

Server-side rendering

Кучеренко Даніїл, Яковенко Єлизавета



Загальний огляд

SSR (Server-Side Rendering) генерує повний HTML для сторінки на сервері у відповідь на запит (перехід по посиланню). Вміст сторінки може включати дані з бази даних або зовнішнього API

Це дозволяє уникнути додаткових запитів даних, наповнення шаблонів на стороні клієнта, оскільки вони обробляються до того, як браузер отримує відповідь. При серверному рендерингу користувачу надсилаються лише текст та посилання. Цей підхід добре працює на широкому діапазоні пристроїв та мережевих умов. Користувачам не потрібно чекати на завершення роботи JavaScript, яка забирає ресурси процесора, перед тим як вони зможуть почати працювати з сайтом.

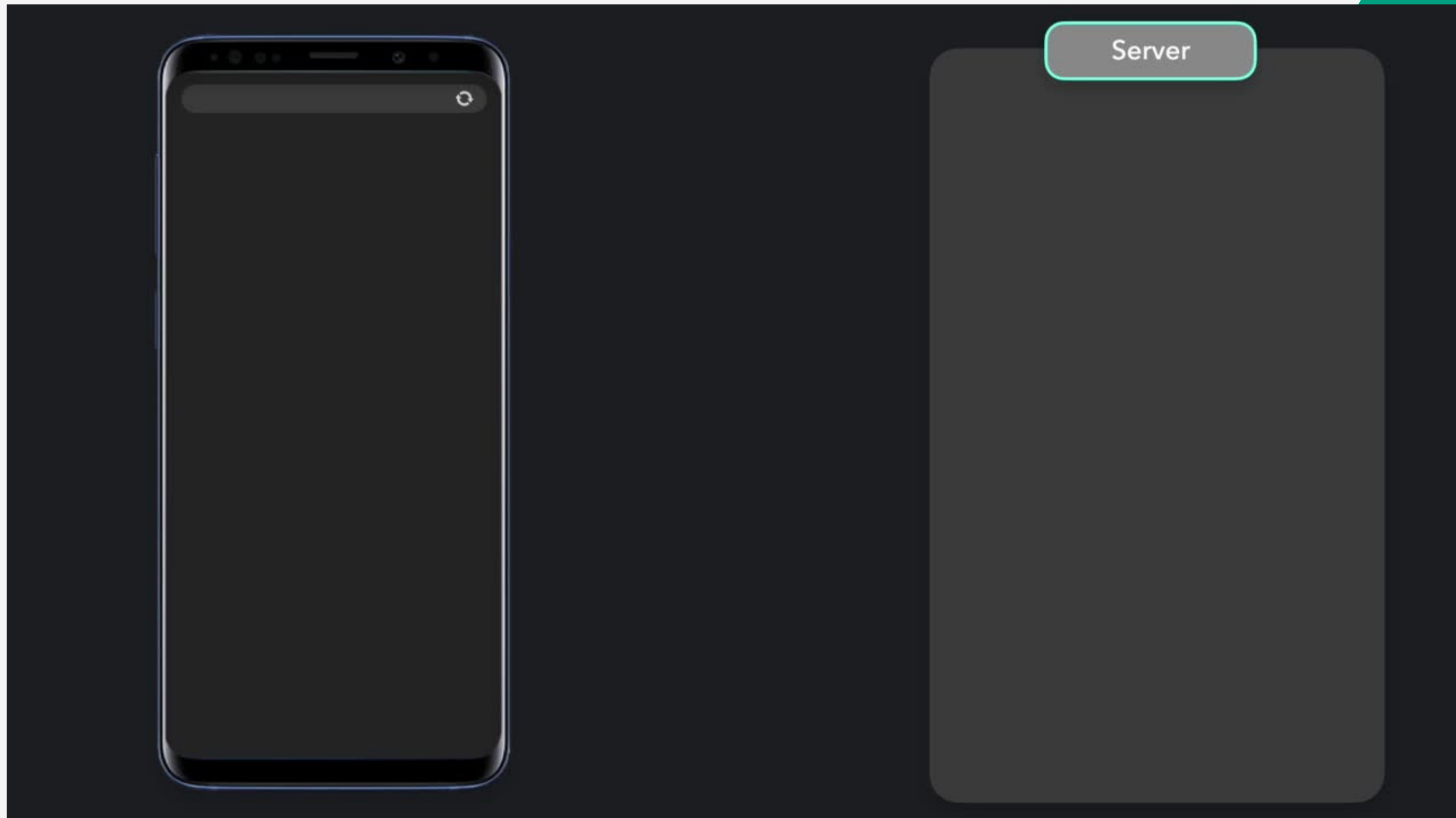


Загальний огляд

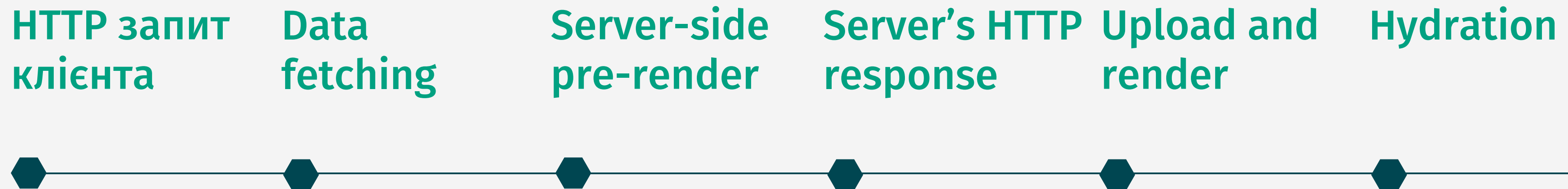
Операції connect та fetch обробляються на сервері + HTML, необхідний для форматування вмісту, також генерується на сервері
=> за допомогою SSR ми можемо уникнути додаткових поворотів для отримання даних і шаблонів
=> код візуалізації не потрібен на клієнті, і відповідний JavaScript не потрібно надсилати клієнту.



Модель роботи



Кроки у процесі SSR



[Back to Agenda Page](#)


Переваги SSR

- Менша кількість JavaScript призводить до швидшого FCP і TTI
- SEO
- Краще для доступності
- Менше проблем з індексацією соціальних мереж

Недоліки SSR

- Уповільнення часу переходу між сторінками
- Більш складна технологія в реалізації
- Кешування
- Навантаження на сервер
- Вартість
- Синхронізація станів на клієнті та сервері

Інші проблеми



**Кінцевий користувач працює з
малопотужним пристроєм, наприклад
зі смартфоном старого покоління**

**Кінцевий користувач має повільне
інтернет-з'єднання**

Засоби реалізації

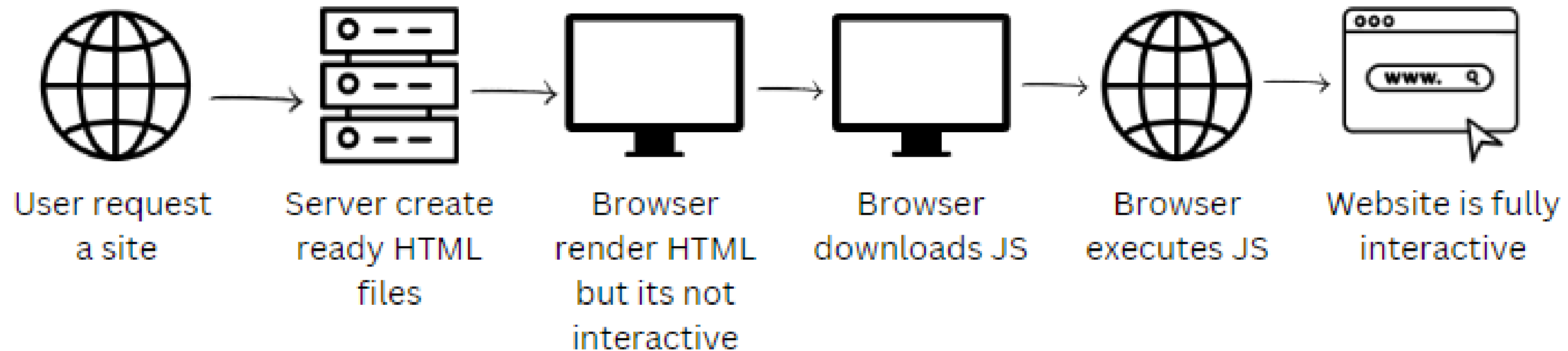
Розглянемо докладніше, як працює SSR:

- У нас має бути сервер, який виконує наш додаток так само, як робив би це користувач у браузері. Робити запити на необхідні ресурси, відображати весь необхідний HTML.
- Сервер віддає клієнту готовий HTML зі станом, всі потрібні скрипти JS, CSS та інші ресурси.
- Клієнт, отримуючи HTML та ресурси, синхронізує стан і працює з додатком як із звичайним Single Page Application. При цьому важливим моментом є те, що стан має синхронізуватись.



Схематично

SSR (Server-side Rendering)



Фреймворки



React	Vue	Angular
<ul style="list-style-type: none">• Next.js• Gatsby• Remix	<ul style="list-style-type: none">• Nuxt.js• Quasar	Universal package