

7

Web-каналы и социальные закладки

Вы только что разместили на своем сайте новую замечательную информацию. *И что теперь?* Конечно, вашим посетителям этот контент нравится. Они даже могут рассказать о нем своим друзьям. Но существуют технологии, которые можно задействовать для того, чтобы они занялись для вас бесплатным маркетингом.

В данной главе рассматриваются Web-каналы и социальные закладки — две технологии, которые могут использовать посетители Web-сайтов для доступа к понравившемуся им контенту и его распространения. Поощрение посетителей на выполнение таких действий — необходимое условие жизнеспособного маркетинга. В данной главе будут рассмотрены различные способы достижения этой цели и предложены три упражнения, в которых вы:

- создадите собственные RSS-каналы;
- будете распространять RSS- и Atom-каналы;
- добавите в свои страницы и каналы значки служб социальных закладок.

Web-каналы

Web-канал (web feed) — это механизм распространения контента по Web-сети в формате на основе XML, без какого-либо визуального сопровождения. Обычно люди читают информацию, посещая Web-страницы с контентом в формате HTML. Но в случае Web-каналов это не так, т.к. они не содержат средств оформления. Вместо этого применяются специальные программы, которые выбирают и отображают данные.

Каналы предназначены для автоматического распространения информации — как для людей, так и для других Web-сайтов. Это очень эффективное средство для распространения информации стало весьма популярным, т.к. позволяет читать новости или свежие сообщения в блогах из понравившихся источников. Хорошее описание Web-каналов имеется по адресу http://en.wikipedia.org/wiki/Web_feed.

Для работы с каналами разработано несколько приложений. Современные Web-браузеры (в том числе Internet Explorer 7 и Firefox 2.0), настольные приложения типа Microsoft Office 2007 и Web-приложения наподобие Google Reader (<http://www.google.com/reader/>) позволяют пользователям просматривать каналы, на которые они подписались, из одного удобного места. Эта приложения называются агрегаторами или программами чтения каналов.

Ваш Web-сайт может обеспечить с помощью Web-каналов доступ к части контента или ко всему контенту. В них могут присутствовать ссылки на сам контент, а также другие ссылки, указывающие на любые места сайта. Постепенно вы накопите трафик и ссылки у пользователей, которые подписались на ваши каналы, а также на различных сайтах, которые *распространяют* (syndicate) информацию.

Web-распространение (web syndication) разрешает другим Web-сайтам продвигать ваш контент. Другие Web-мастера охотно распространяют информацию каналов на своих сайтах в качестве свежего контента, т.к. *умеренное* включение релевантного распространяемого контента может принести существенную пользу. Но, пожалуй, лучше сократить количество информации, поставляемой в канале, т.к. *полный* контент, присутствующий на различных сайтах, может привести к дублированию контента. Вы можете также распространять контент с других Web-сайтов.

“Умеренно” означает, что Web-сайт должен содержать не только чужую распространяемую информацию. В противном случае это может быть сайт рассылки спама.

В настоящее время все основные платформы ведения блогов предоставляют какие-то средства ведения каналов. Такие средства имеются и в большинстве других видов систем управления контентом. Разрабатываемые вами специализированные приложения также можно усовершенствовать, добавив эти средства. Как это сделать, будет показано в данной главе.

Чтобы информацию из каналов могли читать все, она должна быть организована в стандартном формате. Наиболее популярные варианты таких форматов – RSS и Atom.

RSS и Atom

К сожалению, как обычно, и тут не обошлось без войны форматов. Для Web-распространения имеется много конкурирующих форматов. Здесь будут рассмотрены два из них – RSS и Atom.

Оба эти стандарта основаны на XML. Язык XML ценен тем, что он предоставляет общую канву, с помощью которой могут сообщаться приложения, работающие в различных архитектурах и операционных системах. Каналы RSS и Atom можно просматривать и как обычные текстовые файлы, но лучше так не делать, т.к. они предназначены для просмотра с помощью программ чтения каналов или специальных программ, входящих в состав больших приложений. На рис. 7.1 показан канал SEO Egghead Джейми Сировича в списке Google Reader одного из посетителей.

У RSS долгая и запутанная история, полная различных версий и существенных модификаций стандарта. Имеются две фундаментальных разновидности RSS с различными названиями. Вариант RDF Site Summary (RSS 0.9) был разработан компанией Netscape в конце 90-х годов. В ответ на нарекания в излишней сложности была выпущена упрощенная и заметно отличающаяся версия – RSS 0.91. Что интересно, вариант RSS 1.0 в основном базируется на RSS 0.9, а RSS 2.0 больше похож на RSS 0.91. В версии 2.0 “RSS” теперь означает Really Simple Syndication (Действительно простое распространение), а RSS 1.0 по-прежнему означает RDF Site Summary. Поскольку мы пишем не книгу по истории RSS, здесь мы остановимся и скажем лишь, что RSS 2.0 в настоящее время является наиболее популярным и распространенным стандартом. Пока что этот стандарт заморожен, и разработка новых изменений не ведется. Стандарт RSS 2.0 описан по адресу <http://blogs.law.harvard.edu/tech/rss>.

Вариант Atom был разработан из-за проблем, которые постепенно накопились в стандарте RSS. Он появился в 2003 г. и существует в двух версиях – Atom 0.3 и

Atom 1.0. Он гораздо более стандартизован, но и гораздо более сложен и реже применяется. Однако в последнее время Atom начал набирать силу. Более подробное сравнение RSS и Atom см. по адресу <http://www.intertwingly.net/wiki/pie/Rss20AndAtom10Compared>.

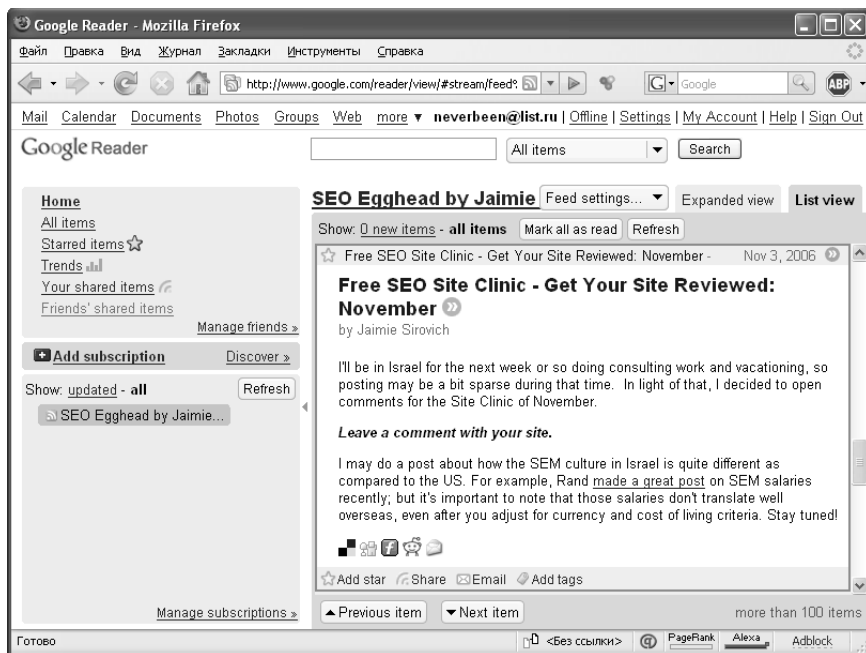


Рис. 7.1. Канал SEO Egghead Джейми Сировича

Мы не отдаем предпочтения ни одному из этих стандартов. RSS 2.0 более распространен, проще по многим параметрам, и потому именно в этом формате мы будем демонстрировать создание Web-канала. Но для распространения каналов вы воспользуетесь PHP-библиотекой под название SimplePie, которая прозрачно читает *все* версии каналов RSS и Atom.

Типичный канал RSS 2.0 может выглядеть примерно так:

```
<rss version="2.0">
  <channel>
    <title>Потрясающие новости на example.com</title>
    <link>http://www.example.org</link>
    <description>Краткое описание этого канала</description>
    <language>ru</language>
    <pubDate>Tue, 12 Sep 2006 07:56:23 EDT</pubDate>
  <item>
    <title>Хлесткий заголовок</title>
    <link>http://www.example.org/catchy-title.html</link>
    <description>
      Описание может содержать любой контент, включая и XHTML.
    </description>
    <pubDate>Tue, 12 Sep 2006 07:56:23 EDT</pubDate>
  </item>
</item>
</rss>
```

```

<title>Еще один хлесткий заголовок</title>
<link>http://www.example.org/another-catchy-title.html</link>
<description>
  Описание может содержать любой контент, включая и XHTML.
</description>
<pubDate>Tue, 12 Sep 2006 07:56:23 EDT</pubDate>
</item>
</channel>
</rss>

```

Канал может содержать любое количество элементов `<item>`, каждый из которых содержит различные новости, или статьи блога, или другой контент, который вы хотите распространять.

Вы можете либо создавать каналы для чтения их другими, либо распространять каналы, созданные другими. В следующем разделе описано, как создавать каналы, а в разделе, следующем за ним — как использовать стороннюю библиотеку SimplePie для распространения каналов.

Создание RSS-каналов

Чтобы облегчить генерацию каналов для вашего контента, вы создадите класс под названием “RSS factory” (Генератор RSS). Здесь мы впервые будем применять объектно-ориентированное программирование (ООП).

В предыдущих упражнениях мы старались не использовать поддержку объектно-ориентированного программирования в PHP. Здесь она применяется потому, что она действительно облегчает нашу задачу. Некоторые объяснения по его использованию будут даны после описания упражнения.

Класс имеет незатейливое название `RSSFactory` и содержит все функции, необходимые для создания RSS-каналов. Вначале вы реализуете этот класс, а затем, в следующем упражнении, создадите с его помощью канал для “новых наворотов SEO”.

Создание генератора RSS

1. Создайте в папке `seophp/include` новый файл с именем `rss_factory.inc.php`. Этот файл будет содержать класс генератора RSS. Поместите в него следующий код:

```

<?php
class RSSFactory
{
  var $_title;
  var $_link;
  var $_description;
  var $_language;
  var $_items;

  // литерализация символов строки для включения в структуру XML
  function _escapeXML($str)
  {
    $translation = get_html_translation_table(HTML_ENTITIES, ENT_QUOTES);
    foreach ($translation as $key => $value)
    {

```

```

    $translation[$key] = '&#' . ord($key) . ';';
}
$translation[chr(38)] = '&';
return preg_replace("/&(?![A-Za-zA-Яа-я]{0,4}\w{2,3};|#[0-9]{2,3};)/",
    "&#38;", strtolower($str, $translation));
}
// конструктор класса - выполняется при создании экземпляра этого класса
function RSSFactory($title, $link, $description,
    $language = 'ru', $items = array())
{
    // сохранение данных канала в локальных членах класса
    $this->_title = $title;
    $this->_link = $link;
    $this->_description = $description;
    $this->_language = $language;
    $this->_items = $items;
}
// добавление в канал нового элемента
function addItem($title, $link, $description, $additional_fields = array())
{
    // добавление элемента канала
    $this->_items[] =
        array_merge(array('title' => $title,
            'link' => $link,
            'description' => $description),
            $additional_fields);
}
// генерация канала
function get()
{
    // подготовка
    ob_start();
    header('Content-type: text/xml');
    // создание заголовка канала
    echo '<rss version="2.0">' .
        '<channel>' .
            '<title>' . RSSFactory::_escapeXML($this->_title) . '</title>' .
            '<link>' . RSSFactory::_escapeXML($this->_link) . '</link>' .
            '<description>' .
                RSSFactory::_escapeXML($this->_description) .
            '</description>';
    // добавление элементов канала
    foreach ($this->_items as $feed_item)
    {
        // добавление элемента канала и его содержимого
        echo '<item>';
        foreach ($feed_item as $item_name => $item_value)
        {
            // вывод элемента канала
            echo "<$item_name>" .
                RSSFactory::_escapeXML($item_value) .
                "</$item_name>";
        }
        echo '</item>';
    }
}

```

```

// закрытие канала и элементов RSS
echo '</channel></rss>';
// возврат данных канала
return ob_get_clean();
}
}
?>

```

- Создайте в папке seophp новый файл с именем feed.php и поместите в него следующий код:

```

<?php
// загрузка библиотеки генератора URL
require_once 'include/rss_factory.inc.php';
// создание канала
$rss_feed = new RSSFactory('Канал новых продуктов SEOEgghead.com',
    'http://www.seoegghead.com/seo-with-php-updates.html',
    'Замечательные новые продукты, ежедневное обновление');
// добавление элемента канала
$rss_feed->addItem('Новый сок из ссылок с ароматом апельсинов!',
    'http://seophp.example.com/Products/SEO-Toolbox-C6/Link-Juice-P31.
html',
    'Новый продукт - сок из ссылок от SEOEgghead.com - преобразит ваш Web-сайт!');
// добавление элемента канала
$rss_feed->addItem('Усовершенствуйте свои PHP-приложения с помощью AJAX!',
    'http://seophp.example.com/Products/Friends-Shed-C2/AJAX-PHP-Book-P42.html',
    'Прочтите это руководство по AJAX для PHP-разработчиков!');
// вывод канала
echo $rss_feed->get('2.0');
?>

```

- Загрузите страницу `http://seophp.example.com/feed.php`. Современный Web-браузер должен спросить, хотите ли вы подписаться на этот канал (рис. 7.2).

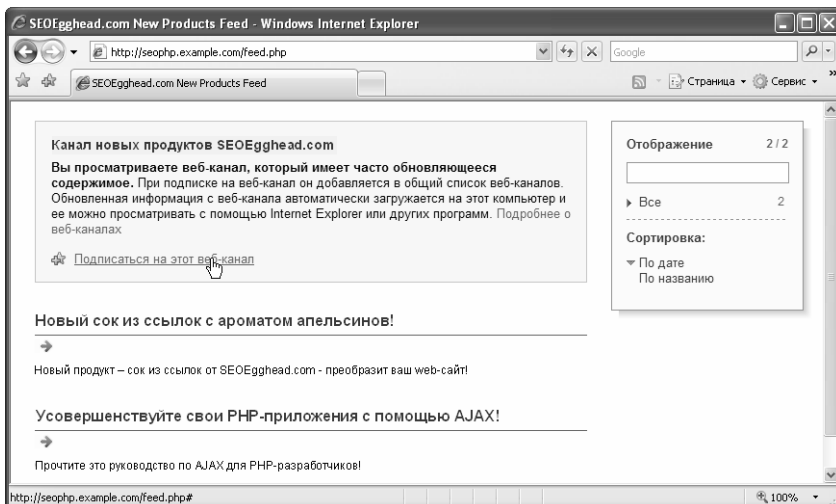


Рис. 7.2. Запрос о необходимости подписки на канал

4. Щелчок на ссылке Подписаться на этот веб-канал открывает диалоговое окно, в котором можно выбрать параметры подписки. В Mozilla Firefox этот диалог выглядит так, как показано на рис. 7.3, но его вид зависит от применяемого браузера.

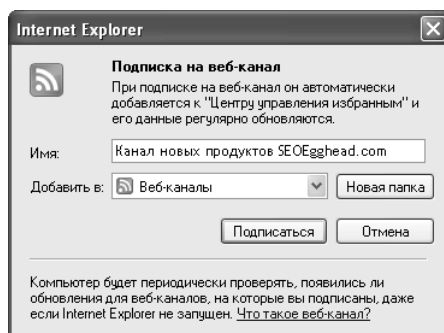


Рис. 7.3. Диалоговое окно подписки на канал

5. После оформления подписки на канал у вас будет быстрый доступ к последним новостям с помощью вашего приложения чтения каналов.

Вы только что создали *класс* RSSFactory и быстренько проверили его, для чего объект RSSFactory (\$rss_feed) сгенерировал простой канал в формате RSS.

Класс? Объект? Введение в ООП

Термин *класс* взят из лексикона объектно-ориентированного программирования (ООП). На случай, если вы не слишком знакомы с ООП, мы приводим краткое введение, которое поможет разобраться в RSSFactory, а также в других примерах этой книги, в которых используется эта возможность.

Как можно догадаться из названия, в модели ООП все основано на объектах. *Объект* — наиболее важная концепция в мире ООП — представляет собой изолированную сущность, у которой есть состояние и поведение, как и у объектов реального мира.

Класс выступает в качестве “шаблона” объекта, а объект является экземпляром класса, определенным его состоянием. Можно иметь много объектов, к примеру, для класса Автомобиль и создать столько объектов Автомобиль, сколько требуется — с именами \$мойАвтомобиль, \$васинАвтомобиль, \$петинАвтомобиль и т.д. Но \$васинАвтомобиль может проходить техосмотр, в то время как \$петинАвтомобиль будет уходить от погони по скоростной трассе, выжимая 180 км/ч.

Ну, вы поняли: класс определяет *функциональность*, предоставляемую объектами. Все объекты типа Автомобиль будут иметь одни и те же основные возможности — например, возможность изменять скорость движения. Но каждый отдельный объект Автомобиль в любой конкретный момент времени может иметь на спидометре отличную от других скорость.

Состояние объекта описывается различными его полями, называемыми также “свойствами”. А его возможности описываются его “методами”. Эти методы представляют собой функции внутри класса, но только они вызываются через (->) объект и существуют в контексте состояния, определяемого средствами объекта.

В нашем упражнении класс называется RSSFactory, а создаваемый объект — \$rss_feed. Если нужно вызвать метод addItem объекта \$rss_feed, то это выполняется с помощью записи \$rss_feed->addItem. В упражнении такое действие выполняется дважды для добавления двух элементов канала. И, наконец, для вывода канала используется оператор echo \$rss_feed->get(), который отображает канал в соответствии с добавленными в него элементами.

Использовать этот класс совсем не трудно. Вначале необходимо указать файл `rss_factory.inc.php` (в котором описан класс) и создать объект `rss_feed`:

```
<?php
// загрузка библиотеки генератора URL
require_once 'include/rss_factory.inc.php';

// создание канала
$rss_feed = new RSSFactory('Канал новых продуктов SEOEgghead.com',
    'http://www.seoegghead.com/seo-with-php-updates.html',
    'Замечательные новые продукты, ежедневное обновление');
```

Ваш класс называется `RSSFactory`, а объект — `rss_feed`. Объект создается с помощью операции `new` и, как вы видите, в скобках после имени класса передаются параметры.

Эти параметры, указываемые при создании объекта, передаются *конструктору* класса. Конструктор — это специальный метод класса, который автоматически вызывается при создании объекта. Он позволяет выполнить начальные настройки на основе переданных параметров. В нашем случае при создании объекта `RSSFactory` конструктору передается название канала, ссылка и описание. Эти атрибуты характеризуют сам канал вообще, а не его отдельные элементы. Определение конструктора в классе `RSSFactory` выглядит так:

```
// конструктор класса - выполняется при создании экземпляра этого класса
function RSSFactory($title, $link, $description,
    $language = 'ru', $items = array())
{
    // сохранение данных канала в локальных членах класса
    $this->_title = $title;
    $this->_link = $link;
    $this->_description = $description;
    $this->_language = $language;
    $this->_items = $items;
}
```

Так что конструктор сохраняет значения параметров в свойствах объекта и таким образом задает его состояние. Обратите внимание на использование `$this`, что внутри класса означает “данный экземпляр класса”. (При создании нескольких объектов типа `RSSFactory` ссылка `$this` будет различной в каждом из этих объектов.)

После создания объекта `$rss_feed` в сценарии `feed.php` два раза вызывается метод `addItem`, чтобы добавить в канал два элемента:

```
// создание канала
$rss_feed = new RSSFactory('Канал новых продуктов SEOEgghead.com',
    'http://www.seoegghead.com/seo-with-php-updates.html',
    'Замечательные новые продукты, ежедневное обновление');

// добавление элемента канала
$rss_feed->addItem('Новый сок из ссылок с ароматом апельсинов!',
    'http://seophp.example.com/Products/SEO-Toolbox-C6/Link-Juice-P31.html',
    'Новый продукт - сок из ссылок от SEOEgghead.com - преобразит ваш Web-сайт!');

// добавление элемента канала
$rss_feed->addItem('Усовершенствуйте свои PHP-приложения с помощью AJAX!',
    'http://seophp.example.com/Products/Friends-Shed-C2/AJAX-PHP-Book-P42.html',
    'Прочтите это руководство по AJAX для PHP-разработчиков!');
```


И, наконец, вызывается метод `get()`, чтобы вывести структуру RSS-канала. Выходные данные этого упражнения будут следующими:

```
<rss version="2.0">
  <channel>
    <title>Канал новых продуктов SEOEgghead.com</title>
    <link>http://www.seoegghead.com/blog/seo-with-php-updates/</link>
    <description>Замечательные новые продукты, ежедневное обновление</description>
    <item>
      <title>Новый сок из ссылок с ароматом апельсинов!</title>
      <link>
        http://seophp.example.com/Products/SEO-Toolbox-C6/Link-Juice-P31.html
      </link>
      <description>
        Новый продукт - сок из ссылок от SEOEgghead.com - преобразит ваш Web-сайт!
      </description>
    </item>
    <item>
      <title>Усовершенствуйте свои PHP-приложения с помощью AJAX!</title>
      <link>
        http://seophp.example.com/Products/Friends-Shed-C2/
        ↪ PHP-E-Commerce-Book-P42.html
      </link>
      <description>
        Прочтите это руководство по AJAX для PHP-разработчиков!
      </description>
    </item>
  </channel>
</rss>
```

Чтобы раскрутить свой Web-канал, вы должны сделать его хорошо заметным на вашем сайте. В главе 18 вы также узнаете, как поместить в блог WordPress “чиклеты” (chicklet), которые облегчают добавление вашего Web-сайта в отдельные средства чтения каналов на базе Web-приложений.

Распространение RSS- и Atom-каналов

Как уже было сказано, для Web-каналов используется много различных стандартов. Однако, как вы видели, имеется и множество средств, которые помогают следить за поступлениями выбранных каналов. Эти средства позволяют полностью забыть о войнах форматов. Но если нужно читать и анализировать информацию из внешних каналов программным способом, то положение усложняется.

К счастью, компания Skyzyx Technologies (<http://www.skyzyx.com/>) разработала PHP-библиотеку под названием SimplePie (<http://simplepie.org/>), которая абстрагирует для программиста все подробности и предоставляет общий API-интерфейс, пригодный для всех типов и версий каналов. Разработчики говорят вот что:

“SimplePie — очень быстрый и легкий для применения класс, написанный на PHP, для чтения RSS- и Atom-каналов. Уделив основное внимание простоте, и оставив только то, что важно, мы создали очень удобный маленький API. Цель создания SimplePie состоит в скорости и удобстве, и нам удалось и то, и другое”.

Инструкции по инсталляции SimplePie находятся по адресу <http://simplepie.org/docs/installation/getting-started/>, а руководство — по адресу <http://simplepie.org/docs/installation/from-scratch/>.

Сейчас вы выполните небольшое упражнение по SimplePie. В нем вы создадите страницу, которая использует SimplePie для чтения только что созданного канала и выводит информацию для ваших посетителей.

Чтение каналов с помощью SimplePie

1. Для работы SimplePie требуется несколько библиотек (рис. 7.4). Если вы подготовили Apache и PHP так, как было объяснено в главе 1, у вас должны быть установлены и активизированы все необходимые библиотеки, кроме cURL. Для активации cURL откройте конфигурационный файл `php.ini` (по умолчанию находится в папке `\xampp\apache\bin`), снимите комментарий со следующей строки, удалив в ее начале точку с запятой, и затем перезапустите Apache:

```
extension=php_curl.dll
```
2. Загрузите пакет SimplePie по адресу <http://simplepie.org/downloads/>.
3. Распакуйте загруженный файл с именем наподобие `simplepie_версия.zip` куда-нибудь на жесткий диск, а потом скопируйте нужный нам PHP-файл с именем `simplepie.inc` в папку `seophp/include`.
4. SimplePie очень дружелюбен к разработчику. Кроме отличной документации, с SimplePie поставляется также сценарий, который проверяет, поддерживает ли ваша инсталляция PHP эту библиотеку. Если вы не уверены, поддерживает ли ваша машина SimplePie, скопируйте в папку `seophp` файл `sp_compatibility_test.php` из загруженного пакета. Затем загрузите этот файл в Web-браузер, и вы получите полную оценку (см. рис. 7.4).



Рис. 7.4. Библиотеки, необходимые для работы SimplePie

- Создайте в папке seophp папку с именем cache. Эта папка будет использована SimplePie для кэширования своих данных.
- Создайте в папке seophp новый файл с именем read_feed.php и поместите в него следующий код:

```

<?php
// загрузка библиотеки SimplePie
require_once 'include/simplepie.inc';

// создание и настройка объекта SimplePie
$feed = new SimplePie();
$feed->feed_url('http://seophp.example.com/feed.php');
$feed->cache_location('cache');
$feed->init();
$feed->handle_content_type();
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html>
  <head>
    <title>Проверка чтения канала</title>
  </head>
  <body>

<?php
if ($feed->data)
{
  // вывод названия
  echo '<h1>' .
    '<a href="' . $feed->get_feed_link() . '"' .
      $feed->get_feed_title() .
    '</a>' .
    '</h1>';

  // вывод не более 5 элементов канала
  $max = $feed->get_item_quantity(5);
  for ($x=0; $x<$max; $x++)
  {
    $item = $feed->get_item($x);

    // вывод ссылки и названия канала
    echo '<h2>' .
      '<a href="' . $item->get_permalink() . '"' .
        $item->get_title() .
      '</a>' .
      '</h2>';

    // вывод описания канала
    echo '<p>' . $item->get_description() . '</p>';
  }
}
?>

</body>
</html>

```

- Загрузите страницу http://seophp.example.com/read_feed.php – вы должны получить результаты, показанные на рис. 7.5.

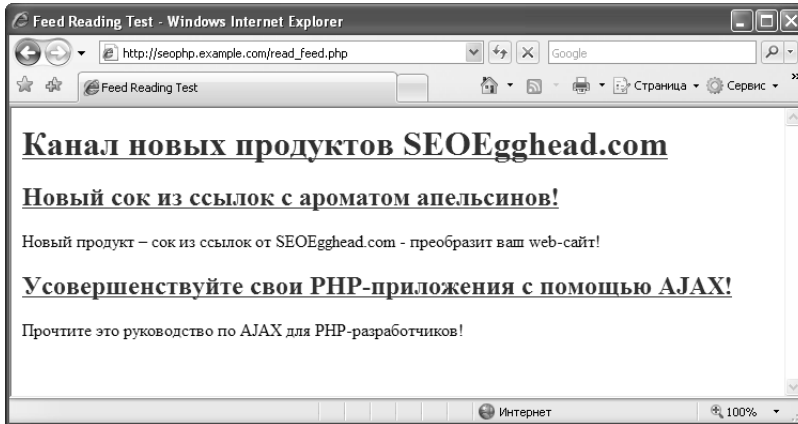


Рис. 7.5. Результат загрузки страницы http://seophp.example.com/read_feed.php

Ну вот — читать данные из внешнего канала с помощью SimplePie оказалось не просто, а очень просто.

Здесь опять проявились преимущества ООП. Вся библиотека SimplePie основана на классе с названием SimplePie, и вы должны лишь создать объект этого класса, а затем использовать нужные методы:

```
<?php
// загрузка библиотеки SimplePie
require_once 'include/simplepie.inc';

// создание и настройка объекта SimplePie
$feed = new SimplePie();
$feed->feed_url('http://seophp.example.com/feed.php');
$feed->cache_location('cache');
$feed->init();
$feed->handle_content_type();
?>
```

Чтобы узнать больше о работе с SimplePie, обратитесь к документации, доступной по адресу <http://simplepie.org/docs/installation/getting-started/>.

Другие источники распространяемого контента

Имеется множество других источников распространяемого контента, обычно в виде *Web-служб*. Их рассмотрение выходит за рамки данной книги, и упоминаются они здесь только для полноты изложения.

Web-службы несколько похожи на RSS- и Atom-каналы тем, что поставляют по запросу данные из внешних источников, но они предоставляют более сложные механизмы коммуникации. Сообщение между клиентами и Web-службами происходит также в форматах, основанных на XML. Наиболее распространены в Web-службах протоколы SOAP и REST.

Многие большие Web-компании, такие как Amazon, eBay, Yahoo!, Google, MSN и Alexa, предоставляют доступ к своим огромным массивам информации с помощью собственных Web-служб. Так что и вы можете предоставлять свои Web-службы. Более подробную информацию по Web-службам можно прочитать в книге *Professional Web APIs with PHP: eBay, Google, Paypal, Amazon, FedEx plus Web Feeds* (Wiley Publishing, Inc., 2006).

Социальные закладки

Web-сайты социальных закладок предлагают пользователям удобное средство для удаленного хранения своих закладок, к которым можно обращаться из любого места. Примерами таких сайтов могут быть del.icio.us, digg, Reddit и т.д. Обычно эти сайты позволяют хранить закладки для личного использования, но у многих имеется возможность сделать их общедоступными. Если закладка на какую-то Web-страницу становится общедоступной для многих пользователей, она становится крупным положительным фактором в алгоритме оценки ранга на сайте социальных закладок. А хороший ранг в такого рода поисках становится дополнительным *крупным* источником естественного трафика. Более того, если закладки на Web-страницу установило большое количество людей, то эта страница может располагаться на первой странице такого сайта. А это обычно приводит к резкому увеличению трафика.

Во многих блогах есть ссылки для упрощения процесса создания закладок на странице. И, как обычно бывает при упрощении какого-то действия, которое нужно пользователям Web-сайта, это может привести к увеличению количества закладок на страницу, находящуюся на вашем сайте. На рис. 7.6 приведен пример страницы со значками для создания закладок на страницу; обведены прямоугольником (слева направо) del.icio.us, digg, Furl и Reddit.

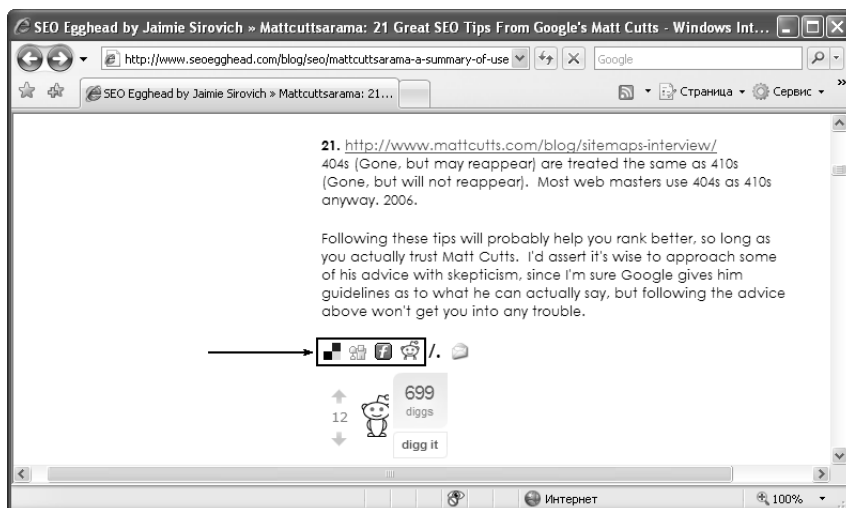


Рис. 7.6. Пример страницы со значками для создания закладок

Эти значки дают людям возможность легче заходить на ваш сайт и бесплатно выполнять для вас маркетинг — если, конечно, им нравится контент по этому адресу, и они захотят поместить на него закладку. Для облегчения добавления таких значков вы создадите класс, которые будет работать с любым Web-приложением. Мы взяли значки и список сайтов социальных закладок из Sociable — подключаемого модуля для WordPress (он использован для этой цели в блоге “SEO Egghead”) — который будет рассмотрен в главе 16. Большое спасибо Питеру Харкинсу (Peter Harkins) за то, что он собрал все эти значки в одном месте.

Библиотека работы с социальными закладками будет создана в следующем упражнении, где вы добавите все значки в страницу вашего каталога `catalog.php`. Данный

сценарий было создан в главе 3, но если вы пропустили эту главу, можете загрузить код со страницы загрузки для главы 7. Страница каталога доступна по адресу <http://seophp.example.com/catalog.html>.

Учитите, что обычно элементы создания социальных закладок не помещаются на страницах каталогов электронной коммерции — ну разве что для очень новых и замечательных продуктов. Мы выбрали этот пример для упрощения реализации демонстрационного примера.

Добавление поддержки социальных закладок

1. Создайте в папке seophp папку с именем social_icons.
2. Загрузите архив с кодом для данной книги и скопируйте из него изображения для сайтов социальных закладок в папку social_icons. (Документ welcome.html из загружаемого архива кода содержит подробное описание о точном местоположении нужных файлов.)
3. Создайте в папке seophp/include файл с именем social_bookmarking.inc.php и поместите в него следующий код:

```
<?php
// +-----+
// | SocialBookmarking |
// | Выводит ссылки на различные службы социальных закладок |
// +-----+
// | Copyright (c) 2005 Jaimie Sirovich |
// +-----+
// | Автор: Jaimie Sirovich <jsirovic@gmail.com> |
// | Значки взяты из модуля Sociable для WordPress, составитель |
// | Peter Harkins (http://push.cx) |
// +-----+

class SocialBookmarking
{
    var $_link;
    var $_title;
    var $_site_name;

    var $_templates = array(
        'blinkbits' => array(
            'icon' => 'blinkbits.png',
            'url' => 'http://www.blinkbits.com/bookmarklets/save.php?v=1&source_url=
{LINK}&title={TITLE}&body={TITLE}'),
        'BlinkList' => array(
            'icon' => 'blinklist.png',
            'url' => 'http://www.blinklist.com/index.php?Action=Blink/addblink.
php&Description=&Url={LINK}&Title={TITLE}'),
        'blogmarks' => array(
            'icon' => 'blogmarks.png',
            'url' => 'http://blogmarks.net/my/new.php?mini=1&simple=1&url={LINK}&
title={TITLE}'),
        'co.mments' => array(
            'icon' => 'co.mments.gif',
            'url' => 'http://co.mments.com/track?url={LINK}&title={TITLE}'),
```

```

'connotea' => array(
  'icon' => 'connotea.png',
  'url' => 'http://www.connotea.org/addpopup?continue=confirm&uri={LINK}&
↳title={TITLE}'),
'del.icio.us' => array(
  'icon' => 'delicious.png',
  'url' => 'http://del.icio.us/post?url={LINK}&title={TITLE}'),
'De.lirio.us' => array(
  'icon' => 'delirious.png',
  'url' => 'http://de.lirio.us/rubric/post?uri={LINK}&title={TITLE};
↳when_done=go_back'),
'digg' => array(
  'icon' => 'digg.png',
  'url' => 'http://digg.com/submit?phase=2&url={LINK}&title={TITLE}'),
'Fark' => array(
  'icon' => 'fark.png',
  'url' => 'http://cgi.fark.com/cgi/fark/edit.pl?new_url=
↳{LINK}&new_comment={TITLE}&new_comment={SITENAME}&linktype=Misc'),
'feedmelinks' => array(
  'icon' => 'feedmelinks.png',
  'url' => 'http://feedmelinks.com/categorize?from=toolbar&op=submit&
↳url={LINK}&name={TITLE}'),
'Furl' => array(
  'icon' => 'furl.png',
  'url' => 'http://www.furl.net/storeIt.jsp?u={LINK}&t={TITLE}'),
'LinkaGoGo' => array(
  'icon' => 'linkagogo.png',
  'url' => 'http://www.linkagogo.com/go/AddNoPopup?url={LINK}&title={TITLE}'),
'Ma.gnolia' => array(
  'icon' => 'magnolia.png',
  'url' => 'http://ma.gnolia.com/beta/bookmarklet/add?url={LINK}&
↳title={TITLE}&description={TITLE}'),
'NewsVine' => array(
  'icon' => 'newsvine.png',
  'url' => 'http://www.newsvine.com/_tools/seed&save?u={LINK}&h={TITLE}'),
'Netvouz' => array(
  'icon' => 'netvouz.png',
  'url' => 'http://www.netvouz.com/action/submitBookmark?url={LINK}&
↳title={TITLE}&description={TITLE}'),
'Reddit' => array(
  'icon' => 'reddit.png',
  'url' => 'http://reddit.com/submit?url={LINK}&title={TITLE}'),
'scuttle' => array(
  'icon' => 'scuttle.png',
  'url' => 'http://www.scuttle.org/bookmarks.php/maxpower?action=add&
↳address={LINK}&title={TITLE}&description={TITLE}'),
'Shadows' => array(
  'icon' => 'shadows.png',
  'url' => 'http://www.shadows.com/features/tcr.htm?url={LINK}&title={TITLE}'),

```

```

'Simpy' => array(
    'icon' => 'simpy.png',
    'url' => 'http://www.simpy.com/simpy/LinkAdd.do?href={LINK}&title={TITLE}'),
'Smarking' => array(
    'icon' => 'smarking.png',
    'url' => 'http://smarking.com/editbookmark/?url={LINK}&description={TITLE}'),
'Spurl' => array(
    'icon' => 'spurl.png',
    'url' => 'http://www.spurl.net/spurl.php?url={LINK}&title={TITLE}'),
'TailRank' => array(
    'icon' => 'tailrank.png',
    'url' => 'http://tailrank.com/share/?text=&link_href={LINK}&title={TITLE}'),
'Wists' => array(
    'icon' => 'wists.png',
    'url' => 'http://wists.com/r.php?c=&r={LINK}&title={TITLE}'),
'YahooMyWeb' => array(
    'icon' => 'yahoomyweb.png',
    'url' => 'http://myweb2.search.yahoo.com/myresults/bookmarklet?u={LINK}&
t={TITLE}')
);
// конструктор
function SocialBookmarking($link, $title, $site_name)
{
    $this->_link = $link;
    $this->_title = $title;
    $this->_site_name = $site_name;
}
// возвращает HTML с символами служб социальных закладок
function getHTML($sites =
    array('del.icio.us', 'digg', 'Furl', 'Reddit', 'YahooMyWeb'))
{
    // создание выходных данных
    $html_feed = '<ul class="social_bookmarking">';
    // создание HTML для каждого сайта, принятого в качестве параметра
    foreach($sites as $s)
    {
        if ($_site_info = $this->_templates[$s])
        {
            $html_feed .= '<li class="social_bookmarking">';
            $url = str_replace(array('{LINK}', '{TITLE}', '{SITENAME}'),
                array(urlencode($this->_link),
                    urlencode($this->_title),
                    urlencode($this->_site_name)),
                $_site_info['url']);
            $html_feed .= '<a rel="nofollow" href="' . $url . '" title="' . $s . '">';
            $html_feed .= '';
            $html_feed .= '</a></li>';
        }
    }
    $html_feed .= '</ul>';
    return $html_feed;
}

```



```

// возвращает HTML со ссылками на службы социальных закладок
// для включения их в каналы
function getFeedHTML($sites =
    array('del.icio.us', 'digg', 'Furl', 'Reddit', 'YahooMyWeb'))
{
    // инициализация $html_feed
    $html_feed = '';
    // создание HTML-канала
    foreach($sites as $s)
    {
        if ($_site_info = $this->_templates[$s])
        {
            $url = str_replace(array('{LINK}', '{TITLE}', '{SITENAME}'),
                array(urlencode($this->_link),
                    urlencode($this->_title),
                    urlencode($this->_site_name)),
                $_site_info['url']);
            $html_feed .= '<a rel="nofollow" href="' . $url .
                '" title="' . $s . '>';
            $html_feed .= '';
            $html_feed .= '</a> ';
        }
    }
    // возврат HTML-канала
    return '<p>' . $html_feed . '</p>';
}
}
?>

```

4. Измените файл seorphp/catalog.php следующим образом:

```

<?php
// загрузка библиотеки генератора ссылок
require_once 'include/url_factory.inc.php';

// загрузка вспомогательного класса работы с социальными закладками
require_once 'include/social_bookmarking.inc.php';

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html>
...
...
...

<center>
<?php
// создание экземпляра класса с указанием ссылки, названия и имени сайта
$social = new SocialBookmarking('http://seorphp.example.com/catalog.html',
    'Замечательные продукты в Наворотах SEO!',
    'SEOegghed');

// вывод ссылок на службы социальных закладок
echo $social->getFeedHTML();

?>
</center>

```

```

<center>
  <a href="popup.php" target="_blank">
    Узнайте больше о Профессиональной поисковой оптимизации на PHP!
  </a>
</center>
...
</html>

```

5. Загрузите страницу <http://seophp.example.com/catalog.html> — вы должны получить результат, показанный на рис. 7.7.

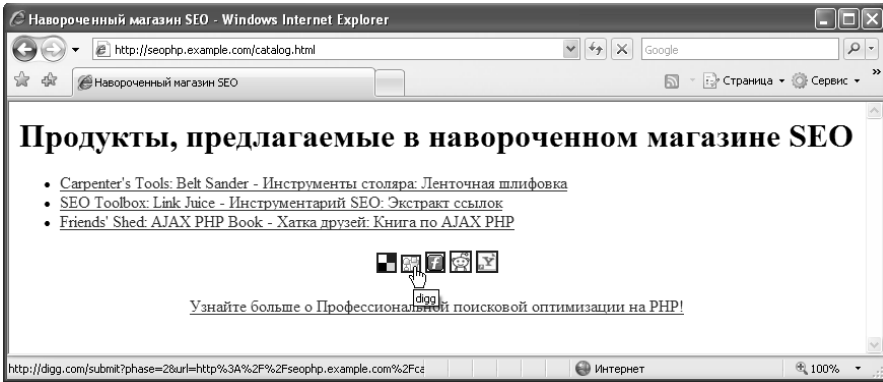


Рис. 7.7. Результат загрузки страницы <http://seophp.example.com/catalog.html>

6. Теперь посмотрим, как добавить такую же возможность в каналы. В классе `SocialBookmarking` имеется метод `getFeedHTML()`, который как раз и предназначен для этой цели. По сути, методы `getHTML()` и `getFeedHTML()` очень похожи и отличаются форматом выводимых данных. В данном случае `getHTML()` выводит ссылки в виде неупорядоченного списка, а `getFeedHTML()` просто разделяет ссылки пробелами. Для добавления ссылок на службы социальных закладок добавьте в сценарии `rss_factory.inc.php` в конец каждого канала следующий код:

```

<?php
// загрузка вспомогательного класса социальных закладок
require_once 'social_bookmarking.inc.php';

class RSSFactory
{
  ...
  ...
  ...
  // генерация канала
  function get()
  {
    ...
    ...
    ...
    // добавление элементов канала
    foreach ($this->_items as $feed_item)
    {
      // добавление элемента канала и его содержимого

```

```

echo '<item>';
foreach ($feed_item as $item_name => $item_value)
{
    // добавление в описание канала значков служб социальных закладок
    if ($item_name == 'description')
    {
        // создание экземпляра класса на основе ссылки, названия и имени сайта
        $social = new SocialBookmarking($feed_item['link'],
            $feed_item['title'],
            $this->_title);

        // добавление в канал значков служб социальных закладок
        $item_value = $item_value . $social->getFeedHTML();
    }

    // вывод элемента канала
    echo "<$item_name>" .
        RSSFactory::_escapeXML($item_value) .
        "</$item_name>";
}
echo '</item>';
}
...
...
...
}
}
?>

```

7. Снова загрузите в браузер канал по адресу <http://seophp.example.com/feed.php> – теперь вы должны видеть новые ссылки на службы социальных закладок, как показано на рис. 7.8.

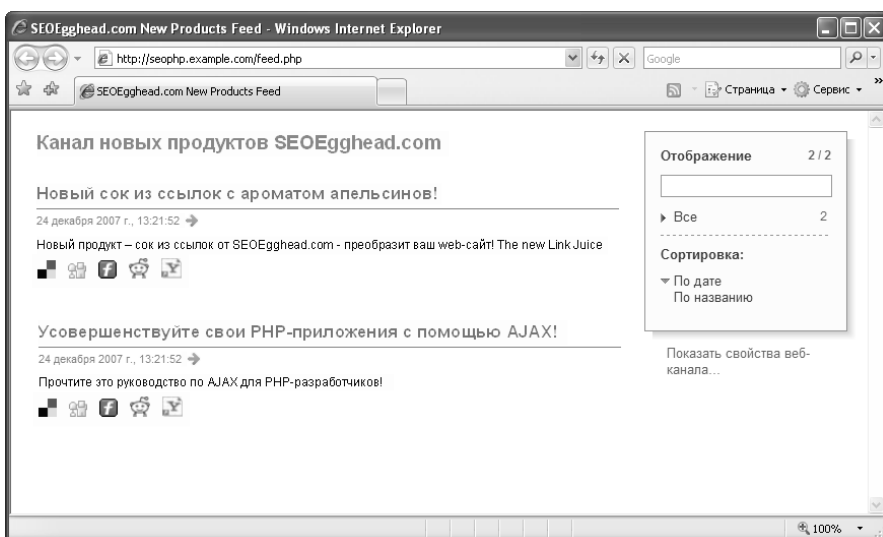


Рис. 7.8. Новые ссылки на службы социальных закладок

Только что созданная библиотека умеет создавать ссылки на множество Web-сайтов социальной сети: blinkbits, blinklist, blogmarks, co.mments, connotea, del.icio.us, de.lirio.us, digg, Fark, feedmelinks, Furl, LinkaGoGo, Ma.gnolia, NewsVine, NetVouz, Reddit, scuttle, Shadows, Simply, Smarking, Spurl, TailRank, Wists и YahooMyWeb. Весьма внушительный список, не так ли?

Если на вашем сайте появится что-то интересное, и вы хотите, чтобы об этом говорило множество людей, нужно облегчить им эту задачу. После создания класса SocialBookmarking и его применения на странице, где находится ваш контент, его нужно просто использовать так, как это сделано в сценарии catalog.php:

```
<center>
  <?php
    // создание экземпляра класса с указанием ссылки, названия и имени сайта
    $social = new SocialBookmarking('http://seophp.example.com/catalog.html',
                                   'Замечательные продукты в Наворотах SEO!',
                                   'SEOEgghead');
    // вывод ссылок на службы социальных закладок
    echo $social->getFeedHTML();
  ?>
</center>
```

Конечно, генерируемый HTML-код при необходимости можно изменять. Но здесь мы не настаиваем на выполнении таких изменений.

Обратите также внимание, что у метода getHTML() имеется необязательный параметр-массив, содержащий информацию о службах, для которых создаются ссылки. По умолчанию этот массив имеет значение array('del.icio.us', 'digg', 'Furl', 'Reddit', 'YahooMyWeb'), но при желании вы можете указать в нем любые известные вам службы.

Резюме

Каналы представляют собой упрощенный способ доступа пользователей к контенту; они также позволяют распространять этот контент другим сайтам. Ссылки, включенные в каналы, увеличивают трафик как непосредственно, так и в дальнейшем в результате дополнительных эффектов. Пользователи будут щелкать на ссылках, встроенных в распространяемый контент, а поисковые механизмы будут регистрировать постепенное увеличение количества ссылок. В данной главе представлен класс, позволяющий легко создавать каналы в формате RSS 2.0, а также сторонний класс для чтения всех различных форматов, применяемых в настоящее время.

Службы социальных закладок предлагают другой вид естественного повышения трафика, который также не стоит игнорировать. Облегчение процесса создания закладок, скорее всего, увеличит количество закладок, устанавливаемых на ваш сайт. Значит, увеличится и ранг в функции поиска по сайтам социальных закладок — возможно, ваш сайт даже займет почетное место на главной странице службы.