next/streaming

The experimental `next/streaming` module provides streaming related APIs to port the existing functionality of Next.js apps to streaming scenarios and facilitate the usage of React Server Components.

unstable_useWebVitalsReport

Next.js provides an `_app` component-level function, `reportWebVitals`, for tracking performance metrics. With Server Components, you may have a pure server-side custom `_app` component (which doesn't run client effects) so the existing API won't work.

With the new `unstable_useWebVitalsReport` API, you're able to track Core Web Vitals in client components:

```
// pages/_app.js
import { unstable_useWebVitalsReport } from 'next/streaming'

export default function Home() {
  unstable_useWebVitalsReport((data) => {
    console.log(data)
  })

  return <div>Home Page</div>
}
```

This method could also be used to replace statically exported `reportWebVitals` functions in your existing `_app`:

```
// pages/_app.server.js
import Layout from '../components/layout.client.js'

export default function App({ children }) {
  return <Layout>{children}</Layout>
}
```

```
// components/layout.client.js
import { unstable_useWebVitalsReport } from 'next/streaming'

export default function Layout() {
  unstable_useWebVitalsReport((data) => {
    console.log(data)
  })

  return (
    <div className="container">
      <h1>Hello</h1>
      <div className="main">{children}</div>
    </div>
  )
}
```

unstable_useRefreshRoot

Since Server Components are rendered on the server-side, in some cases you might need to partially refresh content from the server.

For example, a search bar (client component) which displays search results as server components. You'd want to update the search results while typing and rerender the results list with a certain frequency (e.g. with each keystroke or on a debounce).

The `unstable_useRefreshRoot` hook returns a `refresh` API to let you re-render the React tree smoothly without flickering. This is only allowed for use on the client-side and will only affect Server Components at the moment.

```
// pages/index.server.js
import Search from '../components/search.client.js'
import SearchResults from '../components/search-results.server.js'

function Home() {
  return (
    <div>
      <Search />
      <SearchResults />
    </div>
  )
}
```

```
// components/search.client.js
import { unstable_useRefreshRoot as useRefreshRoot } from 'next/streaming'

export default function Search() {
  const refresh = useRefreshRoot()

  return (
    <SearchUI
      onChange={() => {
        refresh()
        // Or refresh with updated props:
        // refresh(nextProps)
      }}
    />
  )
}
```