

Асинхронные API – все, что вам нужно знать

22.06.2023

Возможно, вы слышали об интерфейсе прикладного программирования (API), но знаете ли вы, что их существует несколько видов? Оказывается, не каждый API является REST API, который возвращает результаты немедленно. (Это, кстати, самый распространенный тип API в мире веб-сервисов). Скорее всего, вы сталкивались со многими видами API, даже не подозревая об этом. В конце концов, это способ общения с приложениями, поэтому они почти всегда работают за кулисами. Даже щелчок на профиле Facebook отправляет вызовы API, которые возвращают почти всю информацию, отображаемую на следующей странице. Итак, каковы другие виды API? Существует несколько различных архитектур и протоколов. В этой статье мы сосредоточимся на асинхронных API и их сравнении с синхронными API.

Что такое асинхронный API?

Асинхронный API, или сокращенно асинхронный API, – это API, который не предоставляет данные немедленно. Подразумевается, что пользователю, скорее всего, придется подождать, пока данные из его запроса будут доставлены. Это может быть несколько минут, часов или больше. Это зависит от конкретной службы или приложения. Есть много ситуаций, когда это имеет смысл, например, когда спрос на услугу настолько велик, что она не может мгновенно доставить все данные пользователям. Иногда просто нет смысла в том, чтобы услуга была мгновенной, потому что цель или функция коммуникации сама по себе не является мгновенной. Например, отправка и получение электронной почты – это услуга, которая не обязательно должна быть мгновенной. Конечно, кто-то может ответить практически мгновенно, используя электронную почту. Однако ожидается, что для получения ответа потребуется некоторое время.

Асинхронные и синхронные API

Как вы уже догадались, если асинхронный API – это API, который возвращает данные позже, то синхронный API должен возвращать данные очень быстро, если не мгновенно. Синхронные API также обычно используют HTTP или HTTPS для передачи данных – протоколы, которые мы используем для навигации в Интернете. Вы вводите URL-адрес веб-сайта. Затем он отправляет HTTP или HTTPS запрос на сервер. После этого вы получаете ответ HTTP или HTTPS. Это и есть синхронный API в действии! Еще одно отличие: синхронные API, как правило, однонаправленные, а асинхронные – двунаправленные. Это означает, что, хотя синхронные API предоставляют вам данные с сервера, вы не будете одновременно отправлять данные на сервер и скачивать их с него.

Асинхронные API, с другой стороны, являются двунаправленными. Это означает, что данные могут идти в обоих направлениях одновременно. Вспомните систему мгновенного обмена сообщениями, где вы можете одновременно отправлять и получать сообщения. Вам не гарантирован немедленный ответ (хотя обычно вы получаете уведомление, когда данные готовы). Некоторые API не укладываются в эти рамки. Возможно, API может использовать HTTP, но при этом поддерживать форму мгновенного обмена сообщениями. В качестве примера можно привести XMLHttpRequest Web API и GraphQL API. Оба они поддерживают синхронную и асинхронную форму коммуникации.

Когда использовать асинхронные API

Говоря о случаях использования, давайте обсудим несколько возможных причин, по которым вы можете рассмотреть возможность использования асинхронного API. Первый и наиболее очевидный пример – это использование электронной почты или другой формы отложенной связи, например, мгновенного обмена сообщениями. Для API можно использовать несколько различных протоколов асинхронной передачи сообщений, таких как Webhooks,

WebSockets, подписки GraphQL и события, отправляемые сервером (SSE). Асинхронные API также могут управлять транзакциями в мире банковского дела. Кредитные транзакции могут обрабатываться с помощью асинхронного API для обеспечения единого метода облегчения платежей между несколькими продавцами и банком.

Поскольку часто можно запланировать платежи, использование асинхронных API имеет смысл. Мир работал бы по-другому, если бы все происходило мгновенно, особенно в банковской сфере. Асинхронные API также используются в многопользовательских играх. Часто многопользовательские игры собирают данные от нескольких игроков и возвращают ответ всем игрокам после обработки данных от каждого, обычно это происходит с заданным интервалом или “тиком”. Это означает, что пользователи постоянно отправляют и получают данные от сервера одновременно. Хотя может показаться, что это происходит мгновенно, на самом деле обмен данными не происходит в реальном времени из-за различной задержки между игроками.

Примеры асинхронных API

Мессенджеры



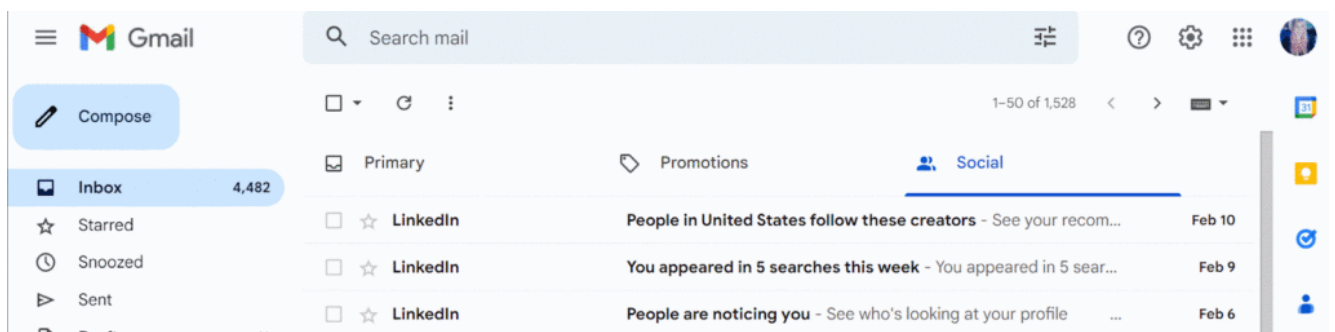
Одним из чрезвычайно популярных примеров асинхронного API в действии является Facebook Messenger. Это клиент для обмена мгновенными сообщениями, доступный как в браузере, так и в виде отдельного приложения на мобильных устройствах. В основном он использует API Send для отправки сообщений пользователям, включая текст, вложения и многое другое. Для отправки сообщения используется POST-запрос на одной из конечных точек API. Различные параметры также используются для указания подробностей о сообщении, таких как тип сообщения, наличие или отсутствие вложений, или даже то, набирает ли кто-то сообщение в данный момент или уже прочитал его.

Многопользовательские видеоигры



Многопользовательские видеоигры – еще один популярный способ использования асинхронных API в реальном мире. Скорее всего, вы слышали о вышеупомянутом примере – World of Warcraft, или сокращенно WoW. Как упоминалось ранее, многопользовательские видеоигры часто используют асинхронные API для обеспечения связи с игроками и между ними. Хотя WoW использует несколько типов API для реализации всех своих функций, Event API является примером асинхронного API, используемого для передачи сообщений. Сюда входит внутриигровая почтовая система и чат, используемый для общения с другими игроками. Конечная точка также работает с POST-запросами, как и в предыдущем примере.

Gmail



Gmail – еще один популярный пример асинхронного API в действии. Хотя некоторые части API Gmail могут функционировать

аналогично синхронному API, в основном это асинхронный API, поскольку при работе с электронной почтой не ожидается немедленного ответа. Сообщения можно создавать, изменять и отправлять через соответствующую конечную точку API, используя запрос POST или PUT, а также все важные HTTP-заголовки, указывающие количество отправляемых байт и тип загружаемого содержимого. После успешной отправки сообщения вы получите в ответ код состояния HTTP 200 OK, а также любые метаданные в виде ответа JavaScript Object Notation (JSON).

В ответе JSON будут возвращены такие данные, как заголовки запроса, тип многоцелевых расширений интернет-почты (MIME), предполагаемый размер файла и т. д. Эта информация может быть использована в других приложениях или запросах. Если бы вам нужно было выбрать только один пример из списка для изучения, это был бы лучший выбор. Он имеет самую лучшую документацию и настолько вездесущ, что даже самый случайный пользователь может придумать, как использовать его в повседневной жизни. Многие другие продукты Google также имеют асинхронные аспекты, например, Google Sheets и Google Docs.

Asana

Task name	Assignee	Due date	Team	
Business				
✓ Customer base grows by 15% 👍 3		Sep 8	Sales	
✓ Customer churn decreases by 8% 💬 2 🗒️ 5		Sep 12	Support	
Internal				
✓ Open NYC office		Sep 16	Facilities	
✓ Hire 18 new team members 👍 8		Sep 6	Recruiting	
Product				
✓ Increase usability with app refresh 🗒️ 3		Sep 14	Mobile	
✓ Secure new integration partnerships		Aug 30	Platform	
Company				
✓ Raise Series B financing 🗒️ 7		Sep 1	Executive	

Asana – это в первую очередь способ асинхронного общения по поводу работы. Это инструмент управления проектами или задачами с множеством функций, таких как сроки выполнения, формы, комментарии, календари, поиск, отчеты и управление командой. Это еще один RESTful API, использующий возможности HTTP. Однако он не является полностью синхронным. Некоторые асинхронные функции включают использование веб-крючков, управление пользовательскими полями, отправку/получение вложений и управление задачами. Как и в других API в этом списке, методы HTTP обычно используются для создания, чтения, обновления или удаления данных через API, при этом для более детального уточнения можно использовать различные заголовки.

Асинхронные и синхронные API – выбор лучшего для вашего проекта

При выборе синхронного или асинхронного API для вашего проекта важно учитывать несколько моментов. Это во многом зависит от

проекта и случая использования, но давайте рассмотрим несколько важнейших особенностей и различий, которые следует учитывать при принятии решения. Асинхронные API подразумевают отложенный ответ. Другие функции могут продолжать работать, пока асинхронная функция ожидает ответа в фоновом режиме. Это идеально подходит для систем обмена сообщениями, но не очень хорошо, если вы просто читаете данные в базе данных. Асинхронные API также являются двунаправленными, в отличие от односторонних.

Опять же, это идеально подходит для случаев, когда между сервером и клиентом будет происходить много обменов данными, но, вероятно, не обязательно, если связь односторонняя. Также следует учитывать, нужен ли ответ на каждый запрос или можно обрабатывать несколько запросов параллельно. Асинхронные API отлично подходят для последнего, в то время как синхронные API могут быть лучшим вариантом для первого. Наконец, возможно, решите, насколько сложным вы хотите сделать API. Асинхронные API более сложны для всех участников процесса, но это не обязательно лишает их права быть лучшим выбором во многих сценариях.

Начало работы

В заключение можно сказать, что существует множество различных вариантов использования как синхронных, так и асинхронных API. Основные различия заключаются в том, что синхронные API обычно отвечают почти мгновенно и являются односторонними, а асинхронные API отвечают позже и являются двунаправленными. Асинхронные API также могут параллельно обрабатывать большое количество запросов, чтобы ответить на них позже. Многие популярные веб-сервисы используют асинхронные API для своей работы, большинство из них являются своего рода службами обмена сообщениями, хотя существуют и другие асинхронные службы. Не бойтесь играть с их API, чтобы узнать, как все это работает на практике.