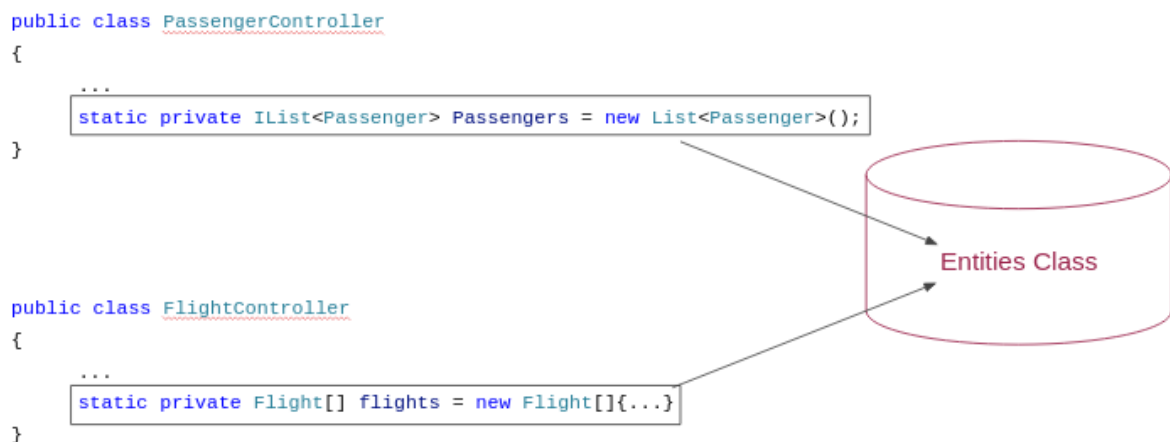# Day 15 - Angular & Asp in 21 days Handouts

⊘ Status

Day 15: Singleton Service Dependency

To store data in memory, do not store your entity collections in your controllers. You can create a class that acts as a container for all of the entity collections in your project.

```
public class Entities
{
    public static List<Passenger> Passengers = new List<Passenger>();
    public static List<Flight> Flights = new List<Flight>();
}
```

You don't need to specify the entity collections as static fields.

A better solution is to keep a single instance of the Entities class in the application and pass it to all clients (Controllers) so that all share the same database.

To share the Entities class you can create a static instance of it, but a more robust alternative is to inject it into the service collection as a singleton.

```csharp
builder.Services.AddSingleton<Entities>();
```

```csharp
public class FlightController
{
    public FlightController(Entities entities)
    {
        this.entities = entities;
    }
    ...
}
```

```csharp
public class PassengerController
{
    public PassengerController(Entities entities)
    {
        this.entities = entities;
    }
    ...
}
```