В «джентльменському наборі» будь-якого користувача всесвітньої мережі повинен бути якійсь месенджер (сервіс миттєвого обміну повідомлень). Саме тому їх є незліченна кількість. Це і telegram, і Whatsapp, і Signal та багато інших Особливістю програм такого типу є те, що користувач має бажання отримати доступ до цього сервісу з будь-якого пристрою, який зараз під рукою у користувача. Саме тому виникла ідея використати для доступу до сервісу в якості клієнта звичайний браузер.

Проблема розробки програми такого типу полягає в тому, що ми створюємо все більш складні веб-додатки на основі протоколу, який, як це не парадоксально, був задуманий і розроблений для набагато простіших форм взаємодії. HTTP не має вбудованої підтримки стану чи навіть безпеки. Його основне припущення полягає в тому, що клієнт надсилає запит, а веб-сервер видає відповідь. Загалом, це означає, що немає запиту, немає відповіді.

Опитування

Враховуючи обмеження HTTP-запиту/відповіді, опитування є єдиним можливим способом налагодити живий зв’язок між клієнтом і веб-сервером. Клієнт надсилає запити, коли йому зручно, а сервер старанно відповідає. Ключем до оцінки ефективності опитування є фактичне значення, яке ви надаєте виразу «як йому зручно». В основі будь-якого рішення, заснованого на опитуванні, завжди лежить запит клієнта. Запит залишається до тих пір, поки сервер не відповість. Будь-який запит, що очікує на розгляд, споживає з’єднання браузера і, що більш важливо, залучає потік сервера, роблячи цей цінний ресурс недоступним для інших запитів. Коротше кажучи, занадто часті запити створюють тиск на веб-сервер.

Але хіба здатність сервера обробляти зростаючу кількість запитів не є ідеєю масштабованості? Насправді опитування не створює нових проблем щодо масштабованості, але додає до сервера значну кількість нових запитів, які необхідно враховувати, щоб забезпечити масштабований і високопродуктивний сервер. Давайте подивимося, як ми можемо реалізувати опитування.

Опитування AJAX

У своїх колонках за грудень 2011 ( msdn.microsoft.com/magazine/hh580729 ) і січень 2012 ( msdn.microsoft.com/magazine/hh708746 ) я представив структуру для контролю ходу віддаленої операції. Вся структура була заснована на викликах AJAX. По-перше, клієнт викликає кінцеву точку сервера, щоб почати потенційно тривале завдання; далі він встановлює таймер для здійснення одночасних викликів до іншої кінцевої точки кожні 500 мілісекунд. Коли завдання сервера виконує свою роботу, воно оновлює відоме розташування. Кінцева точка сервера, яка періодично викликається, просто перевіряє дані в цьому місці та повертає клієнту будь-який вміст, який вона знаходить.

Цей шаблон опитування AJAX чудово працює і широко використовується в багатьох галузевих сценаріях. Більшість сайтів, які надають оновлення новин або прямі результати, дотримуються цієї схеми. У своїх статтях я просто адаптував шаблон до конкретного сценарію моніторингу ходу віддаленої операції.

Зрештою, проблема опитування AJAX полягає у визначенні правильного інтервалу оновлення — такого, який представляє гарний баланс між тиском на сервер і точністю інформації, яка повідомляється користувачам. Ефективне рішення AJAX-опитування надає конкретне значення «на зручність клієнта» під час надсилання запитів.

Довге опитування

До AJAX розробники використовували мета-тег оновлення HTML, щоб інструктувати браузери оновлювати сторінку кожні задану кількість секунд. За допомогою опитування AJAX ви досягаєте того ж результату, але набагато плавніше.

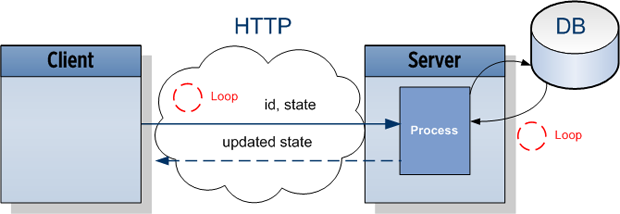
Однак для збільшення кількості заяв цієї форми опитування недостатньо. Опитування AJAX майже неминуче вносить затримку між появою події на сервері та повідомленням про подію клієнту. Налаштувавши частоту оновлень, ви можете знайти хороші рішення, але опитування AJAX не може забезпечити постійне й безперервне з’єднання, яке, здається, потрібне багатьом (переважно соціальним) додаткам сьогодні.

Рішенням цих днів, здається, є довгі опитування. Цікаво, що коли AJAX з'явився багато років тому, розумніші опитування AJAX замінили довгі опитування, оскільки останні були не дуже ефективними в Інтернеті. Тривале опитування полягає в тому, щоб робити через Інтернет те ж саме, що й у настільному сценарії. При тривалому опитуванні клієнт розміщує запит, а сервер не відповідає, поки не отримає інформацію для повернення. Веб-клієнт зберігає з'єднання в очікуванні, яке закривається лише тоді, коли може бути повернута дійсна відповідь. Це саме те, чого ви хочете сьогодні, за винятком того, що це може сповільнити роботу веб-сервера.

Довгий опитування надсилає на сервер меншу кількість запитів у порівнянні з опитуванням AJAX, але кожен запит може зайняти набагато більше часу. З часом це може завдати шкоди здоров’ю веб-сервера, оскільки він утримує потоки сервера задіяними, поки не буде згенеровано відповідь. З меншою кількістю робочих потоків, доступних у певний момент часу, веб-сервер неминуче повільніше відповідає на будь-які інші запити, які він отримує. Щоб бути ефективним, тривале опитування потребує серйозної роботи з впровадження та розширених навичок багатопотокового та паралельного програмування. Тож роками тривале опитування насправді не було варіантом, але це не мало значення, оскільки не було попиту на безперервне підключення.

**Long pooling**

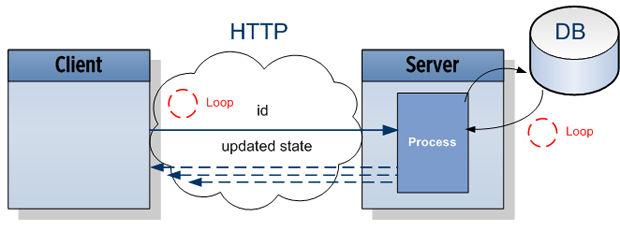
Пояснення техніки long pooling



Як це працює

Клієнтська програма (браузер) надсилає запит із ідентифікатором одержувача події та поточним станом на сервер через HTTP. Він створює окремий процес на сервері, який багаторазово перевіряє стан поки він не зміниться. Коли стан зрештою змінився, клієнт отримує відповідь сервера і надсилає наступний запит до сервера.

**Події, надіслані сервером (Server Side Event)**



**Як це працює**

Традиційно веб -сторінка повинна надсилати запит на сервер для отримання нових даних; Тобто сторінка запитує дані з сервера. За допомогою подій сервера, сервер може в будь-який час надсилати нові дані на веб-сторінку, натиснувши повідомлення на веб-сторінку. Ці вхідні повідомлення можна трактувати як події + дані всередині веб -сторінки.

Длинные опросы

Длинные опросы – это самый простой способ поддерживать постоянное соединение с сервером, не используя при этом никаких специфических протоколов (типа WebSocket или Server Sent Events).

Его очень легко реализовать, и он хорошо подходит для многих задач.

Частые опросы

Самый простой способ получать новую информацию от сервера – периодический опрос. То есть, регулярные запросы на сервер вида: «Привет, я здесь, у вас есть какая-нибудь информация для меня?». Например, раз в 10 секунд.

В ответ сервер, во-первых, помечает у себя, что клиент онлайн, а во-вторых посылает весь пакет сообщений, накопившихся к данному моменту.

Это работает, но есть и недостатки:

Сообщения передаются с задержкой до 10 секунд (между запросами).

Даже если сообщений нет, сервер «атакуется» запросами каждые 10 секунд, даже если пользователь переключился куда-нибудь или спит. С точки зрения производительности, это довольно большая нагрузка.

Так что, если речь идёт об очень маленьком сервисе, подход может оказаться жизнеспособным, но в целом он нуждается в улучшении.

Длинные опросы

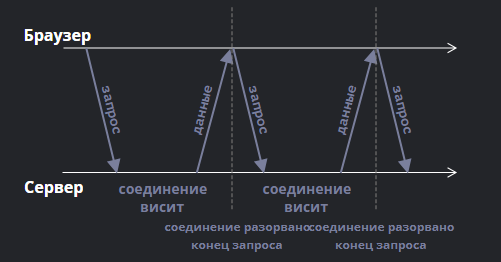
«Длинные опросы» – гораздо лучший способ взаимодействия с сервером.

Они также очень просты в реализации, и сообщения доставляются без задержек.

Как это происходит:

1. Запрос отправляется на сервер.
2. Сервер не закрывает соединение, пока у него не возникнет сообщение для отсылки.
3. Когда появляется сообщение – сервер отвечает на запрос, посылая его.
4. Браузер немедленно делает новый запрос.

Для данного метода ситуация, когда браузер отправил запрос и удерживает соединение с сервером в ожидании ответа, является стандартной. Соединение прерывается только доставкой сообщений.



Если соединение будет потеряно, скажем, из-за сетевой ошибки, браузер немедленно посылает новый запрос.