

# In Memory DataBase

É um provider do Entity Framework instalado através de um pacote de extensão que permite a utilização do armazenamento em memória ao invés do banco relacional.

- Objetivo de uso para testes da aplicação e do ORM
- É necessário instalar o pacote *Microsoft.EntityFrameworkCore.InMemory*
- Na configuração do contexto o uso do banco relacional deve ser removido
- Facilita a execução de testes sejam automatizados ou manuais
- A carga inicial do Entity permite criar cenários de testes em memória

# Data Seeding - Inicialização

É um recurso que permite popular o banco de dados com um conjunto de registros iniciais.

- Executado junto com a criação do modelo
- Valida e cria as entidades ainda respeitando as validações
- Inclusão dos dados de teste
- Inclusão e ajustes de informações pouco alteradas
- Transformações de dados na inicialização



# Data Seeding - Inicialização

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity<Filme>().HasData(
        new Filme { Categoria = "Ação",Codigo = 1, EmCartaz = true, Lancame
        new Filme { Categoria = "Ação",Codigo = 2, EmCartaz = true, Lancame
        new Filme { Categoria = "Ação",Codigo = 3, EmCartaz = true, Lancame
        new Filme { Categoria = "Ação",Codigo = 4, EmCartaz = true, Lancame
        new Filme { Categoria = "Ação",Codigo = 5, EmCartaz = true, Lancame

    );

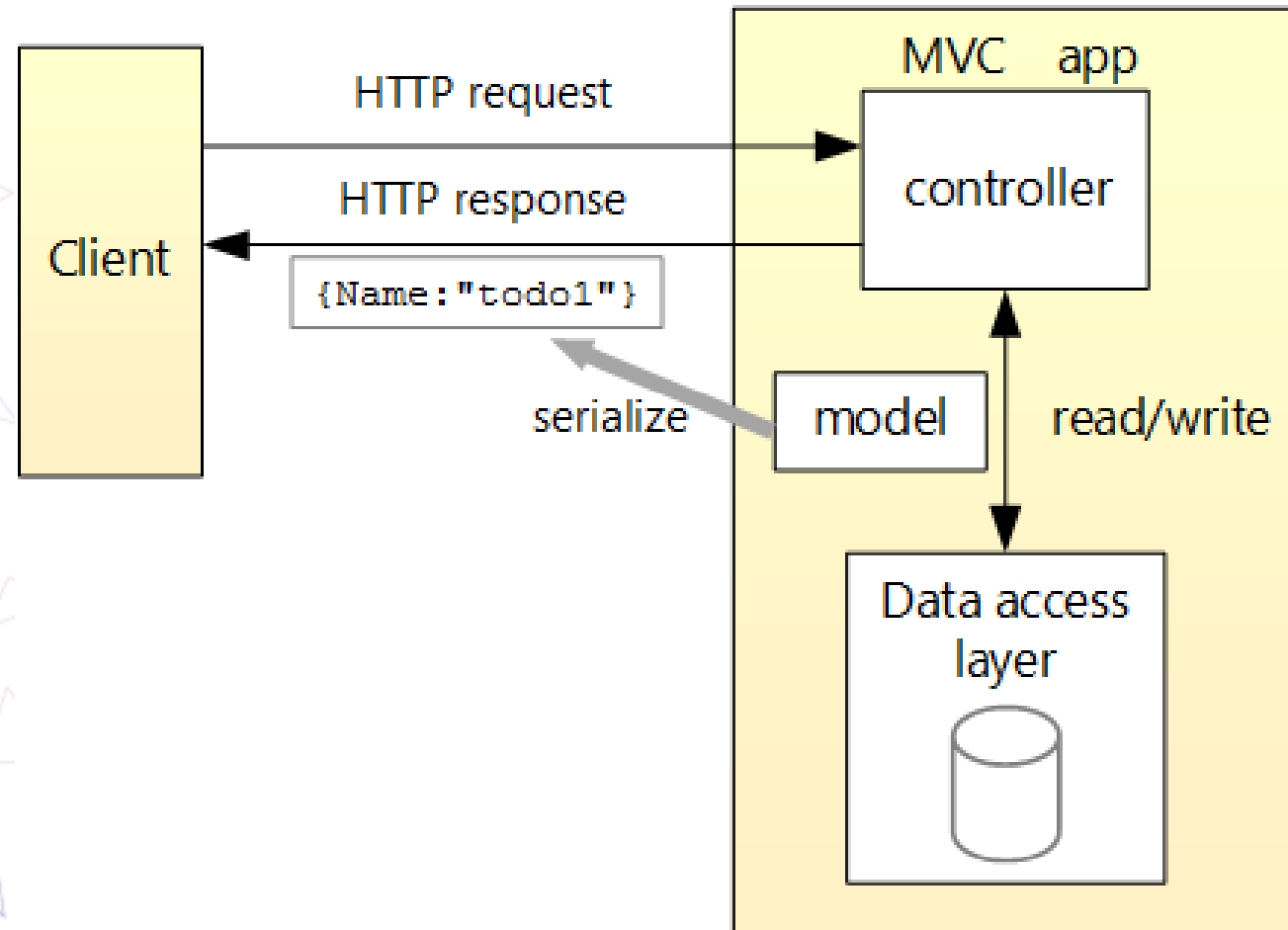
    base.OnModelCreating(modelBuilder);
}
```

# Construindo APIs

Os projetos de API dentro do ASP.NET, também chamados de Web API, permitem a criação de APIs ricas e complexas onde o retorno da chamada será uma estrutura de dados, e não uma View.

- API: Construção de uma interface para *servir* outras aplicações
- Gerar serviços que atendem a requisições *HTTP*
- Segue a arquitetura do projeto MVC
- As rotas são diferenciadas normalmente “/api/{controller}”
- Padrão de retorno é gerado pela framework

# Construindo APIs



# Construindo APIs

JSON

```
[
  {
    "date": "2019-07-16T19:04:05.7257911-06:00",
    "temperatureC": 52,
    "temperatureF": 125,
    "summary": "Mild"
  },
  {
    "date": "2019-07-17T19:04:05.7258461-06:00",
    "temperatureC": 36,
    "temperatureF": 96,
    "summary": "Warm"
  },
  {
    "date": "2019-07-18T19:04:05.7258467-06:00",
    "temperatureC": 39,
    "temperatureF": 102,
    "summary": "Cool"
  },
]
```

# Postman



POSTMAN

https://localhost:7055/api/Filmes

GET https://localhost:7055/api/Filmes **Send**

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Headers 6 hidden

KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
Key	Value	Description			

Body Cookies Headers (4) Test Results 200 OK 308 ms 829 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "codigo": 1,
3   "titulo": "Título",
4   "sinopse": "TESTE",
5   "lançamento": "2022-10-27T09:03:35.4551921-03:00",
6   "categoria": "Ação",
7   "emCartaz": true
8 },
9 {
10  "codigo": 2,
```

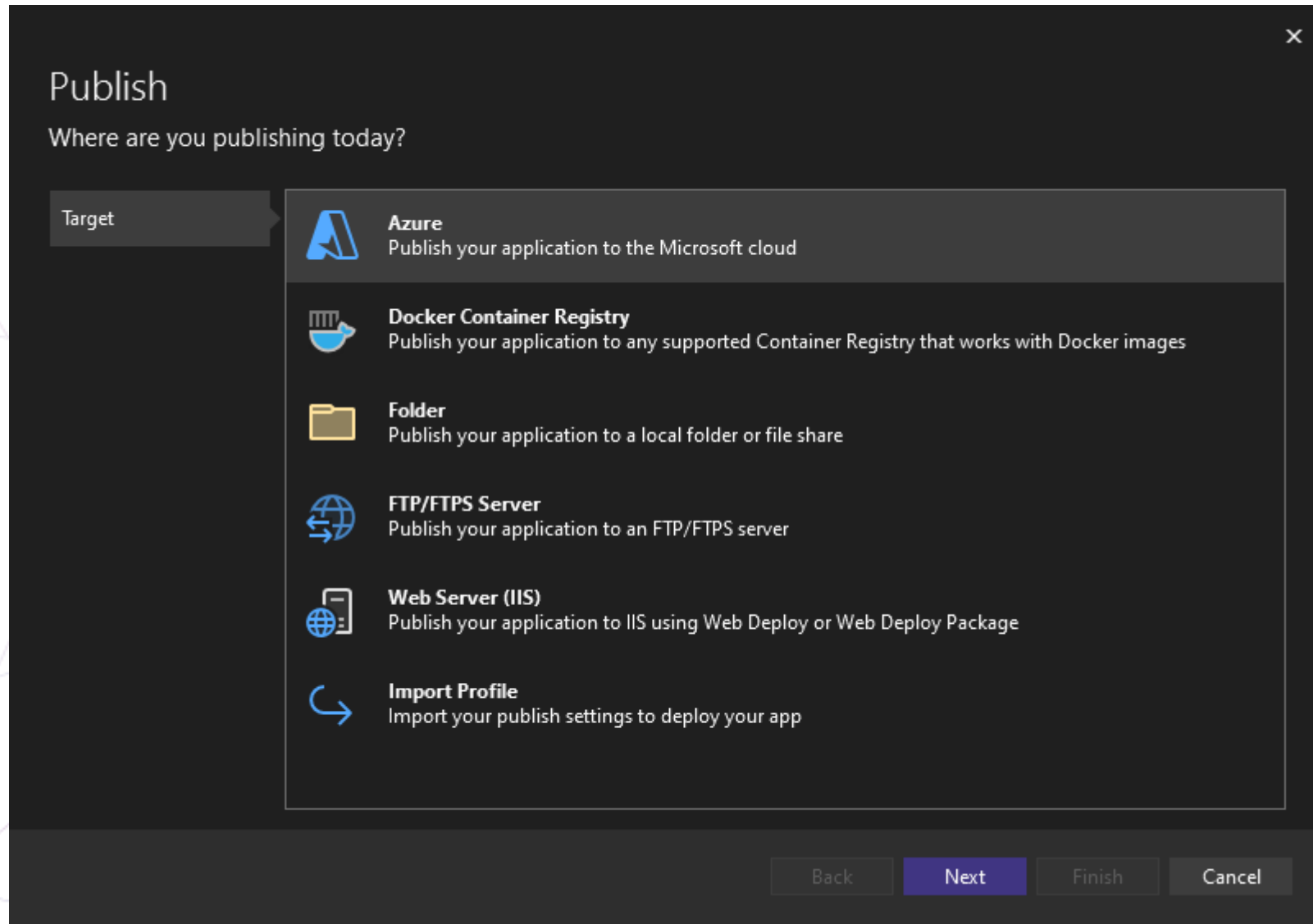
# Publicando um Projeto

Processo de cópia e preparação da aplicação para execução em outro ambiente.

- A publicação envia arquivos para execução
- Estrutura diferente da estrutura de desenvolvimento
- Deve-se definir o parâmetro de compilação *“debug”/“release”*
- A versão do .NET deve estar instalado no servidor de destino
- Conferir a variável de ambiente ASPNETCORE\_ENVIRONMENT
- No Visual Studio criamos um “perfil” de publicação
- O uso do Web Deploy é recomendado para gerenciamento



# Publicando um Projeto



# Hospedagem na Nuvem

A nuvem podem disponibilizar a função de hospedagem de aplicação como serviço, ou seja, não há um sistema operacional envolvido, apenas a contratação da hospedagem.

- Fácil gerenciamento
- Configuração pelo painel de monitoramento
- Agiliza e facilita a escalabilidade
- Atualização do backend de responsabilidade do host da Cloud
- Aumenta a segurança

# Microsoft Azure

**Services**

App Services Subscriptions

**ti | Configuration** ☆ ...

Refresh Save Discard Leave Feedback

**Application settings** **General settings** Default documents Path mappings

**Stack settings**

Stack .NET

.NET version

ASP.NET V4.8  
.NET 7 (preview)  
.NET 6 (LTS)  
**ASP.NET V4.8**  
ASP.NET V3.5  
.NET Core (3.1, 2.1)

**Platform settings**

Platform

Managed pipeline version Integrated

FTP state All allowed

FTP based deployment can be disabled or configured to accept FTP

HTTP version 1.1

**Application settings** \* General settings Default documents Path mappings

Application settings are encrypted at rest and transmitted over an encrypted channel. You can choose to environment variables for access by your application at runtime. [Learn more](#)

+ New application setting Hide values Advanced edit

Filter application settings

Name	Value	Source
ASPNETCORE_ENVIRONMENT	Staging	App

**Connection strings**

Connection strings are encrypted at rest and transmitted over an encrypted channel.



# Exercício 18

Objetivos Finais no Projeto

