

# Why ViewModels Matter

---



**Shawn Wildermuth**

MICROSOFT MVP, SPEAKER AND INSTRUCTOR

@shawnwildermuth <http://wilder minds.com>



# Overview



## Why ViewModels Matter

- What are ViewModels
- Entities or ViewModels
- Why Mapping
- Using ViewModels in Views
- Using ViewModels in Forms



# What are ViewModels

---



define: ViewModel

A use-case specific structure of data for binding to physical views, logical views, or APIs.

\* Also Known As: Data Transfer Object (DTO), Data Model, and Data View





In ASP.NET Core — Any data that is passed is considered a model in ModelBinding — Even if it's your Entities



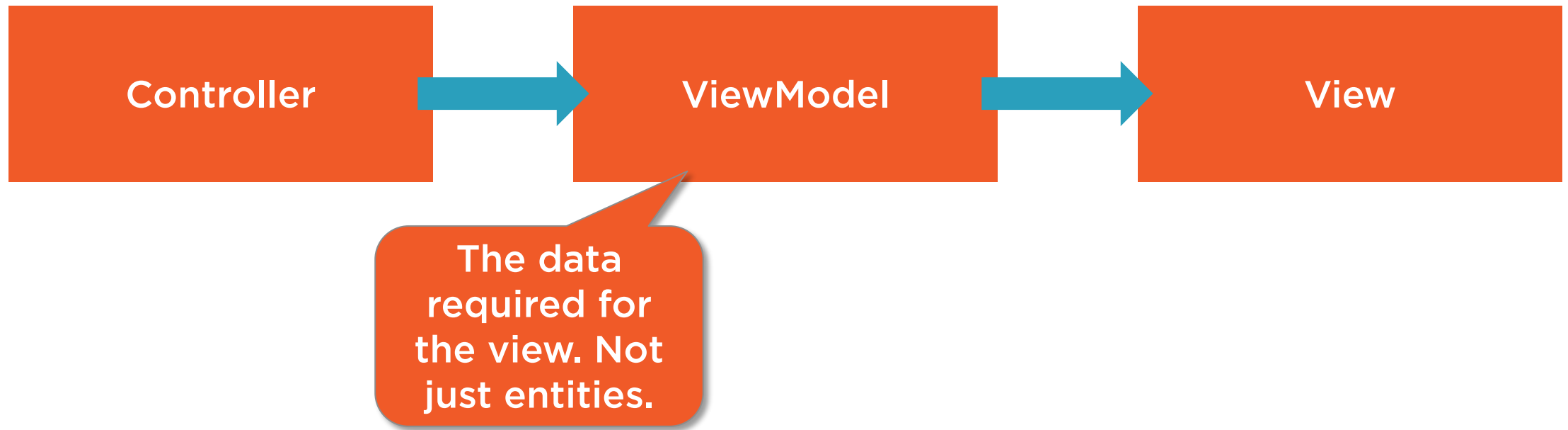


## Why ViewModels?

- Used for customized use-cases
- Used for limiting sensitive data
- Used for streamlining binding
- Used for more complex validation



# The “M” in MVC





## Pragmatism

- ViewModels aren't always required
- Preemptive creation is an anti-pattern
- Don't need to look like your entities
- They can add unnecessary complexity
- Can Loosely type if schedule requires
  - But risks must be weighed



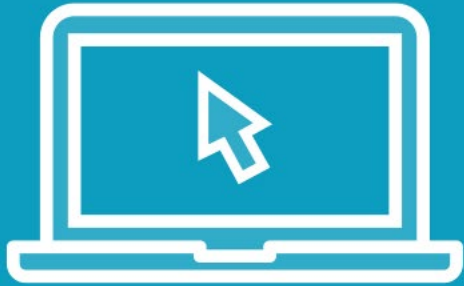
# Demo



## First ViewModel



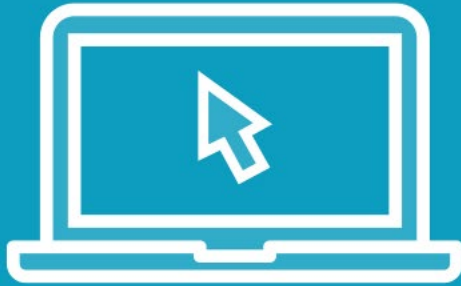
# Demo



## Introducing Mapping



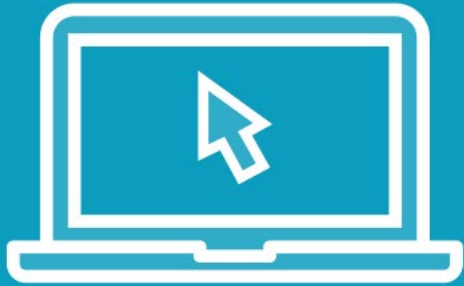
# Demo



## Complex Mapping Scenarios



# Demo



## Mapping Collections



# Demo



## Using ViewModels in Forms



# What We've Learned



ViewModels can allow us to simplify the code in our views and forms



Mapping can simplify using ViewModels in our code



Mapping is a commodity – AutoMapper isn't the only solution



# Coming Up Next: Server-side Validation

