

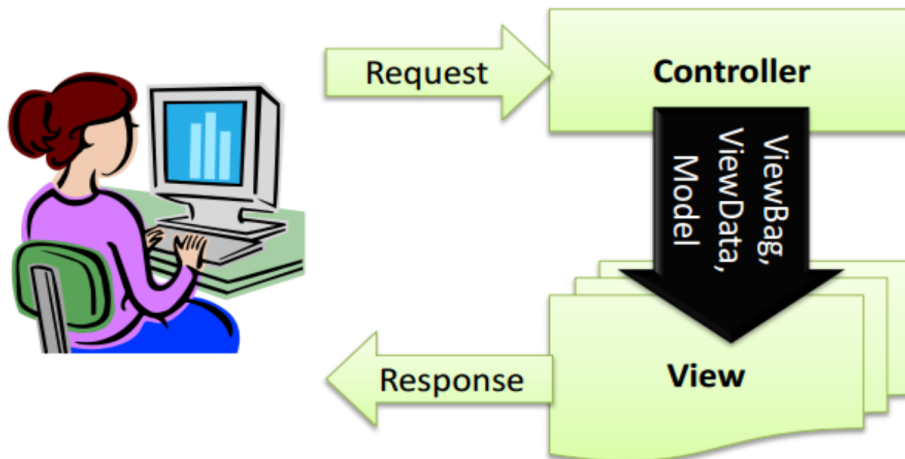
Tương tác dữ liệu giữa controller và view

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	3
2. Truyền dữ liệu từ view sang controller	5
2.1. Tham số	5
2.2. Tiếp nhận tham số	5
2.2.1. Sử dụng Request	6
2.2.2. Sử dụng tham số FormCollection	6
2.2.3. Sử dụng tham số của Action	7
2.2.4. Sử dụng model	7
2.3. Ví dụ ứng dụng	7

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:

+ Gán giá trị:

ViewBag.Thuộc_tính=Giá_trị;

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

+ Lấy giá trị:

@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"]**

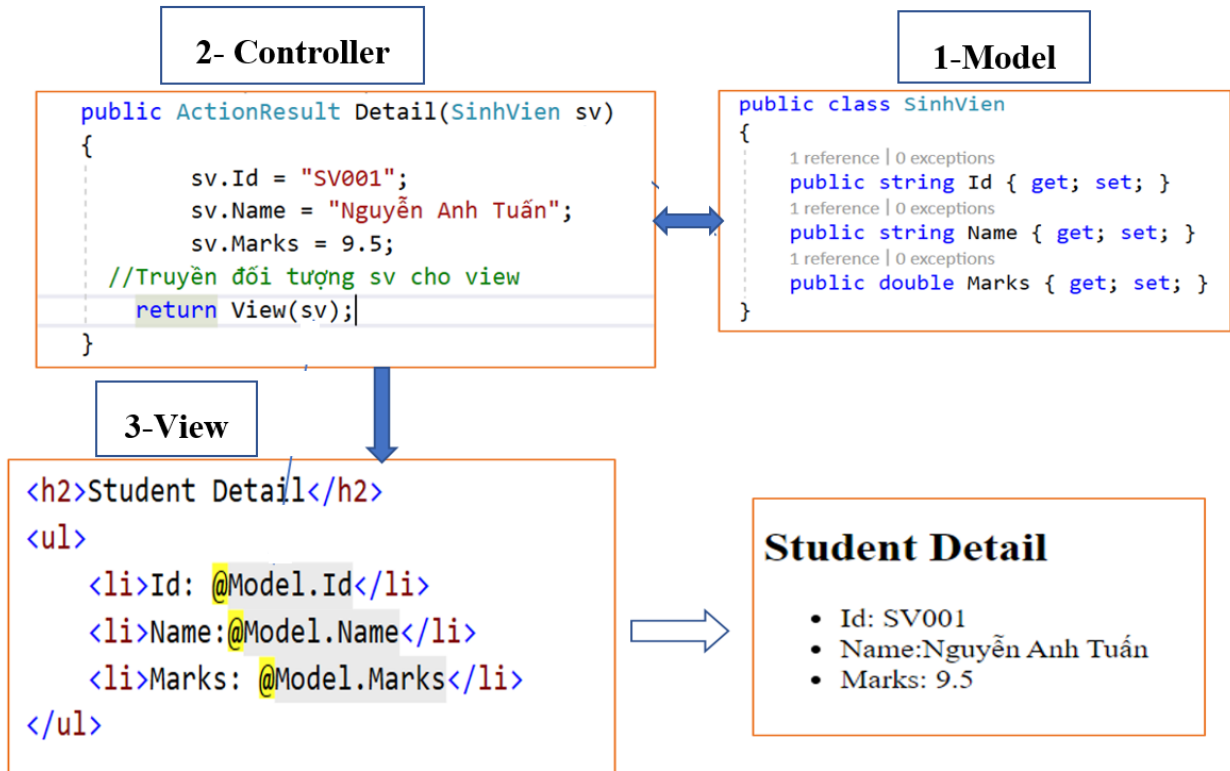
Detail.cshtml

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

Student Detail

- Id: SV001
- Name: Nguyễn Anh Tuấn
- Marks: 9.5

1.2 Sử dụng Model



1.3 Ví dụ ứng dụng 1

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

NhapDiemController

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

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

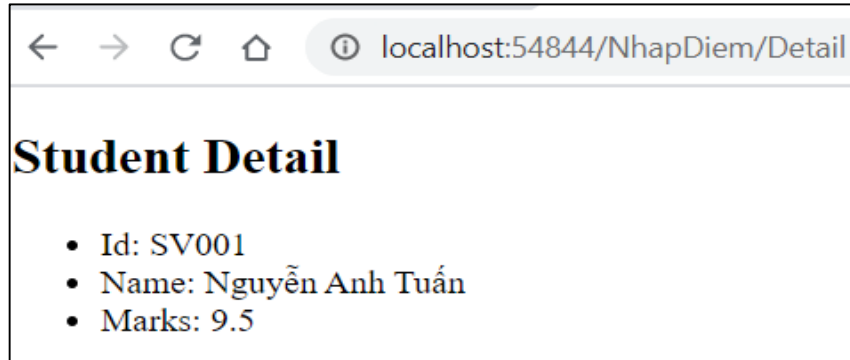
```

    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 Controller sang View 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:

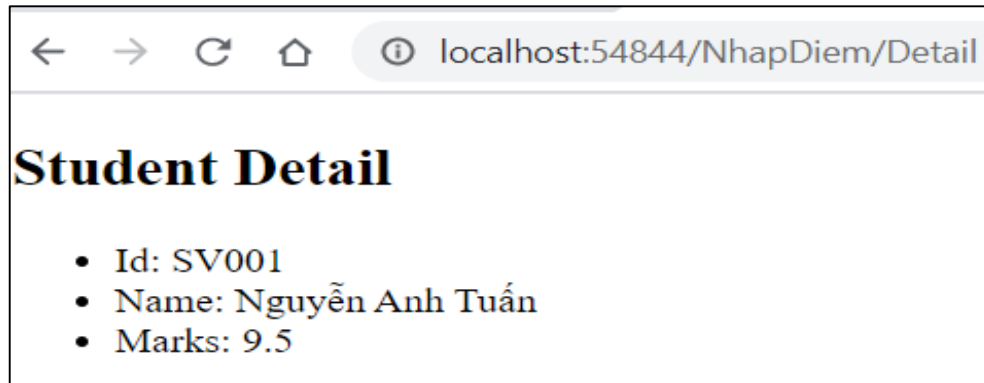
using Lab02.Models;

```
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**(lấy theo link href) hoặc **Form field**(lấy theo tên của **html control**)

 **Query String**

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

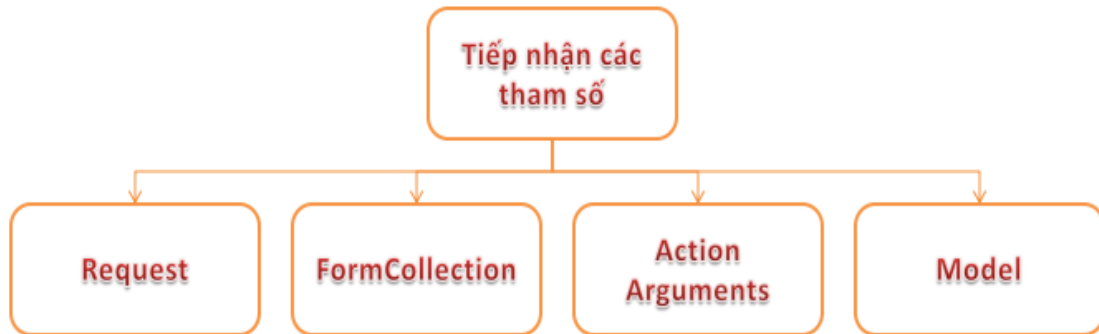
 **Form field**

```
<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 tham số FormCollection
- ✓ Sử dụng tham số của Action
- ✓ 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>"];`
 - Nếu **href** hoặc **method=get** → `Request.QueryString`
 - Nếu **method=post** → `Request.Form`

Ví dụ nhận tham số:

```

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

2.2.2. Sử dụng tham số FormCollection

- ✓ Tập hợp các tham số form vào lớp **FormCollection** của Action. Lớp này chỉ nhận được các trường form (Form field)
- ✓ Khai báo tham số: `FormCollection <Tên đối tượng>`
`Biến=<Tên đối tượng>["Form field"];`
- ✓ Ví dụ nhận tham số từ các trường form: Id, Name, Marks

```

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

2.2.3. Sử dụng tham số của Action

Thay vì ta khai báo **FormCollection** trong các **Action**, ta thay thế bằng các tên biến cụ thể (**name** của các **html control** trong form). Khi **html control** ít thì làm cách này, tuy nhiên nếu nhiều ta nên dùng **FormCollection**

✓ Định nghĩa các tham số cho **Action** phải trùng tên với các trường trên Form.

```
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 tham 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:

Họ tên:

Điểm:

← → ↻ 🏠 ⓘ 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**

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 **NhapDiemController** 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 tham số của Action

- 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:

using Lab02.Models;

```
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 hệ số a, b của phương trình bậc 1 $AX+B=0$.

Khi người dùng nhấn nút **Giải** sẽ hiển thị kết quả nghiệm sang một trang khác. (sử dụng cả 4 cách)

Giải phương trình bậc 1

Vào hệ số a:

Vào hệ số b:

Bài 3: 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. (sử dụng Model)

Index

nhập a:

nhập b:

☐ Cộng
 ☐ Trừ
 ☐ Nhân
 ☐ Chia

Bài 4: Xây dựng trang web ASP.NET MVC tính tiền điện như sau:

TÍNH TIỀN ĐIỆN		
Họ tên chủ hộ:	<input type="text" value="Lê Thu Hà"/>	<input checked="" type="checkbox"/> Hộ ưu tiên
Loại điện tiêu thụ:	<input type="text" value="Sinh hoạt"/>	
Vào chỉ số cũ:	<input type="text" value="100"/>	KW
Vào chỉ số mới:	<input type="text" value="320"/>	KW
<input type="button" value="TÍNH"/>		

Yêu cầu: Khi người dùng click vào nút **Tính** thì gửi kết quả sang trang khác như sau:

Chủ hộ:	Lê Thị Hà
Số điện tiêu thụ:	220 KW
Tiền phải trả:	499500 Đồng

Biết rằng: đơn giá điện **"Sinh hoạt"** được tính như sau:

- 100 kw đầu giá là: 2000đ/kw
- 50 kw tiếp theo giá là: 2500đ/kw
- 50 kw tiếp theo giá là: 3000đ/kw
- Lớn hơn 200 kw giá là: 4000đ/kw

Nếu loại điện dùng là **"Kinh doanh"** thì tiền điện phải trả sẽ tăng 20%. Nếu loại điện dùng là **"Sản xuất"** thì tiền điện phải trả sẽ tăng 30%. Với những **"hộ ưu tiên"** thì tiền điện phải trả sẽ được giảm 10%.