

Microservice project

Feladat célja

- Két különálló .NET microservice:
 - a. **Product Service** – termékek kezelése (CRUD).
 - b. **Order Service** – rendelések kezelése, és termékadatok lekérése a Product Service-től.
- Kommunikáció: **REST API hívások** (Order Service → Product Service).
- Adatbázis: minden szolgáltatás saját adatbázissal rendelkezik (pl. PostgreSQL).

Architektúra áttekintés

• Product Service:

- Felelős a termékek kezeléséért.
- Adatbázis: **ProductsDB** (PostgreSQL).

• Order Service:

- Felelős a rendelések kezeléséért.
- Adatbázis: **OrdersDB** (PostgreSQL).
- Ha új rendelés érkezik, lekéri a termék adatait a Product Service-től (pl. ár, név).

API definíciók

Product Service API

Base URL: /api/products

• GET /api/products

- Leírás: Visszaadja az összes terméket.
- Válasz:

```
1 [  
2 {  
3   "id": 101,  
4   "name": "Keyboard",  
5   "price": 25.99,  
6   "stock": 100  
7 }  
8 ]
```

• GET /api/products/{id}

- Leírás: Visszaadja az adott termék adatait.
- Válasz:

```
1 {  
2   "id": 101,  
3   "name": "Keyboard",  
4   "price": 25.99,  
5   "stock": 100  
6 }  
7  
8 ]
```

• POST /api/products

- Leírás: Új termék létrehozása.
- Request:

```
1 {  
2   "name": "Mouse",  
3   "price": 15.49,  
4   "stock": 200  
5 }
```

```
6 }  
7
```

- **DELETE /api/products/{id}**

- Leírás: töröljön egy terméket

Order Service API

Base URL: /api/orders

- **GET /api/orders**

- Leírás: Visszaadja az összes rendelést.

- Válasz:

```
1 [  
2   {  
3     "id": 5001,  
4     "totalPrice": 51.98,  
5     "products":  
6       [  
7         {  
8           "productId": 101,  
9           "quantity": 2  
10        }  
11      ]  
12    }  
13 ]
```

- **POST /api/orders**

- Leírás: Új rendelés létrehozása.

Az Order Service lekéri a termék árát a Product Service-től, majd kiszámolja a `totalPrice`-t.

- Request:

```
1 [  
2   {  
3     "productId": 101,  
4     "quantity": 2  
5   }  
6 ]
```

- Válasz:

```
1 {  
2   "  
3     "id": 5001,  
4     "totalPrice": 51.98,  
5     "products":  
6       [  
7         {  
8           "productId": 101,  
9           "quantity": 2  
10        }  
11      ]  
12    }  
13 }
```

Adatbázis sémák

ProductsDB (Product Service)

```
1 CREATE TABLE Products (  
2   Id SERIAL PRIMARY KEY,  
3   Name VARCHAR(100) NOT NULL,  
4   Price DECIMAL(10,2) NOT NULL,  
5   Stock INT NOT NULL  
6 );
```

OrdersDB (Order Service)

```
1 CREATE TABLE Orders (  
2   Id SERIAL PRIMARY KEY,  
3   ProductId INT NOT NULL,  
4   Quantity INT NOT NULL,
```

```
5     TotalPrice DECIMAL(10,2) NOT NULL,  
6     CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
7 );
```

Kommunikációs logika

- Order Service → Product Service:

- HTTP GET `/api/products/{id}` hívás a termék adatainak lekéréséhez.
- Ha a termék nem található, hibát dob.
- Ha található, kiszámolja a `totalPrice = price * quantity`, majd menti az OrdersDB-be.
- Ha 3s alatt nem válaszol a product service akkor timeoutoljon az api
- Circuit breaker és retry logika implementálása szükséges
- Extra: caching logika implementálása a két microservice között

Acceptance Criteria

Általános követelmények

- Mindkét microservice különálló .NET alkalmazásként fut.
- Mindkét szolgáltatás saját adatbázissal rendelkezik (PostgreSQL).
- Unit test lefedettség 90%+
- 0 Code smell
- API-k dokumentálva vannak (Swagger/OpenAPI).
- Kommunikáció REST API-n keresztül történik (Order Service → Product Service).
- A szolgáltatások Docker konténerekben futnak, és Kubernetesben deployolhatók. (sprint 3)
- API válaszidő < 500ms normál terhelés mellett.