

РЕНДЕРИНГ ВЕБ-СТОРИНОК

Розробникам часто доводиться приймати рішення, які вплинуть на всю архітектуру програми. Веб-розробникам важливо вибрати правильне місце для реалізації логіки і рендерингу додатка. Це може бути непросто, так як сайт можна створити різними способами. При виборі підходу для рендерингу потрібно розуміти різницю між можливими варіантами, щоб не прогадати з продуктивністю.

Серверний рендеринг - рендеринг на сервері клієнтської частини або універсального додатка в HTML (Server-Side Rendering). При серверному рендерингу у відповідь на запит на сервері генерується весь HTML сторінки. Це виключає необхідність додаткових запитів даних з боку клієнта, так як сервер бере всю роботу на себе перш ніж відправити відповідь. Такий підхід дозволяє домогтися швидкого відтворення змістовного контенту. Виконання логіки сторінки і рендеринг на сервері дозволяють уникнути відправки на клієнт великої кількості JavaScript-коду, це призводить до того, що час коли сторінка стає інтерактивною - мінімальний. І це логічно, адже при серверному рендерингу користувачеві відсилаються тільки текст і посилання. Цей підхід добре спрацює на широкому діапазоні пристроїв, а також відкриє можливості для цікавих браузерних оптимізацій на кшталт потокового парсинга документа.

При використанні серверного рендерингу користувачам не потрібно чекати завершення роботи JavaScript, що віднімає ресурси процесора, перш ніж вони зможуть почати працювати з сайтом. Навіть якщо не можна уникнути використання стороннього JavaScript, серверний рендеринг дозволяє зменшити кількість вашого власного JavaScript і залишає більше ресурсів для всього іншого. Однак у цього підходу є один недолік: формування сторінки на сервері займає певний час, що може призвести до більшого часу для отримання першого байта з серверу.

Клієнтський рендеринг - рендеринг додатка на стороні клієнта (в браузері), зазвичай за допомогою DOM (Client-Side Rendering). Клієнтський рендеринг передбачає рендеринг сторінок прямо в браузері за допомогою JavaScript. Вся логіка, отримання даних, шаблонізація і маршрутизація обробляються на клієнті, а не на сервері. При такому підході складно підтримувати високу швидкість на мобільних пристроях. Основний недолік клієнтського рендерингу полягає в тому, що кількість необхідного JavaScript зазвичай збільшується разом зі зростанням додатка. Ситуація погіршується з підключенням нових JavaScript-бібліотек, поліфілів та іншого стороннього коду, який змагається між собою за обчислювальні потужності і часто вимагає обробки, перш ніж вміст сторінки можна буде відобразити. Для рішення з клієнтським рендерингом, які покладаються на великі JavaScript-файли, варто розглянути розділення коду і ледяче завантаження JavaScript - «завантажуйте тільки те, що вам потрібно і тільки коли це потрібно».

Універсальний рендеринг намагається усунути недоліки серверного та клієнтського рендерингу, використовуючи обидва підходи. Навігаційні запити на кшталт повного завантаження або перезавантаження сторінки обробляються сервером, який рендерить додаток в HTML, потім JavaScript і дані, використовувані для рендеринга, вбудовуються в документ. При правильній реалізації час до першого змістовного відтворення буде як при серверному рендерингу, а повторний рендеринг буде проводитися на клієнті за допомогою техніки, званої (ре)гідратацією. Це нове рішення, тим не менш не позбавлене певних проблем з продуктивністю.

Основний недолік універсального рендерингу з регідратацією полягає в тому, що такий підхід може негативно вплинути на час до інтерактивності – готовності до взаємодії. Сторінки часто виглядають оманливо готовими і інтерактивними, але по факту не можуть ніяк реагувати на дії користувача до виконання JavaScript на стороні клієнта і приєднання обробників подій. Це може зайняти кілька секунд або навіть хвилин на мобільних пристроях.

Розробники часто враховують вплив SEO при виборі стратегії для рендерингу. Серверний рендеринг часто вибирають для формування сторінки з «завершеним виглядом», який пошукові роботи можуть легко інтерпретувати. Пошукові роботи можуть виконувати Javascript, але найчастіше в їх механізмах рендерингу сторінок є обмеження, про які варто пам'ятати. Клієнтський рендеринг може спрацювати, але не без додаткового тестування. У разі сумнівів для перевірки можна скористатись інструментом Mobile Friendly Test від Google, щоб переконатися, що обраний підхід робить саме те, на що ви розраховуєте. Він показує сторінку такою, якою її бачить робот Google, а також знайдений серіалізований HTML-зміст (після виконання JavaScript) і всі помилки, що виникли під час рендерингу.