

Bài 4. Tương tác dữ liệu giữa controller và view

- **Mục đích, yêu cầu:** Cung cấp kiến thức về tham số và cách truyền tham số từ View sang Controller . Sau khi học xong bài học này sinh viên có thể xây dựng được các ứng dụng truyền tham số.

- **Hình thức tổ chức dạy học:** Lý thuyết, trực tiếp, trực tuyến + tự học

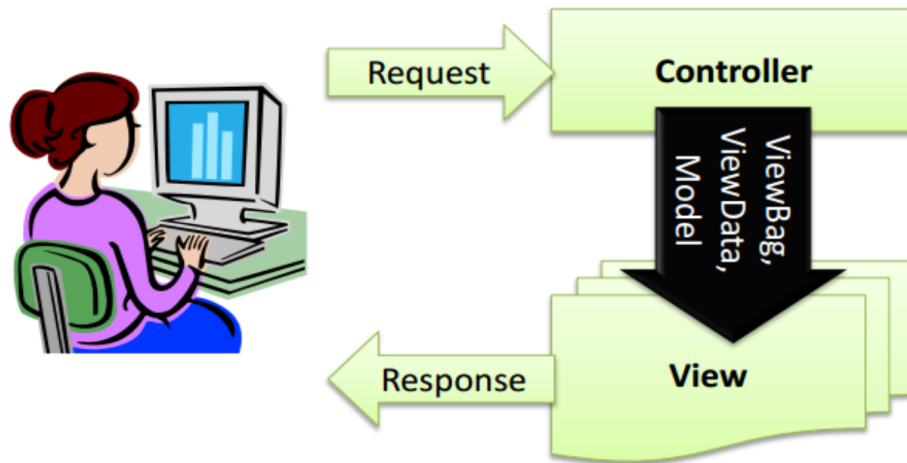
- **Thời gian:** Lý thuyết(trực tiếp 3, online: 3) Tự học, tự nghiên cứu: 12

- **Nội dung:**

1. Truyền dữ liệu từ controller sang view	2
1.1 Sử dụng ViewBag/ViewData	2
1.2 Sử dụng Model.....	3
1.3 Ví dụ ứng dụng 1 (Xem video)	3
2. Truyền dữ liệu từ view sang controller	6
2.1. Tham số.....	6
2.2. Tiếp nhậntham số	6
2.2.1. Sử dụng Request.....	7
2.2.2. Sử dụng FormCollection.....	7
2.2.3. Sử dụng đối số action.....	8
2.2.4. Sử dụng model.....	8
2.3. Ví dụ ứng dụng.....	8

1. Truyền dữ liệu từ controller sang view

ViewBag/ViewData và **Model** được sử dụng để chia sẻ dữ liệu giữa Controller và View



1.1 Sử dụng ViewBag/ViewData

ViewBag là một đối tượng động, có thể đặt bất cứ thứ gì vào đó; Đối tượng **ViewBag** không có thuộc tính xác định nào cho đến khi đặt thứ gì đó vào bên trong nó.

Cú pháp:

ViewBag.Thuộc_tính=Giá_trị;

Hoặc **ViewData["Thuộc_tính"]=Giá_trị;**

Lấy giá trị từ ViewBag/ViewData:

@ViewBag.Thuộc_tính

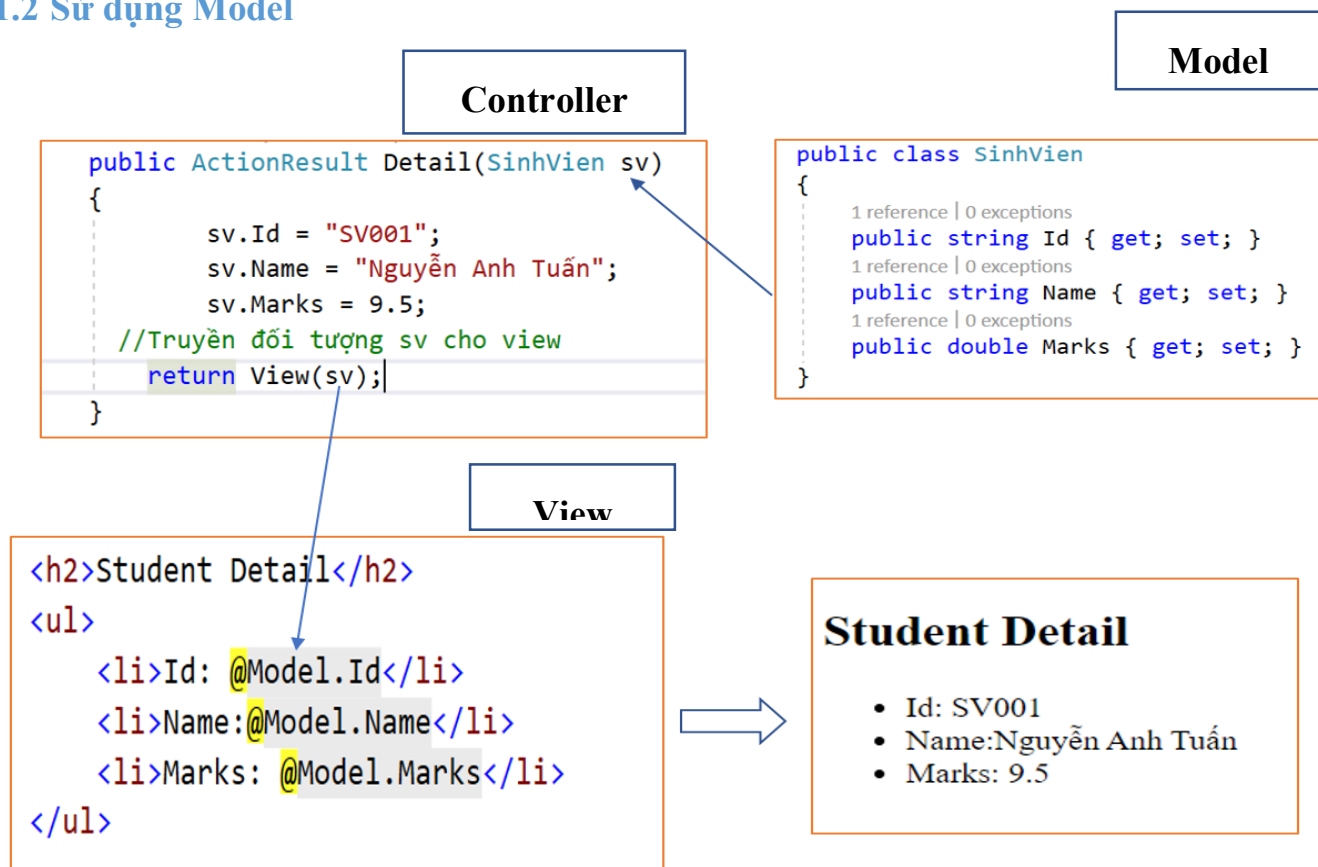
Hoặc **@ViewData["Thuộc_tính"]**

```
public ActionResult Detail()
{
    ViewBag.Id = "SV001";
    ViewBag.Name = "Nguyễn Anh Tuấn";
    ViewData["Marks"] = 9.5;
    return View();
}
```

**ViewBag.Id ~
ViewData["Id"]**

```
<h2>Student Detail</h2>
<ul>
    <li>Id: @ViewBag.Id</li>
    <li>Name: @ViewData["Name"]</li>
    <li>Marks: @ViewBag.Marks</li>
</ul>
```

1.2 Sử dụng Model



1.3 Ví dụ ứng dụng 1 (Xem video)

- Tạo 1 Project mới
- Tạo Controller chọn mẫu **Empty Controller**, đặt tên Controller là **NhapDiemController**

Cách 1: 1 Truyền dữ liệu từ View sang Controller sử dụng ViewBag/

ViewData

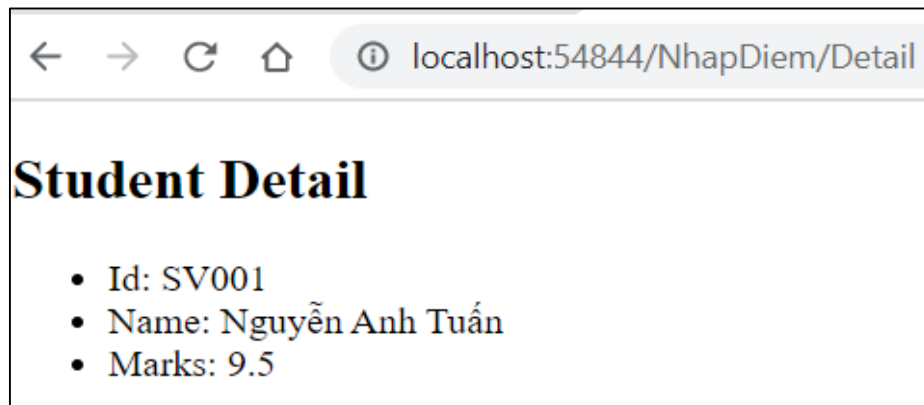
- Trong **NhapDiemController** tạo 1 action method có tên là **Detail** và một View tương ứng.

```
public ActionResult Detail()
{
    //Truyền DL từ Controller sang View sử dụng ViewBag/ViewData
    ViewBag.Id = "SV001";
    ViewBag.Name = "Nguyễn Anh Tuấn";
    ViewData["Marks"] = 9.5;
    return View();
}
```

Detail.cshtml

```
<h2>Student Detail</h2>
*@Sử dụng ViewBag/ViewData*@
<ul>
    <li>Id: @ViewBag.Id</li>
    <li>Name: @ViewData["Name"]</li>
    <li>Marks: @ViewBag.Marks</li>
</ul>
```

=> Chạy xem kết quả:



Cách 2: Truyền dữ liệu từ View sang Controller sử dụng Model

- Tạo một class trong folder Model và đặt tên là **SinhVien**. Mở file **SinhVien.cs** khai báo các thuộc tính cho class này:

```
public class SinhVien
{
    1 reference | 0 exceptions
    public string Id { get; set; }
    1 reference | 0 exceptions
    public string Name { get; set; }
    1 reference | 0 exceptions
    public double Marks { get; set; }
}
```

- Trong **NhapDiemController** sửa code cho **Action Detail** như sau:

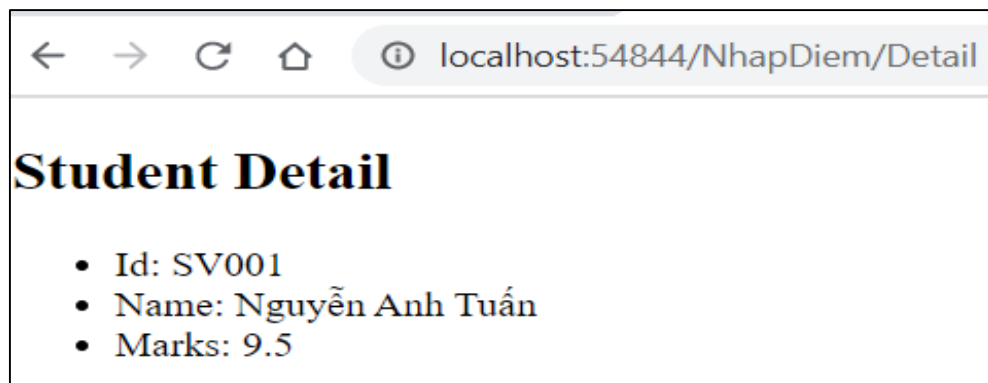
using Lab02.Models;

```
public ActionResult Detail(SinhVien sv)
{ //Truyền DL từ Controller sang View sử dụng Model
    sv.Id = "SV001";
    sv.Name = "Nguyễn Anh Tuấn";
    sv.Marks = 9.5;
    //Truyền đối tượng sv cho view
    return View(sv);
}
```

- Sửa lại View **Detail.cshtml** như sau:

```
<h2>Student Detail</h2>
@*Sử dụng Model*@
<ul>
    <li>Id: @Model.Id</li>
    <li>Name: @Model.Name</li>
    <li>Marks: @Model.Marks</li>
</ul>
```

=> Chạy xem kết quả:



2. Truyền dữ liệu từ view sang controller

2.1. Tham số

Tham số yêu cầu từ người dùng được cung cấp dưới 2 dạng: **Query String** hoặc **Form field**

Query String

```
<a href="/NhapDiem/Xuly?Id=SV01&Name=Tuấn&Marks=9">Tuấn</a>  
<a href="/NhapDiem/Xuly/SV02/?Name=Hải&Marks=8">Hải</a>
```

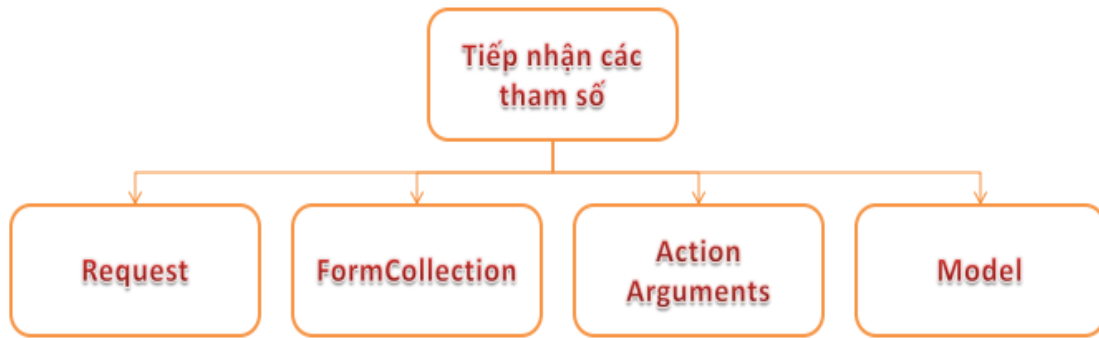
Form field

```
<form action="/NhapDiem/Xuly" method="post">  
    <div> Mã SV:</div><input name="Id" />  
    <div> Họ tên:</div><input name="Name" />  
    <div> Điểm:</div><input name="Marks" />  
    <hr/>  
    <input type="submit" value="Nhập" />  
</form>
```

2.2. Tiếp nhận tham số

Trong MVC có 4 cách để nhận tham số:

- ✓ Sử dụng đối tượng ngầm định Request
- ✓ Sử dụng đối số của Action
- ✓ Sử dụng tham số FormCollection
- ✓ Sử dụng Model



2.2.1. Sử dụng Request

Trong phương thức hành động có thể viết một trong số cách sau đây để nhận tham số

- ✓ String value = Request["<tham số>"];
- ✓ String value = Request.QueryString["<tham số>"];
- ✓ String value = Request.Form["<tham số>"];
- ✓ String value = Request.Params["<tham số>"];

Ví dụ sau sẽ nhận tham số:

```

string Ma = Request["Id"];
string Ten = Request["Name"];
double Diem = Convert.ToDouble(Request["Marks"]);
  
```

2.2.2. Sử dụng FormCollection

- ✓ Tập hợp các tham số form vào đối số **FormCollection** của Action. Chỉ nhận được các trường form.
- ✓ Ví dụ nhận tham số form có tên txtName

```

public ActionResult Xuly(FormCollection data)
{
    string Ma = data["Id"];
    string Ten = data["Name"];
    double Diem = Convert.ToDouble(data["Marks"]);
    return View();
}
  
```

Cách lấy tương đương với **Request.Form**

```
var value = Request.Form["Name"];
```

2.2.3. Sử dụng đối số action

- ✓ Định nghĩa tham số cho Action để nhận tham số cùng tên.
- ✓ Ví dụ nhận 2 tham số txtUserName và txtPassword

```
public ActionResult Xuly
(
  string Id = "",
  string Name = "",
  double Marks = 0
){...}
```

```
<form action="/NhapDiem/Xuly" method="post">
  <input name="Id" />
  <input name="Name" />
  <input name="Marks" />
  <input type="submit" value="Nhập" />
</form>
```

2.2.4. Sử dụng model

- ✓ Tạo lớp trong Model chứa thuộc tính cùng tên với tham số.
- ✓ Sử dụng lớp này làm đối số cho Action để nhận tham số cùng tên với thuộc tính.

```
public class SinhVien
{
  public string Id { get; set; }
  public string Name { get; set; }
  public double Marks { get; set; }
}
```

```
public ActionResult Xuly(SinhVien sv){...}
```

2.3. Ví dụ ứng dụng

Bài 1: Tạo 1 giao diện web cho phép người dùng nhập vào: mã sinh viên, họ tên, điểm. Khi người dùng nhấn nút **Nhập** thì sẽ hiển thị kết quả sang 1 trang khác.

Nhập điểm sinh viên

Mã SV:

SV01

Họ tên:

Lê Thu Hà

Điểm:

8.5

Nhập

← → ↻ 🏠 ⓘ localhost:53568/NhapDiem/Xuly

Mã SV:	SV01
Họ tên:	Lê Thu Hà
Điểm:	8.5

- Tạo 1 Project mới
 - Tạo Controller chọn mẫu **Empty Controller**, đặt tên Controller là **NhapDiemController**
 - Tạo View không sử dụng Layout tương ứng với action Index trong controller **NhapDiemController**
- + Code view **Index** hiển thị Form nhập như sau:

```
<h2>Nhập điểm sinh viên</h2>
<form action="/NhapDiem/Xuly" method="post">
  <div> Mã SV:</div><input name="Id" />
  <div> Họ tên:</div><input name="Name" />
  <div> Điểm:</div><input name="Marks" />
  <hr/>
  <input type="submit" value="Nhập" />
</form> <br/>
```

- Trong **NhapDiemController** tạo 1 action method có tên là **Xuly** và một View tương ứng.

Quá trình hoạt động: người dùng sẽ nhập dữ liệu cho các tham số *Id*, *Name* và *Mark* từ View **Index.cshtml**. **Controller** sẽ lấy dữ liệu từ View **Index** và chuyển kết quả sang View **Xuly**.

Cách 1: Truyền tham số từ **View** sang **Controller** bằng **Request**

- Trong **NhapHoSoController** viết code cho **Action Xuly** như sau:

```
public ActionResult Xuly()
{
  //Lấy dữ liệu từ View bằng Request["<tham số>"]
  string Ma = Request["Id"];
  string Ten = Request["Name"];
  double Diem = Convert.ToDouble(Request["Marks"]);
  ViewBag.ma = Ma;
  ViewBag.ten = Ten;
  ViewBag.diem = Diem;
  return View();
}
```

- Trang **Xuly.cshtml** viết code để hiển thị giá trị được truyền qua **ViewBag**

```

<body>
  <div>
    <table border="1">
      <tr>
        <td>Mã SV:</td>
        <td>@ViewBag.ma</td>
      </tr>
      <tr>
        <td>Họ tên:</td>
        <td>@ViewBag.ten</td>
      </tr>
      <tr>
        <td>Điểm:</td>
        <td>@ViewBag.diem</td>
      </tr>
    </table>
  </div>
</body>

```

=>Chạy thử xem kết quả.

Cách 2: Truyền tham số từ View sang Controller bằng FormCollection

- Trong **NhapDiemController** sửa code cho **Action Xuly** như sau:

```

public ActionResult Xuly(FormCollection data)
{
  //Lấy dữ liệu từ View bằng FormCollection
  string Ma = data["Id"];
  string Ten = data["Name"];
  double Diem = Convert.ToDouble(data["Marks"]);
  ViewBag.ma = Ma;
  ViewBag.ten = Ten;
  ViewBag.diem = Diem;
  return View();
}

```

=>Chạy thử xem kết quả

Cách 3: Truyền tham số từ View sang Controller bằng Action argument

- Trong **NhapDiemController** sửa code cho **Action Xuly** như sau:

```

public ActionResult Xuly(string Id="", string Name = "", double Marks = 0)
{
  //Lấy dữ liệu từ View bằng Action Arguments
  ViewBag.ma = Id;
  ViewBag.ten = Name;
  ViewBag.diem = Marks;
  return View();
}

```

- Đặt *tên các tham số* của **Action Xuly** trong Controller theo đúng *tên các phần tử* trên form.

=>Chạy thử xem kết quả

Cách 4: Truyền tham số từ View sang Controller bằng Model

- Tạo một class trong folder Model và đặt tên là **SinhVien**. Mở file **SinhVien.cs** khai báo các thuộc tính **Id, Name, Marks** cho class này.

```
namespace Lab02.Models
{
    0 references
    public class SinhVien
    {
        0 references | 0 exceptions
        public string Id { get; set; }
        0 references | 0 exceptions
        public string Name { get; set; }
        0 references | 0 exceptions
        public double Mark { get; set; }
    }
}
```

- Trong **NhapDiemController** sửa code cho **Action Xuly** như sau:

```
public ActionResult Xuly(SinhVien sv)
{ //Lấy dữ liệu từ View bằng Model
    ViewBag.ma = sv.Id;
    ViewBag.ten = sv.Name;
    ViewBag.diem = sv.Marks;
    return View();
}
```

=>Chạy thử xem kết quả

Bài 2: Tạo 1 giao diện web cho phép người dùng nhập vào 2 số, sau đó thực hiện chọn 1 phép tính.

Khi người dùng nhấn nút **Caculate** thì phép tính sẽ được thực hiện và hiển thị kết quả sang một trang khác.

Application name

Index

nhập a:

nhập b:

☐ Cộng ☐ Trừ ☐ Nhân ☐ Chia

24