Корпоративные информационные системы. Создание приложений ASP.NET

Жуков Артем Владимирович
zhukov@sampo.ru

План лекции

- Веб-сервер IIS
- Архитектура ASP.NET 2.0
- Объектная модель страницы

Где и что посмотреть:

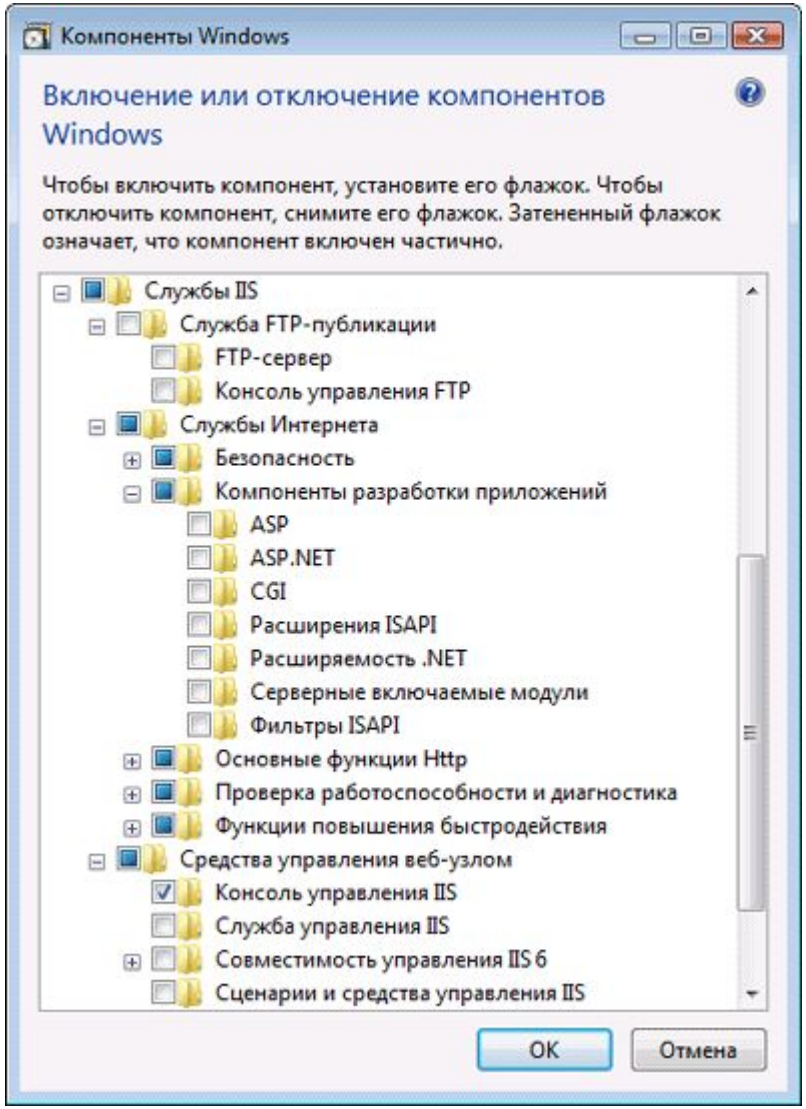
- http://www.MSDN.com/
- http://www.aspnetmania.com/
- Сообщество разработчиков .NET:
- http://www.gotdotnet.ru/
- http://www.sql.ru/
- http://dotsite.ru/
- Сообщество программистов:
- http://www.rsdn.ru/
- Вы можете завести пробный хостинг на http://europe.webmatrixhosting.net/russia/default.aspx.

Веб сервер IIS

- Версии:
  - Windows Server 2008 - IIS 7.0
  - Windows Server 2003 - IIS 6.0
  - Windows XP Professional - IIS 5.1
  - Windows 2000 - IIS 5.0
- IIS 7.0 разбивает веб-сервер на небольшое ядро сервера и более чем 40 модулей компонентов, подключаемых к этому ядру.

- StaticFileModule, который позволяет загружать статическое веб-содержимое,
- WindowsAuthModule, поддерживающий встроенную проверку подлинности NTLM,
- И т.п.

Модульная структура IIS 7.0



Настройка и конфигурирование IIS

- Централизованное хранилище конфигураций (метабаза) в IIS < 7
- Для IIS 7.0 характерна новая система делегированной настройки, основанная на иерархии распределенных файлов настройки в формате XML.
  - Данная иерархия обобщена в глобальном файле applicationHost.config, в котором содержатся значения по умолчанию для настройки уровня сервера, и распределенных файлах web.config, находящихся в структуре каталогов приложения.

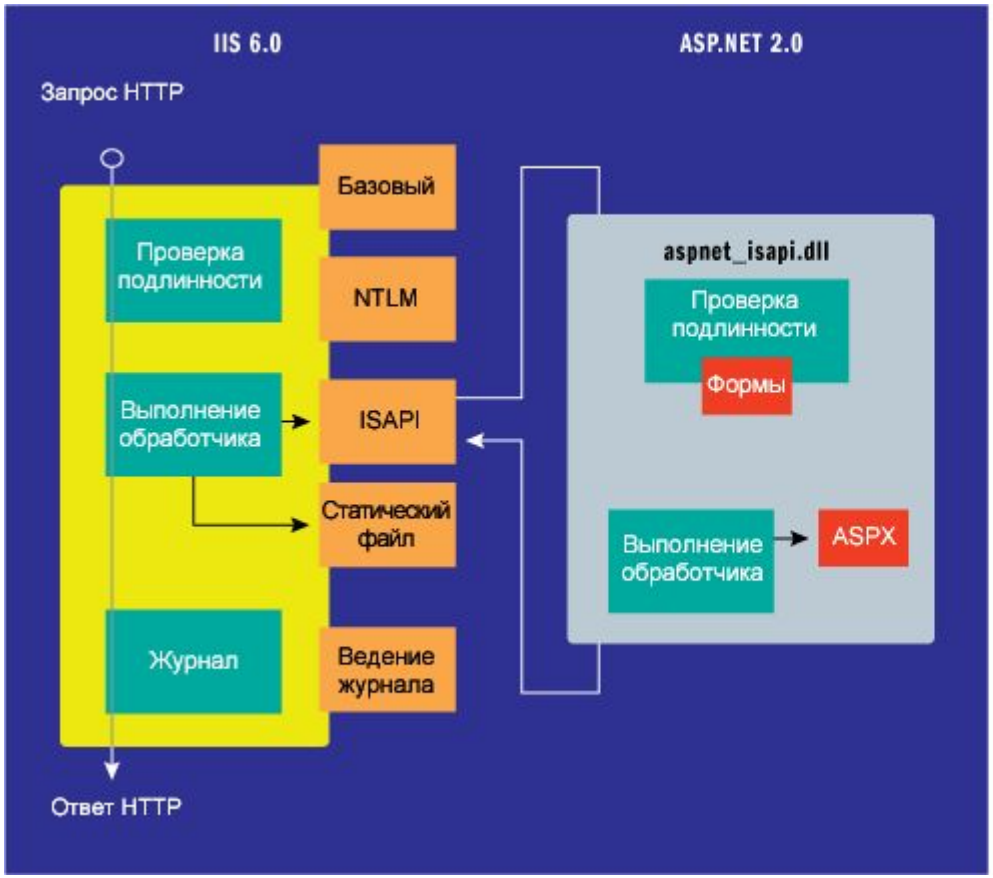
Обработка ASP.NET запросов в IIS 5.0

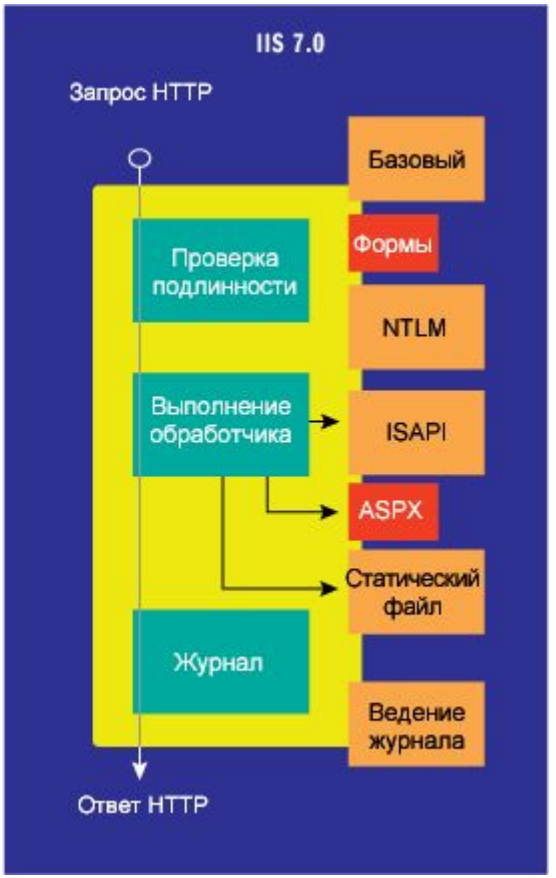
- IIS получает запрос, определяет тип ресурса и, если данный тип связан с ASP.NET, передает его на обработку расширению aspnet_isapi.dll. ISAPI расширение передает запрос на дальнейшую обработку рабочему процессу ASP.NET.
- После получения запроса, рабочий процесс передает сообщение ISAPI расширению, сообщая о том, что запрос будет обработан.
- Запрос выполняется в контексте рабочего процесса ASP.NET.
- После окончания выполнения, рабочий процесс передает данные ISAPI расширению, которое отвечает за освобождение ресурсов, занятых рабочим процессом.

Обработка ASP.NET запросов в IIS 6.0

- Используется модель пула приложений – отдельного рабочего процесса, который обслуживает одно или несколько веб-приложений.
- Каждый пул приложений обслуживается отдельным экземпляром рабочего процесса w3wp.exe.
- Рабочий процесс, обслуживающий пул приложений, загружает необходимые ISAPI расширения.
- В случае ASP.NET это рашсирение aspnet_isapi.dll, которое в свою очередь загружает CLR и начинает обработку HTTP запроса. Рабочие процессы выполняются под учетной записью NetworkService.

Обработка ASP.NET запросов в IIS 7.0





Обработка ASP.NET запросов в IIS 7.0

- При работе в режиме Integrated в IIS 7.0 модули ASP.NET выполняются в рамках единого конвейера обработки запросов одновременно с модулями IIS на C++ в машинном коде

Основные определения ASP.NET

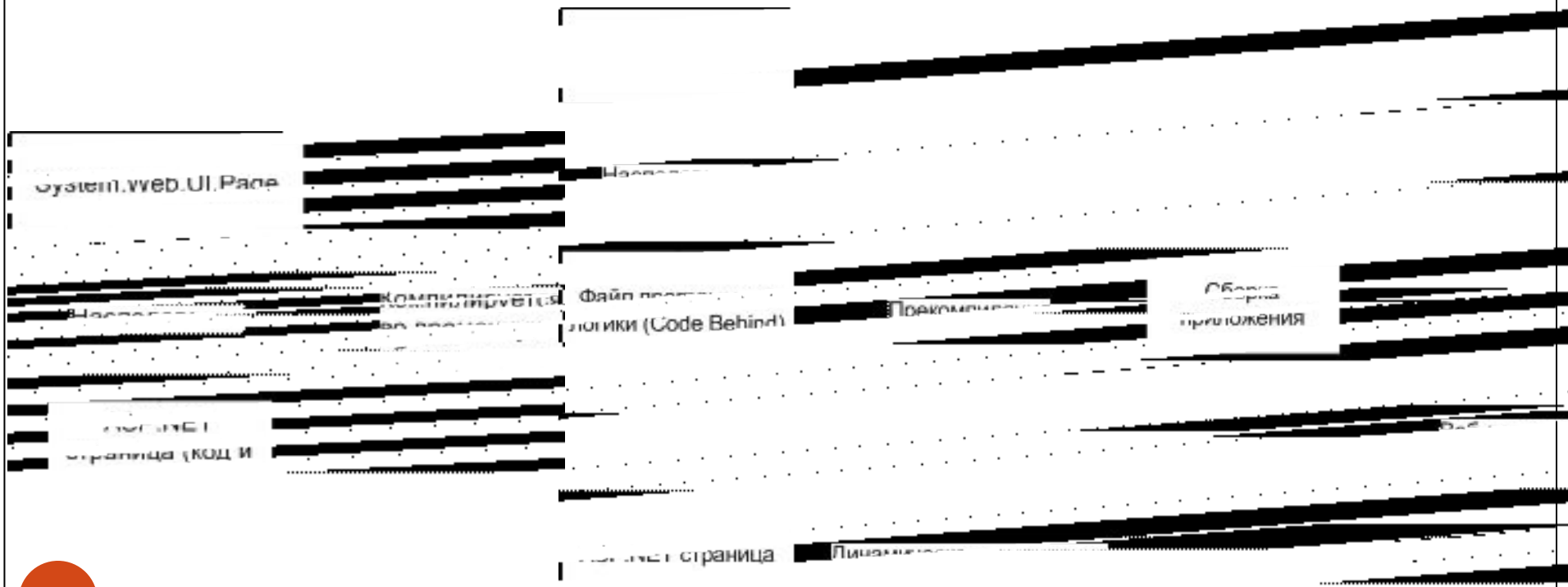
- Microsoft .NET Framework — это платформа для создания, развертывания и запуска web-сервисов и приложений. (intuit.ru)
- .NET Framework состоит из трех основных частей — общеязыковой среды выполнения (common language runtime), иерархического множества унифицированных библиотек классов и компонентной версии ASP, называемой ASP .NET. (intuit.ru)

ASP.net - динамически формируемые страницы

- В .NET Framework применяется двухэтапная компиляция,
  - то есть первый этап — это компиляция в MSIL, а
  - вторая — компиляция "just-in-time" компилятором во время исполнения.
- JIT-компилятор оптимизирует код для той машины, на которой он исполняется. В ASP .NET страница компилируется в MSIL при первом обращении клиента к странице. Вместе с ней компилируются классы, которые она использует.

Разделение программной логики и представления

- ASP – модель смежного (встроенного, смешанного) кода
- ASP.NET – модель раздельного и смежного кода.



Архитектура ASP.NET 2.0

Компиляция ASP.NET 2.0

- для каждой ASP.NET страницы создается своя собственная сборка. Эта модель компиляции открывает возможность не перекомпилировать все приложение при изменении одного файла исходного кода, а осуществлять компиляцию только измененных файлов. Поэтому ASP.NET 2.0 предлагает три основных стратегии компиляции приложений:
- Перекомпиляция файлов программной логики. В этом случае создаются сборки в директории bin, которые во время выполнения объединяются с динамическими сборками, генерируемыми для страниц с использованием механизма разделения классов.
- Полная пре-компиляция. Абсолютно новая возможность, появившаяся в ASP.NET 2.0 и позволяющая создать одну сборку для всех файлов приложения, включая файлы ASPX, содержащие HTML разметку. Сборка помещается в директорию bin веб-приложения, а содержимое всех ASPX файлов замещается на стоку «This is a marker file generated by the precompilation tool, and should not be deleted!».
- Динамическая компиляция. Эта стратегия аналогична используемой в ASP.NET стратегии динамической компиляции по запросы, с одним исключением, что страницы компилируются не одновременно, а по мере поступления запросов к каждой конкретной странице.
- Утилита компиляции aspnet_compiler.exe

Объектная модель ASP

- Каждая страница ASP .NET — это объект класса, который автоматически генерируется ядром ASP .NET.
- Класс наследуется от ассоциированного со страницей класса System.Web.UI.Page

События и методы страницы ASP.NET

Событие	Описание
InitComplete	Возникает после инициализации объекта страницы.
LoadComplete	Происходит после окончания загрузки страницы.
PreInit	Первое событие в жизненном цикле страницы.
PreLoad	Возникает перед загрузкой информации о состоянии.
PreRenderComplete	Происходит после создания всех дочерних элементов управления.

Метод	Описание
EnsureID	Создает уникальный идентификатор для элемента управления (для страницы практической пользы не имеет).
Focus	Устанавливает фокус ввода для элемента управления (для страницы практической пользы не имеет).
GetValidators	Возвращает коллекцию влидаторов, относящихся к определенной группе.
RegisterRequiresControlState	Устанавливает для переданного в качестве параметра элемента управления необходимость управления состоянием элемента управления (control state).
SetFocus	Устанавливает фокус ввода на определенный элемент управления.
TestDeviceFilter	Проверяет, является ли текущий браузер, используемый для просмотра страницы, указанным в качестве параметра метода.

Объекты ASP.NET

Объект	Описание
Application	
Session	
Request	
Response	
ClientScript	Экземпляр класса ClientScriptManager, предоставляющего набор методов для работы с клиентскими сценариями.
Header	Экземпляр объекта HtmlHead, представляющего заголовок HTML страницы.
Master	Ссылка на объект, представляющий шаблон, используемый для данной страницы

Жизненный цикл страницы

Этап	Описание
Запрос станицы	Запрос страницы осуществляется ядо начала жизненного цикла страницы. Когда пользователь осуществляет запрос, среда выполнения ASP.NET устанавливает, необходимо ли осуществить компиляцию страницы и начать жизненный цикл, либо можно выдать в качестве ответа страницу из кеша, таким образом, не выполняя страницы.
Начало жизненного цикла	На этом этапе происходит установка свойств Response и Request и свойства UICulture. Также, на этом этапе устанавливается, была ли эта страница запрошена в результате постбэка (отправления данных на сервер) и соответствующим образом устанавливается свойство IsPostBack.
Инициализация страницы	Ко времени инициализации страницы все дочерние пользовательские элементы управления уже созданы и имеют установленный свойства UniqueID. В это же время к странице применяются темы оформления. Если страница вызвана в результате постбэка, то данные, отправленные на сервер, еще не загружены в свойства элементов управления, на этом этапе.

Жизненный цикл страницы

Загрузка	Если страница вызвана в результате постбэка, то на этом этапе устанавливаются свойства элементов управления, на основании информации о состоянии (ViewState и ControlState).
Валидация	Вызывается метод Validate() для всех, находящихся на странице валидаторов.
Обработка постбэка	Вызываются обработчики событий (при условии, что постбэк произошел).
Рендеринг	Сохраняется информация о состоянии, затем класс страницы вызывает соответствующие методы дочерних элементов управления для генерации HTML представления и передачи его в Response.OutputStream.
Выгрузка	Выгрузка происходит после того, как создано HTML представление для всей страницы.

Жизненный цикл страницы

- Для того, чтобы проследить жизненный цикл страницы и последовательность возникновения событий, можно установить атрибут Trace директивы @Page в true, а атрибут TraceMode в "SortByTime". Тогда в разделе Trace Information можно найти список произошедших событий (колонка Message).

Жизненный цикл страницы

Category	Message	From First(s)	From Last(s)
aspx.page	Begin PreInit		
aspx.page	End PreInit	0.0364973314167865	0.036497
aspx.page	Begin Init	0.0379050459346291	0.001408
aspx.page	End Init	0.047693704143491	0.009789
aspx.page	Begin InitComplete	0.0477864508468221	0.000093
aspx.page	End InitComplete	0.0481875670270608	0.000401
aspx.page	Begin PreLoad	0.0489879732516718	0.000800
aspx.page	End PreLoad	0.0494462283607275	0.000458
aspx.page	Begin Load	0.0494924892194238	0.000046
aspx.page	End Load	0.0553441897381414	0.005852
aspx.page	Begin LoadComplete	0.0554711043059809	0.000127
aspx.page	End LoadComplete	0.055942153615399	0.000471
aspx.page	Begin PreRender	0.0561455634022874	0.000203
aspx.page	End PreRender	0.0618604874695332	0.005715
aspx.page	Begin PreRenderComplete	0.06269871008062	0.000838
aspx.page	End PreRenderComplete	0.0633259746265858	0.000627
aspx.page	Begin SaveState	0.080360541216174	0.017035
aspx.page	End SaveState	0.213795377788888	0.133435
aspx.page	Begin SaveStateComplete	0.213911298043872	0.000116
aspx.page	End SaveStateComplete	0.214385763389788	0.000474
aspx.page	Begin Render	0.214440078745078	0.000054
aspx.page	End Render	0.315044337228923	0.100604

Директива	Атрибуты	Описание
@Assembly	Name Src	Импортирует на страницу или в элемент управления сборку с заданным именем
@Control	такие же, как у Page	Применяется для задания свойств при создании собственных пользовательских элементов управления
@Implements	Interface	Указывает, что класс данной страницы реализует данный интерфейс
@Import	Namespace	Импортирует пространство имен
@Master	такие же, как у Page	Применяется на страницах шаблона дизайна (Master page). Новая в ASP .NET 2.0
@MasterType	TypeName VirtualPath	Дает строго типизированную ссылку на класс, содержащийся в шаблоне дизайна. Позволяет обращаться к свойствам этого класса
@OutputCache	Duration Location VaryByCustom VaryByHeader VaryByParam VaryByControl	Управляет кэшированием страницы или элемента управления. Подробнее описана в лекции 15
@Page	см. ниже	Атрибуты, относящиеся к данной странице. Употребляется только в файлах с расширением aspx
@PreviousPageType	TypeName VirtualPath	Страница, с которой были отправлены данные, введенные пользователем. Новая в ASP .NET 2.0. Раньше страницы отправляли пост только самим себе
@Reference	Page Control	Страница или элемент управления, который нужно компилировать вместе с данной