Creating the List Page

Overview

- Hello MVC
- Creating the model and the repository
- Creating the controller
- Adding the view
- Styling the view

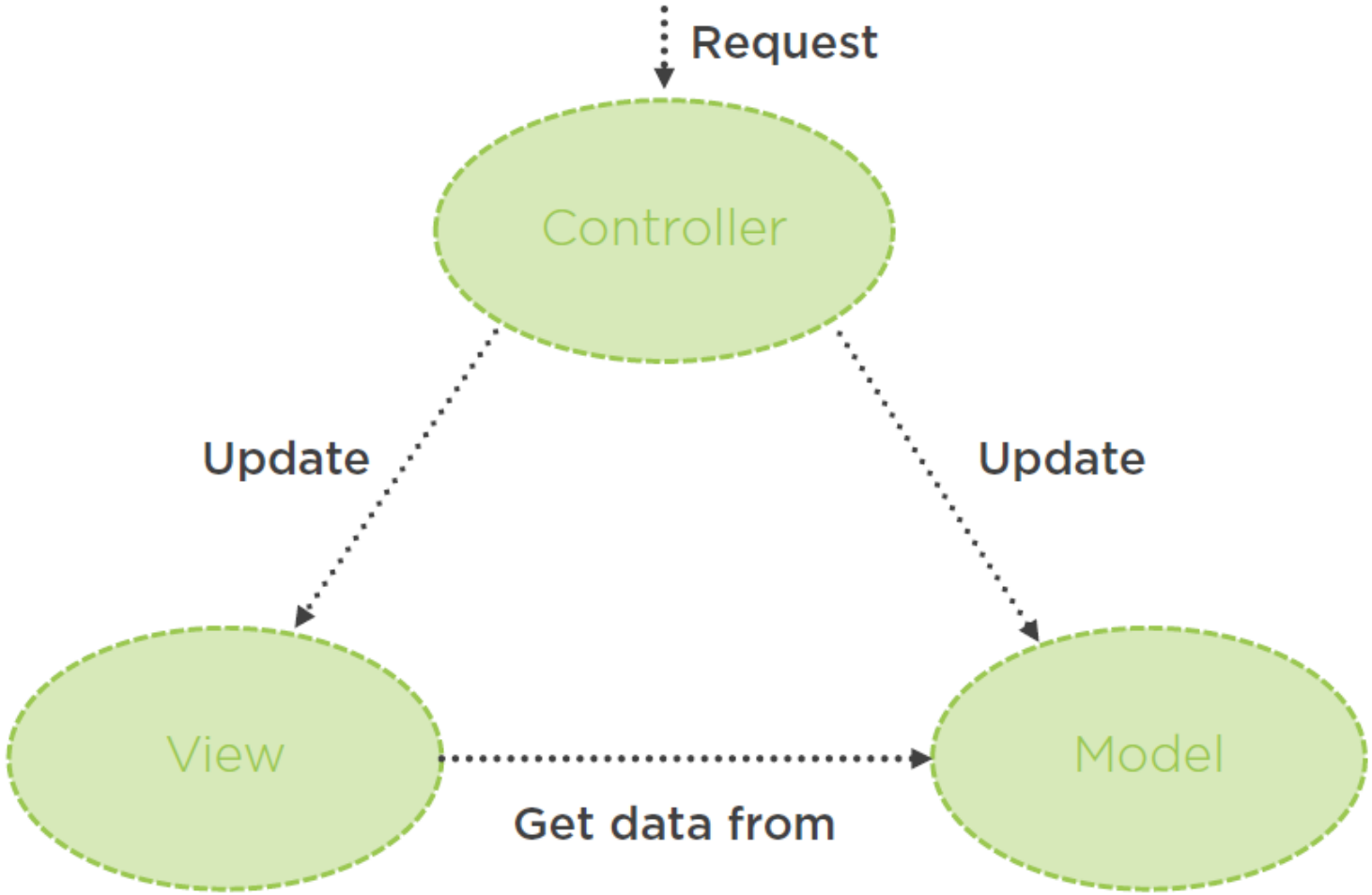
Hello MVC

The MVC in ASP.NET Core MVC

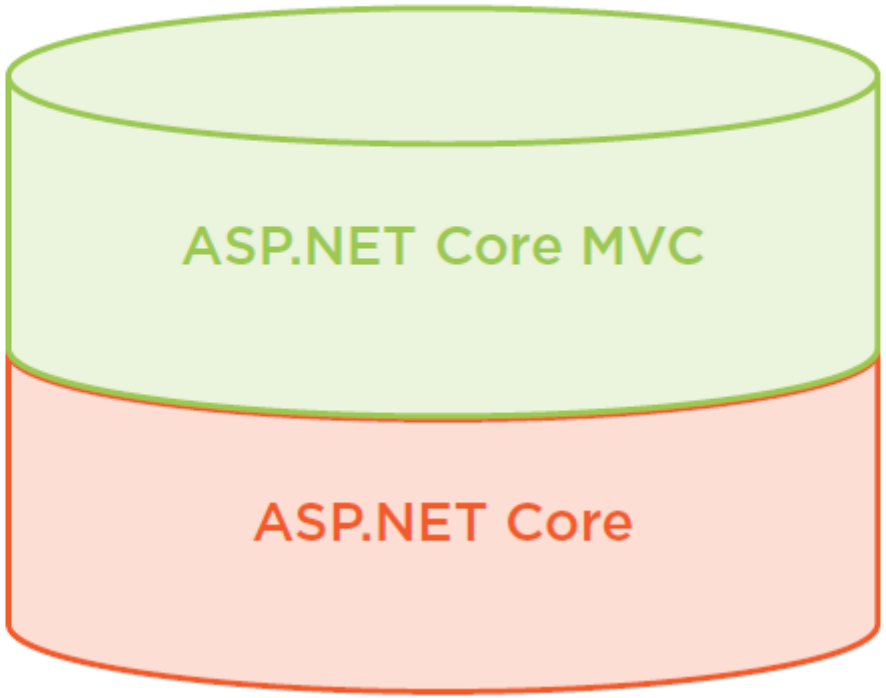
Model-View-Controller

- Architectural pattern
- Separation of concerns
- Promotes testability and maintainability

The MVC in ASP.NET Core MVC



The MVC in ASP.NET Core MVC



Creating the Model and the Repository

The Model

- Domain data + logic to manage data
- Simple API
- Hides details of managing the data

Sample Model Class

```
14 references
public class Pie
{
    5 references
    public int Id { get; set; }
    4 references
    public string Name { get; set; }
    4 references
    public string ShortDescription { get; set; }
    4 references
    public string LongDescription { get; set; }
    0 references
    public string AllergyInformation { get; set; }
    4 references
    public decimal Price { get; set; }
    4 references
    public string ImageUrl { get; set; }
    4 references
    public string ImageThumbnailUrl { get; set; }
    4 references
    public bool IsPieOfTheWeek { get; set; }
    4 references
    public bool InStock { get; set; }

    0 references
    public int CategoryId { get; set; }
    4 references
    public Category Category { get; set; }
}
```

The repository allows our code to use objects without knowing how they are persisted

Repository Interface

```
namespace PieShop.Models
{
    4 references
    public interface IPieRepository
    {
        2 references
        IEnumerable<Pie> AllPies { get; }
        0 references
        Pie GetPieByIdAsync(int pieId);
    }
}
```

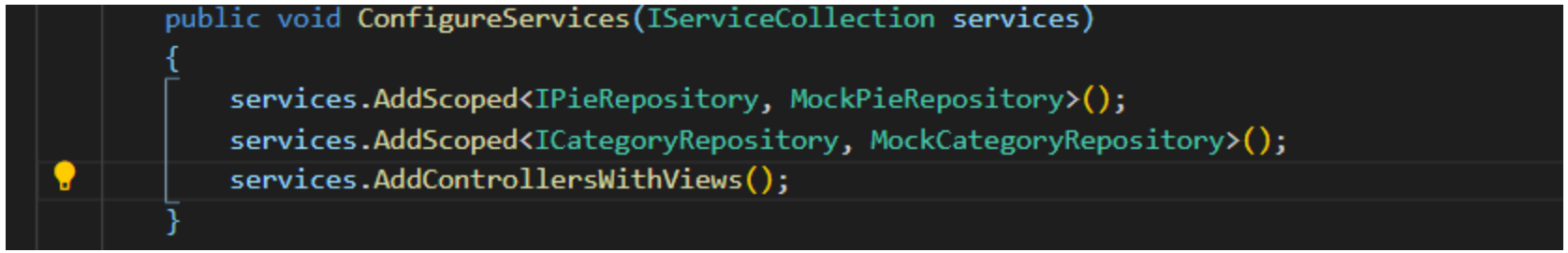
Mock Implementation

```
namespace PieShop.Models
{
    1 reference
    public class MockPieRepository : IPieRepository
    {
        4 references
        private readonly ICategoryRepository _categoryRepository = new MockCategoryRepository();
        2 references
        public IEnumerable<Pie> AllPies =>
            new List<Pie>
            {
                new Pie {Id = 1, Name="Strawberry Pie", Price=15.95M, ShortDescription="Lorem Ipsum"
                new Pie {Id = 2, Name="Cheese cake", Price=18.95M, ShortDescription="Lorem Ipsum",
                new Pie {Id = 3, Name="Rhubarb Pie", Price=15.95M, ShortDescription="Lorem Ipsum",
                new Pie {Id = 4, Name="Pumpkin Pie", Price=12.95M, ShortDescription="Lorem Ipsum",
            };

        0 references
        public IEnumerable<Pie> PiesOfTheWeek { get; }

        0 references
        public Pie GetPieByIdAsync(int pieId)
        {
            return AllPies.FirstOrDefault(p => p.Id == pieId);
        }
    }
```

Registering Services in ConfigureServices



Registration Options

AddTransient

AddSingleton

AddScoped