

Mô hình hoạt động của Blazor Server

[Hướng dẫn tự học lập trình ASP.NET Core toàn tập](#) > [Mô hình hoạt động của Blazor Server](#)

Blazor là loại công nghệ cho phép phát triển web client UI hoàn toàn với .NET và C#. Blazor hiện đang có hai mô hình: Blazor Server và Blazor WebAssembly.

Trong bài viết trước bạn đã trải nghiệm sơ lược cách làm việc với ứng dụng [Blazor](#). Bài viết này sẽ giúp bạn hiểu rõ hơn về cách hoạt động của mô hình Blazor Server, những ưu nhược điểm của nó, và giúp bạn đánh giá xem nó có phù hợp với yêu cầu của mình hay không.

NỘI DUNG CỦA BÀI [Ấn]

1. Blazor Server
2. Lựa chọn Blazor Server
3. Về khả năng chịu tải của Blazor Server
4. Kết luận

Blazor Server

Blazor Server app chứa các component (thành phần cơ bản của ứng dụng Blazor) trên server và xử lý tương tác trên UI thông qua SignalR theo thời gian thực.

Nếu bạn chưa biết: SignalR là một framework khác của Microsoft cho phép client (trên trình duyệt) và server tương tác với nhau theo thời gian thực và theo hai chiều. Trong tương tác Request/Response, client luôn là người phát yêu cầu và server trả lời cho yêu cầu đó. Server không thể tự mình “bắt chuyện” trước với client. SignalR thay đổi cách thức tương tác Request/Response của ứng dụng web truyền thống, trong đó server có thể đẩy (push) nội dung tới client, và tương tác diễn ra theo thời gian thực.

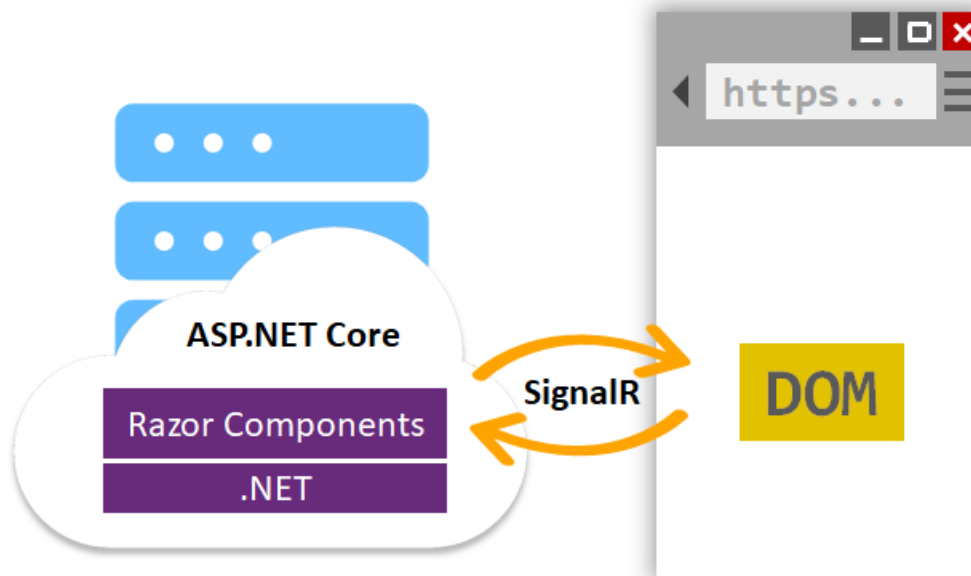
Khi người dùng tương tác với ứng dụng (client chạy trên trình duyệt), các sự kiện của UI (giao diện) được gửi về server (theo thời gian thực) để các component của ứng dụng xử lý. Tức là, sự kiện của UI không được xử lý “tại chỗ” trong trình duyệt như các web client viết bằng JavaScript.

Mỗi khi component xử lý một sự kiện (ở server), hình thức nó được “vẽ” (render) lại theo nội dung mới. Blazor so sánh với hình thức cũ của chính component đó để lấy những thay đổi. Những thay đổi này được gửi ngược lại cho chương trình client trên trình duyệt. Sau khi nhận được những thay đổi này, client mới thực sự cập nhật một phần DOM (tức là những gì người dùng nhìn thấy).

Toàn bộ quá trình tương tác này được thực hiện thông qua SignalR. Quá trình xử lý hoàn toàn tiến hành ở server.

Diễn đạt theo cách khác, Blazor Server nghĩa là “UI được tạo ra ở Server và gửi cho client theo thời gian thực”, và “Client chỉ áp dụng những thay đổi do Server cung cấp”.

Lưu ý rằng, Blazor Server không tạo lại trang mới mà chỉ tính toán ra những thay đổi cần thiết trên DOM (giống như những gì JavaScript framework thực hiện), đóng gói những thay đổi này và gửi lại cho trình duyệt. Page không được tải lại, tương tự như đối với ứng dụng SPA (Single Page Application).



Để thực hiện quá trình tương tác qua mạng, ứng dụng Blazor Server sẽ tải lên trình duyệt file `blazor.server.js`. File này chịu trách nhiệm thiết lập liên kết client với server, đồng thời duy trì liên kết, khôi phục trạng thái của ứng dụng (ví dụ, khi bị đứt mạng).

Với mô hình này, Blazor Server app được thừa hưởng tất cả những lợi ích của việc chạy trên .NET core, như hiệu suất xử lý, công cụ hỗ trợ, thư viện .NET Standard. Blazor Server app do đó cũng không chịu những hạn chế của trình duyệt.

Đồng thời, Blazor Server app không hề bộc lộ code cho trình duyệt. Những gì trình duyệt nhận được đơn giản là các cập nhật DOM đã được tính toán sẵn.

Lựa chọn Blazor Server

Trước hết cần khẳng định, Blazor Server được sử dụng để viết các client web có tính tương tác cao cũng như các Single-page app (SPA). Ở khía cạnh này, Blazor Server tạo ra hiệu quả tương tự như Angular hay React.

Blazor Server cũng có thể tạo ra những app mới hỗ trợ cho các app có sẵn trên MVC và Razor Page. Lý do là Blazor được thiết kế để hoạt động cùng với các mô hình phát triển ứng dụng web sẵn có trên Asp.net Core chứ không phải thay thế chúng.

Với đặc thù hoạt động của mình, Blazor Server phụ thuộc nhiều vào đường truyền mạng. Nếu bạn có đường truyền tốc độ cao và ổn định, Blazor Server sẽ đem lại hiệu quả tốt nhất. Cũng vì lý do này, Blazor Server không hoạt động được nếu mất kết nối mạng. Bản thân Blazor Server app cũng không đòi hỏi phải tải nhiều file nặng nề về browser mà chỉ cần một vài file đơn giản để thiết lập liên kết giữa trình duyệt với server.

Do thời gian trao đổi dữ liệu giữa client và server (để tính ra các cập nhật DOM), Blazor Server không thực sự phù hợp với một số ứng dụng đòi hỏi rất cao về tốc độ cập nhật UI như game trong thời gian thực hoặc các ứng dụng vẽ trên trình duyệt.

Cấu hình máy khách không ảnh hưởng đến hiệu suất của Blazor Server app. Mô hình của Blazor Server đẩy hết các công việc nặng lên server, do đó giúp client hoạt động nhẹ nhàng hơn rất nhiều. Tất cả những công việc nặng nề nhất liên quan đến logic của ứng dụng hoặc render UI đều thực hiện trên server. Điều này càng ý nghĩa hơn khi mà app liên tục được bổ sung thêm những tính năng mới nhưng vẫn hoạt động nhẹ nhàng và tốc độ như trước.

Blazor Server chỉ là một trong các mô hình hoạt động của Blazor. Hiện tại Microsoft đang tiếp tục phát triển mô hình Blazor WebAssembly. Trong tương lai các mô hình mới của Blazor sẽ tiếp tục được bổ sung như: Blazor PWAs, Blazor Hybrid, Blazor Native.

Blazor được thiết kế nhằm đảm bảo khả năng chuyển đổi giữa các mô hình. Ví dụ, nếu bạn xuất phát với Blazor Server, bạn hoàn toàn có thể chuyển đổi sang Blazor WebAssembly hiện nay.

Về khả năng chịu tải của Blazor Server

Blazor được thiết kế để đáp ứng nhu cầu đa dạng, từ các ứng dụng quản lý nội bộ cỡ nhỏ đến các ứng dụng cỡ lớn hoạt động trên Internet.

Các thử nghiệm với hosting trên đám mây Azure cho kết quả như sau:

Standard_D1_v2 (1 vCPU, 3.5 GB memory) có thể tải trên 5000 người dùng đồng thời.

Standard_D3_V2 (4 vCPU, 14GB memory) có thể tải trên 20 000 người dùng đồng thời.

Các thử nghiệm cũng chỉ ra rằng, nút thắt lớn nhất trong việc phục vụ đồng thời nhiều người dùng nằm ở dung lượng bộ nhớ.

Như vậy, với các thử nghiệm đã tiến hành, bạn có lẽ hoàn toàn yên tâm về khả năng chịu tải của ứng dụng Blazor Server.

+ Nếu bạn thấy site hữu ích, trước khi rời đi hãy **giúp đỡ** site bằng một hành động nhỏ để site có thể phát triển và phục vụ bạn tốt hơn.
+ Nếu bạn thấy bài viết hữu ích, hãy giúp **chia sẻ** tới mọi người.
+ Nếu có thắc mắc hoặc cần trao đổi thêm, mời bạn viết trong phần **thảo luận** cuối trang.
Cảm ơn bạn!

Kết luận

Mô hình Blazor Server có những ưu điểm sau:

- Kích thước file tải về trình duyệt rất nhỏ, do đó ứng dụng tải về trình duyệt nhanh hơn.
- Ứng dụng có thể khai thác hết các khả năng của server và các tính năng của .NET Core.
- Các thiết bị hiệu suất thấp hơn (như các máy đời cũ, thiết bị di động, các trình duyệt cũ) hoạt động tốt hơn với Blazor Server.
- Toàn bộ code của ứng dụng không được cung cấp cho client.

Mô hình này cũng có những nhược điểm:

- Yêu cầu đường truyền mạng tốc độ cao và ổn định.
- Không hỗ trợ chế độ hoạt động offline (ngắt kết nối internet).
- Yêu cầu cao hơn với server nếu cần phục vụ nhiều người dùng đồng thời.
- Bắt buộc phải dùng Asp.net Core server để phục vụ.