

Chủ đề 2. CONTROLLERS

- Chứa các phương thức Actions cho project (file **.cs**).

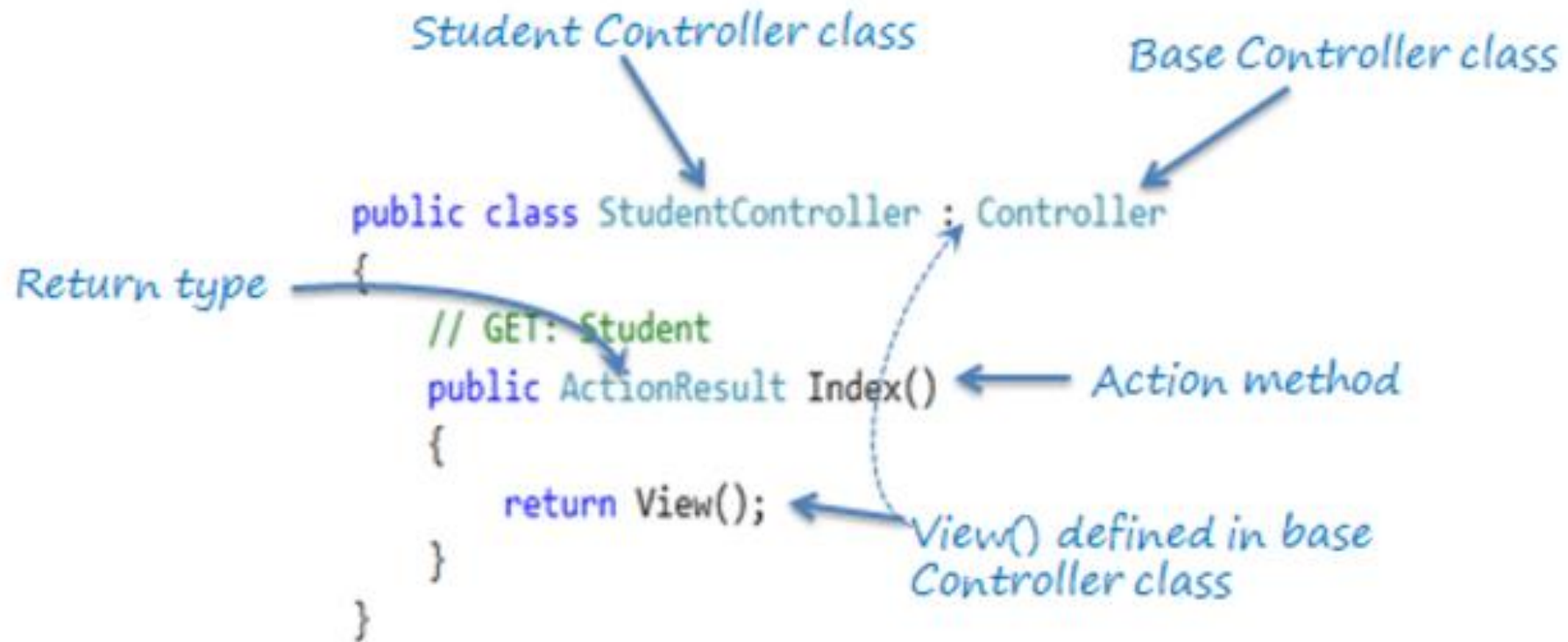
Ví dụ: Account**Controller**.cs, Helloworld**Controller**.cs.

- Tất cả các file **.cs** đều phải kết thúc bởi chuỗi **Controller**.
- Tất cả các controller chứa trong thư mục Controllers

Action:

- Là nơi đến của mỗi Action cho Controller
- Luôn luôn là Public
- Không hạn chế giá trị trả về
- Kết quả trả về của Action là 1 ActionResult, với kiểu này có thể trả về nhiều kết quả khác nhau tùy thuộc vào mục đích sử dụng

Action



Action Method

Action Result

```
graph TD; AR[Action Result] --> View[View()]; AR --> PartialView[PartialView()]; AR --> Content[Content()]; AR --> File[File()]; AR --> Json[Json()]; AR --> JavaScript[JavaScript()]; AR --> Redirect[Redirect()]; AR --> RedirectToAction[RedirectToAction()];
```

View()

PartialView()

Content()

File()

Json()

JavaScript()

Redirect()

RedirectToAction()

Action

1. **View()** trả về View được bọc trong layout (Master Page)
2. **PartialView()** giống như View, nhưng sẽ không được bọc trong layout, ví dụ nếu như trong Action Index return PartialView(), ta sẽ không nhìn thấy footer hoặc navbar nữa, chỉ có View Index thôi.
3. **Content()** trả về text.
4. **Json()** trả về dữ liệu Json.

Action

- 5. **Javascript()** trả về nội dung javascript
- 6. **File()** trả về nội dung file không bao gồm layout.
- 7. **RedirectToAction()** chuyển sang 1 Action khác.
- 8. **Redirect()** chuyển sang một url khác.

| Tên Result | Mô tả | Hàm sử dụng |
|---|--|---|
| ContentResult | Trả về chuỗi | Content() |
| FileContentResult/ FilePathResult/ FileStreamResult | Trả về nội dung file | File() |
| JavaScriptResult | Trả về nội dung JavaScript | JavaScript() |
| JsonResult | Trả về dữ liệu dạng Json | Json() |
| RedirectResult | Chuyển sang URL mới | Redirect() |
| RedirectToRouteResult | Chuyển sang 1 action hoặc 1 action của controller khác | RedirectToRoute() RedirectToAction() |
| ViewResult | Chuyển sang View để hiển thị | View() |
| PartialViewResult | Chuyển sang View để hiển thị không layout | PartialView() |

Action

```
public ActionResult Index()
{
    ViewBag.hello = "Hello world";

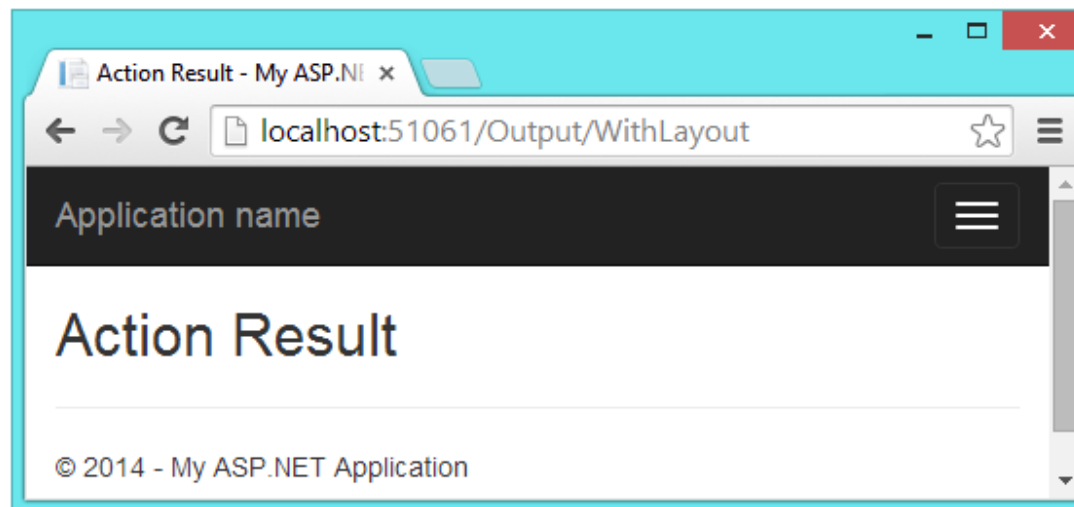
    return View();
}
```

```
public ActionResult Edit(Student std)
{
    // Cập nhật đối tượng Student vào cơ sở dữ liệu
    return RedirectToAction("Index");
}
[HttpDelete]
0 references
public ActionResult Delete(int id)
{
    // Xóa Student từ cơ sở dữ liệu với id xác định
    return RedirectToAction("Index");
}
```


Action

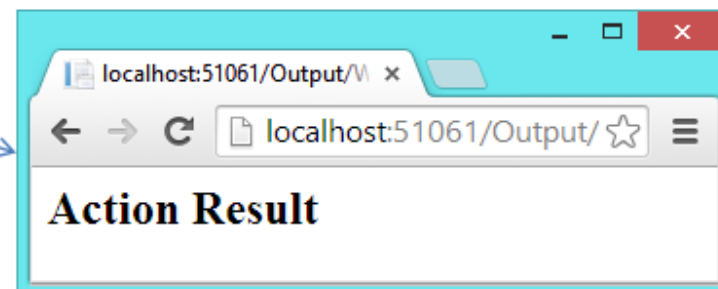
```
public ActionResult DanhsachSinhvien()  
{  
    IList<Student> studentList = new List<Student>() {  
        new Student(){ StudentID=1, StudentName="Steve", Age = 21 },  
        new Student(){ StudentID=2, StudentName="Bill", Age = 25 },  
        new Student(){ StudentID=3, StudentName="Ram", Age = 20 },  
        new Student(){ StudentID=4, StudentName="Ron", Age = 31 },  
        new Student(){ StudentID=5, StudentName="Rob", Age = 19 }  
    };  
  
    ViewData["students"] = studentList;  
    return View(studentList);  
}
```

Ví dụ



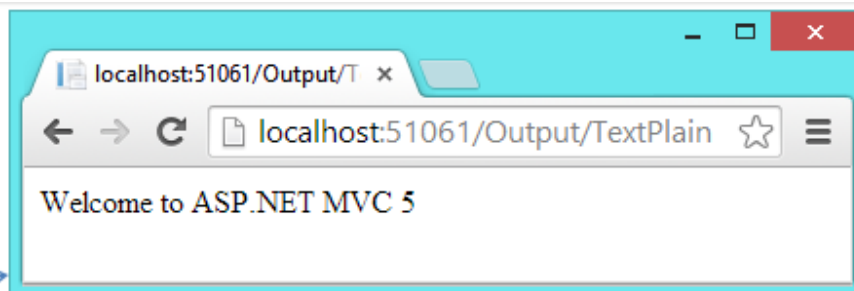
```
public ActionResult WithLayout()
{
    return View("Index");
}

public ActionResult WithoutLayout()
{
    return PartialView("Index");
}
```

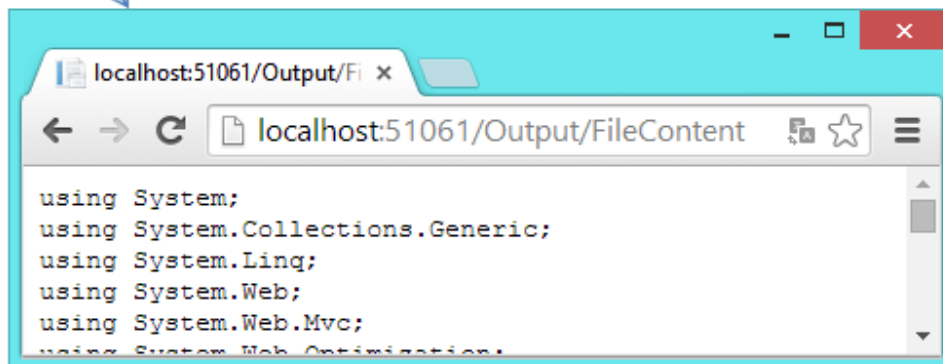


Ví dụ

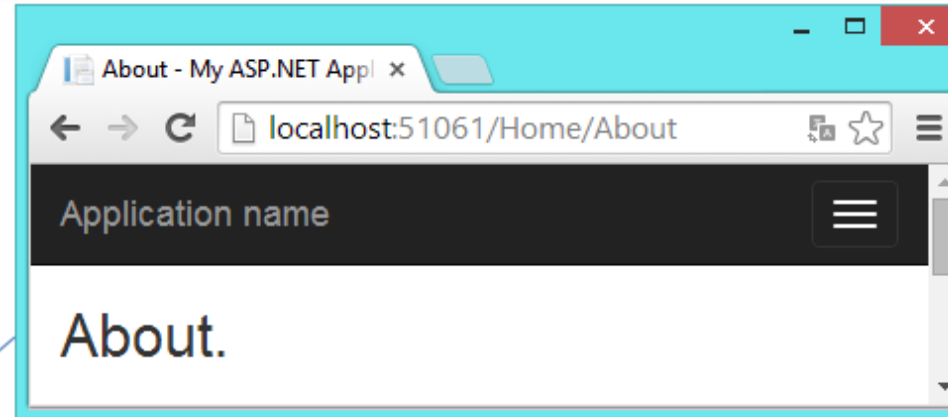
```
public ActionResult TextPlain()  
{  
    return Content("Welcome to ASP.NET MVC 5");  
}
```



```
public ActionResult FileContent()  
{  
    return File("~/Global.asax.cs", "text/plain");  
}
```

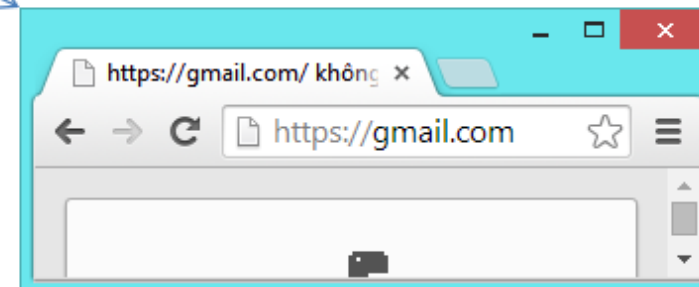


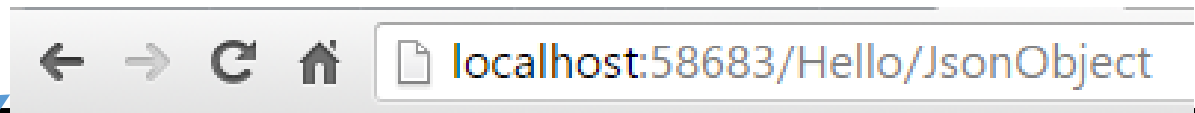
Ví dụ



```
public ActionResult RedirectToAction()  
{  
    return RedirectToAction("About", "Home");  
}
```

```
public ActionResult RedirectToUrl()  
{  
    return Redirect("http://gmail.com");  
}
```

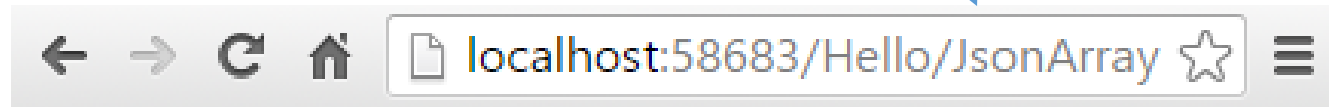




{"Name": "Minh", "Year": 1978}

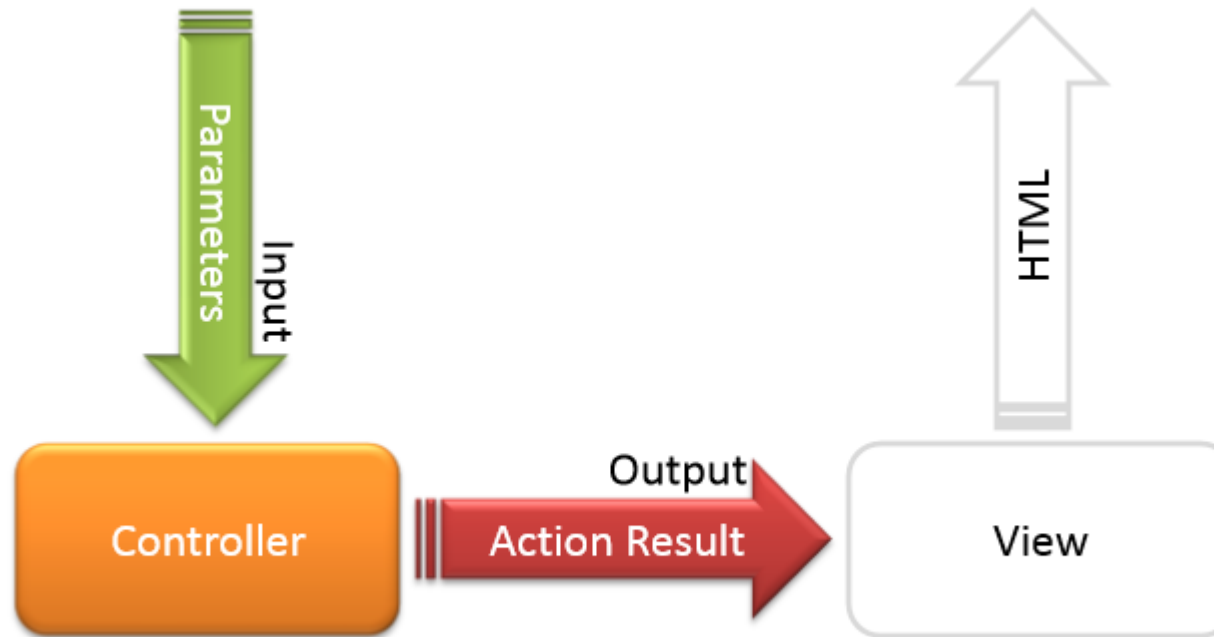
```
public ActionResult JsonObject()
{
    var data = new { Name="Minh", Year=1978 };
    return Json(data, JsonRequestBehavior.AllowGet);
}

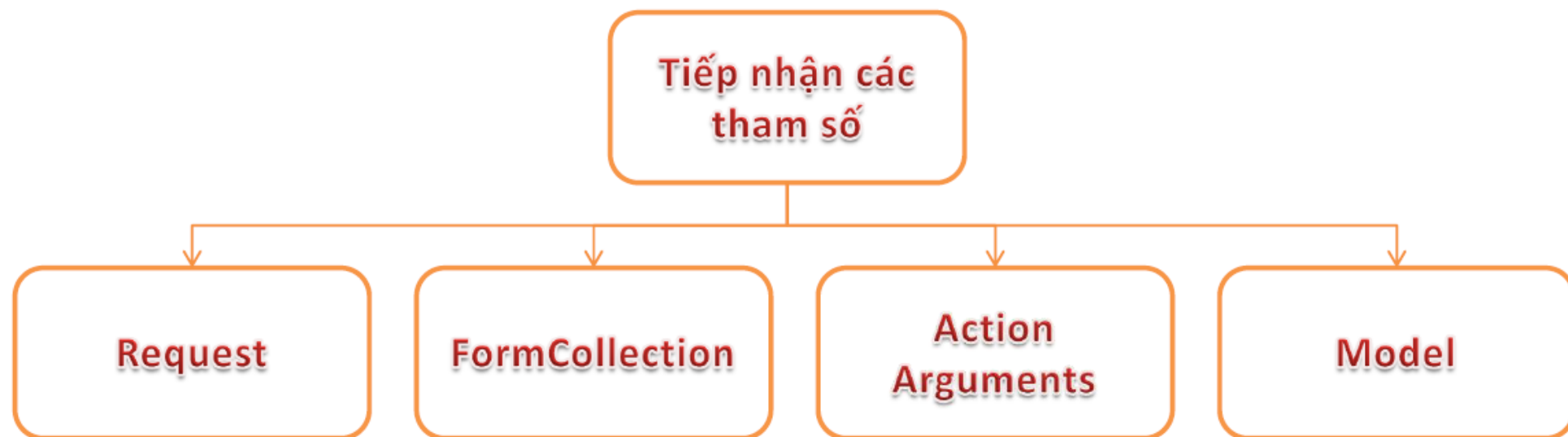
public ActionResult JsonArray()
{
    var data = new ArrayList();
    data.Add(new { Name = "Minh", Year = 1978 });
    data.Add(new { Name = "Hà", Year = 1979 });
    data.Add(new { Name = "Vân", Year = 1974 });
    return Json(data, JsonRequestBehavior.AllowGet);
}
```



[{"Name": "Minh", "Year": 1978}, {"Name": "Hà", "Year": 1979}, {"Name": "Vân", "Year": 1974}]

Action IO: Truyền/Nhận tham số Input/Output





❑ Trong MVC có 4 cách để nhận tham số yêu cầu

✎ Sử dụng đối tượng ngầm định **Request**

✎ Sử dụng **đối số của Action**

✎ Sử dụng tham số **FormCollection**

✎ Sử dụng **Model**

Sử dụng Request:

- ❑ Trong phương thức hành động bạn 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 và lưu dữ liệu vào ViewBag:

```
,  
[HttpPost]  
0 references  
public ActionResult Register1()  
{  
    string Id = Request["Id"];  
    string Name = Request["Name"];  
    double Marks = Convert.ToDouble(Request["Marks"]);  
    ViewBag.Id = Id;  
    ViewBag.Name = Name;  
    ViewBag.Marks = Marks;  
    return View(ViewBag);  
}
```

Sử dụng FormCollection:

- Tập hợp các tham số của Form vào đối số của FormCollection của Action, chỉ nhận được các trường Form.
- Ví dụ, nhận tham số của các trường Form và lưu vào ViewBag:

```
[HttpPost]
0 references
public ActionResult Register2(FormCollection field)
{
    string Id = field["Id"];
    string Name = field["Name"];
    double Marks = Convert.ToDouble(field["Marks"]);
    ViewBag.Id = Id;
    ViewBag.Name = Name;
    ViewBag.Marks = Marks;
    return View(ViewBag);
}
```

Sử dụng Action Arguments:

- Định nghĩa tham số cho Action để nhận tham số cùng tên.

- Ví dụ:

[HttpPost]

0 references

```
public ActionResult Register3(string Id, string Name, double Marks)
{
    ViewBag.Id = Id; ViewBag.Name = Name; ViewBag.Marks = Marks;
    return View(ViewBag);
}
```

```
<form action="/Student/Register" method="post">
    <input name="Id" />
    <input name="Name" />
    <input name="Marks" />
    <input type="submit" value="Register" />
</form>
```

Sử dụng Model:

- ❑ Tạo lớp Model chứa các 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 StudentInfo
{
    public string Id { get; set; }
    public string Name { get; set; }
    public double Marks { get; set; }
}
```

```
public ActionResult UseModel(StudentInfo model){...}
```

