ЛЕКЦИЯ

ASP .Net MVC. 1 часть

Лектор Крамар Ю.М.

Содержание

1. Введение в ASP.NET MVC
2. Контроллеры
3. Методы действия
4. Результаты действий методов контроллера

ВВЕДЕНИЕ В ASP.NET MVC

Введение в ASP.NET MVC

ASP.NET MVC — платформа для создания сайтов и веб-приложений с использованием паттерна MVC (model - view - controller).

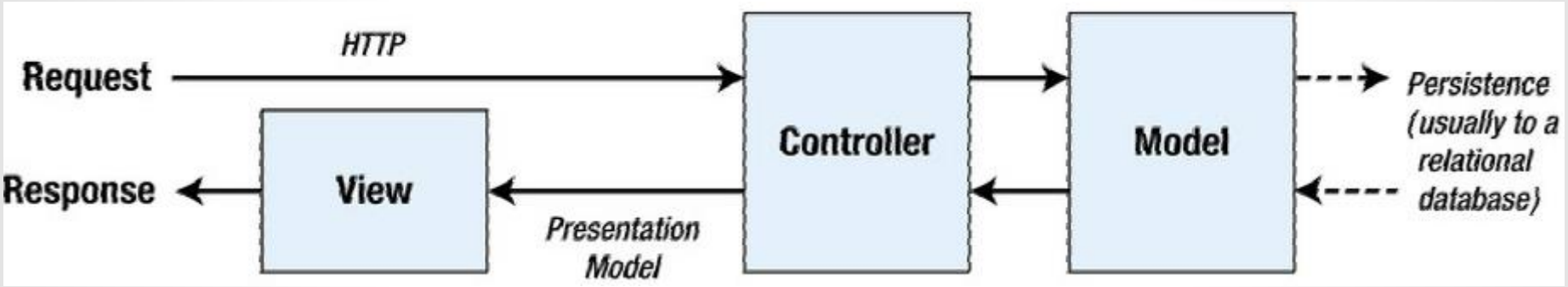
Шаблон MVC, лежащий в основе новой платформы, подразумевает взаимодействие трех компонентов:

контроллера (controller),

модели (model),

представления (view).

Введение в ASP.NET MVC



Паттерн ASP.NET MVC

Введение в ASP.NET MVC

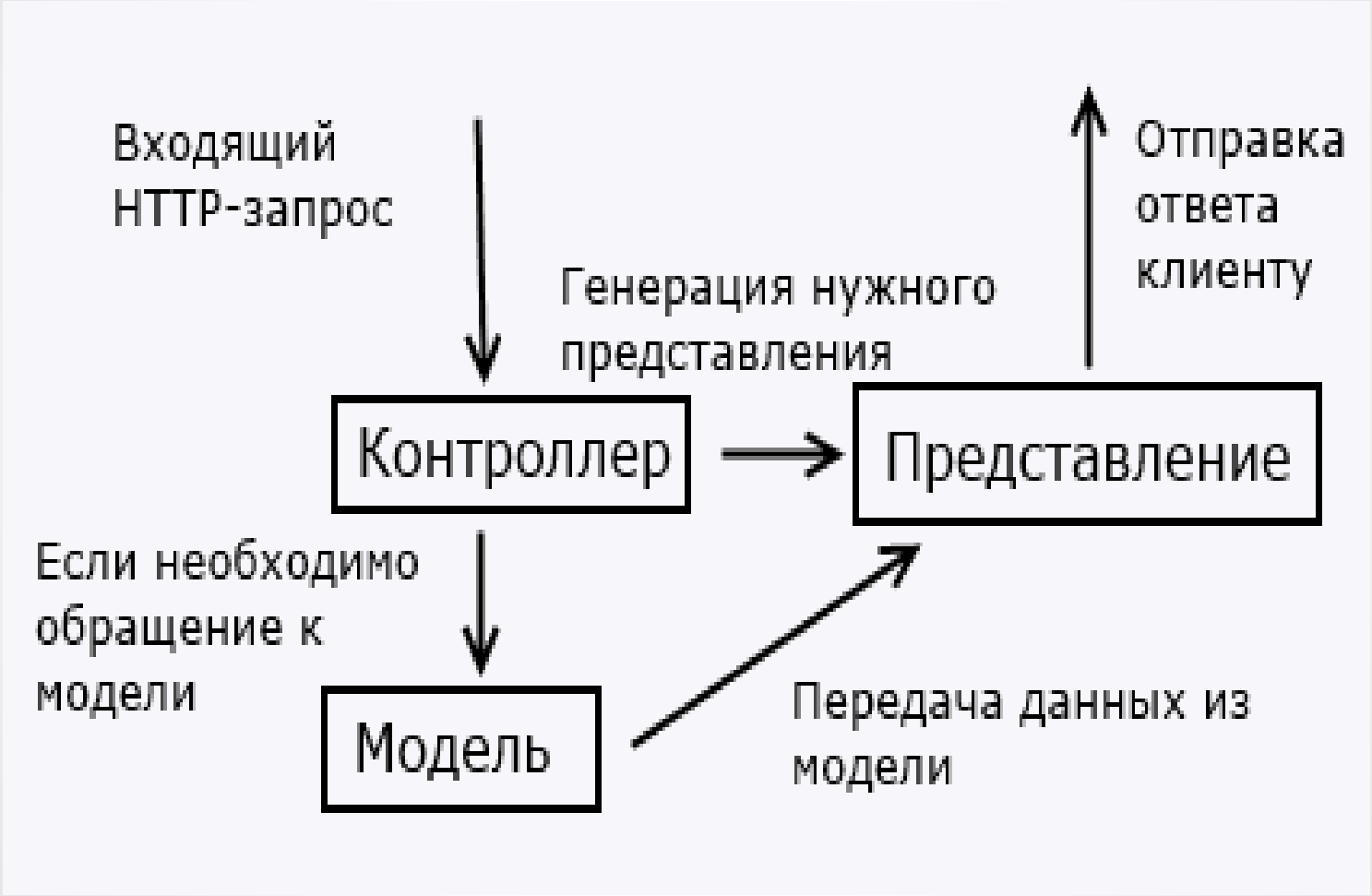


Схема взаимодействия компонентов ASP.NET MVC

Введение в ASP.NET MVC

- **Контроллер** (controller) представляет класс, обеспечивающий связь между моделью и представлением. Получая вводимые пользователем данные, контроллер, исходя из внутренней логики, при необходимости обращается к модели и генерирует соответствующее представление.

- **Представление** (view) – это собственно визуальная часть или пользовательский интерфейс приложения – например, html-страница, через которую пользователь, зашедший на сайт, взаимодействует с веб-приложением.

- **Модель** (model) представляет набор классов, описывающих логику используемых данных.

Введение в ASP.NET MVC

- **Разделение ответственности**: В MVC приложение состоит из трех частей: контроллера, представления и модели, каждая из которых выполняет свои специфичные функции..
- **Тестирование**: В силу разделения ответственности приложения mvc обладают лучшей тестируемостью. Можно тестировать отдельные компоненты независимо друг от друга.
- **Соответствие протоколу HTTP**: Приложения MVC в отличие от веб-форм не поддерживают объекты состояния (ViewState).
- **Гибкость**: Можно настраивать различные компоненты платформы: изменять какие-либо части конвейера работы MVC или адаптировать его к своим нуждам и потребностям.

Введение в ASP.NET MVC

ASP .Net Web Application
ASP .Net Core Web Application

Шаблоны Web

- Empty
- Web Forms
- MVC
- Web API
- Single Page Application

- Azure API\Mobile App

Движки MVC

- Razor
- ASPX

Введение в ASP.NET MVC

ASP .Net MVC 5

Особенности

- Аутентификация с помощью **ASP.NET Identity**
- Использование адаптивной верстки с помощью **Bootstrap**
- Фильтры аутентификации
- Настраиваемый роутинг

Введение в ASP.NET MVC

Структура проекта MVC

- **App_Data**
- **App_Start**
- **Content**
- **Controllers**
- **Models**
- **Views**
- **Images**
- **Scripts**
- Web.config
- Global.asax
- Packages.config

КОНТРОЛЛЕРЫ

Контроллеры

Контроллер – компонент в архитектуре MVC, получающий ввод пользователя, обрабатывающий его, отправляющий результат обработки (например, в представление).

Основные свойства контроллеров:

- Хранятся в папке Controllers
- Наследуют класс System.Web.Mvc.Controller
- Имя класса заканчивается суффиксом Controller
- Обращение к контроллеру в строке адреса: URL/имя_контроллера
- Обращение к методу контроллера с передачей в него аргументов:

URL /имя_контроллера/имя_метода/значение_аргумента (если параметр только один и его имя – id)

Контроллеры

Объект **Controller.** Контекст запроса – RequestContext
ControllerContext – контекст контроллера
HttpContext – связанные с HTTP сведения о HTTP-запросе
Profile
Request
Response
Server
Session
RouteData – данные маршрута для текущего запроса
User
Url – объект вспомогательного метода URL-адреса, используемый для создания URL-адресов с помощью маршрутизации
TempData – динамический словарь временных данных
ViewData – словарь данных предсталения
ViewBag – динамический словарь данных представления

МЕТОДЫ ДЕЙСТВИЙ КОНТРОЛЛЕРОВ

Методы действий контроллера

Методы действий (action methods) представляют такие методы контроллера, которые обрабатывают запросы по определенному URL.

Для каждого типа запроса существует свой атрибут: [HttpGet], [HttpPost], [HttpDelete] или [HttpPut].

Методы действий всегда имеют модификатор public.

Закрытые методы – служебные методы контроллера

```
[HttpPost]
public string Buy(Purchase purchase)
{
    purchase.Date = getToday();
    …
}
private DateTime getToday()
{
    return DateTime.Now;
}
```

Методы действий контроллера

Передача параметров [HttpGet] – методу действия через адресную строку:

My2/Square?a=10&h=3

```
public class My2Controller : Controller{
…
[HttpGet]
public string Square(int a, int h)
{
    double s = a*h/2;
    return "<h2>Площадь треугольника с основанием " + a +
            " и высотой " + h + " равна " + s + "</h2>";
// или int a = Int32.Parse(Request.Params["a"]);
}
```

Если параметры – по умолчанию, то возможно обращение:

My2/Square?h=10

My2/Square

РЕЗУЛЬТАТЫ ДЕЙСТВИЙ МЕТОДОВ КОНТРОЛЛЕРА

Результаты действий

На стороне сервера метод контроллера, получая параметры, обрабатывает их и формирует некоторый ответ в виде результата действия. Им является объект класса, производного от **ActionResult**. ActionResult представляет собой абстрактный класс, в котором определен один метод ExecuteResult, переопределяемый в классах-наследниках

```
public abstract class ActionResult
{
    public abstract void ExecuteResult(ControllerContext context);
}
```

Результаты действий

Создание результата действия:

```
public class HtmlResult : ActionResult{
        private string htmlCode;

        public HtmlResult(string html)
        {
            htmlCode = html;
        }

        public override void ExecuteResult(ControllerContext
context){
            string fullHtmlCode = "<!DOCTYPE html><html><head>";
            fullHtmlCode += "<title>Главная страница</title>";
            fullHtmlCode += "<meta charset=utf-8 />";
            fullHtmlCode += "</head> <body>";
            fullHtmlCode += htmlCode;
            fullHtmlCode += "</body></html>";
            context.HttpContext.Response.Write(fullHtmlCode);
        }
    }
```

Результаты действий

Запрос результата действия :

```
public class MyController : IController
{
public ActionResult GetHtml()
{
    return new HtmlResult("<h2>Привет мир!</h2>");
}
}
```

Обращение к контроллеру:

URL/My/GetHtml

Результаты действий

Фрейморк **ASP.NET MVC** содержит набор классов результатов действий, которые охватывают большинство возможных вариантов:

- ContentResult
- EmptyResult
- FileResult и его производные
- HttpStatusCodeResult и его производные
- JavaScriptResult
- JsonResult
- PartialViewResult
- RedirectResult
- RedirectToRouteResult
- **ViewResult**

Результаты действий

Результат метода действия **ViewResult**

```
public class HomeController : Controller
{
    public ViewResult SomeMethod1()
    {
        return View();
    }
    public ViewResult SomeMethod2()
    {
        return View(“Index”);
    }


    public ViewResult SomeMethod3()
    {
        return View("~/Views/Some/Index.cshtml");
    }
}
```

Результаты действий

Результат метода действия **RedirectResult**

```
public class HomeController : Controller
{
    public RedirectResult SomeMethod()
{
    return Redirect("/Home/Index");
}

public RedirectResult SomeMethod()
{
    return RedirectPermanent("/Home/Index");
}
}
```

Результаты действий

Результат метода действия **RedirectToRouteResult**

```
public class HomeController : Controller
{
  public RedirectToRouteResult SomeMethod()
{
    return RedirectToRoute(new { controller="Home",
action="Index"});
}

public RedirectToRouteResult SomeMethod()
{
    return RedirectToAction("Square", "Home", new { a=10,h=12});
}

}
```

Результаты действий

Результат метода действия **HttpStatusCodeResult**

```
public class HomeController : Controller
{
public ActionResult Check1(int? age)
        {
            if ((age == null) || (age < 21))
            {
                return new HttpUnauthorizedResult(); // 401
            }
            else
            {
                return new HttpStatusCodeResult(404);
            }

        }
}
```

Результаты действий

Результат метода действия **FileResult**

```
public class HomeController : Controller
{
public FileResult GetFile()
        {
             // Путь к файлу
             string file_path = Server.MapPath
("~/Images/Winnie_the_Pooh.jpg");
            // Тип файла - content-type
            string file_type = "application/jpg";
            // Имя файла
             string file_name = "Winnie_the_Pooh.jpg";
            return File(file_path, file_type, file_name);
        }}
```

ASP .Net MVC. 1 часть

Ресурсы:

http://smarly.net/pro-asp-net-mvc-4

http://www.metanit.com/sharp/mvc5