УДК 004

Парсинг сайтов при помощи phpQuery

Круглик Роман Игоревич
Приамурский государственный университет им. Шолом-Алейхема
Студент

Аннотация
В статье рассматривается поэтапно создание парсера с помощью библиотеки phpQuery. В качестве примера были скопированы статьи новостного веб-ресурса.
Ключевые слова: парсинг, PHP, phpQuery.

Parsing sites using phpQuery

Kruglik Roman Igorevich
Sholom-Aleichem Priamursky State University
Student

Abstract
In article describes the phased creation of the parser using the phpQuery library. As an example, articles from a news web resource were copied.
Keywords: Parsing, PHP, phpQuery.

Сбор информации в интернете – трудоемкая, рутинная, отнимающая много времени работа, но которую необходимо выполнять при разработки собственного веб-ресурса. Помимо этого, разработчик сталкивается с другими задачами, такими как:

Большие объёмы. Успешный веб-проект немыслим без размещения большого количества информации на сайте. Современные темпы жизни приводят к тому, что контента бывает слишком много.

Частое обновление. Обслуживание огромного потока динамично меняющейся информации не в силах обеспечить одним человеком или даже слаженной командой операторов. Порой информация изменяется ежеминутно и в ручном режиме обновлять её вряд ли целесообразно.

Все эти задачи очень важны и требуют радикального решения. Именно поэтому были разработаны парсеры.

Парсинг – это автоматизированный сбор неструктурированной информации, ее преобразование и выдача в структурированном виде.

По сравнению с человеком, парсер:

1. **быстро** обойдёт тысячи веб-страниц;
2. **аккуратно** отделит техническую информацию от «человеческой»;
3. **безошибочно** отберёт нужное и отбросит лишнее;
4. **эффективно** упакует конечные данные в необходимом виде.

Исследования в области парсинга сайтов актуальны и по сей день. Статья В.А. Ильин, Д.А. Скоселев [1] посвящена методу парсинга веб-страниц, который определяет ссылки на определенные HTML элементы. В статье М.Е. Кочитов [2] рассматривает парсинг сайтов с помощью технологии cURL и библиотеки phpQuery. В работе Р.М. Яхшисарова [3] описываются методы и инструменты для парсинга данных с web-сайта. Просветов В.Л., Конева Н.Е. [4] рассматривают наиболее распространённые методы и средства агрегации данных, одним из которых является - парсинг данных

В данной статье будет наглядно показана реализация парсера на примере новостного сайта [5] eaomedia.ru.

Логика реализована в 5 шагах:

1. С помощью библиотеки phpQuery достаётся весь HTML код страницы, которую необходимо спарсить,
2. Вытащить нужные данные из всего массива,
3. Структурированно сохранить в другой массив,
4. Вывести полученный данные на новой странице.

Для начала создадим файл, который будет доставать и формировать эти данные (см. рис. 1).

```php
<?php
include './phpQuery.php';
// Общие данные
$site = 'https://eaomedia.ru/news/incidents/';
$protocol = 'https:';
$html = file_get_contents($site);
// Документ phpQuery
$doc = phpQuery::newDocument($html);
// Главные статьи
$articlesItems = $doc->find( selectors: '.newslist-blk');
// Перебираем статьи, вытаскиваем картинки, заголовки, тексты и ссылки
$articles = array();
foreach ($articlesItems as $articleItem) {
    // Находим нужный элемент
    $articleElem = pq($articleItem);
    // Вытаскиваем данные
    $image = $articleElem->find( selectors: 'a img')->attr( attr: 'src');
    $title = $articleElem->find( selectors: 'a')->text();
    $text = $articleElem->find( selectors: 'span:first')->text();
    $link = $articleElem->find( selectors: 'a')->attr( attr: 'href');
    // Добавляем протокол или url сайта при необходимости
    if (strpos($image, $protocol) === false) {
        $image = $protocol . $image;
    }
    if (strpos($link, $site) === false) {
        $link = $site . $link;
    }
    // Сохраняем результаты в массив
    array_push( &array: $articles, array(
        'image' => $image,
        'title' => $title,
        'text' => $text,
        'link' => $link
    ));
}
include './template.php';
```

Рисунок 1. Сборщик данных

Откроем страницу с новостями и посмотрим от куда будут браться статьи. (см. рис. 2).

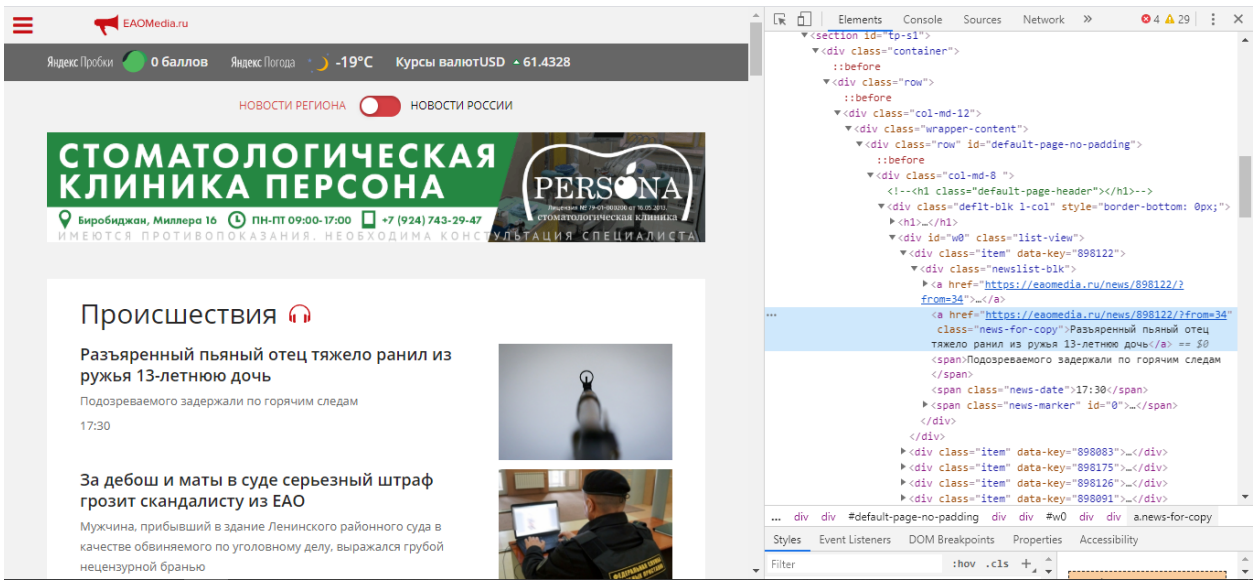

Рисунок 2. Новостной сайт

Слева находится список статей, которые будут скопированы, а справа код, который нужен, чтобы доставать данные. В конце все данные сохраняются в массив $articles и передаются в файл template.php, где выводятся уже в структурированном виде (см. рис. 3).

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Новости Биробиджана</title>
    <link href="css/main.css" rel="stylesheet">
</head>
<body>
    <table>
        <tbody><tr><td class="articles">
                    <h2>Интересные статьи</h2>
                    <ul>
                        <?php
                        foreach ($articles as $item) { ?>
                        <li>
                            <a href="<?php echo $item['link']; ?>" target="_blank"><?php echo $item['title']; ?></a>
                            <p>
                                <img src="<?php echo $item['image']; ?>" alt="">
                                <?php echo $item['text']; ?>
                            </p>
                        </li>
                        <?php} ?>
                    </ul></td></tr>
        </tbody>
    </table>
</body>
</html>
```

Рисунок 3. Структура сайта

Разбивается массив и статьи выводятся в том же порядке, в котором сохранялись. Так же был добавлен файл main.css для более наглядного структурирования (см. рис. 4).

```
li {
  list-style: none;
  clear: both;
  padding-left: 40px;}
a {
  display: block;
  color: steelblue;
  text-decoration: none;}
a:hover {
  color: navy;
  text-decoration: underline;
}
.wbd-link {
  text-align: center;
  font-size: 0.9em;
}
table td {
  vertical-align: top;
}
.articles a {
  font-size: 1.1em;
  font-weight: 400;}
.articles img {
  float: left;
  max-width: 150px;
  margin: 0 10px 10px 0;}
.articles p {
  font-size: 0.9em;}
```

Рисунок 4. Файл со стилями.

Теперь можно посмотреть на результат работы (см. рис. 5).

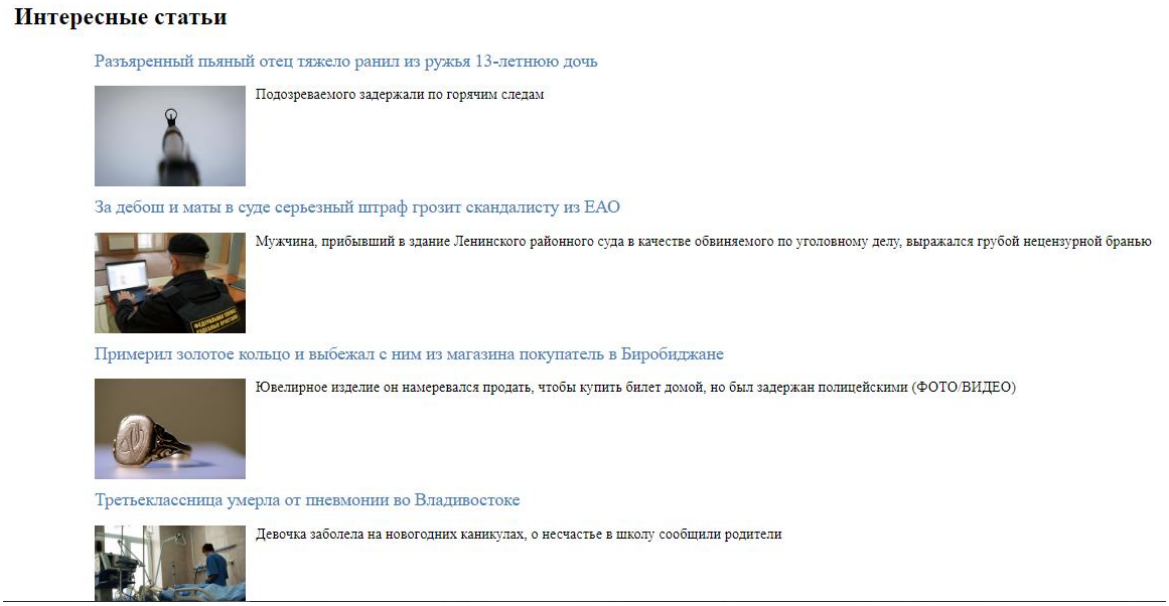

Рисунок 5. Созданный сайт с новостями

Статьи полностью совпадают с официальным новостным сайтом, с которого был произведён парсинг. При добавлении новых статей, они будут дублироваться на сайт. Так же мы не храним данные на сервере, что улучшает производительность будущей системы.

Рассмотренный функционал можно использовать в различных целях, где нужны данные с других источников.

Библиографический список

1. Ильин В.А., Скоселев Д.А. Парсинг веб-сайтов с использованием шаблонов // Мир современной науки. 2018. № 2 (48). С. 8-12.
2. Кочитов М.Е. Парсинг сайтов с помощью curl и phpquery // Постулат. 2018. № 8 (34). С. 1.
3. Яхшисарова Р.М. Программная реализация парсинга данных раздела поиска недвижимости сайта avito.ru // Математическое моделирование процессов и систем Материалы VII Международной молодежной научно-практической конференции. Ответственный редактор С.А. Мустафина. 2017. С. 437-441.
4. Просветов В.Л., Конева Н.Е. Анализ методов и средств автоматизации процессов обработки данных веб-сайтов // Евразийское Научное Объединение. 2019. № 1-2 (47). С. 89-94.
5. Происшествия // eaomedia.ru URL: https://eaomedia.ru/news/incidents/ (дата обращения: 15.01.2020).