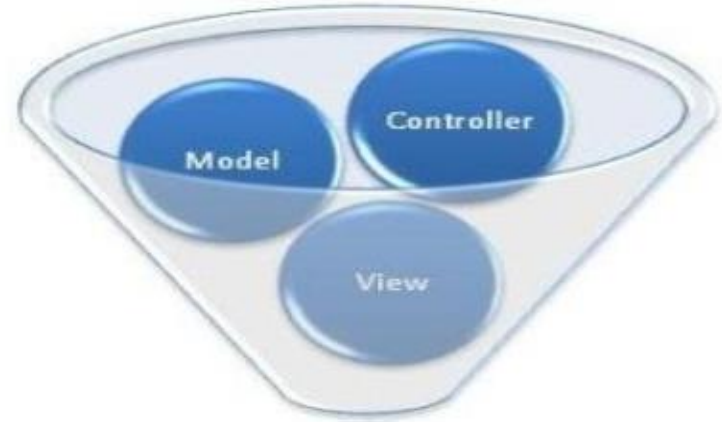
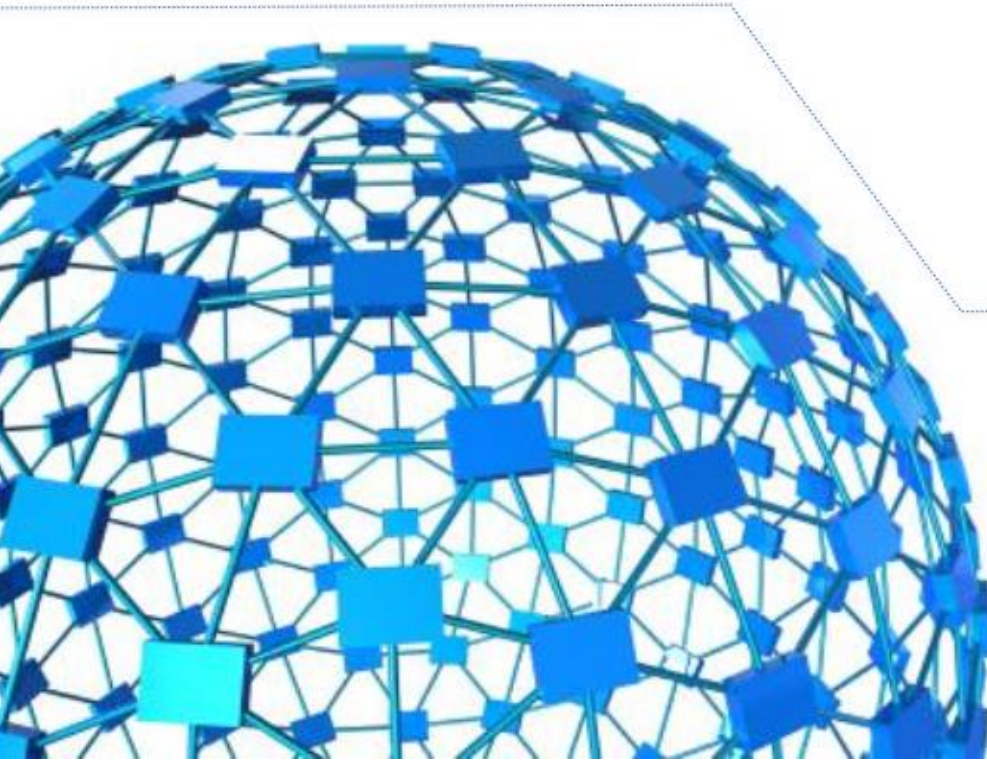


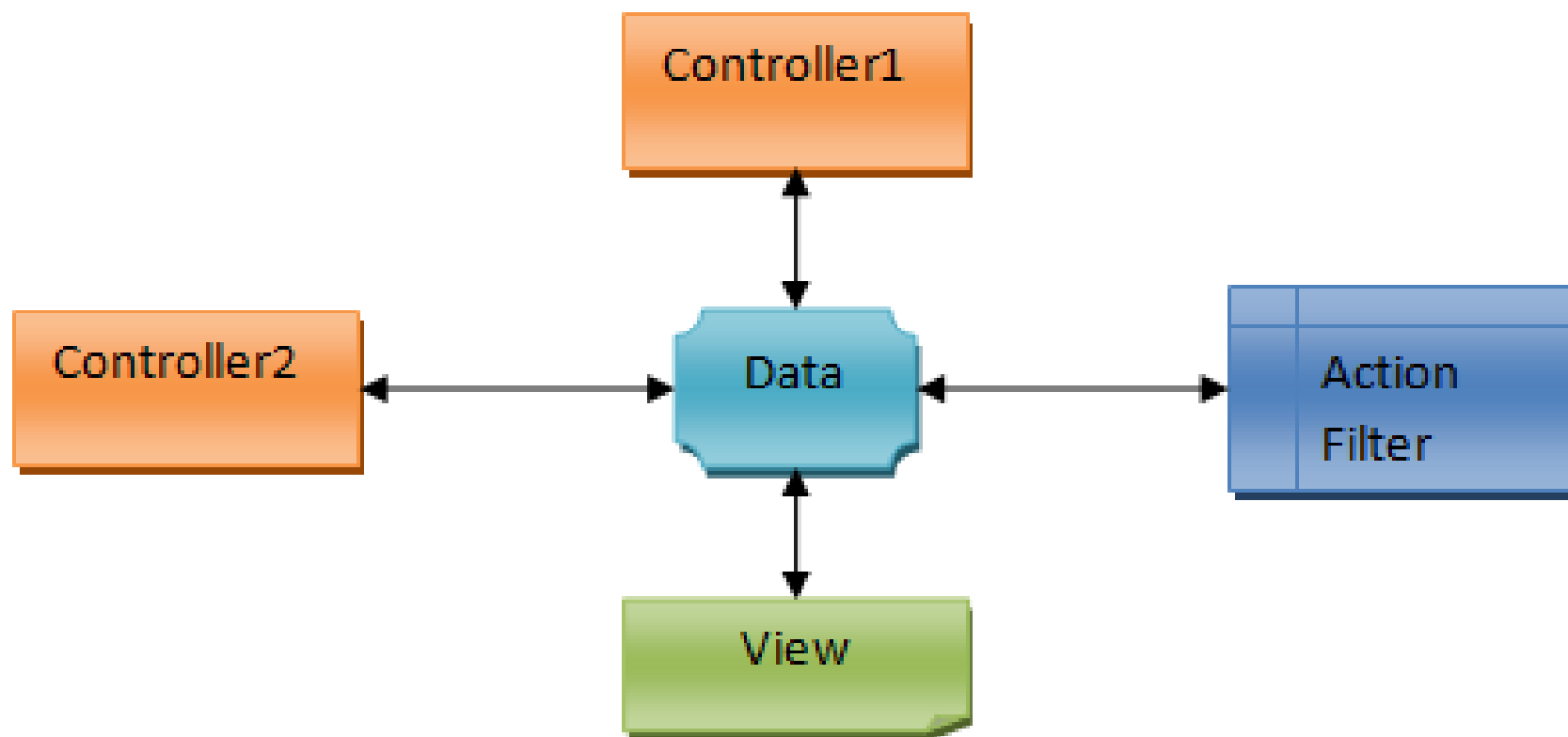
Microsoft®
ASP.net MVC **5**



DATA SHARING

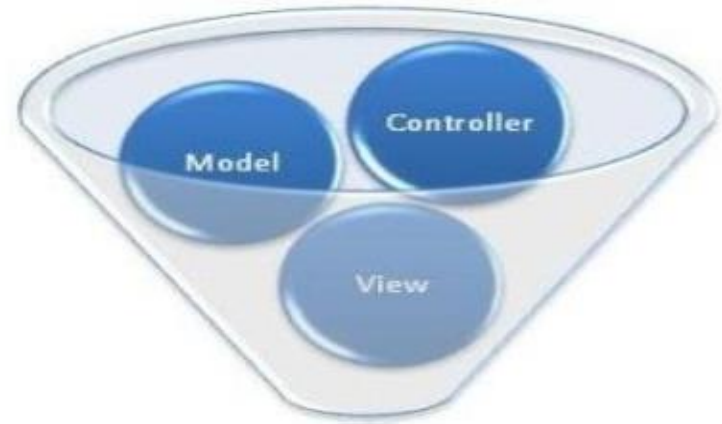


- ❑ Dữ liệu được tạo ra bởi thành phần này và được sử dụng bởi thành phần khác



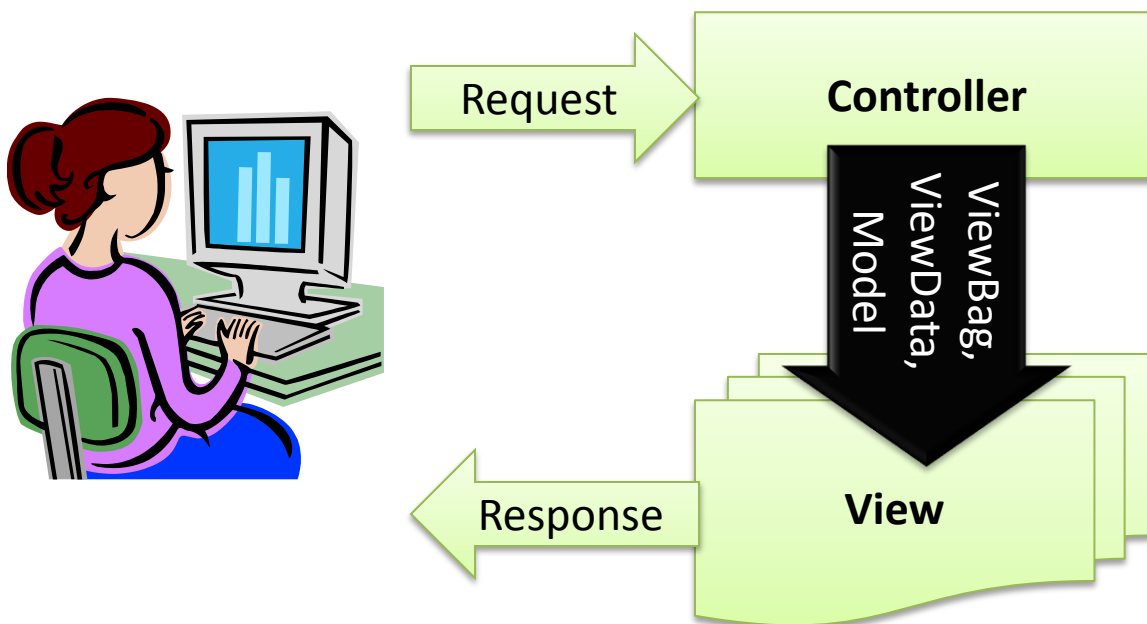
- ☐ ViewBag, Model
- ☐ Session
- ☐ Application
- ☐ Cookie
- ☐ Global.asax

Microsoft®
ASP.net MVC **5**



VIEWBAG, MODEL

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



- ❑ **ViewBag** và **ViewData** đã được sử dụng để truyền dữ liệu từ Controller sang View. View sẽ sử dụng để xây dựng giao diện phù hợp trả lại cho người dùng.

```
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>
```


Action

```
public ActionResult Detail()
{
    // Tạo đối tượng
    var model = new StudentInfo
    {
        Id = "SV001",
        Name = "Nguyễn Anh Tuấn",
        Marks = 9.5
    };
    // Truyền đối tượng model cho view
    return View(model);
}
```

Model

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

Khai báo kiểu dữ liệu của đối tượng Model để tận dụng intelligence (chấm xỏ)

View

```
@model Mvc5.Models.StudentInfo

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

Student Detail

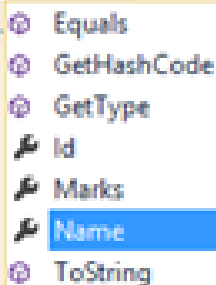
- Id: SV001
- Name: Nguyễn Anh Tuấn
- Marks: 9.5

```
@model Mvc5.Models.StudentInfo
@{
    ViewBag.Title = "Student Detail";
}
```

Chỉ thị **@model** được sử dụng để khai báo kiểu cho đối tượng **Model**

```
<h2>Student Detail</h2>
```

```
<ul>
    <li>Id: @Model.Id</li>
    <li>Name: @Model.</li>
    <li>Marks: @Model
</ul>
```



Equals
GetHashCode
GetType
Id
Marks
Name
ToString

Tận dụng được tính năng intelligence của công cụ VS2013 tránh được sai sót

❑ Chú ý: phân biệt @model và @Model

✂ @model: dùng để khai báo kiểu của @Model

✂ @Model: đối tượng chứa dữ liệu truyền từ Controller


```
public ActionResult Browse()
{
    ViewBag.Title = "List of your mails";

    Mail mail1 = new Mail
    {
        From = "receiver1@gmail.com",
        To = "sender1@gmail.com",
        Subject = "Mail subject 1",
        Body = "Mail content 1"
    };
    Mail mail2 = new Mail
    {
        From = "receiver2@gmail.com",
        To = "sender2@gmail.com",
        Subject = "Mail subject 2",
        Body = "Mail content 2"
    };

    List<Mail> mails = new List<Mail>();
    mails.Add(mail1);
    mails.Add(mail2);

    return View(mails);
}
```

```
@model IEnumerable<BasicDemo.Models.Mail>
```

```
<h2>@ViewBag.Title</h2>

<table border="1" style="width:100%">
    <tr style="background:yellow;">
        <th>FROM</th>
        <th>TO</th>
        <th>SUBJECT</th>
        <th>BODY</th>
        <th>&nbsp;</th>
    </tr>
    @foreach (var item in Model) {
        <tr>
            <td>@item.From</td>
            <td>@item.To</td>
            <td>@item.Subject</td>
            <td>@item.Body</td>
            <td><a href="Home/Single">Single</a></td>
        </tr>
    }
</table>
```

List of your mails









FROM	TO	SUBJECT	BODY	
receiver1@gmail.com	sender1@gmail.com	Mail subject 1	Mail content 1	Single
receiver2@gmail.com	sender2@gmail.com	Mail subject 2	Mail content 2	Single

Index - My ASP.NET Appli x

localhost:51114/Product/Index

Application name Home About Contact

Product List

<p>Aniseed Syrup</p>  <p>\$ 1012 Add To Cart</p>	<p>Chef Anton's Cajun Seasoning</p>  <p>\$ 1011 Add To Cart</p>	<p>Chef Anton's Gumbo Mix</p>  <p>\$ 1010 Add To Cart</p>	<p>Grandma's Boysenberry Spread</p>  <p>\$ 1009 Add To Cart</p>
<p>Uncle Bob's Organic Dried Pears</p>  <p>\$ 1008 Add To Cart</p>	<p>Northwoods Cranberry Sauce</p>  <p>\$ 1007 Add To Cart</p>	<p>Mishi Kobe Niku</p>  <p>\$ 1006 Add To Cart</p>	<p>Queso Cabrales</p>  <p>\$ 1005 Add To Cart</p>

Detail - My ASP.NET Appli x

← → ↻

localhost:51114/Product/Detail/1002

☆

≡


Application name

Home

About

Contact

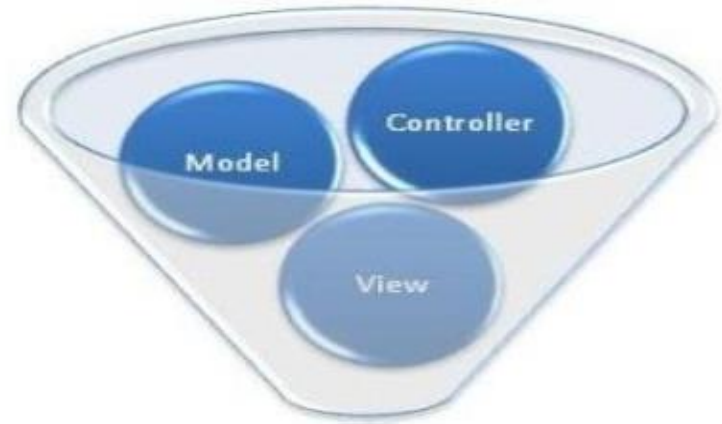
Product Detail



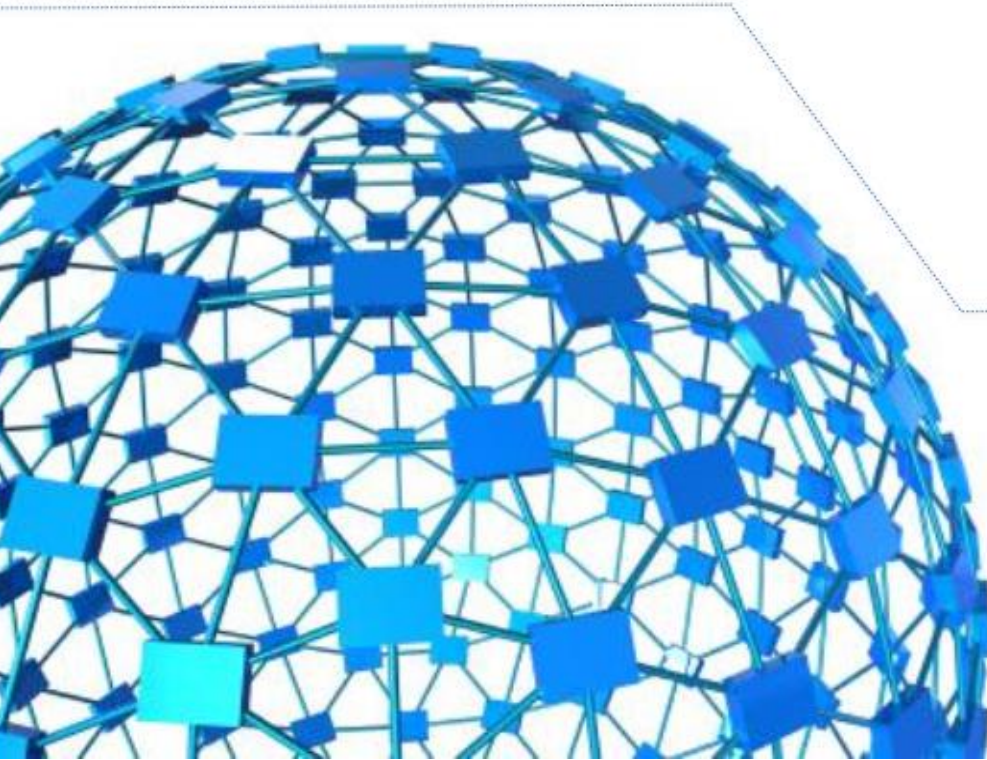
- Id: 1002
- Name: Chef Anton's Cajun Seasoning
- Unit Price: 1011
- Image: 1002.jpg

© 2014 - My ASP.NET Application

Microsoft®
ASP.net MVC **5**



Session





- ❑ Phiên làm việc (session) được tính từ lúc bắt đầu truy cập website cho đến khi đóng cửa sổ trình duyệt hoặc hết hạn sử dụng.
- ❑ Dữ liệu được lưu vào session sẽ được duy trì trong suốt phiên làm việc và được truy cập bởi bất kỳ thành phần nào hoạt động trong phiên làm việc đó.

□ Ứng dụng

- ✎ Duy trì giỏ hàng: giỏ hàng cần duy trì trong suốt quá trình tìm kiếm và chọn mua hàng hóa.
- ✎ Duy trì tài khoản đăng nhập: tài khoản đăng nhập cần được duy trì trong suốt phiên làm việc để khi cần có thể sử dụng.

□ Mã lấy đối tượng session

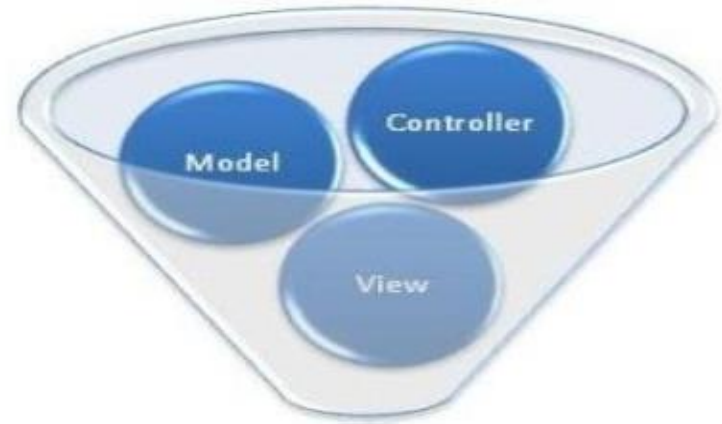
Nơi viết mã	Mã lệnh	Ví dụ
Controller	<code>Session</code>	<code>Session["A"]="Hello"</code>
View	<code>@Session</code>	<code>@Session["A"]</code>
Class bất kỳ	<code>HttpContext.Current.Session</code>	<code>HttpContext.Current.Session["A"]="Hello"</code>

Method/Property	Mô tả	Ví dụ
Add(Key, Value)	Thêm mới một đối tượng vào Session	Session.Add("Now", DateTime.Now)
[Key]=Value	Thêm mới hoặc thay thế một đối tượng	Session["Cart"] = new ShoppingCart()
Remove(Key)	Xóa đối tượng	Session.Remove("Cart")
Clear()	Xóa sạch các đối tượng trong Session	Session.Clear()
Abandon()	Hủy session	Session.Abandon()
SessionID	Mã của phiên làm việc	Var id = Session.SessionID

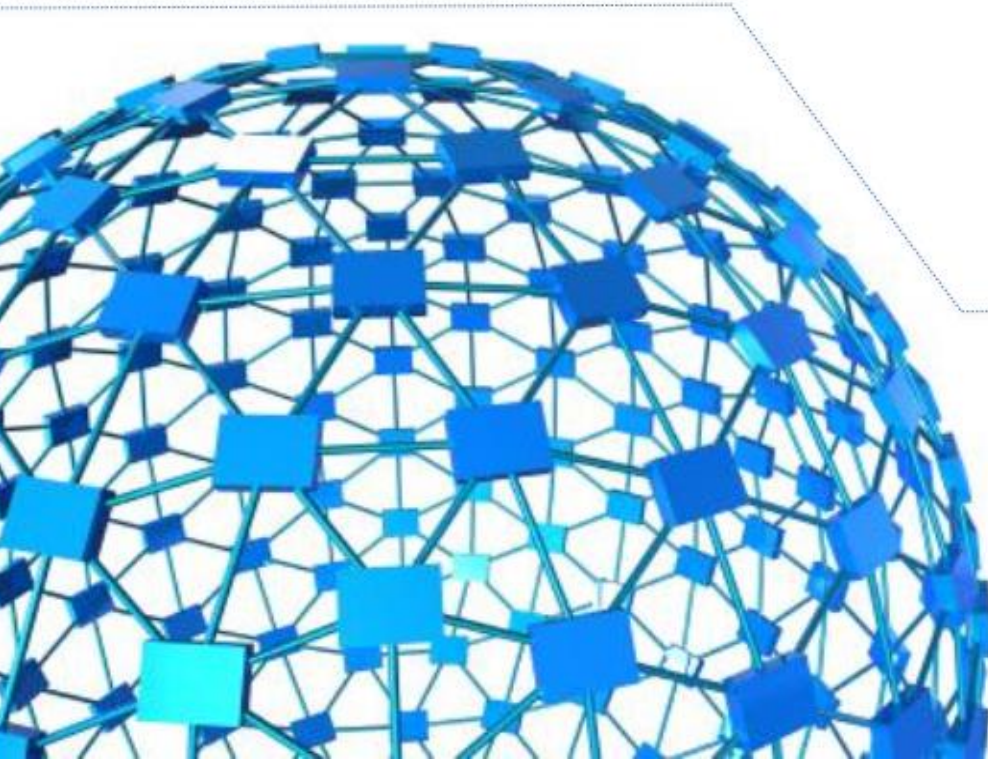
☐ CRUD: duy trì dữ liệu trong List

- ❑ Shopping Cart: duy trì danh sách hàng đã chọn trong session
- ❑ CRUD: duy trì dữ liệu trong List

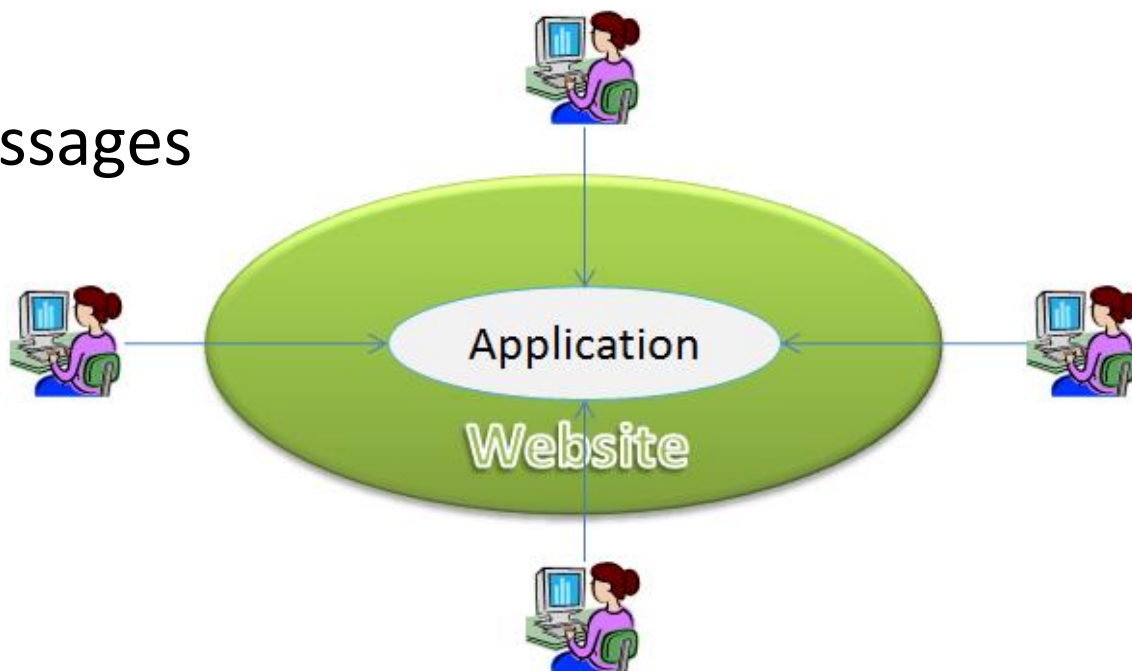
Microsoft®
ASP.net MVC **5**



APPLICATION



- ❑ Application là phạm vi chia sẻ dữ liệu trên toàn ứng dụng (tất cả mọi user có thể tạo và sử dụng)
- ❑ Ứng dụng: Cache dữ liệu cho toàn ứng dụng
 - ✗ Bộ đếm số người truy cập
 - ✗ Queue mails
 - ✗ Queue chat messages



❑ Truy xuất đối tượng Application

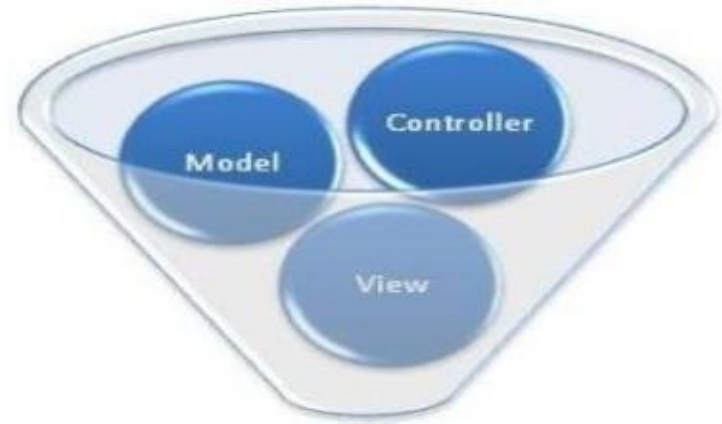
- ✍ Trong Controller: `HttpContext.Application`
- ✍ Trong View: `@HttpContext.Current.Application`
- ✍ Trong lớp bất kỳ: `HttpContext.Current.Application`

❑ Thao tác

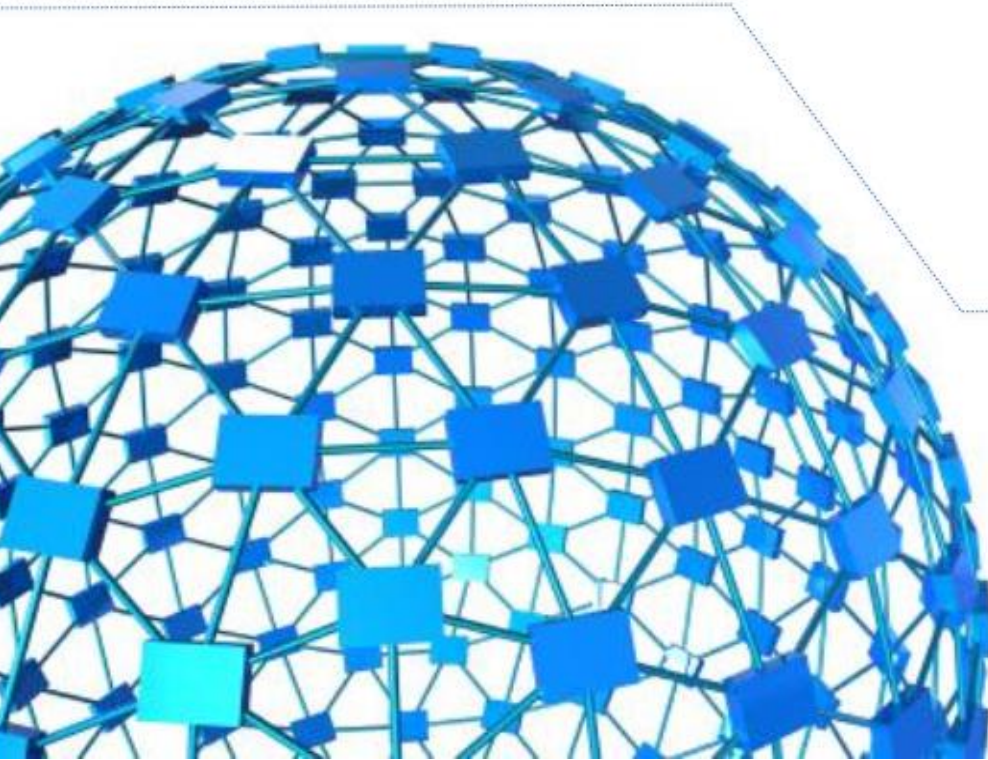
- ✍ Application. `Add` (name, value)
- ✍ Application[`name`] = <value>
- ✍ Application. `Remove` (name)
- ✍ Application. `Clear` ()
- ✍ Application. `Lock` ()
- ✍ Application. `Unlock` ()

- ☐ Bộ đến khách thăm web
- ☐ Chat
- ☐ Queue mails

Microsoft®
ASP.net MVC **5**



COOKIE



- ❑ Cookie là mẫu tin nhỏ được lưu trên máy client và truyền thông với server trong các request và response.
- ❑ Ứng dụng: chia sẻ dữ liệu giữa các trang trong website được truy cập từ máy
 - ✍ Tài khoản đăng nhập
 - ✍ Hàng hóa đã xem



❑ Lấy cookie từ client

✍ Trong controler: `Request.Cookies[name]`

✍ Trong View: `@Request.Cookies[name]`

❑ Gửi cookie về client

✍ `Response.Cookies.Add(cookie)`

❑ Tạo cookie

✍ `HttpCookie cookie = new HttpCookie(name, value)`

✓ Tạo cookie với tên và giá trị

✍ `HttpCookie cookie = new HttpCookie(name)`

✓ Tạo cookie với tên

☐ `cookie.Expires`

~~✎~~ Thời hạn của cookie.

☐ `cookie.Name`

~~✎~~ Tên của cookie

☐ `cookie.Value`

~~✎~~ Giá trị đơn của cookie

☐ `cookie.Values`

~~✎~~ Các giá trị của một cookie

☐ `cookie.Values.Add(Key, Value)`

~~✎~~ Thêm một giá trị vào cookie

☐ `cookie.Values[Key] = <Value>`

~~✎~~ Thêm hoặc thay thế 1 giá trị của cookie

☐ Duy trì hàng hóa đã xem

- ❑ Tập tin Global.asax chứa các điều khiển sự kiện quản lý vòng đời của application, session và request

```
public class MvcApplication : System.Web.HttpApplication
{
    // Chạy sau khi ứng dụng start thành công
    protected void Application_Start()...

    // Chạy trước khi ứng dụng shutdown
    protected void Application_End()...

    // Chạy sau khi có một phiên làm việc được tạo
    protected void Session_Start()...

    // Chạy trước trị 1 phiên làm việc hết hạn
    protected void Session_End()...

    // Chạy trước khi request chưa được phục vụ
    protected void Application_BeginRequest()...

    // Chạy trước khi request đã được phục vụ
    protected void Application_EndRequest()...
}
```

☐ HitCounter