszok

- ☐ HTTP GET idempotens
- ☐ HTTP PUT idempotens
- ☐ Idempotens végpontok válaszai cache-elhetőek
- ☐ Az idempotens HTTP metódusokat automatikusan megismételhetjük kommunikációs hiba esetén

## 6. feladat 6 pont

Adott ez a Background service .NET Core alkalmazásunkban:

```
public class MyBackgroundService : BackgroundService
{
    private readonly ILogger<MyBackgroundService> _logger;

    public MyBackgroundService(ILogger<MyBackgroundService> logger)
    {
        _logger = logger;
    }

    protected override async Task ExecuteAsync(CancellationToken stoppingToken)
    {
        while (!stoppingToken.IsCancellationRequested)
        {
            _logger.LogInformation("Háttérszolgáltatás fut...");
            await Task.Delay(5000, stoppingToken);
        }
    }
}
```

```

    }
}

public static async Task Main(string[] args)
{
    var hostBuilder = CreateHostBuilder(args).Build();

    using var serviceScope = hostBuilder.Services.CreateScope();
    var services = serviceScope.ServiceProvider;

    var myBackgroundService = services.GetRequiredService<MyBackgroundService>();
    // helyes kódrész

    await hostBuilder.RunAsync();
}

```

Az alábbiak közül melyiknek kell kerülnie a helyes kódrész helyére a kódban, hogy a szervízünk elinduljon hiba nélkül?

## Válasz

☐

`await myBackgroundService.StartAsync();`

☐

`await myBackgroundService.StartAsync(CancellationToken.None);`

☐

`await myBackgroundService.RunAsync(CancellationToken.None);`

☐

`await myBackgroundService.Run();`

## 7. feladat 4 pont

Egy naptáras alkalmazást fejlesztesz, amely lehetővé teszi a felhasználók számára az események és találkozók rögzítését. Az eseményekhez dátumokat és időpontokat is tárolsz. Az alábbi kód bemutat egy példát az esemény osztályra:

```
public class Event
{
    public string Name { get; set; }
    public DateTime StartDate { get; set; }
    public DateTime EndDate { get; set; }
}
```

Az alkalmazásban az események kezdeti és végdátumai lehetnek különböző időzónákban. A felhasználók az alkalmazásban előzetesen kiválaszthatják az időzónájukat, amelyet figyelembe kell venni a dátumok megjelenítésekor.

Melyik kódrészlet mutatja be helyesen, hogyan lehet a helyi időzóna alapján megjeleníteni egy esemény kezdeti dátumát és időpontját?

## Válasz

☐

```
string localStartDate = event.StartDate.ToString();
```

☐

```
TimeZoneInfo userTimeZone = GetUserTimeZone();
DateTime localStartDate = TimeZoneInfo.ConvertTimeFromUtc(event.StartDate, userTimeZone);
```

☐

```
DateTime localStartDate = event.StartDate.ToLocalTime();
```

☐

```
DateTime localStartDate = TimeZoneInfo.ConvertTime(event.StartDate, TimeZoneInfo.Local);
```

## 8. feladat 2 pont

Melyik kódrészletnek nagyobb a memória felhasználása?

## Válasz

☐

```
public class EmployeeClass
{
    public int Id;
    public DateTime Birthday;
```

```
        public ushort AreaId;
    }
}
```

```
public struct EmployeeStruct
{
    public int Id;
    public DateTime Birthday;
    public ushort AreaId;
}
```

## 9. feladat 1 pont

Hívható-e újabb command egy command handlerből CQRS pattern esetén?

### Válasz

- ☐ Nem, mert minden commandnak saját felelősségi területe van illetve a tranzakciók kezelése nem oldható meg.
- ☐ Igen, mivel a commandok végrehajtása aszinkron, fire-and-forget folyamat így lényegtelen, hogy egymásba ágyazottak-e.
- ☐ Igen, de minden esetben meg kell fontolni a felelősség, konzisztencia és teljesítmény szempontokat.

Megoldások beküldése