



Mentoring: ASP.NET MVC

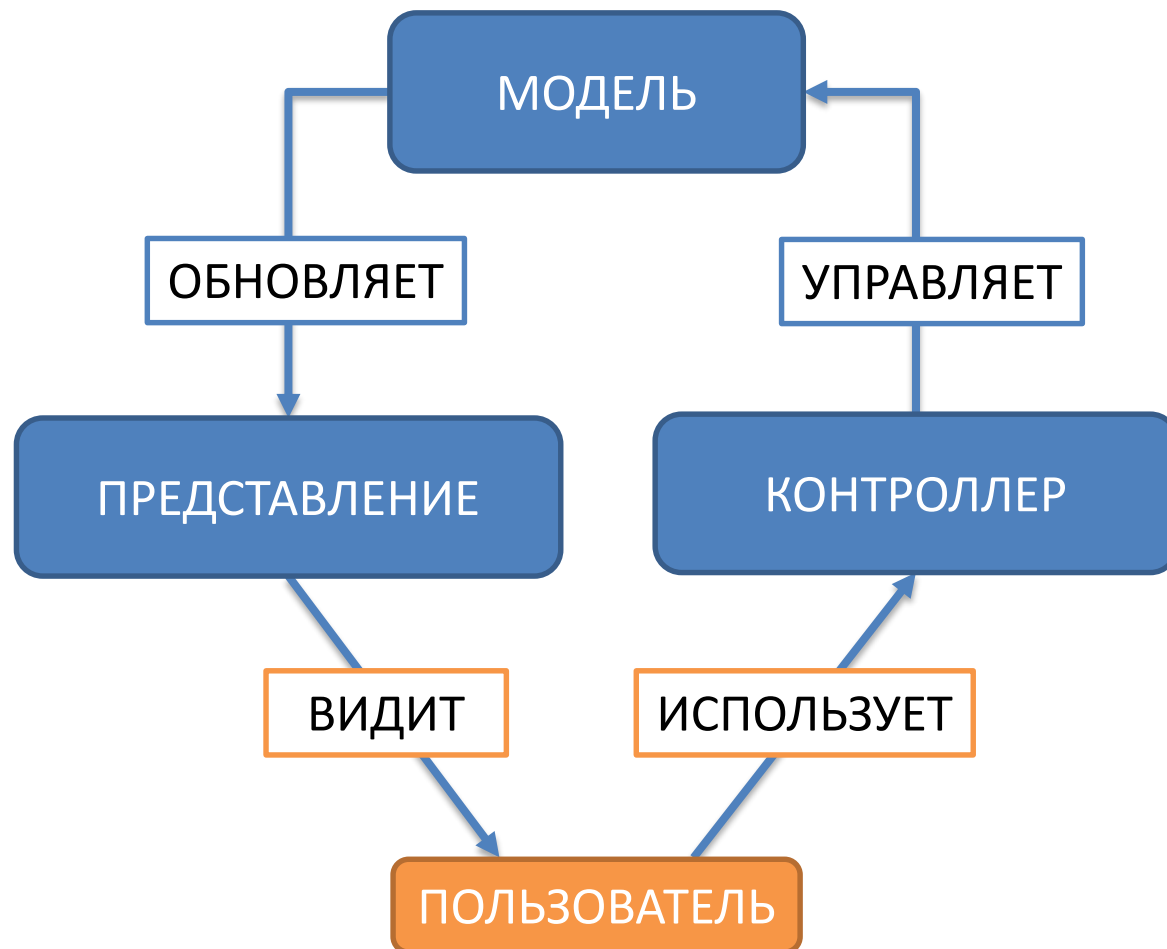
ASP.NET MVC Principles

- Шаблон проектирования MVC
- Структура приложения ASP.NET MVC
- Интеграция в трёхслойную архитектуру

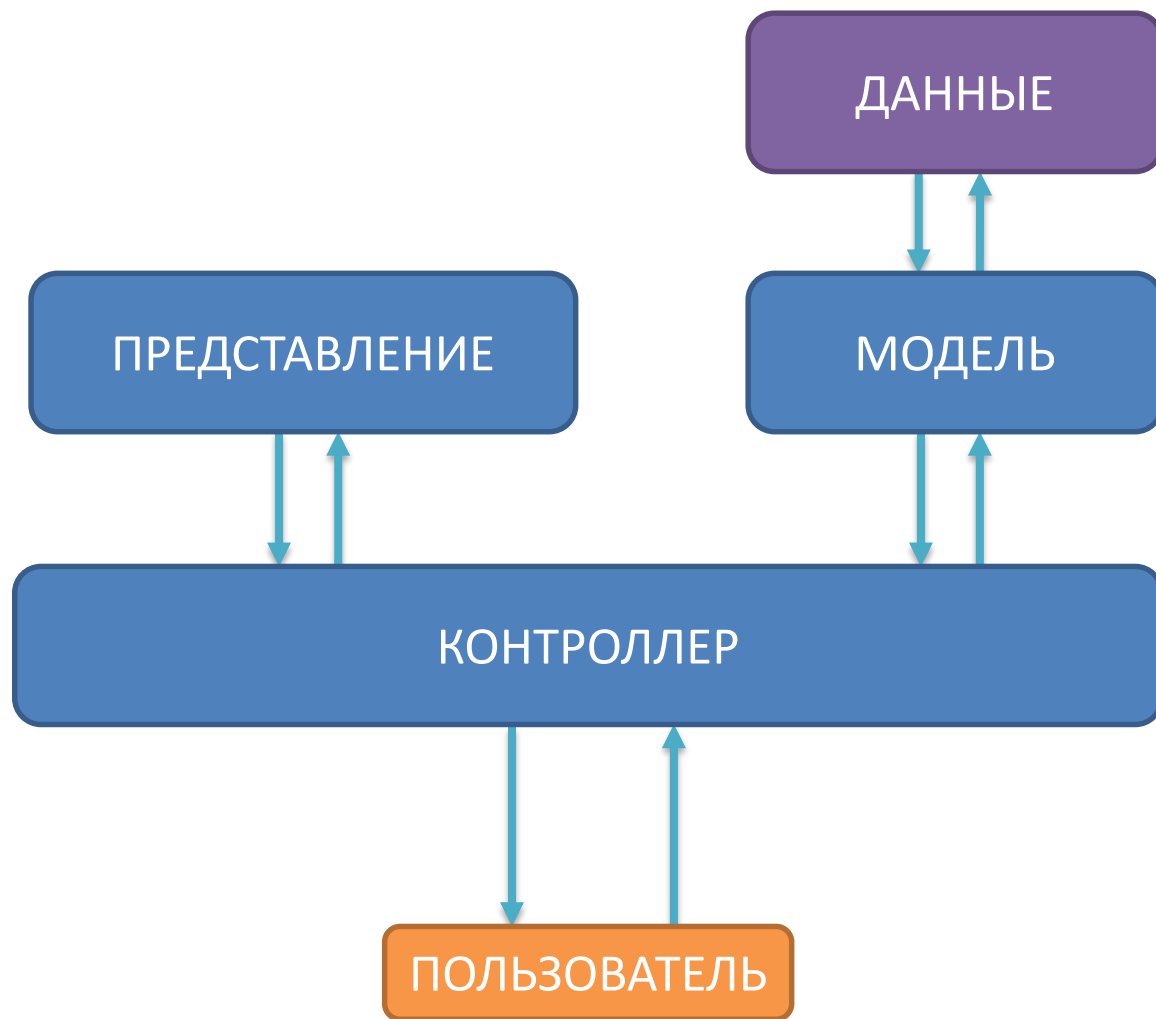
Что такое MVC?

- Model (Модель)
 - Данные приложения, методы для работы с ними, механизмы для их получения и актуализации.
- View (Представление)
 - Визуализация данных модели.
- Controller (Контроллер)
 - Обработка пользовательского ввода, выбор модели и представления.

Принцип функционирования MVC



Принцип взаимодействия компонентов



Эволюция ASP.NET

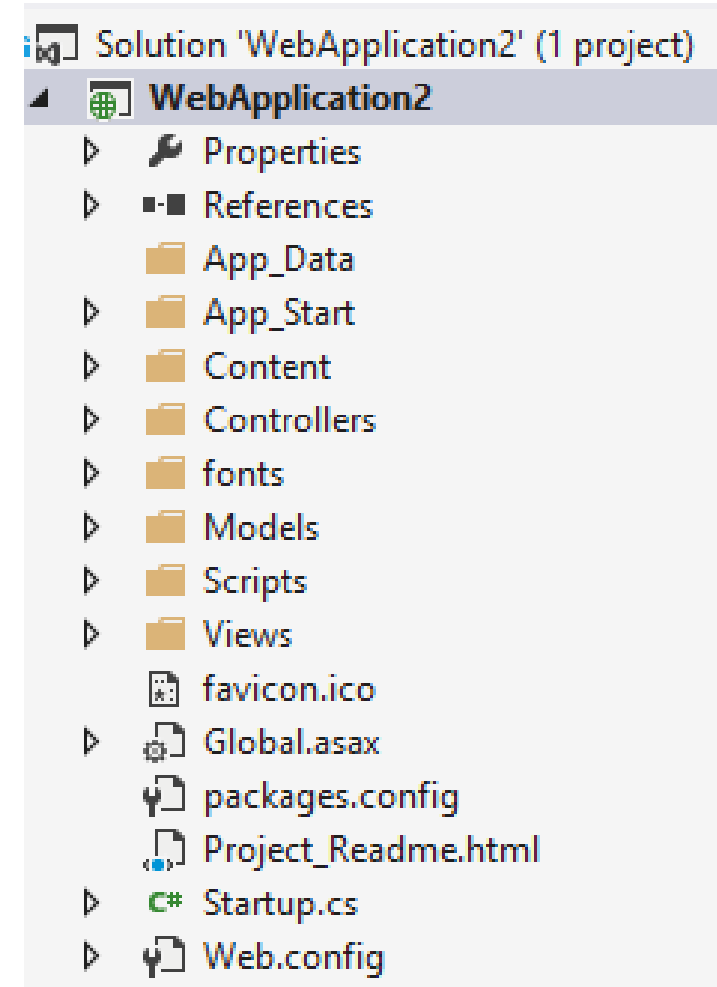
- 1996 — ASP (Active Server Pages)
- 2002 — ASP.NET Web Forms
- 2007 — ASP.NET AJAX
- **2008 — ASP.NET MVC**
- 2013 — ASP.NET Web API
- 2014(?) — ASP.NET vNext

Сильные стороны ASP.NET MVC

- Архитектурный шаблон MVC
- Расширяемость
- Жёсткий контроль над HTML и HTTP
- Тестируемость
- Маршрутизация
- Платформа ASP.NET на базе .NET
- Открытый исходный код

Структура приложения ASP.NET MVC

- **App_Data** — локальная БД
- **App_Start** — конфигурирование приложения
- **Content** — веб-контент (стили, изображения)
- **Controllers** — контроллеры
- **Fonts** — шрифты
- **Models** — модели
- **Scripts** — скрипты js
- **Views** — представления
- **Global.asax** — инициализатор приложения
- **Web.config** — конфигурация приложения



- Model — обычные классы C#.
- View — активные страницы:
 - ASPX + code-behind;
 - Razor;
 - Прочие.
- Controller — классы, являющиеся потомками класса Controller. Открытые методы контроллера называются действиями (action). Имя должно заканчиваться на Controller.

Пример модели

```
public class Student
{
    public string Name { get; set; }

    public int Age { get; set; }

    public bool EnglishSpeaker { get; set; }

    public static Student GetByName(string name)
    {
        return StudentLogic.Find(name);
    }

    public void Save()
    {
        StudentLogic.Update(this);
    }
}
```

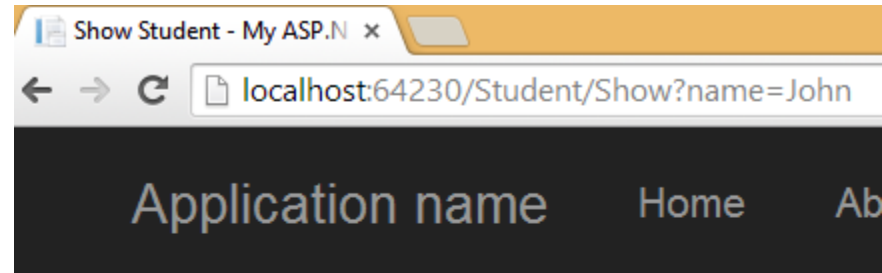
Пример представления (Razor)

```
@model WebApplication1.Models.Student
```

```
@{  
    ViewBag.Title = "Show Student";  
}
```

```
<h2>Show Student</h2>
```

```
<div>  
    <h4>@Model.Name</h4>  
    <h5>Age: @Model.Age</h5>  
  
    @if (Model.EnglishSpeaker)  
    {  
        <span>English speaker</span>  
    }  
</div>
```



Show Student

John

Age: 27

English speaker

Пример контроллера

```
public class StudentController : Controller
{
    public ActionResult Show(string name)
    {
        var model = Student.GetByName(name);
        if (model == null)
        {
            return HttpNotFound();
        }

        return View(model);
    }
}
```

Стандартная маршрутизация в ASP.NET MVC

- <http://localhost:64230/Student/Show>
 - Controller: Student
 - Action: Show
- <http://localhost:64230/Student/Edit/12>
 - Controller: Student
 - Action: Edit
 - Id: 12
- <http://localhost:64230/Student>
 - Controller: Student
 - Action: *Index*
- <http://localhost:64230/>
 - Controller: *Home*
 - Action: *Index*

```
url: "{controller}/{action}/{id}",  
defaults: new  
{  
    controller = "Home",  
    action = "Index",  
    id = UrlParameter.Optional  
}
```

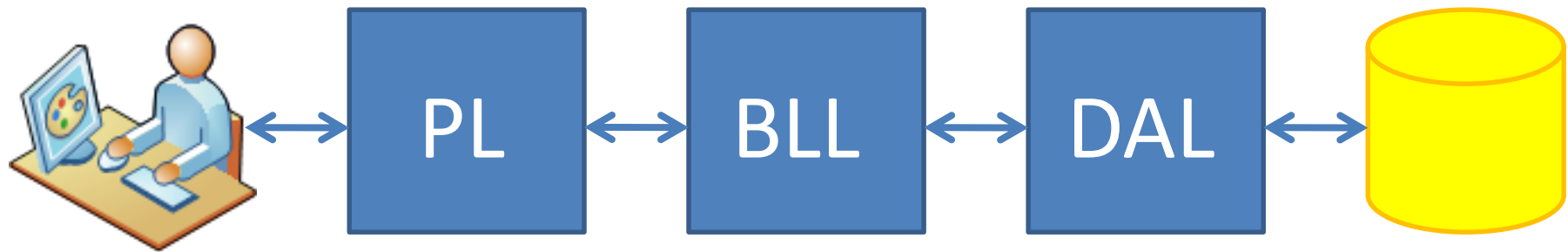
Инициализатор приложения Global.asax

```
public class MvcApplication : System.Web.HttpApplication
{
    protected void Application_Start()
    {
        AreaRegistration.RegisterAllAreas();
        FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
        RouteConfig.RegisterRoutes(RouteTable.Routes);
        BundleConfig.RegisterBundles(BundleTable.Bundles);
    }
}
```

Ещё полезные методы этого класса:

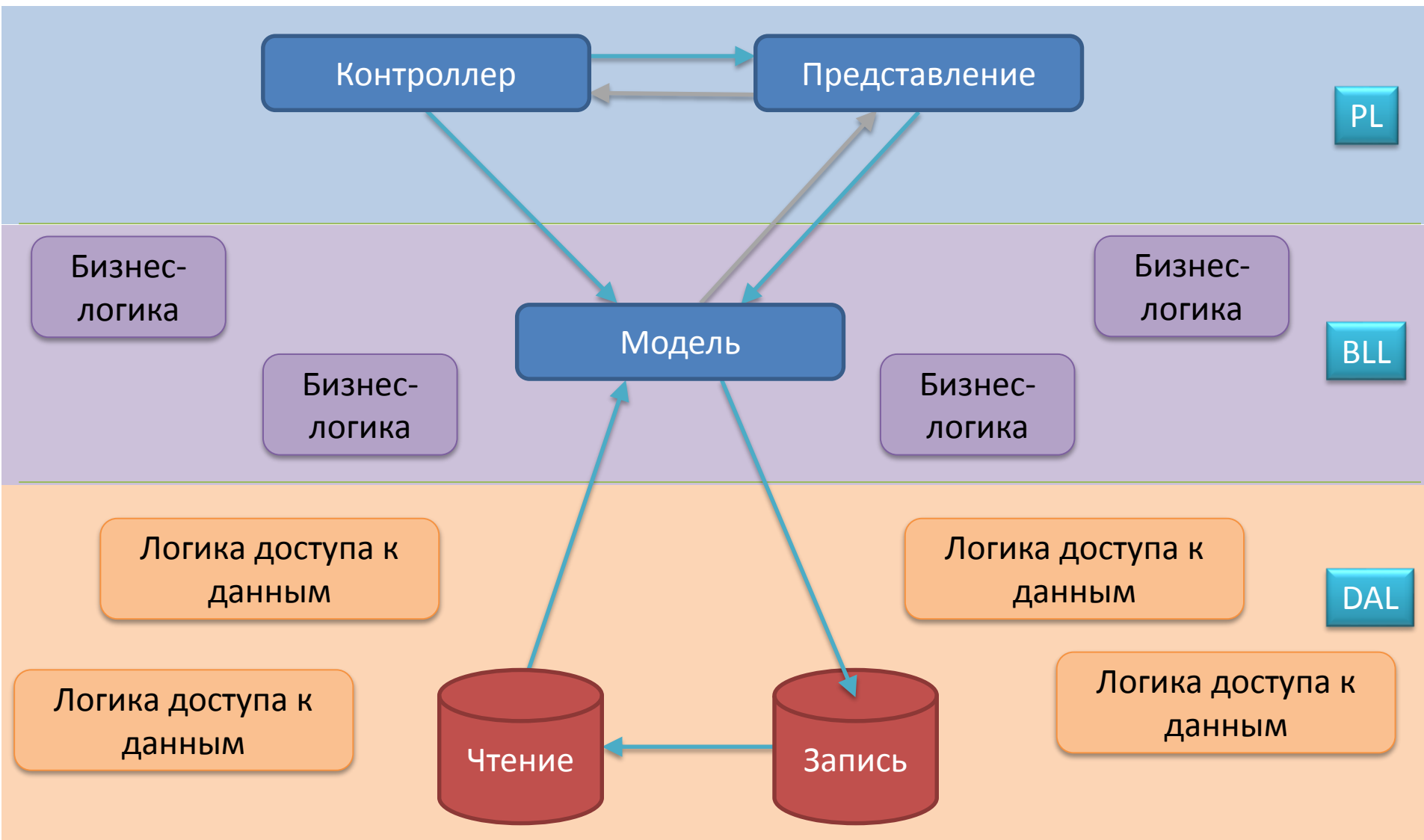
- **Application_Error** — произошло необработанное исключение;
- **Application_AuthenticateRequest** — выполняется после успешной аутентификации;
- **Session_Start** — открывается новая сессия;
- и многие другие...

Трехслойная архитектура



- **Presentation Layer** — слой представления. Отвечает за взаимодействие с пользователем.
- **Business Logic Layer** — слой бизнес-логики. Отвечает за непосредственную реализацию внутренних алгоритмов приложения.
- **Data Access Layer** — слой доступа к данным. Отвечает за хранение и предоставление данных.

Интеграция MVC в трёхслойную архитектуру



Что почитать?



«ASP.NET MVC 4 с примерами на C# 5.0 для профессионалов»

Адам Фримен

Официальный сайт:

<http://www.asp.net/mvc>

Цикл статей на Хабрахабре:

<http://habrahabr.ru/post/175999/>

Спасибо за внимание!

Контактная информация:

Дмитрий Верескун

Инструктор

EPAM Systems, Inc.

Адрес: Саратов, Рахова, 181

Email: Dmitry_Vereskun@epam.com

<http://www.epam.com>