

ПАВЕЛ АГУРОВ



# ASP.NET

## СБОРНИК РЕЦЕПТОВ



АРХИТЕКТУРА  
ВЕБ-ПРИЛОЖЕНИЙ

ВЕБ-ФОРМЫ  
И КОМПОНЕНТЫ ASP.NET

РАБОТА С БАЗАМИ ДАННЫХ

ОТЛАДКА  
И ОБРАБОТКА ОШИБОК

ЗАЩИТА ПРИЛОЖЕНИЙ  
ASP.NET

СОЗДАНИЕ ОТЧЕТОВ  
В MS Excel

**PRO**

ПРОФЕССИОНАЛЬНОЕ  
ПРОГРАММИРОВАНИЕ

Павел Агуров

# ASP.NET

## СБОРНИК РЕЦЕПТОВ

Санкт-Петербург

«БХВ-Петербург»

2010

УДК 681.3.06  
ББК 32.973.26-018.2  
А27

## **Агуров П. В.**

А27 ASP.NET. Сборник рецептов. — СПб.: БХВ-Петербург, 2010. — 528 с.: ил. — (Профессиональное программирование)

ISBN 978-5-9775-0521-5

В книге собраны практические советы и примеры, которые помогут при создании веб-приложений с использованием ASP.NET: разработка архитектуры веб-приложения, его отладка, профилирование, защита, конфигурирование, работа с данными и многое другое. Рассмотрены специальные инструменты и утилиты, которые позволяют ускорить и упростить разработку и отладку веб-приложений. Уделено внимание обработке исключений в веб-приложениях. Отдельная глава посвящена созданию отчетов в MS Excel. Книга будет полезна не только программистам, которые уже используют в своих разработках ASP.NET, но и тем, кто переходит на технологию ASP.NET с классической ASP или языка PHP.

*Для программистов*

УДК 681.3.06  
ББК 32.973.26-018.2

### **Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Екатерина Капальгина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 31.03.10.

Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 42,57.

Тираж 1500 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов  
в ГУП "Типография "Наука"  
199034, Санкт-Петербург, 9 линия, 12

# Оглавление

<b>Введение.....</b>	<b>1</b>
Для кого эта книга.....	2
О программном коде.....	2
Краткое описание глав.....	3
Благодарности.....	4
Обратная связь.....	4
<b>Глава 1. Архитектура и общие вопросы.....</b>	<b>5</b>
1.1. Основные отличия ASP.NET 1.1 и 2.0.....	5
1.2. В ASP было.....	7
1.3. Можно ли запустить приложение ASP.NET под Apache.....	8
1.4. Где найти исходный код Framework.....	8
1.5. Использование SSI <i>include</i> в ASP.NET.....	8
1.6. Как узнать версию ASP.NET, под которой работает сайт.....	9
1.7. Как узнать браузер и версию клиента, запустившего сайт.....	9
1.8. Как узнать параметры компьютера, на котором работает сайт.....	9
1.9. Где расположен временный каталог.....	9
1.10. Как изменить временный каталог ASP.NET.....	10
1.11. Информация о соединении.....	10
1.12. Зачем создается пользователь <i>ASPNET</i> .....	10
1.13. Где сохранить данные при переходе между страницами.....	10
1.13.1. Адресная строка.....	11
1.13.2. Куки.....	11
1.13.3. Скрытые поля.....	12
1.13.4. Состояние страницы.....	12
1.13.5. Сессия.....	12
1.13.6. Память приложения.....	13
1.13.7. Что же выбрать.....	14

1.14. Не используйте подчеркивание в имени серверов .....	14
1.15. Общие правила создания страниц.....	14
1.15.1. Не путайте разметку и код.....	14
1.15.2. Не встраивайте C#-код в ASPX-файл.....	15
1.15.3. Используйте отдельные JS-файлы .....	15
1.15.4. Используйте отдельные CSS-файлы.....	16
1.15.5. Используйте мастер-страницы.....	16
1.15.6. Создавайте базовые классы страниц .....	16
1.15.7. Используйте свойства для обращения к сессии и состоянию .....	16
1.15.8. Не используйте <i>Convert.To</i> -методы, когда этого не требуется .....	17
1.15.9. Не используйте <i>TryParse</i> , когда не проверяется результат .....	18
1.15.10. Проверяйте данные не только на клиенте, но и на сервере.....	18
1.15.11. Не создавайте глубокой иерархии элементов управления .....	18
1.15.12. Используйте события для обмена между элементами управления и их контейнерами .....	19
1.15.13. Будьте аккуратны со статическими переменными.....	19
1.15.14. Правила обработки исключений.....	20
1.15.15. Не выводите входные данные напрямую на страницу.....	20
1.15.16. Подготовка к тестированию .....	20
1.16. Как заблокировать одновременный доступ <i>Application</i> .....	20
1.17. Простой класс доступа к данным.....	20
1.18. Реализация единого дизайна страниц.....	30
1.19. Процедура восстановления пароля .....	30
1.20. Файл <i>global.asax</i> и события.....	31
1.20.1. Наиболее важные методы <i>global.asax</i> .....	31
1.20.2. Можно ли создать <i>global.asax</i> в виде CS-файла.....	32
1.20.3. Определение причины закрытия сайта.....	33
1.21. Модули HTTP и обработчики HTTP .....	34
1.21.1. Модули HTTP.....	35
1.21.2. Обработчики HTTP .....	37
1.21.3. Стандартные обработчики <i>HTTP</i> .....	38
1.22. Как сделать иконку для сайта (Favicon) .....	38
1.23. Ввод чисел с плавающей точкой.....	39
1.24. В чем разница между <i>CurrentCulture</i> и <i>CurrentUICulture</i> .....	41
1.25. Отправка почты из ASP.NET-приложения.....	41
1.25.1. Использование класса <i>SmtpClient</i> .....	41
1.25.2. Стандартные настройки SMTP в ASP.NET 2.0 .....	47
1.25.3. Решение проблемы с русскими символами .....	48
1.25.4. Игнорирование проверки сертификата SSL .....	49
1.25.5. Как отправить событие в календарь Outlook .....	49
1.26. Проверка орфографии.....	50
1.27. Как задать допустимое время выполнения скриптов.....	51
1.28. Почему установка <i>executionTimeout</i> не работает .....	51
1.29. Offline-режим для приложения .....	51

1.30. Кроссбраузерность .....	52
1.30.1. Проблемы не IE-браузеров .....	52
1.30.2. Фильтры браузеров .....	52
1.30.3. Применение стилей только для IE .....	53
1.30.4. Условные выражения в CSS .....	53
1.30.5. Прыгающая ширина поля ввода в IE .....	54
1.30.6. Прозрачная PNG-картинка в браузере IE .....	54
1.30.7. Ограничение на число файлов стилей в браузере IE .....	55
1.30.8. Удаление рамки с активной ссылки .....	55
1.31. Разное про HTML .....	55
1.31.1. Лишний отступ снизу и подчеркивание изображения внутри ссылки .....	55
1.31.2. Вставка флэш-файлов в страницу .....	55
1.31.3. "Отладка" верстки .....	56
1.31.4. Преобразование HTML-текста .....	56
1.31.5. Как передать значение из JS-кода на сторону сервера .....	57
1.31.6. Как вывести значение в HTML .....	57
1.31.7. Предупреждение о закрытии браузера (только IE и Firefox) .....	57
1.31.8. Двигающийся текст .....	58
1.31.9. Блокировка экрана на время длительной операции .....	58
1.31.10. Определение текущей кодировки страницы .....	61
1.32. Совместимость .....	61
1.32.1. Как определить, поддерживает ли браузер пользователя ActiveX .....	61
1.32.2. Как определить, поддерживает ли браузер пользователя JavaScript .....	61
1.33. Дни месяца по-русски .....	61
1.34. Запуск задач по расписанию .....	62
1.34.1. Запуск задач в ASP.NET с помощью таймера или потока .....	62
1.34.2. Запуск задач в ASP.NET с помощью кэша .....	63
1.34.3. Вызов определенного URL через Windows-планировщик .....	64
1.35. Использование встроенных ресурсов .....	65
1.35.1. Встроенные изображения .....	65
1.35.2. Строковые ресурсы .....	67
1.36. Работа с изображениями и пиктограммами .....	68
1.36.1. Изменение размера изображения .....	68
1.36.2. Создание пиктограммы .....	69
1.36.3. Анимированный GIF .....	70
1.36.4. Обрезка изображений .....	71
1.37. Разное .....	71
1.37.1. Как преобразовать массив в строку с разделителем .....	71
1.37.2. Перекодировка текста .....	71
1.37.3. Преобразование в Base64 и обратно .....	72
1.37.4. Преобразование из Win1251 в Koï8 и обратно .....	72
1.37.5. Преобразование цвета в строку и обратно .....	73
1.37.6. Преобразование цвета в HTML-формат .....	73
1.37.7. Преобразование цвета в целое число и обратно .....	74
1.37.8. Возможности форматирования методов <i>Format</i> и <i>Eval</i> .....	74

<b>Глава 2. Формы</b> .....	<b>76</b>
2.1. Получение параметров формы.....	76
2.2. Модификация страницы до вызова метода <i>Page_Load</i> .....	77
2.3. Почему <i>Page_Load</i> вызывается два раза.....	77
2.4. Сохранение позиции скроллинга в браузере.....	77
2.5. Отображение данных в строке состояния браузера.....	78
2.6. Программная установка метатегов.....	78
2.7. Установка фокуса на элемент управления.....	79
2.8. Установка фокуса по умолчанию.....	79
2.9. Задание кнопки по умолчанию.....	79
2.10. На странице не отображаются русские буквы.....	79
2.11. Задание фона страницы из кода.....	80
2.12. Комментирование кода внутри ASPX-страницы.....	80
2.13. Комментирование внутри элементов управления.....	80
2.14. Открытие страницы по кнопке в новом окне.....	81
2.15. Использование WinForms-компонентов в веб-проектах.....	81
2.16. Как сделать аналог метода <i>MessageBox.Show</i> .....	84
2.17. Ручное формирование HTML-кода страницы.....	85
2.18. Мастер-страницы (master pages).....	87
2.18.1. Создание шаблона страниц.....	87
2.18.2. Доступ к мастер-странице.....	90
2.18.3. Передача данных из мастер-страницы в контент.....	91
2.18.4. Регистрация JS-скрипта для мастер-страницы.....	93
2.18.5. Доступ к <i>ScriptManager</i> мастер-страницы.....	93
2.18.6. Указание мастер-страницы через <i>web.config</i> .....	94
2.18.7. Динамический выбор мастер-страницы.....	95
2.18.8. Выбор мастер-страницы в зависимости от браузера.....	95
2.18.9. Как задать тему для мастер-страницы.....	96
2.18.10. Установка метатегов мастер-страницы.....	98
2.19. Динамическое добавление JS-файла к странице.....	98
2.20. Динамическое добавление CSS-файла к странице.....	99
2.21. Динамическое добавление HTML-кода на страницу.....	99
2.22. Просмотр исходного кода страницы.....	99
2.23. Получение всех введенных данных формы.....	100
2.24. Как получить значение hidden-поля в коде.....	100
2.25. "Горячие" клавиши страницы.....	100
2.25.1. Использование свойства <i>AccessKey</i> .....	100
2.25.2. Использование JS-скрипта.....	101
2.26. Автоматическое обновление страницы по времени.....	101
2.26.1. Использование JavaScript.....	101
2.26.2. Использование метатегов.....	102
2.27. Печать страницы на принтер по умолчанию.....	103
2.28. Создание PDF-файла из страницы.....	105

<b>Глава 3. Элементы управления .....</b>	<b>109</b>
3.1. Общие вопросы элементов управления.....	109
3.1.1. Регистрация элементов управления в <code>eb.config</code> .....	109
3.1.2. Получение HTML-кода элемента .....	109
3.1.3. Получение клиентских идентификаторов ( <i>ClientID</i> ) .....	110
3.1.4. Удаление ненужных клиентских идентификаторов ( <i>ClientID</i> ) .....	111
3.1.5. Улучшение работы с <i>ClientID</i> в ASP.NET 4.0 .....	112
3.1.6. Отключение табличного представления форм (ASP.NET 4.0).....	112
3.1.7. Использование серверных тегов в серверных элементах управления.....	113
3.1.8. Создание элементов из строки .....	115
3.1.9. Скрыть/показать элемент страницы .....	115
3.2. Элемент выбора файла.....	116
3.2.1. Загрузка нескольких файлов .....	116
3.2.2. Как запретить ввод имени файла .....	119
3.2.3. Проверка типа файла .....	119
3.2.4. Можно ли задать фильтр для выбираемых файлов.....	119
3.2.5. Одновременная загрузка нескольких файлов с отображением процесса... Вариант 1 .....	120
Вариант 2 .....	120
Вариант 3 .....	120
3.3. Элемент управления <i>Label</i> .....	121
3.3.1. Как отобразить текст по вертикали .....	121
3.3.2. Как изменить текст через JavaScript.....	121
3.3.3. Как отобразить текст в несколько строк .....	121
3.4. Элемент управления <i>CheckBoxList</i> .....	121
3.4.1. Выбрать все отмеченные элементы.....	121
3.4.2. Проверить, что ничего не выбрано.....	121
3.5. Элемент управления <i>TreeView</i> .....	122
3.5.1. Ограничение по ширине .....	122
3.5.2. Дерево отображается некорректно .....	122
3.6. Элементы управления <i>ListView</i> и <i>ListBox</i> .....	122
3.6.1. Почему свойство <i>SelectedItem</i> равно <i>null</i> .....	122
3.6.2. Улучшения <i>ListView</i> в ASP.NET 4.0 .....	123
3.6.3. Скроллируемый <i>ListBox</i> .....	123
3.7. Элемент управления <i>TextBox</i> .....	124
3.7.1. Запрет ввода формы по клавише <code>&lt;Enter&gt;</code> .....	124
3.7.2. Указание максимальной длины <i>TextBox</i> для полей ввода .....	124
3.7.3. Подсказка ввода в <i>TextBox</i> .....	126
3.7.4. Выравнивание текста .....	127
3.7.5. Проверка введения корректной даты .....	128
3.8. Отображение графических карт .....	128
3.9. Элемент управления <i>GridView</i> .....	129
3.9.1. Общие вопросы .....	129
А где <i>DataGrid</i> ?.....	129
Добавление прокрутки .....	129



3.9.2. Привязка данных .....	130
Чем свойство <i>DataSource</i> отличается от <i>DataSourceId</i> .....	130
Привязка данных с помощью свойства <i>DataSource</i> .....	130
Привязка данных с помощью свойства <i>DataSourceId</i> .....	131
Форматирование данных при привязке .....	132
3.9.3. Колонки <i>GridView</i> .....	132
Автоматическая генерация колонок.....	132
Декларативное добавление колонок.....	132
Программное добавление колонок.....	133
Проблемы форматирования колонок .....	133
Колонки операций.....	133
Несколько кнопок в одной колонке .....	135
Подтверждение выполнения операции .....	135
Как отобразить картинку.....	135
Как отобразить и заголовок и картинку .....	136
3.9.4. Строки <i>GridView</i> .....	136
Получение номера строки в <i>GridView</i> .....	136
Подсветка строк при наведении курсора мыши.....	137
Удаление строки .....	138
Редактирование строк.....	139
Добавление строк.....	139
Подтверждение при удалении строки .....	139
3.9.5. Сортировка <i>GridView</i> (MS SQL) .....	141
3.9.6. Уменьшение размера <i>ViewState</i> .....	142
3.9.7. Событие выбора строк.....	142
3.9.8. Экспорт в Excel .....	147
3.9.9. Сортировка по нескольким столбцам.....	148
3.10. Элемент управления <i>Repeater</i> .....	148
3.10.1. Привязка списка объектов.....	148
3.10.2. Привязка списка строк.....	150
3.10.3. Вложенные <i>Repeater</i> .....	150
3.10.4. Операции с элементами <i>Repeater</i> .....	152
3.10.5. Цветные строки в <i>Repeater</i> .....	152
3.11. Календарь ( <i>Calendar</i> ) .....	153
3.11.1. Отображение расписания с помощью элемента <i>Calendar</i> .....	153
3.11.2. Валидация данных календаря .....	157
3.12. Реализация закладок ( <i>TabControl</i> ) .....	157
3.13. Кнопки.....	160
3.13.1. Изображение для запрещенной кнопки с картинкой .....	160
3.13.2. Задание кнопки по умолчанию .....	161
3.13.3. Как отключить у кнопки валидацию формы.....	161
3.13.4. Как задать клиентский обработчик кнопки .....	161
3.13.5. Как перейти на другую страницу после возврата страницы.....	161
3.13.6. Почему не вызывается метод <i>Click</i> у кнопки.....	161

3.13.7. Кнопка закрытия окна браузера .....	162
3.13.8. Запрет кнопки на время длительной операции .....	162
3.14. Отображение рекламных объявлений .....	163
<b>Глава 4. Валидация .....</b>	<b>165</b>
4.1. Варианты валидации .....	165
4.1.1. Обязательные поля .....	166
4.1.2. Проверка диапазона данных .....	166
4.1.3. Проверка формата данных .....	166
Примеры регулярных выражений .....	167
4.1.4. Сравнение значений .....	169
4.1.5. Сравнение дат календарей .....	169
4.1.6. Пользовательские процедуры валидации .....	171
4.1.7. Отображение итоговой информации о валидации .....	172
4.2. Установка фокуса на ошибку .....	173
4.3. Элементы, не вызывающие валидацию .....	174
4.4. Валидация групп полей .....	174
4.5. Проблемы валидации данных .....	175
4.5.1. Слишком строгие правила .....	175
4.5.2. Проблема кодировок .....	175
4.5.3. Валидация буквы <i>e</i> .....	176
4.5.4. Только клиентская валидация .....	176
4.6. Валидация переключателей ( <i>CheckBox</i> ) .....	177
4.7. Валидация чисел с плавающей точкой .....	178
4.8. Валидация перед переходом на другую страницу .....	178
4.9. Валидация без использования стандартных валидаторов .....	179
4.10. Клиентская валидация с помощью веб-методов .....	180
<b>Глава 5. Отладка, тестирование, обработка исключений и ошибок.....</b>	<b>184</b>
5.1. Проверка на запуск в отладочном режиме .....	184
5.2. Правила разработки для облегчения тестирования сайтов .....	184
5.2.1. Режим отладки .....	185
5.2.2. Тестирование при Windows-имперсонации .....	185
5.2.3. Тестирование рабочего процесса, зависящего от времени .....	185
5.2.4. Тестирование почтовой рассылки .....	186
5.3. Общие правила обработки исключений .....	187
5.3.1. "Не глотайте" исключения молча .....	187
5.3.2. Не обрабатывайте те исключения, которые не должны быть обработаны в данный момент .....	187
5.3.3. Обходите без исключений, если это возможно .....	188
5.3.4. Сообщайте информацию о коде с помощью исключений .....	188
5.3.5. Не пересоздавайте исключения заново .....	189
5.3.6. Не давайте пользователю приложения лишнюю информацию об исключениях .....	190

5.3.7. Используйте исключения, а не коды ошибок .....	190
5.3.8. Используйте иерархию исключений .....	192
5.4. Обработка ошибок в параметрах URL .....	193
5.5. Отладочная информация (трассировка) для ASP.NET .....	194
5.6. Оценка времени выполнения кода.....	195
5.6.1. Измерение с помощью <i>TickCount</i> (наименьшая точность).....	195
5.6.2. Измерение с помощью <i>Ticks</i> (средняя точность) .....	195
5.6.3. Измерение с помощью <i>QueryPerformance</i> (высокая точность).....	195
5.6.4. Измерение с помощью класса <i>Stopwatch</i> (C# 2.0).....	196
5.7. Вывод сообщений в окно <i>Output</i> среды .....	196
5.8. Запись в Application Log .....	196
5.9. Создание своего Event Log.....	197
5.10. Обработка исключений на странице.....	199
5.11. Глобальная обработка исключений ASP.NET .....	199
5.11.1. Обработка через HTTP-модуль ( <i>IHttpModule</i> ).....	199
5.11.2. Обработка через web.config.....	201
5.11.3. Обработка с помощью монитора здоровья .....	201
5.11.4. Обработка с помощью библиотеки ELMAN.....	202
5.12. Обнаружение причины перезагрузки сайта .....	203
5.13. Отключение перезагрузки сайта при изменениях в каталогах .....	203
5.14. Исключение при перенаправлении на другую страницу .....	204
5.15. Тестирование веб-страниц без веб-сервера.....	205
5.16. Отладка JS-кода.....	205
5.17. Сохранение запроса .....	206

## **Глава 6. Конфигурирование и конфигурационные файлы..... 207**

6.1. Конфигурационный файл web.config.....	207
6.1.1. Общие сведения о web.config.....	207
6.1.2. Где найти класс <i>ConfigurationManager</i> .....	208
6.1.3. Можно ли в web.config использовать символы <, > и т. п. ....	208
6.1.4. Шифрование секций web.config.....	208
6.1.5. Шифрование строки подключения.....	208
6.1.6. Вынесение секции параметров во внешний файл .....	208
6.1.7. Вынесение секций во внешний файл.....	209
6.1.8. Нестандартный конфигурационный файл.....	209
6.1.9. Изменение web.config .....	215
6.1.10. Отключение модулей в web.config.....	216
6.2. Где хранить строку подключения к БД.....	217
6.3. Управление виртуальными директориями IIS .....	218
6.4. Шифрование с помощью DPAPI.....	222
6.5. Хранение паролей в памяти.....	223
6.6. Хранение паролей в файле конфигурации .....	224

<b>Глава 7. Производительность .....</b>	<b>226</b>
7.1. Не кэшируйте соединение с БД .....	226
7.2. Получение статистики о соединении с БД .....	227
7.3. Используйте <i>DataReader</i> для последовательного доступа к данным .....	227
7.4. Используйте хранимые процедуры .....	228
7.5. Управление буферизацией страниц .....	228
7.6. Используйте отдельные JS- и CSS-файлы .....	228
7.7. Отключайте режим отладки .....	228
7.8. Управление кэшированием страниц .....	229
7.8.1. Атрибут <i>Location</i> .....	230
7.8.2. Атрибут <i>Duration</i> .....	230
7.8.3. Атрибут <i>VaryByParam</i> .....	230
7.8.4. Атрибут <i>VaryByControl</i> .....	231
7.8.5. Атрибут <i>VaryByHeader</i> .....	232
7.8.6. Атрибут <i>VaryByCustom</i> .....	232
7.8.7. Использование класса <i>HttpCachePolicy</i> .....	234
7.8.8. Очистка кэша .....	235
7.8.9. Ограничения кэширования .....	236
7.9. Частичное кэширование .....	236
7.9.1. Кэширование элементов управления .....	236
7.9.2. Подстановка вне кэша .....	236
7.10. Создание статического кэша .....	237
7.11. Кэширование в ASP.NET 4.0 .....	239
7.12. Использование кэша ASP.NET .....	240
7.12.1. Чем <i>Cache</i> отличается от <i>Application</i> .....	240
7.12.2. Добавление элементов в кэш .....	240
Простое добавление в кэш .....	241
Задание абсолютного времени устаревания .....	241
Задание скользящего времени устаревания .....	241
Задание зависимостей от другого элемента кэша .....	241
Задание зависимостей от файла или папки .....	242
Задание времени начала контроля зависимостей .....	242
Задание приоритетов освобождения .....	243
Обращение к элементам кэша .....	243
<b>Глава 8. Работа с URL .....</b>	<b>244</b>
8.1. Получение "чистого" пути к странице .....	244
8.2. Получение "чистого" пути к приложению .....	244
8.3. Чем URL отличается от URI .....	245
8.4. Разбор URL на составляющие .....	245
8.5. Преобразование относительного пути в абсолютный .....	246
8.6. Проверка использования защищенного протокола .....	246
8.7. Перенаправление на другую страницу .....	247
8.7.1. Постоянное перенаправление (код 301) .....	247
8.7.2. Перенаправление через определенный интервал времени .....	247

8.7.3. Чем <i>Server.Transfer</i> отличается от <i>Response.Redirect</i> .....	248
Ошибка "View State Is Invalid" .....	249
Почему <i>Server.Transfer</i> ("page.html") дает пустую страницу? .....	249
Как распознать <i>Response.Redirect</i> и <i>Server.Transfer</i> .....	249
Выбор между <i>Response.Redirect</i> и <i>Server.Transfer</i> .....	249
8.7.4. Чем <i>Server.Execute</i> отличается от <i>Server.Transfer</i> ? .....	250
8.7.5. Сравниваем <i>Server.Transfer</i> , <i>Server.Execute</i> и <i>Response.Redirect</i> .....	250
Вызываем <i>Response.Redirect</i> .....	251
Вызываем <i>Server.Transfer</i> .....	251
Вызываем <i>Server.Execute</i> .....	252
8.8. Управление созданием HTTP-обработчиков ( <i>IHttpHandler</i> ) .....	252
8.9. Можно ли задать расширение в диалоге выбора файла .....	254
8.10. Отобразить значок состояния ICQ-пользователя .....	254
8.11. Отобразить значок состояния Skype-пользователя .....	254
8.12. Получение относительного пути .....	254
8.13. Оптимизация ссылок (URL Rewriting) .....	255
8.13.1. Использование свойства <i>PathInfo</i> .....	256
8.13.2. Использование метода <i>RewritePath</i> .....	257
8.13.3. Использование IS7 .....	258
8.13.4. Использование web.config .....	258
8.13.5. Ссылки на картинки, скрипты и др. ....	259
8.13.6. Сравнение вариантов перезаписи ссылок .....	259
<b>Глава 9. Пользователи, имперсонация, авторизация .....</b>	<b>261</b>
9.1. Получение имени текущего пользователя .....	261
9.2. Программная имперсонация .....	261
9.3. Как получить IP-адрес клиента, открывшего сайт .....	263
9.4. Как получить культуру клиента, открывшего сайт .....	264
9.5. Как получить список групп домена, в которые входит пользователь .....	264
9.6. Сохранение данных пользователя и реализация <i>IPrincipal</i> .....	265
<b>Глава 10. Библиотека JQuery .....</b>	<b>276</b>
10.1. Базовые операции .....	276
10.1.1. Подключение библиотеки .....	276
10.1.2. Обработка событий страницы .....	276
10.1.3. Выбор элементов страницы .....	277
Обращение по клиентскому идентификатору .....	277
Выбор элементов по типу .....	277
Выбор элементов по классу .....	277
Специальные символы в идентификаторах .....	278
Выбор элемента в иерархии .....	278
Выбор дочерних элементов .....	278
Выбор элементов по атрибуту .....	278
Примеры .....	278

10.1.4. Проверка существования элемента .....	279
10.1.5. Метод <i>click</i> .....	280
10.1.6. Перебор элементов.....	280
10.1.7. Отмена стандартного обработчика.....	280
10.1.8. Обработка возврата формы (submit).....	281
10.1.9. Генерация возврата формы (submit).....	281
10.1.10. Проверка принадлежности .....	281
10.1.11. Установка и удаление атрибутов элементов.....	282
10.1.12. Загрузка данных из XML-файла .....	282
Проблема в браузере IE.....	283
10.1.13. Чем <i>html()</i> отличается от <i>text()</i> .....	284
10.2. Элементы управления .....	284
10.2.1. Разрешение и запрещение элементов.....	284
10.2.2. Отметка опции ( <i>checkbox</i> ) .....	284
10.2.3. Получение выбранного элемента выпадающего списка.....	285
10.2.4. Получение выбранного переключателя ( <i>radiobutton</i> ).....	285
10.2.5. Изменение атрибутов при подведении курсора .....	285
10.3. Плагины .....	286
10.3.1. Таблица jqGrid.....	286
10.3.2. Графики jqPlot.....	287
10.3.3. Преобразование таблиц в графики .....	288
10.3.4. Таблица tablesorter .....	288
10.3.5. Сортировка таблиц перетаскиванием.....	289
10.3.6. HTML-редакторы текста .....	289
10.3.7. Подсказки qTip .....	289
10.3.8. Подсказки Easy Tooltip .....	289
10.3.9. Кнопки рейтинга .....	290
10.3.10. Загрузка файлов.....	290
10.3.11. Обрезка изображений .....	290
10.3.12. Меню в стиле MS Office .....	290
10.3.13. Ненавязчивые окна jGrow .....	291
10.3.14. Всплывающие подсказки BeautyTips .....	291
10.3.15. Меню в стиле Apple Mac .....	291
10.3.16. Карусель MovingBoxes .....	292
10.3.17. Меню Garage Door .....	292
10.3.18. Подключение Google Maps .....	293
10.3.19. Модуль валидации полей.....	293
10.3.20. Вращение предметов .....	293
10.4. JQuery CDN.....	293
<b>Глава 11. Получение данных из Интернета.....</b>	<b>295</b>
11.1. Получение файла из Интернета.....	295
11.2. Получение любых данных из Интернета.....	296
11.3. Получение веб-страницы .....	296

11.4. Использование <i>прокси-сервера</i> .....	297
11.5. Получить текущий курс валюты .....	297
11.6. Создание простого RSS-канала .....	297
11.7. AJAX .....	301
11.7.1. Что такое AJAX? .....	301
11.7.2. Получение серверных данных без AJAX .....	308
11.7.3. Вывод сообщений с помощью <i>UpdatePanel</i> .....	311
11.7.4. Вызов серверного метода с помощью AJAX .....	313

## **Глава 12. Базы данных, привязка данных..... 316**

12.1. Привязка данных .....	316
12.1.1. Сложная привязка данных с помощью <i>DataBinder</i> .....	316
12.1.2. Зависимая привязка данных .....	317
12.1.3. Привязка XML-данных .....	318
12.1.4. Привязка списка данных к выпадающему списку .....	320
12.1.5. Привязка перечислений ( <i>enum</i> ) .....	321
Добавление дополнительных элементов .....	322
Расширенная привязка перечислений .....	322
Вычисление имен значений перечисления .....	324
12.1.6. Привязка данных Access .....	326
12.1.7. Привязка данных к списку .....	326
12.1.8. Привязка данных с помощью LINQ .....	327
12.1.9. Привязка данных с разбивкой на страницы .....	328
12.2. Передача списка в SQL .....	331
12.3. Создание ссылки на файл, сохраненный в БД .....	332
12.3.1. Код обработчика .....	333
12.3.2. Регистрация обработчика .....	334
12.3.3. Код обработчика (второй вариант) .....	335
12.3.4. Структура БД для хранения файлов .....	336
12.3.5. Базовые классы .....	336
12.3.6. Загрузка файлов в БД .....	338
12.3.7. Получение файлов из БД .....	340
12.3.8. Создание ссылки на файл в БД .....	342
12.3.9. Типы файлов .....	344
12.3.10. Определение типа файла по расширению .....	344
12.3.11. Список поддерживаемых браузером типов .....	345
12.3.12. Регистрация своего расширения файла .....	346
12.3.13. Определить формат файла .....	346
12.3.14. Можно ли использовать в имени файла русские буквы? .....	348
12.3.15. Указание размера файла .....	348
12.3.16. Некоторые ограничения .....	348
12.4. Что следует использовать для закрытия соединения — <i>Close</i> или <i>Dispose</i> ? .....	349
12.5. Получение индекса объекта после добавления его в таблицу MS SQL .....	349

<b>Глава 13. Сессия, куки и хранение данных .....</b>	<b>350</b>
13.1. Как программно завершить сессию .....	350
13.2. Сообщение о завершении сессии .....	350
13.3. Сжатие данных в сессии (ASP.NET 4.0) .....	351
13.4. Отображение окна сообщения о завершении сессии .....	351
13.5. Непредсказуемое поведение сессии .....	352
13.6. Почему не вызывается <i>Session_End</i> .....	352
13.7. Подсчет числа посетителей сайта .....	352
13.8. Как получить доступ к сессии обычного класса .....	356
13.9. Как получить доступ к сессии из <i>HttpHandler</i> .....	356
13.10. Использование cookies .....	356
13.11. Что плохого в использовании сессий? .....	357
13.12. Настройка хранения сессий .....	358
13.13. Создание общей сессии между ASP- и ASP.NET-приложениями .....	359
13.14. Как не допустить закрытия сессии .....	359
13.15. Передача между страницами значений серверного элемента управления .....	361
13.16. Как перехватить загрузку и сохранение <i>ViewState</i> .....	363
13.17. Управление размером <i>ViewState</i> .....	363
13.18. Сжатие <i>ViewState</i> .....	364
13.19. Хранение <i>ViewState</i> на сервере .....	366
13.20. Управление <i>ViewState</i> в ASP.NET 4.0 .....	367
<b>Глава 14. Защита данных .....</b>	<b>368</b>
14.1. Шифрование конфигурации .....	368
14.1.1. Шифрование строки подключения .....	368
14.1.2. Шифрование секций web.config .....	369
14.1.3. Программное шифрование секций web.config .....	370
14.2. "Защитные" пароли .....	370
14.3. Защита от внедрения в SQL (SQL Injection) .....	370
14.4. Защита от внедрения в XML (XML Injection) .....	373
14.5. Защита от внедрения в строки запуска (DOS Injection) .....	373
14.6. Защита от внедрения кода в HTML (XSS) .....	374
14.6.1. Проверка подверженности сайта XSS .....	376
14.6.2. Защита от XSS .....	376
14.6.3. Защита от XSS в ASP.NET 2.0 .....	376
14.6.4. Библиотека Microsoft AntiXss .....	376
14.7. Ошибки в алгоритмах .....	377
14.8. Защита от разглашения информации .....	377
14.8.1. Забытые комментарии .....	377
14.8.2. Сообщения об ошибках .....	377
14.8.3. Трассировка .....	378
14.9. Защита паролей, хранящихся в БД .....	378
14.10. Защита от отказа в обслуживании (DOS) .....	379



14.11. Защита от перебора данных .....	380
14.11.1. Слабые пароли .....	380
14.11.2. Пароли по умолчанию .....	380
14.11.3. Блокировка подбора .....	381
14.11.4. Замедление проверок .....	381
14.11.5. Время жизни пароля .....	381
14.11.6. Очевидные ответы .....	382
14.12. Пассивная защита .....	382
14.13. Отсутствие автозакрытия сессии .....	382
14.14. Защита от перебора параметров .....	383
14.15. Защита файлов ресурсов .....	383
14.16. Защита ссылок .....	384
14.17. Создание CAPTCHA .....	385
14.18. Защита без CAPTCHA .....	401

## **Глава 15. Данные и отчеты MS Excel для веб-приложений ..... 403**

15.1. Способы взаимодействия с MS Excel .....	403
15.1.1. Использование библиотеки MS Excel .....	404
15.1.2. Формат CSV .....	405
15.1.3. Формат HTML .....	406
15.1.4. Формат XML .....	406
15.1.5. Использование OLE DB-провайдера .....	406
15.1.6. Формат Office 2008 .....	406
15.1.7. Бесплатные библиотеки .....	407
15.1.8. Платные библиотеки .....	407
15.2. Лицензионные ограничения MS Excel .....	407
15.3. Библиотека MS Excel .....	408
15.3.1. Раннее и позднее связывание .....	408
15.3.2. Сборки взаимодействия .....	408
15.3.3. Объектная модель Excel .....	410
Как найти нужные объекты .....	411
15.3.4. Раннее связывание .....	412
15.3.5. Позднее связывание .....	418
Как определять значения констант .....	423
15.3.6. Создание шаблона отчета .....	424
15.3.7. Быстрая вставка данных .....	428
15.4. Excel CSV, HTML и XML .....	429
15.4.1. Формат Excel/CSV .....	429
15.4.2. Формат Excel/HTML .....	432
15.4.3. Формат Excel/XML .....	435
15.4.4. Объединенный формат HTML и XML .....	443
15.4.5. Генерация Excel-документов в ASP.NET .....	445
15.5. Использование OLE DB .....	451
15.5.1. OLE DB-провайдер .....	451
15.5.2. OLE DB для платформы x64 .....	460

15.6. Формирование файлов в формате Excel 2008.....	460
15.6.1. Кратко о формате Office 2008 .....	460
15.6.2. Распаковка документа.....	462
15.6.3. Создание документа.....	464
15.6.4. Запись данных в документ .....	468
15.6.5. Использование блоков кода Microsoft.....	473
15.6.6. Использование утилиты DocumentReflector .....	477
15.6.7. Использование утилиты OpenXmlDiff.....	478
15.6.8. Описание констант MS Office 2008 .....	479

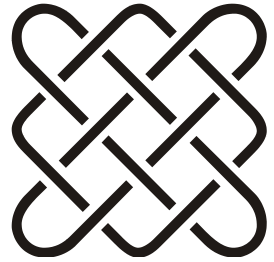
## **Глава 16. Инструменты и библиотеки..... 480**

16.1. Инструменты .....	480
16.1.1. Fiddler.....	480
16.1.2. Firebug.....	481
16.1.3. YSlow .....	482
16.1.4. Wireshark.....	482
16.1.5. SQL Server Profiler .....	484
16.1.6. Анализатор ссылок Xenu.....	484
16.1.7. HttpWatch.....	485
16.1.8. Отладчик Web Development Helper .....	485
16.1.9. Internet Explorer Developer Toolbar .....	485
16.1.10. Сайт Site-Perf.....	486
16.1.11. Doloto .....	487
16.1.12. Редактор IxEdit .....	487
16.1.13. jQueryPad .....	488
16.1.14. CSS-утилиты.....	489
16.1.15. UrlScan .....	489
16.1.16. Microsoft Ajax Minifier .....	489
16.2. Редакторы HTML-текста .....	490
16.2.1. CKEditor.....	490
16.2.2. Damn Small Rich Text Editor .....	490
16.2.3. TinyMCE .....	491
16.2.4. WYMeditor.....	491
16.2.5. widgEditor.....	491
16.2.6. jwysiwyg .....	492
16.2.7. NicEdit.....	492
16.2.8. Whizzywig.....	493
16.2.9. Yahoo! .....	493
16.2.10. markItUp!.....	493
16.2.11. eIRTE.....	494
16.3. Архиваторы.....	494
16.3.1. Библиотека SharpZipLib.....	494
16.3.2. Библиотека CAKE3 .....	495

---

16.4. Таблицы .....	495
16.4.1. Плагины jQuery .....	495
16.4.2. Таблица <i>AjaxDataControl</i> .....	495
16.5. Графики и диаграммы.....	496
16.5.1. Плагины jQuery .....	496
16.5.2. Графики и диаграммы MS Chart .....	496
16.5.3. Графики Google Chart .....	497
16.6. Разное .....	499
16.6.1. Библиотека Google Gears .....	499
16.6.2. CDN-сервисы.....	499
<b>Предметный указатель .....</b>	<b>504</b>

# ГЛАВА 1



## Архитектура и общие вопросы

"В будущем, компьютеры будут весить не больше 1,5 тонн".

*Журнал "Popular Mechanics", 1949.*

### 1.1. Основные отличия ASP.NET 1.1 и 2.0

- Для разработки проектов не требуется IIS и виртуальная директория.
- Для веб-проектов не используются файлы `.csproj`, которые раньше создавались для каждого из типов проектов ASP.NET. Вместо файлов проекта Visual Studio использует структуру директорий, таким образом, для того чтобы включить в проект существующий файл, достаточно просто скопировать его в директорию проекта. Если файл или директория удаляется из дерева файлов проекта, то они физически удаляются из файловой системы.
- Директива `@Page` имеет новые атрибуты:
  - `Async` — если значение этого атрибута равно `true`, то страница будет реализовывать интерфейс `IHttpAsyncHandler`, а иначе — интерфейс `IHttpHandler`. В первом случае код страницы может выполняться асинхронно;
  - `AsyncTimeout` — задает ограничение по времени, отведенное для выполнения асинхронных операций. По умолчанию этот параметр равен 45 секундам;
  - `Culture` — устанавливает набор региональных параметров, используемый для страницы (см. разд. 1.24);
  - `UICulture` — устанавливает набор региональных параметров, используемый для пользовательского интерфейса страницы (см. разд. 1.24);

- `EnableTheming` — позволяет включить или выключить поддержку тем оформления. По умолчанию включено;
- `StyleSheetTheme` — позволяет установить идентификатор темы оформления, которая будет использоваться для изменения установленной темы оформления (в атрибуте `Theme` или в файле `web.config`). Таким образом можно установить общую тему оформления для всего сайта, а с помощью атрибута `StyleSheetTheme` вносить некоторые изменения в общее оформление страницы и/или некоторых элементов управления, содержащихся на странице;
- `Theme` — указывает название темы оформления, которая будет использована для оформления кода данной страницы;
- `MasterPageFile` — указывает путь к шаблону, который будет использован для создания кода этой страницы. Шаблон называется "мастер-страницей" (см. разд. 2.18).

□ Класс `Page` имеет новые свойства и методы. Вот некоторые из них:

- `ClientScript` — объект типа `ClientScriptManager` для работы с клиентскими скриптами;
- `RegisterRequiresControlState` — регистрирует элемент, требующий сохранения состояния;
- `EnableTheming` — если значение равно `false`, то поддержка тем на странице отключена;
- `GetValidators` — метод, возвращающий коллекцию валидаторов данной страницы (см. главу 4);
- `Header` — ссылка на объект `HtmlHead`, позволяющий контролировать содержимое раздела `<head>` страницы, при условии, что он отмечен как серверный (`runat="server"`);
- `IsAsync` — если `true`, то страница будет работать в асинхронном режиме;
- `IsCrossPagePostBack` — свойство, позволяющее определить, была ли данная страница запрошена в ответ на отправку данных с другой страницы;
- `Master` — ссылка на экземпляр мастер-страницы (см. разд. 2.18);
- `MasterPageFile` — свойство, содержащее имя файла мастер-страницы (см. разд. 2.18);
- `MaxPageStateFieldLength` — см. разд. 3.17;
- `PreviousPage` — ссылка на экземпляр объекта страницы, с которой была осуществлена отправка формы;

- `SetFocus` — см. разд. 2.7;
  - `Title` — свойство, позволяющее получить и изменить заголовок страницы.
- Сменился цикл жизни страницы. В ASP.NET 1.1 последовательность событий была следующей: инициализация (`Init`), загрузка (`Load`), создание (`PreRender`) и завершение (`Unload`). После этого страница возвращалась клиенту. В ASP.NET 2.0 добавилось еще несколько событий:
- `PreInit` — вызывается до начала инициализации страницы;
  - `InitComplete` — вызывается после завершения инициализации страницы;
  - `LoadComplete` — вызывается после завершения загрузки страницы;
  - `PreRenderComplete` — вызывается после завершения создания всех элементов страницы.
- В ASP.NET 2.0 применена новая модель компиляции страниц "на лету". Код страниц представляет собой частичный класс (`partial class`) без описания серверных элементов управления. Эти описания вынесены в другую часть класса страницы, которая генерируется автоматически во время выполнения. Это позволяет разделить автоматически генерируемый код и код пользователя.

## 1.2. В ASP было...

Несколько ответов на вопрос "вот в ASP было так, а где это в ASP.NET":

- файл `global.asa` теперь называется `global.asax` и имеет значительно больше функциональности (см. разд. 1.20);
- эквивалент функции `date()` выглядит как:  

```
System.DateTime.Now.ToShortDateString()
```
- эквивалент функции `time()` выглядит как:  

```
System.DateTime.Now.ToShortTimeString()
```
- созданные объекты не нужно уничтожать самостоятельно, это делает сборщик мусора (однако файлы и другие ресурсы нужно освобождать сразу после завершения их использования).

Приведу несколько статей, описывающих миграцию с ASP на ASP.NET:

- <http://www.asp.net/downloads/archived/migration-assistants/asp-to-aspnet/>
- <http://www.asp101.com/articles/john/asptoaspnet/default.asp>
- <http://www.asp101.com/articles/paolo/asp2aspnet/default.asp>

## 1.3. Можно ли запустить приложение ASP.NET под Apache

Да, можно. См. [http://httpd.apache.org/cli/mod\\_aspdotnet](http://httpd.apache.org/cli/mod_aspdotnet).

## 1.4. Где найти исходный код Framework

См. ссылку <http://referencesource.microsoft.com/netframework.aspx>.

## 1.5. Использование SSI *include* в ASP.NET

Common\SSI<sup>1</sup>

Оператор `include` из языка SSI (Server Side Include, включение на стороне сервера) можно использовать, например, для создания блока скриптов, которые повторяются на всех страницах или регистрации элементов управления. В общем, для создания блоков текстовой информации, которая повторяется на многих страницах.

Например, каждая страница сайта должна содержать набор скриптов, описанный в файле `scripts.inc`:

```
<script type="text/javascript" src="Scripts/common.js"></script>
... другие файлы скриптов ...
```

Для включения содержимого этого файла в страницу используется конструкция `include`, заключенная в знаки HTML-комментариев:

```
<head runat="server">
  <title></title>
  <!-- #include virtual="~/Include/scripts.inc" -->
</head>
```

Теперь содержимое файла `scripts.inc` будет подставлено в указанное место страницы, и обновлять необходимый список скриптов будет легко и просто. Аналогично можно вставлять любые блоки текста. Только учтите, что реализовывать с помощью SSI единый дизайн страниц (как это делалось в классическом ASP) необходимости нет. Для этого есть более мощный механизм мастер-страниц (см. разд. 2.18). И, кроме того, учтите, что вставка происходит *после* компиляции страницы, поэтому использовать при этом серверный код не получится.

---

<sup>1</sup> Таким образом указывается название папки на прилагаемом к книге компакт-диске, содержащем полный и компилируемый исходный код.

И еще одно замечание. Включение файла можно сделать, например, так:

```
<% Response.WriteFile("~/Include/scripts.inc"); %>
```

Данная запись более близка к ASP.NET, но, как мне кажется, менее красива и проста.

## 1.6. Как узнать версию ASP.NET, под которой работает сайт

Вызов `System.Environment.Version.ToString()` возвращает версию ASP.NET.

## 1.7. Как узнать браузер и версию клиента, запустившего сайт

Свойство `Request.Browser.Browser` возвращает название браузера клиента, а `Request.Browser.Version` — полную строку версии браузера (свойства `MajorVersion` и `MinorVersion` возвращают соответственно старшую и младшую части строки версии браузера). Название браузера может быть, например, IE, Netscape, Opera и т. д.

## 1.8. Как узнать параметры компьютера, на котором работает сайт

С помощью класса `System.Environment` можно узнать информацию о компьютере, на котором работает сайт:

- `System.Environment.MachineName` — имя машины;
- `System.Environment.OSVersion` — версия операционной системы;
- `System.Environment.WorkingSet` — объем памяти;
- `HttpContext.Current.Server.MachineName` — имя машины;
- `HttpContext.Current.Request.ServerVariables["LOCAL_ADDR"]` — локальный IP-адрес машины.

## 1.9. Где расположен временный каталог

Определить расположение временного каталога можно с помощью вызова

```
Environment.GetEnvironmentVariable("TEMP")
```



## 1.10. Как изменить временный каталог ASP.NET

Временный каталог ASP.NET можно изменить с помощью настройки в глобальном файле `web.config`, который расположен в папке `%windows%\Microsoft.NET\Framework\v2.0.50727\CONFIG`.

Эта настройка задается параметром `tempDirectory`, например:

```
<compilation tempDirectory="C:\MyTemp">
```

## 1.11. Информация о соединении

Переменная `Request.IsLocal` возвращает `true`, если запрос локальный. Эта информация полезна, если нужно отобразить некоторые данные только для локальных пользователей (например, отладочную информацию, см. *разд. 14.8*).

Переменная `Request.IsSecureConnection` возвращает `true`, если запрос сделан через защищенное соединение. Проверка этой переменной — более правильный путь, чем сравнение начала URL-адреса со строкой `https`.

## 1.12. Зачем создается пользователь ASPNET

После установки ASP.NET вы можете увидеть созданную учетную запись пользователя ASPNET. Он используется в IIS5 процессом `aspnet_wp.exe`. Если вы не используете IIS5 (или IIS6 в режиме совместимости с IIS5), то можете просто удалить или запретить этот аккаунт. Процессом `w3wp.exe` это имя не используется.

## 1.13. Где сохранить данные при переходе между страницами

Для сохранения данных есть следующие варианты:

- адресная строка (URL);
- куки (cookies);
- скрытые поля (hidden fields);
- состояние страницы (ViewState);
- сессия (Session);
- приложение (Application).

Я имею в виду текущие данные пользователя, а не данные вообще, поэтому просто хранение данных в базе данных я здесь не рассматриваю (вопрос хранения данных в БД будет разбираться в *главе 12*).

Каждый из перечисленных вариантов имеет свои сильные и слабые стороны и при выборе нужно четко понимать их различия и в каких случаях какой из вариантов лучше.

### 1.13.1. Адресная строка

Некоторые данные можно передавать между страницами в *адресной строке*. Точнее сказать, эти данные будут являться параметрами вызывающей страницы, но это вполне можно считать передачей данных. При этом следует учитывать, что, во-первых, эти данные открыты для пользователя (некоторые соображения по безопасности я расскажу в *разд. 14.14*), а во-вторых, длина адреса не может превышать 1024 символа.

Считать данные из параметров страницы можно с помощью свойств `QueryString` и `PathInfo` (см. *разд. 8.13.1*).

### 1.13.2. Куки

Куки хранятся на клиентской стороне. В этом их преимущество — если работа с данными идет на клиентской стороне, то не нужно передавать данные на сервер. В этом и недостаток — клиент может легко найти сохраненные данные и подменить их. Не стоит хранить в куках пароли и другую конфиденциальную информацию.

Для работы с куками предусмотрены методы `Add` (добавить), `Get` (получить), `Clear` (очистить), `Remove` (удалить), `Set` (задать) класса `Request.Cookies`.

Например:

```
HttpCookie langCook = new HttpCookie("Language");
langCook.Value = newLang;
Response.Cookies.Add(langCook);
.....
if (Request.Cookies["Language"] != null)
{
if (Request.Cookies["Language"].Value != null)
    lang = string.Format(Request.Cookies["Language"].Value);
}
```

Такой пример я привел только ради простоты. В реальном коде обращение к кукам по имени лучше оборачивать в свойства, иначе одна опечатка в строке имени будет стоить не мало времени отладки (см. совет в *разд. 1.15.6*).

### 1.13.3. Скрытые поля

*Скрытые поля* использовались в классическом ASP для хранения данных пользователя. Точно так же можно использовать их и в ASP.NET. Нужно понимать, что скрытые поля хранятся в самой странице, а значит, во-первых, легко доступны пользователю, а во-вторых, большой объем данных утяжеляет страницу и увеличивает объем данных, передаваемых между клиентом и сервером. Не стоит класть внутрь страницы большие объемы данных. Не стоит хранить в скрытых полях пароли и секретную информацию.

Для доступа к скрытому полю из CS-кода нужно установить скрытому полю атрибут `runat="server"` и обращаться к такому полю по имени.

### 1.13.4. Состояние страницы

*Состояние страницы* (`ViewState`) по сути является тем же скрытым полем, поэтому к нему относится все то, что я перечислил ранее. Преимуществ состояния страницы несколько.

- Все серверные компоненты умеют работать с ним автоматически.
- Доступ к состоянию страницы очень простой — через индексатор `ViewState[имя]`.
- Состояние страницы сжимается, что значительно уменьшает объем передаваемых данных.
- ASP.NET 2.0 может производить контроль изменений состояния страницы. Если данные изменились будет сгенерировано исключение защиты.

К минусам относится увеличивающийся размер страницы (см. также *разд. 3.9.6* и *главу 13*).

### 1.13.5. Сессия

*Сессия* (`Session`) позволяет сохранять данные пользователя на то время, пока пользователь работает с сервером. Доступ к данным производится с помощью индексатора `Session[имя]`.

Приведу несколько полезных фактов "из жизни сессии".

- Данные в сессии живут ровно столько, сколько живет сама сессия пользователя. По умолчанию это 20 минут. Если в течение 20 минут пользователь не общается с сервером, сервер закрывает сессию и все данные удаляются. Очень сильно увеличивать это время не хорошо из соображений безопасности (см. *разд. 14.13*). Как одно из решений — на страницах требующих длительных операций можно добавить таймер, периодически "дергающий" сервер (см. *разд. 13.14*).

- ❑ ASP.NET позволяет гибко настраивать хранилище данных сессии. Сессию можно сохранить в куках, в памяти сервера и в БД (см. разд. 13.12).
- ❑ Данные, хранящиеся в сессии, доступны только "владельцу" этой сессии, если, конечно, данные не хранятся в БД, а приложение забирает их в обход всех правил.
- ❑ К плюсам хранения данных в сессии можно отнести возможность хранения существенных объемов информации (если, не хранить сессию в пользовательских куках). К минусам — часто довольно затруднительно определить момент, когда данные из сессии можно удалить.
- ❑ При хранении сессии в памяти сервера возможны проблемы с кластером серверов. Ведь у каждого сервера в кластере будет своя сессия пользователя. Решений этой проблемы два. Либо не хранить сессию в памяти сервера (или вообще не использовать сессию), либо настроить кластер таким образом, чтобы клиент, подключившись к конкретному серверу кластера один раз, всю сессию работал только с этим сервером.
- ❑ В ASP.NET 4.0 добавлен новый параметр `compressionEnabled`, который позволяет сжимать данные в сессии с помощью ZIP-алгоритма (см. разд. 13.3).

## 1.13.6. Память приложения

*Приложение* (Application) позволяет сохранять данные, общие для всех сессий, например, счетчик посетителей сайта. Доступ к данным приложения осуществляется с помощью индексатора `Application[имя]`.

Кроме того, любая переменная, описанная как `static`, будет храниться в области данных приложения, а значит, будет общей для всех пользователей сайта. Учитывайте это при разработке приложений (см. разд. 1.15.13)!

Аналогично сессиям, в использовании памяти приложения есть проблема, связанная с работой на кластере серверов. Данные приложения хранятся в памяти IIS, поэтому для каждого сервера кластера это будет *своя* область памяти. Нет ничего страшного, если это используется для кэширования — все сведется к тому, что на каждом сервере кластера будет загружен свой кэш. Но если данные приложения хранят общую для всех пользователей (точнее — для всех сессий) информацию, например, счетчик пользователей, то это окажется проблемой — каждый сервер кластера будет иметь свой собственный счетчик. В таких случаях придется обойтись без использования памяти приложения и хранить такие счетчики (и вообще общие данные) где-то в БД.

### 1.13.7. Что же выбрать

Выбор конкретного хранилища зависит от нескольких факторов:

1. Должны ли данные быть доступны всем пользователям сервера?
2. Каков объем данных?
3. Есть ли в данных конфиденциальная информация?
4. Должны ли данные храниться дольше, чем время жизни страницы?
5. Будет ли приложение работать на кластере?

Если данные должны быть доступны всем пользователям сервера, то единственный вариант, предоставляемый ASP.NET, — это память приложения (Application). Нужно только учитывать особенность работы приложения на кластере. Если приложение планируется масштабировать, то лучше сразу хранить данные в БД и не использовать другие варианты.

Если объем данных небольшой, эти данные используются только одной страницей и не являются секретом для пользователя (например, номер выбранной страницы в таблице данных), то лучше воспользоваться адресной строкой, данными страницы или куками.

Куки удобно использовать для неконфиденциальных данных, работа с которыми ведется только на стороне клиента.

Если предполагается хранить существенный объем данных или конфиденциальные данные, то лучше хранить их в сессии (конечно, если сессия не хранится в куках).

## 1.14. Не используйте подчеркивание в имени серверов

Согласно стандарту RFC 952 (<http://www.ietf.org/rfc/rfc952.txt>) подчеркивание не является разрешенным символом в имени серверов. При его использовании в ASP.NET будут проблемы с сессией и куками — сессия будет работать не предсказуемо.

## 1.15. Общие правила создания страниц

### 1.15.1. Не путайте разметку и код

Архитектура веб-страницы в ASP.NET предполагает разделение разметки и кода, но очень часто это правило нарушается. Например, приветствие выводится так: