

.NET MICROSERVICES

5. 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 3 pont

Melyik állítás igaz a TimeZoneInfo osztály használatával kapcsolatban a .NET keretrendszerben?

Válasz

- ☐ A TimeZoneInfo osztály csak a Windows időzónák kezelésére használható, és nem támogat más platformokat.
- ☐ A TimeZoneInfo.ConvertTimeToUtc metódus segítségével átalakíthatjuk egy adott időpontot (DateTime) UTC időzónába.
- ☐ A TimeZoneInfo osztály csak a helyi időzónával kapcsolatos információkat képes tárolni.
- ☐ A TimeZoneInfo.FindSystemTimeZoneById metódus segítségével nem találhatunk időzóna definíciót a rendszerben előre definiált időzónák között.

2. feladat 3 pont

Mi a fő elv a Event Sourcing tervezési mintában?

Válasz

- ☐ Az alkalmazás adatbázisában csak az aktuális állapotot kell tárolni.
- ☐ Az alkalmazás állapotát a történelmi események sorozataként tároljuk.
- ☐ Az adatok közvetlenül az alkalmazás memóriájában tárolódnak.
- ☐ Az adatokat egyetlen centralizált adatbázisban kell tárolni.

3. feladat 3 pont

Válaszd ki a helyes állításokat a .NET Core microservice-ekre vonatkozóan!

Válaszok

- ☐ A health checks kizárólag az egyes microservice példányok teljesítményének monitorozására szolgál.
- ☐ A beépített health checks kizárólag HTTP alapú szolgáltatásokra vonatkozik, nem használható más protokollokkal.
- ☐ A health checks ellenőrzéseket általában külső monitorozási eszközök végzik és nincs hatással magukra a microservice szolgáltatásokra.
- ☐ A health checks-ek segítenek előre felismerni a problémákat, ezzel segítve az orkesztrátor működését az általános rendszer megbízhatóságához.
- ☐ A fentiek közül egyik sem.

4. feladat 3 pont

Jelöld meg a helyes állításokat.

Válaszok

- ☐ A REST kommunikáció stateless
- ☐ gRPC esetén is érdemes követni a REST szabályait
- ☐ Az OpenAPI kizárja a REST API használatát
- ☐ HTTP OPTIONS egy listával tér vissza a lehetséges HTTP műveletekről az adott végpontra nézve

5. feladat 4 pont

Adott ez a két entitás:

```
public class Slot1
{
    public string Name { get; }
    public int Threshold { get; }
    public int Position { get; }
    public IReadOnlyList<string> Materials { get; }

    public Slot1(string name, int threshold, int position, IReadOnlyList<string> materials)
    {
        Name = name;
        Threshold = threshold;
        Position = position;
        Materials = materials;
    }
}
```

```
public class Slot2
{
    public string Name { get; }
    public int ThresholdQty { get; }
    public int Position { get; }
    public IReadOnlyList<string> Materials { get; }

    public Slot2(string name, int threshold, int position, IReadOnlyList<string> materials)
```

```
{
    Name = name;
    ThresholdQty = threshold;
    Position = position;
    Materials = materials;
}
```

Automappert használunk, ebben a modulban definiáltunk mappeléseket.

```
public class ModelMapper : Profile
{
    public ModelMapper()
    {
        // Ide jön a kódrészlet
    }
}
```

Az alábbi négy közül melyik a helyes mappelés?

Válasz

☐

CreateMap<Slot1, Slot2>();

☐

Egyik sem helyes

6. feladat 3 pont

Mi a fő különbség a CQS és a CQRS között?

Válasz

☐

A CQRS az adatállapotot két különböző modellben tárolja, míg a CQS nem.

☐

A CQS csak parancsokat használ, a CQRS csak lekérdezéseket.

☐

A CQS az adatot lekérdezési és módosítási modelleken keresztül választja szét, míg a CQRS az MVC tervezési mintát alkalmazza.

☐

A CQRS csak mikroszolgáltatás-architektúrákban használható, a CQS pedig csak monolitikus rendszerekhez.

Megoldások beküldése