Ряшенцев Владимир Сергеевич

Функциональность кино-сервисов, реализованная на JavaScript

**Аннотация**

Рассмотрены основные интерактивные веб-элементы, использующиеся на страницах кино-сервисов, технологии, предназначенные для работы с сервером без перезагрузки документа, взаимодействия со сторонними ресурсами. Исследованы различные виды HTML5 медиаплееров и проведен их сравнительный анализ. Осуществлен обзор многофункционального интерфейса для работы со звуком – Web Audio API.

Ключевые слова: frontend, веб-страница, JavaScript, AJAX, API.

**Введение**

Разработка любой веб-страницы состоит из frontend и backend частей, первая отвечает за функционал клиентской стороны, а именно обработку действий пользователя в веб-браузере. Backend составляющая работает на стороне сервера: принимает запросы пользователя, реализует взаимодействие с базами данных, генерирует HTML страницы.

Неотъемлемой частью frontend разработки является язык программирования JavaScript, который создавался с целью добавить веб-страницам интерактивности. Программы на данном языке называются скриптами. [1] В браузере они напрямую подключаются к HTML, которые выполняются при загрузке страницы.

JavaScript можно использовать не только в веб-разработке, он позволяет решать большой спектр задач в frontend. Ключевыми возможностями рассматриваемого языка являются добавление эффекта анимации различным элементам страницы, реакция на события и их обработка, валидация полей для ввода данных, запись и чтение cookie файлов, отправка запросов к серверу без перезагрузки страницы (AJAX) [2].

С помощью сторонних или же встроенных инструментов браузера, используя JavaScript, также можно управлять аудио и видео контентом на сайте. Благодаря современному Web Audio API разработчикам стала доступна генерация, обработка и объемное распределение звука. Рассматриваемый язык также позволяет создавать игры внутри веб-страницы, графическая часть которых реализуется с помощью современных WebGL и WebRTC.

В данной статье будет рассмотрена функциональность кино-сервисов, реализованная с помощью JavaScript.

**Интерактивные элементы веб-страницы**

Современный веб-сайт не будет выглядеть современно, собственно, не сможет заинтересовать пользователя, если в нем будут отсутствовать эффекты анимации. Онлайн кинотеатры не являются исключением. JavaScript позволяет реализовывать выцветание, плавное увеличение и уменьшение, передвижение объектов, сокрытие и раскрытие элементов. [3] Подобные эффекты можно сделать благодаря CSS3, однако, JS будет более гибким решением таких задач.

Чаще всего на веб-страницах онлайн кинотеатров встречаются следующие элементы: слайдер, диалоговые окна и карты.

Наиболее распространенным инструментом среди перечисленных является «карусель» (слайдер). Данный элемент позволяет на ограниченном месте страницы отображать большое количество составляющих страницы, в случае кино-сервисов – афиш фильмов, также многие сайты используют «карусель» для отображения трейлеров, выстроенных в очередь.

Практически любой слайдер состоит из следующих частей:

– Блок-обертка, который будет скрывать все выходящее за его рамки;

– Контейнер, который будет вмещать в себе слайды;

– Слайды, которые будут хранить в себе афиши или трейлеры;

– Кнопки управления, позволяющие двигать слайды. [4]

Для создания подобного инструмента существует множество готовых решений, некоторые из них являются плагинами для jQuery, такие как: Owl-carousel, Slick, Wow-slider и др. [5] Многие являются самостоятельными библиотеками, например, Vanilla.js. В основном, программисты пользуются встроенными средствами популярных JavaScript фреймворков. В единичных случаях разработчики реализуют «карусель» с нуля.

Существует тенденция использования диалоговых или модальных окон на веб-странице. В онлайн кинотеатрах, в основном, они предназначены для просмотра краткой информации: описание к картине, ее рейтинг, жанр, год премьеры, оценки и отзывы пользователей. В реализации рассматриваемого элемента JS выполняет функции отображения и скрытия окна, в модальных вариантах – он также осуществляет передвижение с помощью технологии Drag & Drop.

На некоторых страницах кино-сервисов, предназначенных для отображения контактных данных или для связи с администрацией, размещаются карты с указанием адреса главного офиса компании. На КиноПоиск существует возможность заказа билетов в кинотеатр, где пользователю, в первую очередь, необходимо на карте указать его местоположение.

Для встраивания управляемой карты на веб-страницу используются наиболее популярные Яндекс.Карты или Google Maps API. Данные интерфейсы помогут установить на сайт или в приложение карту с поиском по топонимам и организациям, с возможностью строить маршруты и смотреть панорамы, а также с другими функциями, доступными на Яндекс.Картах или Google Maps. Картографические данные постоянно обновляются, поэтому созданные с помощью предлагаемых интерфейсов карты отображают только актуальную информацию. [6]

С помощью JavaScript API существует возможность настроить нужную логику взаимодействия пользователя с картой и определить, как эта карта будет выглядеть. Чтобы задать внешний вид объектов на карте, можно выбрать стандартные элементы или создать собственный макет. Функциональность карты будет одинаковой на разных типах устройств.

**Инструменты для работы с сервером**

AJAX (Asynchronous JavaScript And Xml) – технология обращения к серверу без перезагрузки страницы. За счет этого уменьшается время отклика и веб-приложение для пользователя выглядит удобнее.

Живой поиск – классический пример использования данной технологии, который используется в кино-сервисах для поиска фильма к просмотру.

Пользователь начинает печатать поисковую фразу, а JavaScript предлагает возможные варианты, получая список самых вероятных дополнений с сервера.

Алгоритм работает следующим образом:

– Код активируется примерно при каждом нажатии клавиши, но не чаще чем раз в 100 мс;

– Создается скрытый DIV и заполняется ответом сервера: текущий результат подсвечен, можно перемещаться и выбирать с помощью клавиатуры или мыши;

– Результаты запросов кешируются, повторных обращений к серверу не происходит.

Механизм установки пометки «Нравится / Не нравится» под картиной, интересующей пользователя, а также написание отзывов, комментариев, реализовано с помощью AJAX. Если воспользоваться для разработки данных функций средствами серверной стороны – при любом действии пользователя будет перезагружаться весь документ, что ухудшает производительность и увеличивает время отклика страницы. AJAX, с свою очередь, обновляет лишь необходимые элементы.

Как и любые другие веб-сайты, кино-сервисы взаимодействуют со сторонними ресурсами. Источниками такого рода могут служить социальные сети, которые, например, позволяют осуществить пользователю быструю регистрацию в системе, а разработчикам, с свою очередь, получить все необходимые данные в удобном формате.

Для запроса к API социальных сетей существуют множество различных технологий. Подобные системы работают по принципу HTTP запрос – JSON ответ. Есть множество библиотек, которые являются готовыми решениями таких задач, например, restful.js, amygdala. Некоторые используют AJAX с помощью jQuery, которая основана на нативном инструменте XMLHttpRequest. Однако, разработчики отдают предпочтение следующим фреймворкам как jQuery, Vue.js, React.js и др. Такому выбору есть достойное объяснение. Restful.js имеет некачественную архитектуру, где функции реализованы без достойной оптимизации. В amygdala отсутствуют Promise, следовательно, нет возможности реализации async/await. [7] Фреймворк jQuery является устаревшей технологией.

У множества социальных сетей есть собственные SDK (Software Development Kit), который содержит в себе набор классов и отдельные функции, реализующие определенные запросы к ресурсу.

Отдельного внимания заслуживает встроенный в браузеры инструмент fetch, который, в свою очередь, является новым поколением XMLHttpRequest. Он предоставляет улучшенный интерфейс для осуществления запросов к серверу: как в аспекте возможностей и контроля над происходящим, так и по синтаксису, потому что основан на технологии Promise.

Fetch спецификация отличается от jQuery.ajax() в основном двумя способами:

Promise возвращенный fetch() не отклонит состояние ошибки HTTP, даже если ответ будет с кодом 404 или 500. Вместо этого, он будет выполнен с установкой статуса ok в false и будет отклонять только при сбое сети или если что-либо помешало запросу выполниться. [8]

По умолчанию, fetch не будет отправлять или получать файлы cookie с сервера, в результате чего запросы будут без проверки подлинности, если сайт основан на сохранении сессии пользователя (для отправки cookies в опции init должны быть установлены параметры доступа).

**Сравнительный анализ HTML5 медиаплееров**

Основной частью каждого кино-сервиса является механизм, реализующий воспроизведение видео и аудио контента. Для показа фильма требуется создание плеера, который бы осуществлял отображение картины. С появлением в HTML5 тегов <video> и <audio> медиафайлы стали по-настоящему доступными для Интернета.

За короткий срок стандарт HTML5-видео заменил Flash Player и аналогичные сторонние медиаплееры. Ранее, для корректного запуска медиафайлов требовалось приложить немало усилий. Для воспроизведения мультимедиа зачастую нужно было использовать теги <embed> и < object> с внушительным списком параметров.

Разница в том, что HTML5 теги аудио и видео воспринимают файлы как изображения. Такие атрибуты, как высота, ширина и автозапуск задаются в тегах, как и в любом другом HTML элементе.

Существует внушительное количество плееров, предназначенных для корректного и качественного воспроизведения медиафайлов, поддерживающих большинство популярных форматов, сравнительный анализ которых представлен в таблице 1.

Таблица 1 – Сравнительный анализ HTML5 медиаплееров

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Фреймворк | Video.js | JW Player | MediaElement.js | Kaltura | Afterglow | jPlayer | Plyr |
| Свойства |
| Резервное копирование на FLASH | + | + | + | + | + | + | + |
| API | + | + | + | + | + | + | + |

Продолжение таблицы 1.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Интегрируемые платформы | V, Y | Y | Y, V, DM | - | Y | - | Y, V |
| JS Библиотеки | - | - | - | jQuery | - | jQuery | - |
| Hosting | - | + | + | + | + | - | + |
| SDK для моб. устройств | - | + | + | + | + | - | + |
| Бесплатный | + | - | + | + | + | + | + |
| CMS Плагины | W | W, D | W, D, T, J, C | - | - | - | W |
| Полный экран | + | + | + | + | + | + | + |
| Управление клавиатурой | - | - | + | + | + | - | + |
| Аналитика | - | + | + | + | - | - | - |
| Субтитры | + | + | + | + | + | + | + |
| Плейлисты | - | + | + | - | - | - | - |
| Контроль скорости | - | + | + | + | - | - | + |
| Контроль качества | - | + | + | + | + | - | + |

Условные обозначения к таблице: Y – YouTube, V – Vimeo, DM – Dailymotion, W – WordPress, D – Drupal, T – Typo3, J – Joomla, C – Contao.

**Web Audio API**

Главным аспектом конкурентоспособности современных кино-сервисов является качество предоставляемых пользователю картин. Помимо изображения зрителям важно звуковое сопровождение фильма. Одним из типов устройств, на которых можно использовать сервисы онлайн кинотеатра, являются smart-телевизоры. Они обладают мощной аудиосистемой, в связи с этим, необходим плеер, который будет обеспечивать достойное качество звука видео контента.

Web Audio API – это многофункциональный инструмент, без которого сложно будет обойтись при разработке современных игр и интерактивных веб-приложений. API достаточно высокоуровневый, имеет качественно спроектированную архитектуру, оптимизирован и обладает низким порогом вхождения. [10]

Следует понимать, что HTML элементы <audio>, в отличие от рассматриваемого интерфейса, предназначены совершенно для других типов задач. Единственная связь между рассматриваемыми инструментами в том, что элемент <audio> может служить источником звука для web Audio API.

Основные преимущества, которые стимулируют к использованию рассматриваемого интерфейса следующие:

– Работа с многоканальным аудио. Исходя из документации API обязан поддерживать до 32 каналов аудио. Многие пользователи кино-сервисов владеют проигрывающими системами, которые обладают современными звуковыми технологиями. Они, в свою очередь, поддерживают 8 каналов.

– Высокоуровневое 3D распределение аудио по каналам, в зависимости от положения, направления и скорости источника звука и слушателя.

**Заключение**

В ходе работы над статьей, автором были достигнуты следующие результаты.

Проанализировано, что на страницах онлайн кинотеатров присутствует множество различных интерактивных элементов. Чаще всего используются слайдеры, диалоговые окна и динамические карты.

Рассмотрены технологии, которые осуществляют взаимодействие с внешними ресурсами, используя JavaScript. AJAX, позволяющий реализовать функции живого поиска, механизм отзывов и отметок пользователей. С помощью Fetch можно осуществлять запросы к API и получать данные со сторонних источников.

Проведен обзор различных медиаплееров, реализованных с помощью JavaScript и исследована технология Web Audio API, осуществляющая реализацию взаимодействия с современными аудиосистемами. Данный интерфейс предоставляет возможность воспроизведения фильмов с объемным звуком.

Таким образом, функциональность, разработанная с помощью JavaScript, является неотъемлемой частью кино-сервисов.

**Список использованной литературы**

1. Введение в JavaScript. [Электронный ресурс] URL: https://learn.javascript.ru/intro
2. JavaScript: Возможности и ограничения. [Электронный ресурс] URL: https://puzzleweb.ru/javascript/00\_teacher.php
3. Роль JavaScript в природе. [Электронный ресурс] URL: https://javascript.ru/tutorial/foundation/role#chto-umeet-javascript
4. Карусель на Vanilla.JS. [Электронный ресурс] URL: https://habr.com/ru/post/327246/
5. 25 бесплатных адаптивных слайдеров типа Карусель на jQuery. [Электронный ресурс] URL: https://www.internet-technologies.ru/articles/25-besplatnyh-adaptivnyh-slayderov-tipa-karusel-na-jquery.html
6. JavaScript API Яндекс.Карт – Технологии Яндекса. [Электронный ресурс] URL: https://tech.yandex.ru/maps/jsapi/
7. Запросы к Rest API из JavaScript компактно и красиво. [Электронный ресурс] URL: https://habr.com/ru/post/302288/
8. Использование Fetch. [Электронный ресурс] URL: https://developer.mozilla.org/ru/docs/Web/API/Fetch\_API/Using\_Fetch
9. Топ 5 Open Source HTML5 видеоплееров на 2018 год. [Электронный ресурс] URL: https://medium.com/nuances-of-programming/
10. Web Audio API – новые возможности генерации, обработки и объемного распределения звука в браузере. [Электронный ресурс] URL: https://html5.by/blog/audio/