

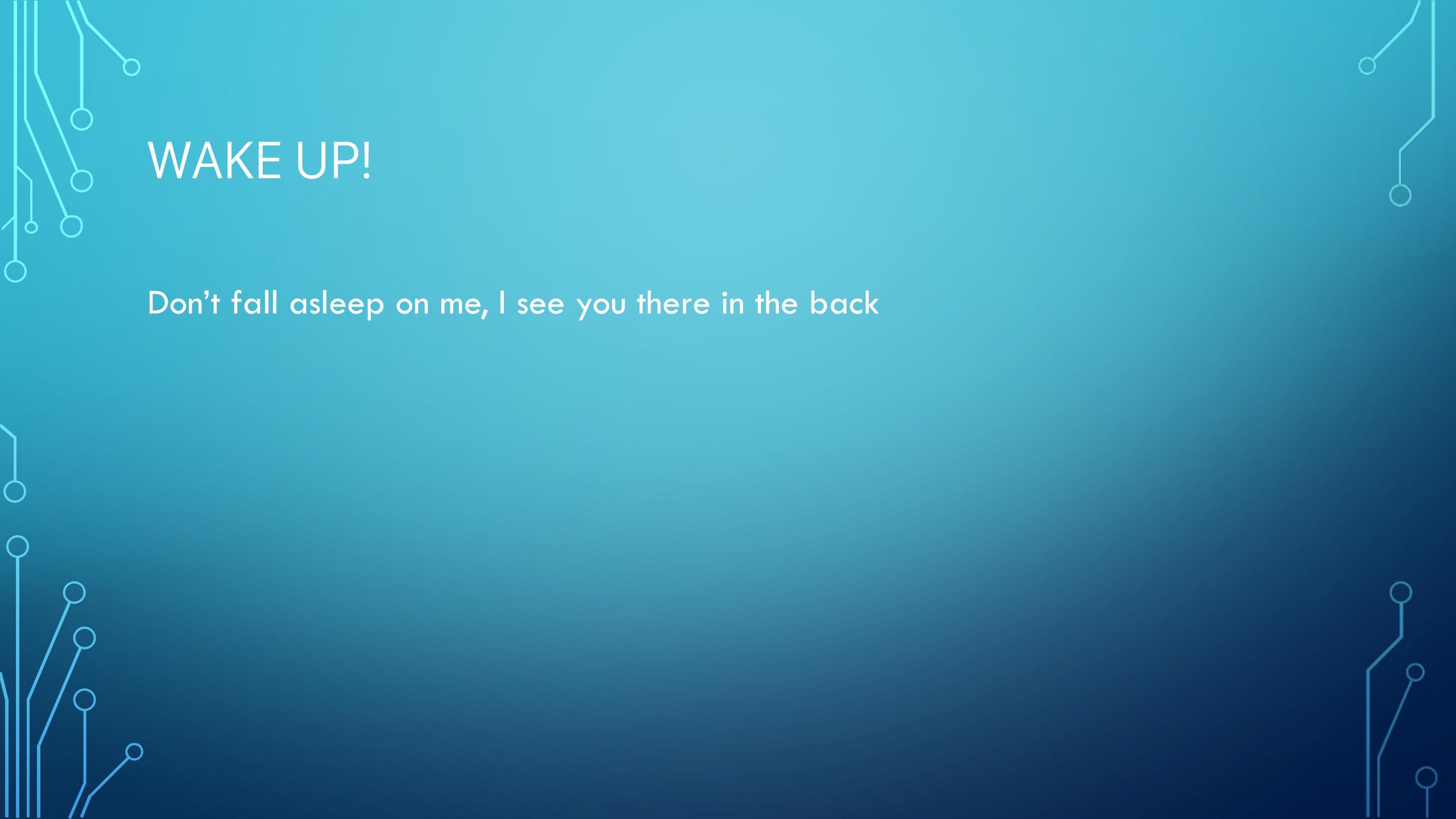
10 5 THINGS I HATE/LOVE ABOUT ENTITY CORE

(maybe less)

AND WHY THAT'S OKAY

Who am I? What am I?

- My name is Daniel Ashcraft
 - I typically work in full stack Javascript
 - I also work/play in ASP.NET Core (which means Entity Core)
 - I have been developing websites/applications for a little over 3 years
 - I am an admin for the Free Code Camp Usergroup here in OKC.
 - I am lazy i.e. I am your average programmer
 - Twitter, medium (@d_ofashandfire)
 - Github (dashcraft)



WAKE UP!

Don't fall asleep on me, I see you there in the back

Entity Core, keep it simple

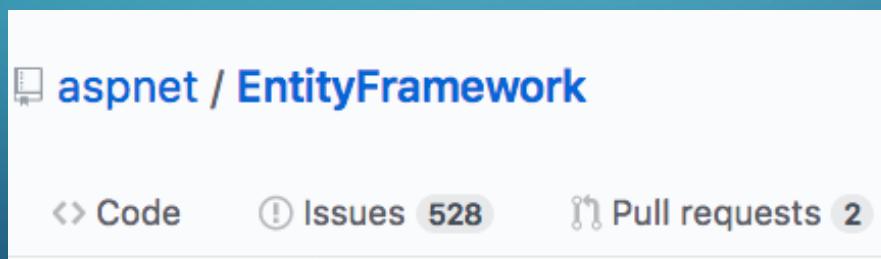
- Entity Framework is Microsoft's Object Relational Mapper that allows you to work with a database using .NET objects. In Visual Studio it has things like scaffolding, migrations and many-to-many relational mapping.
- Entity Core is a lightweight version of Entity Framework, meant to be cross platform and new/improved. Pros and Cons to follow...

The Pros

1. Entity Core allows me to write models and generate tables, using C#, not touching the DB (even though I can, don't look down your noses at me yet purists).
2. With Entity Core, I can manage migrations fairly easily with `dotnet migrations`, a command line tool used when using VC (visual code).
3. Auto-magical code generally doesn't suck (Relationships, migrations).
4. It doesn't try to do “too much”, i.e. scaffolding in EF has never made me happy.
5. Plays nicely with VC, which is a big deal to me.

The Cons

1. Sloooooooooooooow, if you use EC for groupby or other aggregate functions. (like really slow, go google it)
2. .Net Objects not consistent (`IEnumerable`, `List`, `IQueryable`, pick one microsoft)
i.e. `.ToList()` vs `.Select()`
3. Seriously microsoft??? Maybe only part time maintainers?



Wrap Up

All in all, I don't hate it too much. It does what I need it to do, when I explicitly tell it to do that thing. I can also just write raw sql queries in EC, when I need to. If you want something that doesn't try to think or make decisions for you, Entity Core may be a good direction to go. If performance is a major concern for you, avoid using most (if not all) of the prebuilt aggregate methods.

RESOURCES

- <https://docs.microsoft.com/en-us/ef/core/index>
- <https://github.com/aspnet/EntityFramework>