

Веб-программирование

Часть 1

Юрий Литвинов

yurii.litvinov@gmail.com

23.11.2018г

Веб-приложения

Как оно вообще работает

- ▶ Пользователь заходит браузером на определённый URL
 - ▶ На самом деле, выполняя HTTP GET-запрос на порт 80 или 443 (обычно)
- ▶ ОС сервера перенаправляет запрос запущенному там *веб-серверу*
 - ▶ Например, Apache, IIS
- ▶ Веб-сервер — отдельный процесс, в рамках которого запущено несколько *веб-приложений*, веб-сервер по URL запроса определяет, какому веб-приложению он адресован, и передаёт запрос ему
- ▶ Веб-приложение формирует ответ и отправляет его обратно по HTTP в виде HTML-страницы
- ▶ Эта страница и показывается пользователю в браузере

Веб-сервисы

- ▶ *Веб-сервис* — это примерно то же самое, но не для пользователя, а для других приложений
- ▶ Нужны для создания распределённых приложений
- ▶ Общаются не с помощью HTML, а посредством специализированных протоколов
 - ▶ Например, SOAP
 - ▶ Использует синтаксис XML, может использовать HTTP как транспорт
- ▶ Как правило, содержат механизм публикации метаинформации
 - ▶ Например, WSDL
- ▶ Реализуются посредством технологий, например, Windows Communication Foundation

Веб-приложения и .NET

- ▶ Веб-сервер — IIS (Internet Information Services), IIS Express, Kestrel
 - ▶ Есть “из коробки” в Windows, IIS Express поставляется с Visual Studio и используется для отладки
- ▶ Технология для разработки веб-приложений — ASP.NET MVC
 - ▶ ASP.NET MVC 5
 - ▶ ASP.NET MVC Core 2.0
- ▶ Технология для разработки веб-сервисов — WCF
- ▶ Работа с базами данных — MS SQL Server (SQL Server Express)
- ▶ ORM — Entity Framework (Entity Framework Core)
- ▶ Облачный хостинг — Azure

Браузерная часть

- ▶ HTML (HyperText Markup Language) — используется для задания содержимого и структуры веб-страницы
 - ▶ Тут размечаются параграфы, заголовки, списки, таблицы и т.д.
 - ▶ Тут же обычно определяются способы идентифицировать элементы
- ▶ CSS (Cascading Style Sheet) — используется для задания внешнего вида, оформления и расположения элементов
- ▶ JavaScript — используется для задания поведения веб-страницы на клиенте
 - ▶ Интерпретируется браузером
 - ▶ Полноценный язык программирования

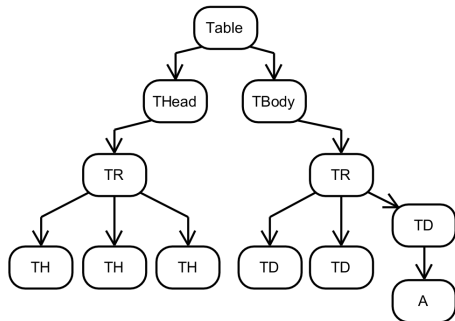
DOM

- ▶ DOM (Document Object Model) — представление HTML-документа в виде дерева объектов и API для доступа к нему
- ▶ JavaScript может манипулировать DOM-деревом, браузер рендерит его “на лету”, что и даёт интерактивность

```

<table class="listing">
  <thead>
    <tr class="odd">
      <th>Выпускник</th>
      <th>Научный руководитель</th>
      <th>Текст</th>
    </tr>
  </thead>
  <tbody>
    <tr class="odd">
      <td>Акбаров Артур Александрович</td>
      <td>д.т.н., проф. Д.В. Кознов</td>
      <td><a href="bmo/441-Akbarov-report.pdf">Текст</a></td>
    </tr>
  </tbody>
</table>

```



HTML-формы

- ▶ Способ получить ввод от пользователя
- ▶ Возможность организовать POST-запрос (GET по умолчанию)

```
<form method="post">
```

```
  First name:<br>
```

```
  <input type="text" name="firstName"><br>
```

```
  Last name:<br>
```

```
  <input type="text" name="lastName"><br><br>
```

```
  <input type="radio" name="gender" value="male" checked>Male<br>
```

```
  <input type="radio" name="gender" value="female">Female<br>
```

```
  <input type="submit" value="Submit">
```

```
</form>
```

CSS

- ▶ Способ задать внешний вид элементов

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
  body {background-color: powderblue;}
```

```
  h1 {color: blue;}
```

```
  p  {color: red;}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
</body>
```

```
</html>
```

© <https://www.w3schools.com>

Или, что более типично

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

© <https://www.w3schools.com>

Селекторы

```
<p id="p01">I am different</p>  
<p class="error">Error message</p>
```

```
#p01 {  
  color: blue;  
}
```

```
p.error {  
  color: red;  
}
```

© <https://www.w3schools.com>

Немного JavaScript-a

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h1>My First JavaScript</h1>
```

```
<button type="button"
```

```
onclick="document.getElementById('demo').innerHTML = Date()">
```

```
Click me to display Date and Time.</button>
```

```
<p id="demo"></p>
```

```
</body>
```

```
</html>
```

© <https://www.w3schools.com>

Или так

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script>
```

```
function doSomething() {
```

```
  document.getElementById("demo").style.fontSize = "25px";
```

```
  document.getElementById("demo").style.color = "red";
```

```
  document.getElementById("demo").style.backgroundColor = "yellow";
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<button type="button" id="demo" onclick="doSomething()">Click me!</button>
```

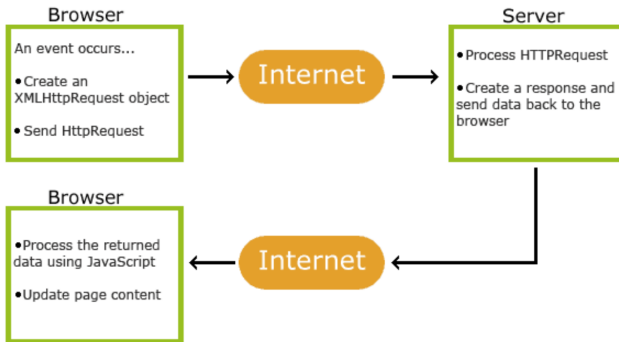
```
</body>
```

```
</html>
```

© <https://www.w3schools.com>

AJAX

Asynchronous JavaScript And XML



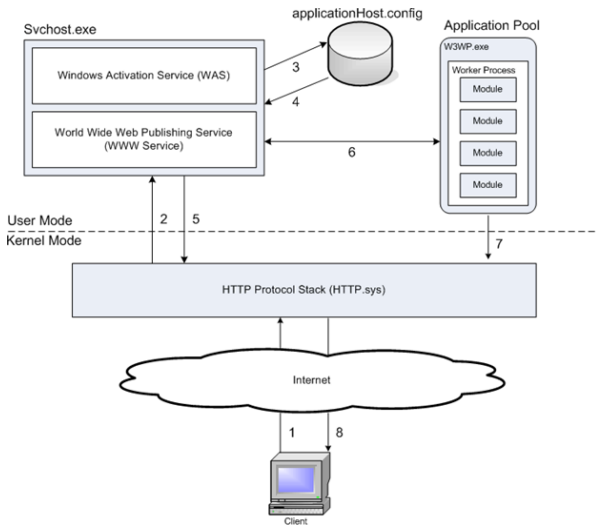
Пример

```
<!DOCTYPE html>
<html>
<body>
<div id="demo">
  <h2>The XMLHttpRequest Object</h2>
  <button type="button" onclick="loadDoc()">Change Content</button>
</div>
<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
        this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
</script>
</body>
</html>
```

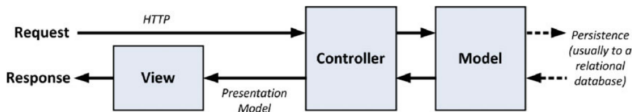
© <https://www.w3schools.com>

Бэкенд

Обработка веб-запроса в Windows



ASP.NET MVC, основные понятия



© A. Freeman, Pro ASP.NET Core MVC

- ▶ **Модель** содержит или представляет данные, с которыми работает приложение
 - ▶ **Domain model** содержит объекты предметной области вместе с бизнес-логикой, механизмами сериализации и т.д.
 - ▶ **View Model** содержит классы, удобные для отображения во View, без бизнес-логики
- ▶ **Представление** (View) отвечает за показ данных из модели пользователю
 - ▶ Работает в браузере, но генерится на сервере
- ▶ **Контроллер** отвечает за обработку входящих запросов, преобразование моделей и формирование данных для видов

Обработка веб-запроса в ASP.NET

