



Платформа ASP.NET MVC

Магомед-Касумов Магомедрасул

rasuldev@gmail.com

Отдел математики и информатики ДНЦ РАН

Хронология технологий веб-разработки Microsoft

Период времени	Технология	Достоинства	Недостатки
Юрский период	Common Gateway Interface (CGI)*	Простота Гибкость Единственный выбор на то время	Выполняется вне веб-сервера, поэтому является ресурсоемкой (порождает по одному отдельному процессу операционной системы на каждый запрос) Является низкоуровневой
Бронзовый век	Microsoft Internet Database Connector (IDC)	Выполняется внутри веб-сервера	Является лишь оболочкой для запросов SQL и шаблонов для формирования результирующего набора
1996 г.	Active Server Pages (ASP)	Общее назначение	Интерпретируется во время выполнения Приводит к появлению "спагетти-кода"
2002/2003 гг.	ASP.NET 1.0/1.1	Компилируемый пользовательский интерфейс с поддержкой состояния Обширная инфраструктура Стимулирует объектно-ориентированное программирование	Требует большой ширины пропускания Порождает корявую HTML-разметку Не является тестируемой
2005 г.	ASP.NET 2.0		
2007 г.	ASP.NET AJAX		
2008 г.	ASP.NET 3.5		

ASP.NET WebForms

Ключевая особенность:

скрыть как протокол HTTP, так и язык HTML посредством предоставления готовых элементов управления

Недостатки WebForms

- Механизм поддержки состояния (ViewState) часто требует передачи огромных блоков данных между клиентом и сервером
- Ограниченный контроль за HTML-разметкой
- Недостаточно четкое разделение ответственностей
- Невозможность тестирования

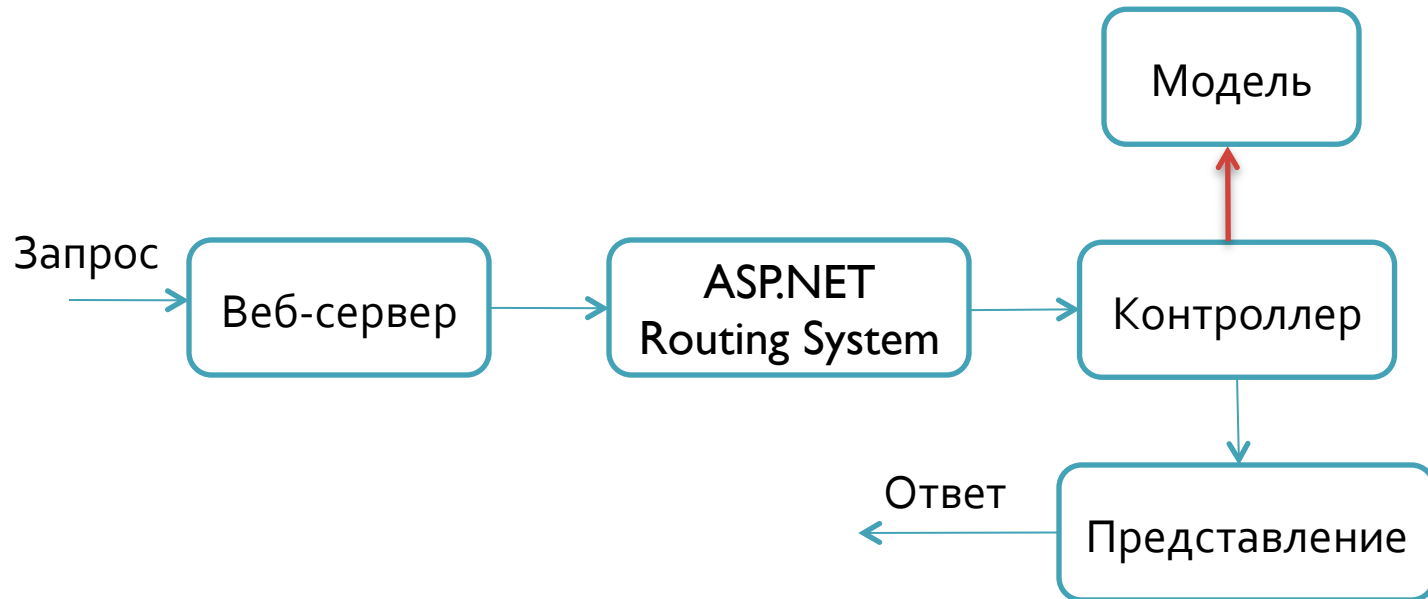
Ключевые преимущества ASP.NET MVC

- Использование модели MVC
- Расширяемость
- Тестируемость
- Полный контроль над HTML
- Мощная новая система маршрутизации
- Возможность использования компонентов платформы ASP.NET WebForms
- ASP.NET MVC имеет открытый исходный код

План

- Представления (Views, Views Engines)
- Контроллеры и фильтры
- Работа с данными, привязка модели
- Формы, ввод данных, валидация

Схема обработки запроса



Представления

- Движок представления Razor
- Master pages
- Helpers
- Partial Views
- Child Actions

View Engines

- WebForms (ASPX)
- Razor
- Spark
- NHaml
- NVelocity

Особенности Razor

- Краткость и выразительность
- Легкость в освоении

Сравнение

Web Forms

```
<ul>
  <% for (int i = 0; i < 10; i++) { %>
    <li><% =i %></li>
  <% } %>
</ul>
```

PHP

```
<ul>
  <?php
    for ($i = 0; $i < 10; $i++) {
      echo("<li>$i</li>");
    }
  ?>
</ul>
```

Razor

```
<ul>
  @for (int i = 0; i < 10; i++) {
    <li>@i</li>
  }
</ul>
```

Примеры использования

if-else

```
@if (Model.IsLoggedIn) {  
    <p>You're logged in!</p>  
} else {  
    <p>You're not logged in.</p>  
}
```

foreach

```
<ul id="products">  
    @foreach (var p in products) {  
        <li>@p.Name ($@p.Price)</li>  
    }  
</ul>
```

Многострочный код

```
@{  
    int n = 1;  
    string message = "Number is " + n;  
}  
<p>@message</p>
```

Блок кода

```
@("Number is " + n)
```

Смешение кода и разметки

HTML Block

```
@{
    var name = "John Doe";
    <div>
        Your name: @name
    </div>
}
```

Text Block

```
@{
    var name = "John Doe";
    <text>
        Your name: @name
    </text>
}
```

Text Block

```
@{
    var name = "John Doe";
    @: Your name: @name
}
```

Master pages

_Layout.cshtml

```
<!DOCTYPE html>
<html>
<head>
    <title>@ViewBag.Title</title>
    @RenderSection("head", false)
</head>

<body>
    <div id="left-menu">
        @RenderSection("menu", true)
    </div>

    <div id="body">
        @RenderBody()
    </div>

    <div id="footer">
        @RenderSection("footer", false)
    </div>
</body>
</html>
```

Index.cshtml

```
@{
    ViewBag.Title = "Index";
}

<h2>Index</h2>
<div>
    Here we can place some content.
</div>
<div>
    We can use code here too.
    Current datetime: @DateTime.Now
</div>

@section menu {
    <ul>
        @for(int i=1; i<6; i++)
        {
            <li>Item @i</li>
        }
    </ul>
}

@section footer {
    <p>Some footer text</p>
}
```

Helpers

- `@Html.TextBox("Name")`
- `@Html.TextBoxFor(m => m.Name)`

```
<input id="Name" name="Name" type="text" value="John" />
```

`ModelState["Name"]`

`ViewData["Name"]`

`ViewBag.Name`

`Model.Name`

Helpers

- `Html.CheckBox("myCheckbox", false)`

`<input name="myCheckbox" type="hidden" value="false" />`

- `Html.TextArea("myTextarea", "val", 5, 20, null)`

`<textarea cols="20" id="myTextarea" name="myTextarea" rows="5">val</textarea>`

- `Html.Hidden("myHidden", "val")`

`<input id="myHidden" name="myHidden" type="hidden" value="val" />`

Helpers

- `Html.DropDownList("myList", new SelectList(new [] {"A", "B"}), "Choose")`

```
<select id="myList" name="myList">  
<option value="">Choose</option>  
<option>A</option>  
<option>B</option>  
</select>
```

Templated view helpers

- `@Html.Display("Name")`
- `@Html.Editor("Name")`
- `@Html.Label("Name")`

Partial Views, Child Actions

- `@Html.Partial("ViewName")`
- `@Html.Action("ActionName")`

Контроллеры

- Действия (action methods)
- Результаты (action results)
- Фильтры (filters)

Получение данных

- Request, Response, RouteData, HttpContext, Server, TempData
- Model Binding

Action Results

- ActionResult
- ViewResult (`return View("ViewName")`)
- RedirectToRouteResult
(`return RedirectToRoute(new {action=""}),
RedirectToAction("action","controller")`)
- RedirectResult (`return Redirect("url")`)
- ContentResult (`return Content()`)
- JsonResult (`return Json(object)`)

Передача данных представлению

- ViewBag
- ViewData
- Model

Фильтры

```
public class MainController : Controller
{
    public ActionResult Index()
    {
        if (!Request.IsAuthenticated)
        {
            FormsAuthentication.RedirectToLoginPage();
        }
        // ...
    }
    public ActionResult Create()
    {
        if (!Request.IsAuthenticated)
        {
            FormsAuthentication.RedirectToLoginPage();
        }
        // ...
    }
    public ActionResult Edit(int productId)
    {
        if (!Request.IsAuthenticated)
        {
            FormsAuthentication.RedirectToLoginPage();
        }
        // ...
    }
    // ...
}
```


Фильтры

```
[Authorize]
public class MainController : Controller
{
    public ActionResult Index()
    {
        // ...
    }
    public ActionResult Create()
    {
        // ...
    }
    public ActionResult Edit(int productId)
    {
        // ...
    }
    // ...
}
```

Фильтры

- Authorization
- Action
- Result
- Exception

Model Binding

- Request.Form
- RouteData.Values
- Request.QueryString
- Request.Files

DefaultModelBinder

```
class Person {  
    public string Name {get; set;}  
    public string LastName {get; set;}  
}  
public ActionResult Index(Person person, int number)  
{ ... }
```

<input type='text' name='LastName' />



Request.Form["LastName"]

<input type='text' name='number' />



Request.Form["number"]

Массивы

@Html.TextBox("movies")

@Html.TextBox("movies")

@Html.TextBox("movies")

<input id="movies" name="movies" type="text" value="" />

<input id="movies" name="movies" type="text" value="" />

<input id="movies" name="movies" type="text" value="" />

public ActionResult Movies(**List<string> movies**) {

...

}

Ввод данных и валидация

- Вручную
- Через атрибуты

Валидация вручную

```
public ActionResult Index(Person person) {  
    if( person.Age<0 ) {  
        ModelState.AddModelError("age", "Age can't be less zero");  
    }  
}
```

@Html.ValidationSummary()

@Html.ValidationFor

@Html.Validation("age")

Валидация с помощью атрибутов

- [Compare("MyOtherProperty")]
- [Range(10, 20)]
- [RegularExpression("pattern")]
- [Required]
- [StringLength(10)]
- [<Attr>(ErrorMessage="Error")]

Валидация с помощью атрибутов

```
public class MustBeTrueAttribute : ValidationAttribute {  
    public override bool IsValid(object value) {  
        return value is bool && (bool)value;  
    }  
}
```