

ASP.NET Core MVC 3.1

Tahaluf Training Center 2021



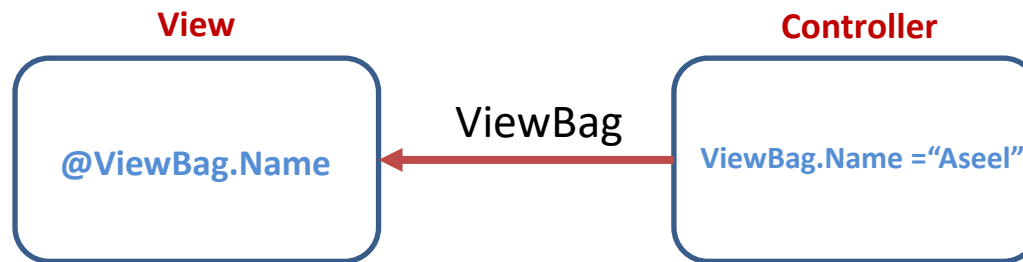
Chapter 8

- 1 View Bag
- 2 View Data
- 3 Display Dynamic data from DB



View Bag

- The ViewBag is used to transfer temporary data from the controller to the view.
- which this data is not included in the model, It is a dynamic type.
- If assign the same property name multiple times to ViewBag, it will only consider the last value assigned to the property.



View Bag

Example:

Let's find the number of customers, products, maximum price, and minimum price.



View Bag

Solution:

- In the admin dashboard controller,

```
public IActionResult Index()
{
    //ViewBag :transfer data from controller to view
    //only not vis versa . {Dynamic Type}
    //return number of customer == select count(*)
    from customer
    ViewBag.numberofcustomer = _context.Customers.Count();
    ViewBag.Sales = _context.Products.Sum(x => x.Sale);
    ViewBag.Customer = _context.Customers.ToList();

    return View();
}
```



View Bag

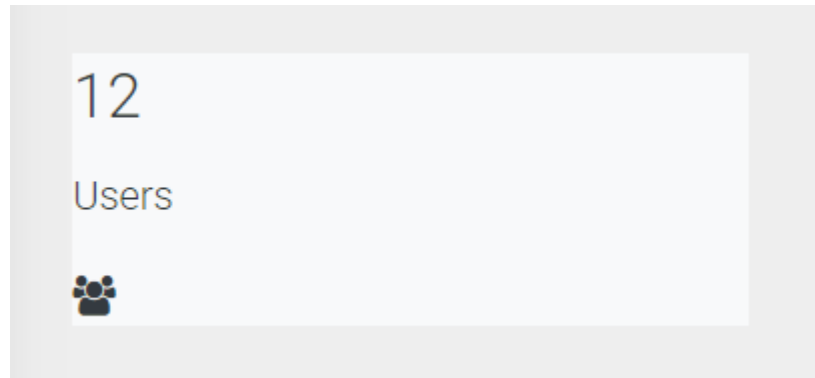
➤ On the view page :

```
<div class="col-lg-3 col-sm-6">
  <div class="card-box bg-light text-dark">
    <div class="inner">
      <h3> @ViewBag.numberofcustomer </h3>
      <p> Users </p>
    </div>
    <div class="icon">
      <i class="fa fa-users" aria-hidden="true"></i>
    </div>
  </div>
</div>
```



View Bag

Result:



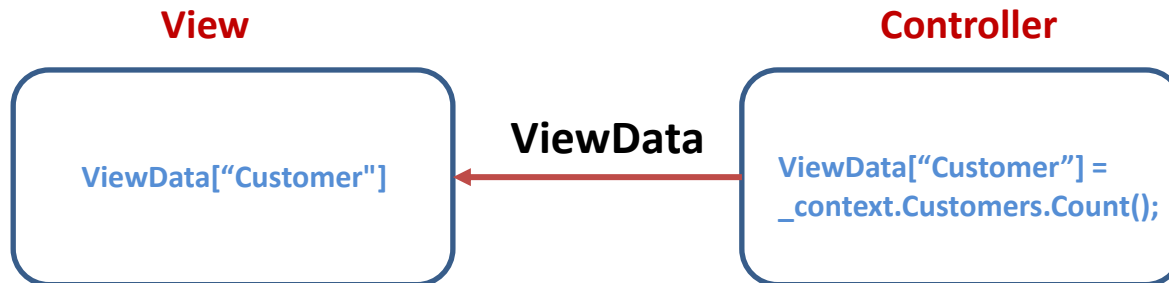
Chapter 8

- 1 View Bag
- 2 View Data
- 3 Display Dynamic data from DB



View Data

- ViewData is the same as ViewBag.
- ViewData is of Dictionary type.
- ViewBag and ViewData store data in the same dictionary internally.



View Data

Solution:

- In the admin dashboard controller,

```
public IActionResult Index()
{
    ViewData["Category"] = _context.Customers.Count();

    return View();
}
```



View Data

➤ On the view page :

```
<div class="col-lg-3 col-sm-6">
  <div class="card-box bg-light text-dark">
    <div class="inner">
      <h3> @ViewData["Category"] </h3>
      <p> Users </p>
    </div>
    <div class="icon">
      <i class="fa fa-users" aria-hidden="true"></i>
    </div>
  </div>
</div>
```



View Data

Result:

12

Users



View Bag

Solution:

- In the admin dashboard controller,

```
public IActionResult Index()
{
    //ViewBag :transfer data from controller to view
    //only not vis versa . {Dynamic Type}
    //return number of customer == select count(*)
    from customer
    ViewBag.numberofcustomer = _context.Customers.Count();
    ViewBag.Sales = _context.Products.Sum(x => x.Sale);
    ViewBag.Customer = _context.Customers.ToList();

    return View();
}
```



Chapter 8

- 1 View Bag
- 2 View Data
- 3 Display Dynamic data from DB



Display Dynamic Data

➤ In home controller :

```
public IActionResult catProduct(int id)//10
{
    var pro = _context.Products.Where(x =>
x.CategoryId == id).ToList();
    return View(pro);
}
```



Display Dynamic Data

➤ In Index View for the Home page :

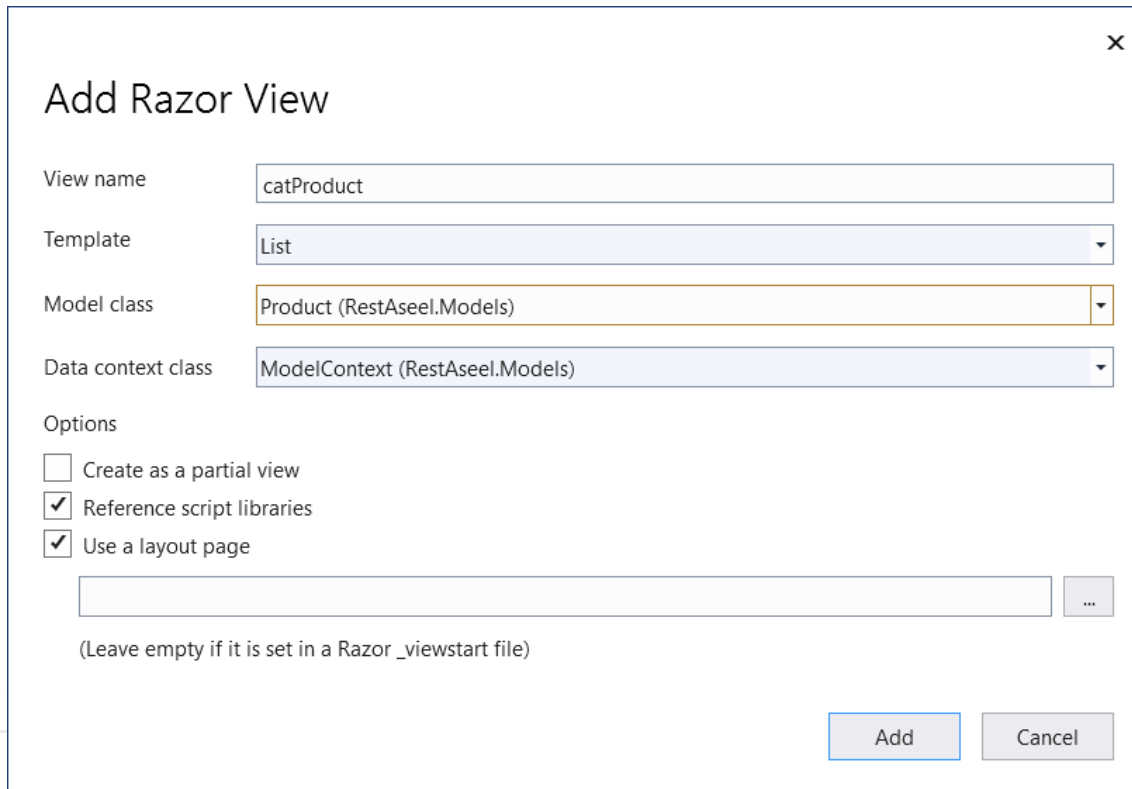
@foreach (var item in Model)

```
{
    <div class="card" style="width: 18rem;">
        
        <div class="card-body">
            <h5 class="card-title">@item.CategoryName</h5>
            <p class="card-text">Some quick example text to build on the
card title and make up the bulk of the card's content.</p>
            <a href="/Home/catProduct/@item.Id" class="btn btn-
primary">More Details</a>
        </div>
    </div>
}
```



Display Dynamic Data

- Create a view page for catproduct action result :
 - Right click => add view => Razor View.



×

Add Razor View

View name:

Template:

Model class:

Data context class:

Options

☐ Create as a partial view

☒ Reference script libraries

☒ Use a layout page

(Leave empty if it is set in a Razor _viewstart file)

...

Add Cancel



Display Dynamic Data

Result :

```
@model IEnumerable<RestAseel.Models.Product>

@{
    ViewData["Title"] = "catProduct";
}

<h1>catProduct</h1>

<p>
    <a asp-action="Create">Create New</a>
</p>
<table class="table">
    <thead>
        <tr>
            <th>
                @Html.DisplayNameFor(model => model.Namee)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Sale)
            </th>
            <th>
                @Html.DisplayNameFor(model => model.Price)
            </th>
        </tr>
    </thead>
</table>
```

No issues found

