**Класс WebResponse**

В классе WebResponse инкапсулируется ответ, получаемый по запросу. Этот класс

является абстрактным. В наследующих от него классах создаются отдельные его версии,

поддерживающие конкретный протокол. Объект класса WebResponse обычно

получается в результате вызова метода GetResponse(), определенного в классе

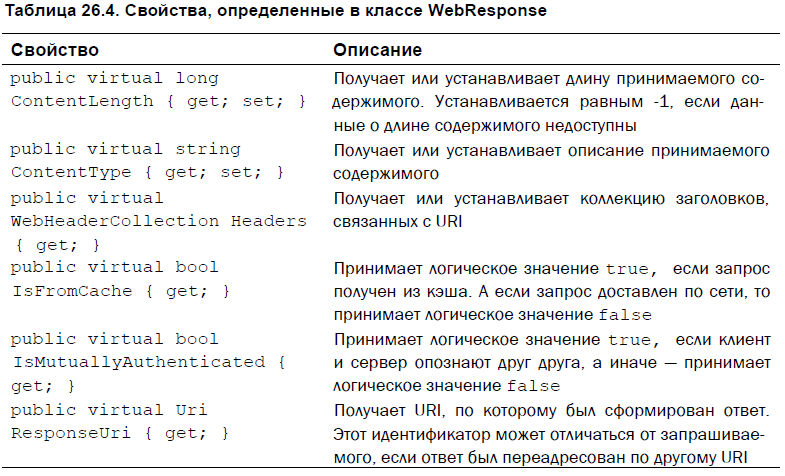
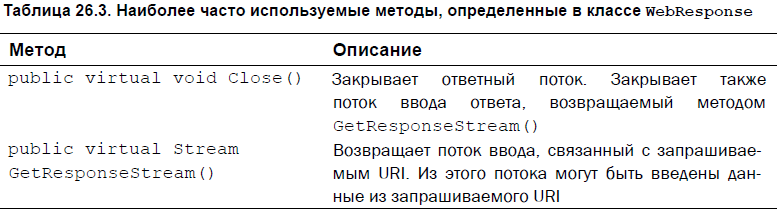
WebRequest. Этот объект будет экземпляром отдельного класса, производного от

класса WebResponse и реализующего конкретный протокол. Методы, определенные

в классе WebResponse, сведены в табл. 26.3, а свойства, объявляемые в этом классе, —

в табл. 26.4. Значения этих свойств устанавливаются на основании каждого запроса в

отдельности. Открытые конструкторы в классе WebResponse не определяются.



**Доступ к дополнительной информации,**

**получаемой в ответ по протоколу HTTP**

С помощью сетевых средств, имеющихся в классе HttpWebResponse, можно получить

доступ к другой информации, помимо содержимого указываемого ресурса.

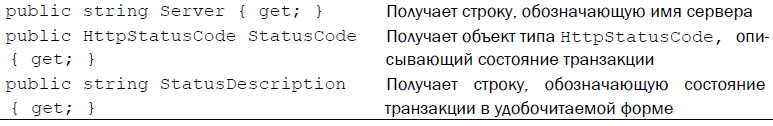
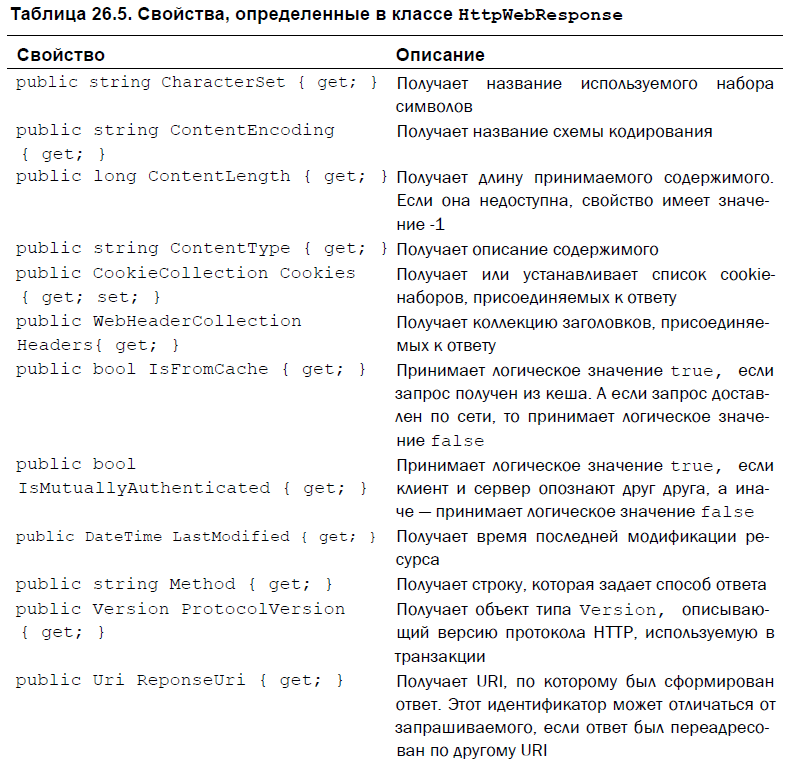
К этой информации, в частности, относится время последней модификации ресурса,

а также имя сервера. Она оказывается доступной с помощью различных свойств, связанных

с подучаемым ответом. Все эти свойства, включая и те что, определены в классе

WebResponse, сведены в табл. 26.5. В приведенных далее примерах программ демонстрируется

применение этих свойств на практике.



**Доступ к заголовку**

Для доступа к заголовку с информацией, получаемой в ответ по протоколу HTTP,

служит свойство Headers, определенное в классе HttpWebResponse.

public WebHeaderCollection Headers { get; }

Заголовок протокола HTTP состоит из пар "имя-значение", представленных строками.

Каждая пара "имя-значение" хранится в коллекции класса WebHeaderCollection.

Эта коллекция специально предназначена для хранения пар "имя-значение" и применяется аналогично любой другой коллекции (подробнее об этом см. в главе 25). Строковый

массив имен может быть получен из свойства AllKeys, а отдельные значения —

по соответствующему имени при вызове метода GetValues(). Этот метод возвращает

массив строк, содержащий значения, связанные с заголовком, передаваемым в качестве

аргумента. Метод GetValues() перегружается, чтобы принять числовой индекс или

имя заголовка.

В приведенной ниже программе отображаются заголовки, связанные с сетевым ресурсом,

доступным по адресу [www.McGraw-Hill.com](http://www.McGraw-Hill.com).

(***glava26\_3***)

class HeaderDemo

{

static void Main()

{

//create request object type WebRequest certain URI

HttpWebRequest req = (HttpWebRequest)

WebRequest.Create("http://www.McGraw-Hill.com");

//send request and get response

HttpWebResponse resp = (HttpWebResponse)

req.GetResponse();

//get name list

string[] names = resp.Headers.AllKeys;

//show pairs name - value from header

Console.WriteLine("{0, -20} {1}\n", "Name", "Value");

foreach (var n in names)

{

Console.Write("{0, -20}", n);

foreach (var v in resp.Headers.GetValues(n))

Console.WriteLine(v);

}

//close stream

resp.Close();

}

}

**Доступ к cookie-наборам**

Для доступа к cookie-наборам, получаемым в ответ по протоколу HTTP, служит

свойство Cookies, определенное в классе HttpWebResponse. В cookie-наборах

содержится информация, сохраняемая браузером. Они состоят из пар "имя-значение"

и упрощают некоторые виды доступа к веб-сайтам. Ниже показано, каким образом

определяется свойство Cookies.

public CookieCollection Cookies { get; set; }

В классе CookieCollection реализуются интерфейсы ICollection и

IEnumerable, и поэтому его можно использовать аналогично классу любой другой

коллекции (подробнее об этом см. в главе 25). У этого класса имеется также индексатор,

позволяющий получать cookie-набор по указанному индексу или имени.

В коллекции типа CookieCollection хранятся объекты класса Cookie. В классе

Cookie определяется несколько свойств, предоставляющих доступ к различным фрагментам

информации, связанной с cookie-набором. Ниже приведены два свойства, Name

и Value, используемые в примерах программ из этой главы.

public string Name { get; set; }

public string Value { get; set; }

Имя cookie-набора содержится в свойстве Name, а его значение — в свойстве

Value.

Для того чтобы получить список cookie-наборов из принятого ответа, необходимо

предоставить cookie-контейнер с запросом. И для этой цели в классе HttpWebRequest

определяется свойство CookieContainer, приведенное ниже.

public CookieContainer CookieContainer { get; set; }

В классе CookieContainer предоставляются различные поля, свойства и методы,

позволяющие хранить cookie-наборы. По умолчанию свойство CookieContainer

содержит пустое значение. Для того чтобы воспользоваться cookie-наборами, необходимо

установить это свойство равным экземпляру класса CookieContainer. Во

многих приложениях свойство CookieContainer не применяется непосредственно,

а вместо него из принятого ответа составляется и затем используется коллекция типа

CookieCollection. Свойство CookieContainer просто обеспечивает внутренний

механизм сохранения cookie-наборов.

В приведенном ниже примере программы отображаются имена и значения cookie-

наборов, получаемых из источника по URI, указываемому в командной строке. Следует,

однако, иметь в виду, что cookie-наборы используются не на всех веб-сайтах, поэтому

нужно еще найти такой веб-сайт, который поддерживает cookie-наборы.

(***glava26\_4***)

using System;

using System.Net;

class CookieDemo

{

static void Main(string[] args)

{

if(args.Length != 1)

{

Console.WriteLine("Do: CookieDemo <uri>");

return;

}

//requets object WebRequest certain URI

HttpWebRequest req = (HttpWebRequest)

WebRequest.Create(args[0]);

//get empty container

req.CookieContainer = new CookieContainer();

//sen request and get response

HttpWebResponse resp = (HttpWebResponse)

req.GetResponse();

//show cookie - collections

Console.WriteLine("Quantity of cookie-collections: "

+ resp.Cookies.Count);

Console.WriteLine("{0, -20} {1}", "Name", "Value");

for (int i = 0; i < resp.Cookies.Count; i++)

Console.WriteLine("{0, -20} {1}",

resp.Cookies[i].Name,

resp.Cookies[i].Value);

//close source

resp.Close();

}

}

**Применение свойства LastModified**

Иногда требуется знать, когда именно сетевой ресурс был обновлен в последний

раз. Это нетрудно сделать, пользуясь сетевыми средствами класса HttpWebResponse,

среди которых определено свойство LastModified, приведенное ниже.

public DateTime LastModified { get; }

С помощью свойства LastModified получается время обновления содержимого сетевого

ресурса в последний раз.

В приведенном ниже примере программы отображаются дата и время, когда был в

последний раз обновлен ресурс, указываемый по URI в командной строке.

(***LastModifiedDemo***)

class LastModifiedDemo

{

static void Main(string[] args)

{

if(args.Length != 1)

{

Console.WriteLine("Do: LastModifiedDemo <uri>");

return;

}

HttpWebRequest req = (HttpWebRequest)

WebRequest.Create(args[0]);

HttpWebResponse resp = (HttpWebResponse)

req.GetResponse();

Console.WriteLine("Last modification: " + resp.LastModified);

resp.Close();

}

}