**Strongly Typed View in ASP.NET Core MVC**

In ASP.NET Core MVC, there are 3 ways to pass data from a controller to a view

1. Using a strongly typed model object. This is also called Strongly typed view.
2. Using ViewData
3. Using ViewBag

**Strongly Typed View - Controller Code**   
  
The preferred approach to pass data from a controller to a view is by using a **strongly typed view.** To create a strongly typed view, in the controller action method, pass the model object to the View() helper method. Notice, in the example below, we are passing the Employee model object to the View() method.

public ViewResult Details()  
{  
    Employee model = \_employeeRepository.GetEmployee(1);  
  
    ViewBag.PageTitle = "Employee Details";  
  
    return View(model);  
}

**Strongly Typed View - View Code**  
  
To create a **strongly typed view**specify the model type in the view using @model directive. In the example below, we told the view that it will using the EmployeeManagement.Models.Employee object as the model by using @model directive. Please note that in the directive(@model), **m**is in lowercase.   
  
To access the model object properties we use @Model. In @Model, M is in uppercase. In the example, below to access the Employee object properties like Name, Email and Department we are using @Model.Name, @Model.Email and @Model.Department respectively.

@model EmployeeManagement.Models.Employee

<html>

<head>

    <title></title>

</head>

<body>

    <h3>@ViewBag.PageTitle</h3>

    <div>

        Name : @Model.Name

    </div>

    <div>

        Email : @Model.Email

    </div>

    <div>

        Department : @Model.Department

    </div>

</body>

</html>

**Strongly Typed View Benefits**  
  
Unlike **ViewData**and **ViewBag**, a **strongly typed view**provides **compile-time type checking**and **intellisense**. With intellisense support we can be more productive and the chances of mis-spelling and making typographical errors are almost nill. If we do make any errors we will come to know about them at compile time rather than at runtime. So always use a **strongly typed view**to pass data from a controller to a view.   
  
**What about PageTitle**  
  
In our example, we are still using **ViewBag**to pass PageTitle from the Controller to the View. How can we use a strongly typed view to pass PageTitile. Well, this is one example, where we can use a view specific model called **ViewModel**.