



ОТКРЫТАЯ НАУКА
издательство

Международный журнал информационных технологий и
энергоэффективности

Сайт журнала:

<http://www.openaccessscience.ru/index.php/ijcse/>



УДК 004.738

ВЫБОР ТЕХНОЛОГИИ АСИНХРОННОЙ ПЕРЕДАЧИ ДАННЫХ В РЕАЛЬНОМ ВРЕМЕНИ ПРИ ПРОЕКТИРОВАНИИ ВЕБ-МЕССЕНДЖЕРОВ

Сергеев Д.Н.

ФГБОУ ВО "КАЗАНСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМ. А.Н. ТУПОЛЕВА-КАИ», Казань, Россия (420111, Республика
Татарстан, город Казань, ул. Карла Маркса, д.10), e-mail: sergeevDanil2301@gmail.com

В данной статье рассматриваются пять основных технологий асинхронной передачи данных для интерактивного взаимодействия клиента с сервером: Long Polling, Server-Sent Events (SSE), WebSocket, WebRTC, WebTransport. Описываются основные принципы работы этих технологий, а также рассматриваются ключевые аспекты взаимодействия с ними, выделяются их сильные и слабые стороны в контексте разработки веб-мессенджеров, анализируется потенциальная сложность проектирования информационной системы, а также по совокупным показателям выбирается наиболее оптимальная технология для проектирования веб-мессенджеров.

Ключевые слова: Передача данных в реальном времени, способы коммуникации, асинхронная передача данных, веб-мессенджер, веб-сайт.

SELECTION OF REAL-TIME ASYNCHRONOUS DATA COMMUNICATIONS TECHNOLOGY IN WEB MESSENGER DEVELOPMENT

Sergeev D.N.

«KAZAN NATIONAL RESEARCH TECHNICAL UNIVERSITY. A.N. TUPOLEV-KAI», Kazan,
Russia (420111, Republic of Tatarstan, Kazan, Karl Marx st., 10), e-mail:
sergeevDanil2301@gmail.com

This article reviews five main technologies of asynchronous data transfer for interactive client-server communication: Long Polling, Server-Sent Events (SSE), WebSocket, WebRTC, WebTransport. Describes the basic principles of operation of these technologies, and also considers the key aspects of interaction with them, highlights their strengths and weaknesses in the context of web messenger development, analyzes the potential complexity of information system design, and by aggregate indicators selects the most optimal technology for the design of web messengers.

Keywords: Real-time data transfer, communication methods, asynchronous data communication, web messenger, website.

С развитием технологий передачи данных возросла популярность мессенджеров. В современном мире веб-мессенджеры стали неотъемлемой частью повседневной жизни – они позволяют пользователям общаться, делиться информацией, решать рабочие вопросы и быть в курсе последних событий. Однако проектирование таких систем требует особого внимания к выбору технологий передачи данных в реальном времени. Существует ряд проблем, которые могут возникнуть при разработке веб-мессенджера. К ним можно отнести следующие проблемы: проблемы с производительностью [1], проблемы с безопасностью [2], проблемы излишней сложности проектирования системы [3]. В данной статье мы рассмотрим основные

аспекты и критерии выбора подходящих технологий для создания эффективных и надёжных веб-мессенджеров.

Веб-разработка часто требует реализации механизмов обновления контента на странице в реальном времени. При работе с современными веб-приложениями реального времени незаменима возможность отправлять события с сервера на клиент. Именно этой необходимостью продиктовано то, что за годы работы было изобретено несколько методов для этой цели, каждый с собственным набором достоинств и недостатков. Есть разные ситуации, когда требуется передача информации в режиме реального времени. Например, при создании веб-мессенджера пользователю нужны постоянные уведомления о том, что сообщение отправлено, прочитано, отредактировано или удалено. Все эти сценарии объединяет одна общая черта: источник обновления данных находится на стороне сервера, поэтому для асинхронного получения информации о событиях требуется взаимодействие с серверной стороной. Передача данных в режиме реального времени в веб-приложениях позволяет отправлять и получать данные без необходимости перезагружать страницу.

В данной статье мы рассмотрим пять подходов к реализации этой функциональности: Long Polling, Server-Sent Events (SSE), WebSocket, WebRTC, WebTransport. Мы проанализируем каждый метод, выявим их плюсы, минусы и сложность реализации в разрезе проектирования веб-мессенджера.

В начале 2000-х годов Long Polling был одним из первых методов обновления контента на странице в режиме реального времени, предшествуя появлению более современных технологий, таких как WebSocket и Server-Sent Events. Этот подход стал популярным благодаря возможности преодолеть ограничения традиционного веб-протокола HTTP, который не поддерживает двустороннюю связь.

Принцип работы заключается в следующем: клиент отправляет HTTP-запрос на сервер, сервер в ответ может отправлять несколько порций данных перед отправкой окончательного результата и закрытием соединения. К преимуществам данного подхода можно отнести простоту реализации. Данный подход не является нативным, то есть HTTP запрос не совсем предусмотрен для такого типа взаимодействия. В связи с этим можно наблюдать следующие недостатки: задержки в передаче данных из-за ожиданий и таймаутов, высокая нагрузка на сервер из-за большого количества открытых соединений. Такой подход неэффективен при постоянном потоке данных, который потенциально может генерировать веб-мессенджер.

Server-Sent Events (SSE) — это технология, позволяющая серверу отправлять клиенту поток событий через однонаправленное соединение. Чтобы поддерживать открытое соединение, сервер может периодически отправлять пустые события, предотвращая закрытие соединения браузером из-за таймаута. SSE определён в спецификации HTML5 и поддерживается большинством современных браузеров. Этот подход можно использовать при создании веб-мессенджера, однако в этом случае сообщения от клиента должны отправляться с использованием стандартных HTTP-запросов.

Преимущества: Простая реализация на стороне сервера и клиента. Широкая поддержка как браузерами, так и веб-фреймворками. Автоматическое восстановление соединения при разрыве связи.

Недостатки: только односторонняя передача данных от сервера к клиенту, отсутствие полноценной поддержки старыми браузерами.

WebSocket стандартизирован в 2011 году как передовое решение для устойчивых двусторонних каналов связи между клиентом и сервером через единственное TCP-соединение. Этот механизм обеспечивает непрерывное соединение браузера с сервером без необходимости постоянного обновления страницы, позволяя данным мгновенно передаваться в обе стороны. После первоначального HTTP/HTTPS-запроса для установки соединения, происходит автоматическое переключение на специализированный бинарный протокол WebSocket через заголовок Upgrade[4]. TCP-поддержка гарантирует надежность и целостность передаваемой информации – ключевой фактор при разработке веб-мессенджеров, так как важность гарантированного уведомления пользователей стоит на первом месте.

Преимущества: WebSocket предлагает гибкую, полноценную двустороннюю асинхронную связь в реальном времени. Это наиболее универсальное решение с минимальными ограничениями для реализации разнообразных интерактивных сценариев на веб-платформах.

Недостатки: WebSocket характеризуется высокой ресурсоемкостью и, как правило, требует разработки отдельного серверного приложения для своей интеграции в проекты.

WebRTC (англ. web real-time communications — коммуникации по сети в реальном времени) - технология с открытым исходным кодом, ориентированная на прямую передачу потоковых данных между браузерами и другими поддерживающими устройствами. Данная технология имеет преимущество при проектировании аудио-видео связей, так как в WebRTC используются два аудиокодека, G.711 и Opus, а также видеокодеки VP8 и H.264. В WebRTC встроен протокол DTLS — протокол датаграмм безопасности транспортного уровня, он позволяет приложениям, основанным на коммуникациях посредством датаграмм, общаться безопасным способом, предотвращающим перехват, прослушивание, вмешательство, не нарушая защиты целостности данных или подделку содержимого сообщения [4].

Техническая особенность данной технологии в том, что она создавалась для взаимодействия между клиентами, без участия сервера, хотя она также применима и для коммуникации вида сервер-клиент, но в роли сервера будет выступать другое клиентское приложение, которое будет выполнять серверные функции, такие как обработка сообщений пользователей, запись данных сообщений в базу данных и т.д. То есть всего лишь будет происходить имитация сервера. Поэтому чтобы разработать структуру, в которой WebRTC будет работать корректно в рамках веб-мессенджеров, всё равно понадобится общий сервер событий, который будет работать по архитектуре веб-сокетов или Server-Sent Events. Именно поэтому WebRTC не может называться полноценной заменой вышеуказанных технологий.

WebTransport API — инновационный стандарт на базе WebSocket для оптимальной связи веб-клиентов и серверов, обеспечивающий сверхэффективную передачу данных за счёт HTTP/3 QUIC протокола. Особенности технологии выступают мультипоточковая отправка (включая неупорядоченные данные) как через надежные[5], так и ненадежные каналы. Для организации надежных каналов по аналогии с WebSocket выступает протокол TCP. Высокая надёжность в каналах TCP обеспечивается проверкой контрольных сумм и подтверждением приёма полученных данных без нарушений их состояния и смысла. Это требует дополнительных запросов между устройствами, но обеспечивает большую стабильность канала связи. А для организации высокопроизводительных, но менее надежных каналов используется протокол UDP. Это идеальный инструмент для разработки приложений с

высокими требованиями к сетевой производительности — в частности, веб-конференций и голосовых звонков в веб-мессенджерах.

WebTransport является обновленной версией WebSocket, поэтому он получил все преимущества вышеуказанной технологии. Но важно отметить, что в настоящее время WebTransport пребывает в состоянии рабочего проекта (working draft)[6] и пока не пользуется широкой популярностью в сообществе. По состоянию на декабрь 2024 года WebTransport не поддерживается в браузере Safari, а также не имеет нативной поддержки в таких популярных фреймворках как Node.js, Laravel. Поэтому при использовании WebTransport сложность проектирования веб-мессенджера значительно увеличивается.

Преимущества: имеет все преимущества WebSocket, работа с аудио-видео связью.

Недостатки: Сложность технологии, отсутствие полной поддержки браузерами.

Мы рассмотрели 5 основных популярных технологий интерактивного взаимодействия клиента с сервером для получения данных. При проектировании веб-мессенджера, среди наиболее удобных решений было выделено две технологии: WebSocket, как основу для передачи уведомлений о получении, прочтении, удалении, редактировании сообщений, благодаря гарантированной передаче данных с наименьшими задержками и полноценной двусторонней асинхронной связью между клиентом и сервером и WebRTC для предоставления возможности для аудио- и видеосвязи, так как данная технология имеет поддержку аудиокодеков G.711 и Opus, а также видеокодеков VP8 и H.264. Комбинация данных технологий способна заложить основу для создания эффективных и надежных веб-мессенджеров.

Список литературы

1. Николай Мациевский. Разгони свой сайт. Методы клиентской оптимизации веб-страниц. – М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2009. – 264 с.
2. Тутубалин П.И. Основные задачи прикладной теории информационной безопасности АСУ. Научно-технический вестник Санкт-Петербургского государственного университета информационных технологий, механики и оптики. 2007. № 39. С. 63-72.
3. Кожевников А.Ю., Тутубалин П.И., Кирпичников А.П., Мокшин В.В. О построении подсистемы удалённого мобильного доступа к информационным ресурсам некоторой организации. Вестник Технологического университета. 2018. Т. 21. № 2. С. 139-147.
4. Моисеев В.С., Тутубалин П.И. К задаче определения вероятностных характеристик информационной безопасности разрабатываемых автоматизированных систем управления. Депонированная рукопись ВИНТИ № 26-В2007 11.01.2007.
5. Кожевников А.Ю., Тутубалин П.И., Кирпичников А.П., Мокшин В.В. О разработке математических моделей, методов и программного обеспечения для проектирования перспективных изделий запрос-ответной аппаратуры. Вестник Технологического университета. 2018. Т. 21. № 2. С. 155-162.
6. Рахманов А.С., Тутубалин П.И. Автоматизация тестирования безопасности веб приложений. В сборнике: Молодёжь и наука: актуальные проблемы фундаментальных и прикладных исследований. Материалы IV Всероссийской национальной научной конференции студентов, аспирантов и молодых учёных. В 4-х частях. Редколлегия: Э.А.

References

1. Nikolai Matsievsky. Disperse your site. Methods of client optimization of web pages. - M.: Internet University of Information Technologies, Binom. Laboratory of Knowledge, 2009. – p.264
 2. Tutubalin P.I. The main tasks of the applied theory of information security of automated control systems. Scientific and Technical Bulletin of the St. Petersburg State University of Information Technologies, Mechanics and Optics. 2007. No. 39. pp. 63-72.
 3. Kozhevnikov A.Yu., Tutubalin P.I., Kirpichnikov A.P., Mokshin V.V. On the construction of a subsystem for remote mobile access to information resources of an organization. Bulletin of the Technological University. 2018. Vol. 21. No. 2. pp. 139-147.
 4. Moiseev V.S., Tutubalin P.I. On the problem of determining the probabilistic characteristics of information security of automated control systems under development. Deposited manuscript of VINITI No. 26-In 2007 11.01.2007.
 5. Kozhevnikov A.Yu., Tutubalin P.I., Kirpichnikov A.P., Mokshin V.V. On the development of mathematical models, methods and software for the design of promising products of request-response equipment. Bulletin of the Technological University. 2018. Vol. 21. No. 2. pp. 155-162.
 6. Rakhmanov A.S., Tutubalin P.I. Automation of web application security testing. In the collection: Youth and science: current problems of fundamental and applied research. Materials of the IV All-Russian National Scientific Conference of students, postgraduates and young Scientists. In 4 parts. Editorial board: E.A. Dmitriev (editor's note), A.V. Kosmynin (Deputy editor's note). Komsomolsk-on-Amur, 2021. pp. 298-301.
-