

# AutoMapper a jak se z něj nezbláznit

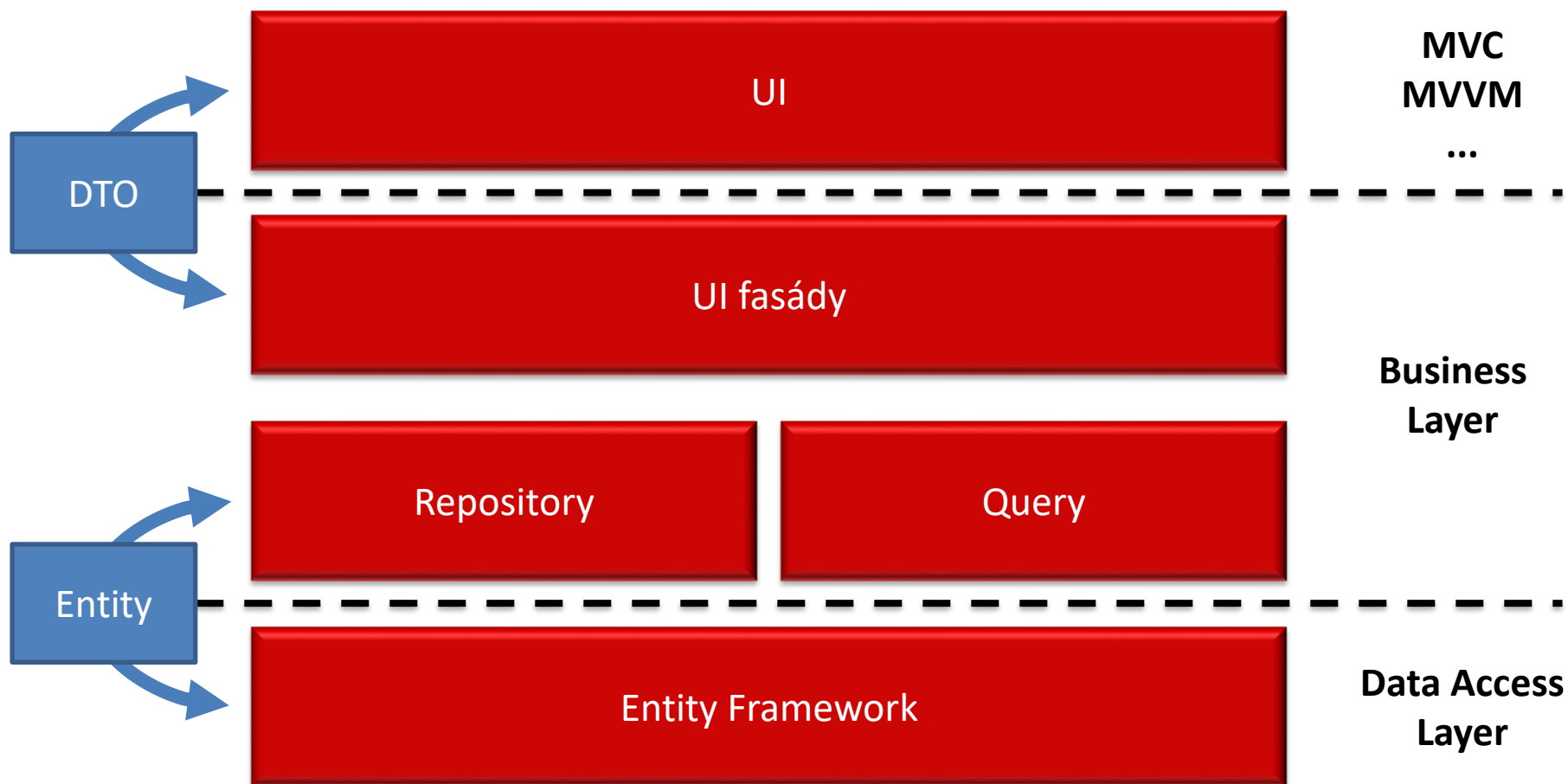
Tomáš Herceg

*Microsoft Most Valuable Professional (ASP.NET)*

*Microsoft Regional Director*

[tomas.herceg@riganti.cz](mailto:tomas.herceg@riganti.cz)

# K čemu mapování



# AutoMapper

- Knihovna pro .NET
  - <https://github.com/AutoMapper/AutoMapper>
- Mapování objektu na jiný objekt
- Jmenné konvence
- Konfigurovatelná pravidla
- Podpora **IQueryable**

# Jak to funguje?

- `Mapper.CreateMap<TSource, TDest>`
- Vestavěné konvence
  - **TSource** a **TDest** má vlastnost stejného jména
  - Flattening (automatické zploštění typů)
    - `Order.Name` → `OrderName`
    - `OrderItems.Count` → `OrderItemsCount`

# DEMO

Základy mapování

# Konfigurace

- `Mapper.CreateMap<TSource, TDest>();`
- `.ForMember(o => o.Property, ...)`
  - Ignorování vlastnosti
    - `m => m.Ignore();`
  - Výraz pro mapování
    - `m => m.MapFrom(x => ...)`
  - Podmíněné mapování
    - `m => m.Condition(x => ...)`

# DEMO

Ukázka mapování

# Mapování

- Vytvoření cílového objektu
  - `Mapper.Map<TDest>(source)`
- Naplnění existujícího objektu (populate)
  - `Mapper.Map(source, dest)`
- Použití v dotazech
  - `IQueryable<TSource> query = ...;`  
`query.ProjectTo<TDest>`



# DEMO

Synchronizace kolekcí

# Na co si dát pozor

- Rekurzivní mapování
  - Funguje, ale komplikuje dotazy v EF
  - U mapování jedné instance možný lazy loading
  - Rozumné maximálně jedno zanoření
- Reusování DTO pro více situací
  - Typicky **UserDTO**...
  - Mnoho vlastností, které nejsou potřeba vždy
  - Lepší jsou cílená DTO (i když je to pracnější)

# Na co si dát pozor

- Všechna pravidla v jednom souboru
  - Rozdělení mapovacích pravidel do skupin
  - Možné využít IoC/DI kontejner
- Zapomenuté vlastnosti
  - `Mapper.AssertConfigurationIsValid()`

# DEMO

Pořádek v mapováních

