**C HA P T E R 1**

**Internet, asynchrony, multiuser… wow!**

- Một ứng dụng kết hợp Internet, không đồng bộ và nhiều người dùng tương tác cùng một lúc luôn xứng đáng là "wow!". Tại một thời điểm nào đó, chúng ta đã vô cùng ngạc nhiên trước sự tương tác mà một số hệ thống web hiện đại có thể cung cấp, chẳng hạn như Facebook, Twitter, Gmail, Google Docs, Office Web Apps,… nơi mà chúng ta nhận được nhiều thông tin mới hầu như trong thời gian thực mà không phải tải lại trang.

- Ví dụ: khi chúng ta đang chỉnh sửa tài liệu trực tuyến bằng Office Web Apps và một người dùng khác cũng truy cập nó, chúng ta có thể thấy rằng họ đã nhập tài liệu và làm theo các thay đổi mà họ đang thực hiện. Ngay cả trong một tình huống hằng ngày hơn chẳng hạn như một cuộc trò chuyện trực tuyến đơn giản, các tin nhắn được gõ bởi bạn của chúng ta chỉ xuất hiện, như thể bằng ma thuật. Cả hai hệ thống sử dụng cùng một loại giải pháp: truyền dữ liệu không đồng bộ giữa máy chủ và máy khách trong thời gian thực.

- Thế giới chắc chắn đòi hỏi tính trực tiếp này: người dùng cần phải biết ngay những gì đang xảy ra trong môi trường của họ, tài liệu họ đang làm việc, mạng xã hội, trò chơi trực tuyến của họ và ngày càng nhiều lĩnh vực trong cuộc sống hàng ngày của họ. Thay vì phải tìm kiếm thông tin như trước đây họ đã từng làm, bây giờ họ muốn thông tin đến với họ ngay khi nó được tạo ra.

- Nhu cầu này rõ ràng ở cấp giao thức web một thời gian, bởi vì HTTP lâu dài, như được định nghĩa trong ngày, không thể đáp ứng chúng một cách hiệu quả. Trên thực tế, các tổ chức định nghĩa các tiêu chuẩn web và các giao thức, cũng như các nhà phát triển trình duyệt, đã nhận thức được điều này và đã làm việc trong nhiều năm với các cơ chế mới để truyền thông giữa máy khách và máy chủ theo hướng ngược lại với quy trình thông thường - nghĩa là, cho phép Máy chủ để chủ động trong truyền thông. Điều này đã được thực hiện thành các giao thức mới có thể được sử dụng với một mức độ tin cậy, mặc dù họ vẫn còn khá xa các giải pháp phổ quát. Sự đa dạng tuyệt vời của nền tảng khách hàng và máy chủ, Và ngay cả cơ sở hạ tầng mạng, làm cho việc áp dụng các cơ chế mới này trở nên khó khăn và chậm chạp. Sau đó, chúng ta sẽ nghiên cứu chi tiết về những khía cạnh này.

- Tuy nhiên, đây không phải là vấn đề duy nhất phải được giải quyết khi phát triển các ứng dụng đa người dùng theo thời gian thực. Truyền thông, như chúng ta biết, tạo nên các biến không ổn định và không thể đoán trước, làm cho việc quản lý và phân phối các thông điệp đến người sử dụng khá phức tạp. Ví dụ: trong ứng dụng phòng trò chuyện, chúng ta có thể có người dùng với băng thông rất khác nhau được kết nối với cùng một phòng, và những băng thông đó có thể dao động trong suốt phiên trò chuyện. Để ngăn chặn sự mất mát các tin nhắn trong kịch bản này, máy chủ có thể lưu trữ chúng tạm thời, gửi chúng tới người nhận và theo dõi những người dùng đã nhận được chúng, luôn luôn có tính đến các điều kiện giao tiếp với từng người dùng và các sự cố có thể xảy ra. Có thể xảy ra trong khi giao hàng, gửi lại dữ liệu nếu cần. Theo một nghĩa nào đó, điều này rất giống với các tính năng mà chúng ta có thể tìm thấy trong các máy chủ SMTP truyền thống, nhưng với yêu cầu bổ sung về tính cấp thiết cần thiết của các hệ thống thời gian thực. Thật dễ dàng để hình dung sự phức tạp và khó khăn liên quan đến việc thực hiện một hệ thống như mô tả.

- Cho đến gần đây, không có thành phần hoặc khuôn khổ trong lĩnh vực công nghệ .NET cung cấp bởi Microsoft có khả năng cung cấp một giải pháp hoàn chỉnh cho các vấn đề của việc áp dụng loại ứng dụng này. Chắc chắn, có rất nhiều công nghệ có khả năng cung cấp các dịch vụ kết nối và ngắt kết nối, chẳng hạn như các dịch vụ Web quen thuộc, WCF, hoặc Web API gần đây hơn. Tuy nhiên, không ai trong số chúng được thiết kế đặc biệt cho các môi trường không đồng bộ với sự cộng tác thời gian thực giữa nhiều người dùng. Trong thực tế, mặc dù nó có thể tạo ra loại hệ thống này với các loại đĩa như vậy, nhưng nó không phải là một công việc tầm thường ngay cả đối với các nhà phát triển có kinh nghiệm nhất và thường sản xuất ra những hệ thống không hiệu quả với nhiều vấn đề về hiệu năng và giới hạn khả năng mở rộng.

- Trong các trang này, chúng ta sẽ học cách thực hiện các tính năng ấn tượng của loại này bằng cách sử dụng SignalR, một khung công tác mạnh mẽ, linh hoạt và có thể mở rộng - điều này sẽ tạo điều kiện cho nhiệm vụ của chúng ta trở nên tầm thường.

- Với mục đích này, trước tiên chúng ta sẽ trình bày một bài đánh giá ngắn gọn về các vấn đề mà chúng ta tìm thấy khi phát triển các ứng dụng đa người dùng theo thời gian thực, một số trong đó chúng ta đã đề cập đến. Chúng tôi sẽ nhanh chóng xem xét các hoạt động HTTP và những hạn chế của nó để hỗ trợ các loại hệ thống, và chúng tôi sẽ giới thiệu khái niệm đẩy. Chúng tôi cũng sẽ mô tả các tiêu chuẩn hiện đang trong quá trình xác định bởi W3C và IETF, cũng như các kỹ thuật mà chúng tôi hiện có thể sử dụng để thực hiện đẩy trên HTTP. Điều này sẽ cho phép chúng ta hiểu sâu hơn về tình huống mà chúng ta đang làm việc và những thách thức xung quanh sự phát triển của các ứng dụng tự hào về tính tương phản và tính tương tác mà chúng tôi đã mô tả. Đổi lại, điều này sẽ giúp chúng ta hiểu rõ hơn về hoạt động của SignalR và cơ sở của nó như thế nào.

- Tiếp theo, tôi sẽ chính thức giới thiệu SignalR, mô tả các tính năng chính của nó, vị trí của nó trong chồng công nghệ của Microsoft để phát triển web và các mức trừu tượng khác nhau mà nó cho phép qua các giao thức cơ bản và sẽ giúp chúng tôi tách biệt với các chi tiết cấp thấp hơn Để chúng tôi chỉ có thể tập trung vào việc tạo ra các tính năng ngoạn mục cho người dùng của chúng tôi. Chúng tôi cũng sẽ có cơ hội để nói về OWIN và Katana, hai đại lý mới ngày càng trở nên nổi bật trong các công nghệ khác nhau, bao gồm cả SignalR.

- Chúng ta sẽ nghiên cứu kỹ lưỡng các kỹ thuật và kỹ thuật khác nhau được cung cấp bởi khuôn khổ này để tạo ra các ứng dụng thời gian thực tương tác đa người dùng tương tác, cả trên máy khách và máy chủ, và chúng ta sẽ học cách tận dụng sức mạnh và tính linh hoạt của chúng. Đương nhiên, chúng ta sẽ cung cấp các ví dụ mã khác nhau để giúp chúng ta hiểu cơ sở của nó một cách thực tế và do đó minh hoạ cách chúng ta có thể sử dụng khung công việc này trong các dự án thực tế.

- Chúng tôi cũng sẽ mô tả cách thức SignalR độc lập với môi trường web: mặc dù chúng dường như tạo thành môi trường tự nhiên của nó nhưng khung công tác này vượt xa chúng, cho phép cung cấp các dịch vụ thời gian thực từ bất kỳ loại ứng dụng nào, và do đó tiêu thụ của chúng từ thực tế Bất kỳ loại hệ thống nào. Chúng ta sẽ thấy một số ví dụ về điều này.

- Một khía cạnh khác rất quan trọng, mà chúng tôi sẽ dành nhiều trang, đang xem xét triển khai và khả năng mở rộng của các ứng dụng SignalR. Chúng tôi sẽ nghiên cứu các công cụ "out-of-the-box" đi kèm với nền tảng này và chỉ ra các giải pháp khả thi khác khi giải quyết các kịch bản mà những công cụ như vậy không đủ mạnh. Ngoài ra, chúng tôi sẽ xem xét các kỹ thuật khác nhau được thiết kế để theo dõi trạng thái của các máy chủ của chúng tôi và cải thiện hiệu suất của chúng trong các môi trường đồng thời cao.

- Cuối cùng, chúng ta sẽ đi sâu vào các khía cạnh tiên tiến của chương trình với SignalR, điều này sẽ cho chúng ta cái nhìn sâu sắc hơn về cách hoạt động của khuôn khổ, bao gồm bảo mật, tạo ra các thành phần tách rời bằng cách sử dụng tiêm phụ thuộc, khả năng mở rộng SignalR, kiểm tra đơn vị và các khía cạnh khác quan tâm.

- Chào mừng bạn đến với nhiều ứng dụng không đồng bộ thời gian thực đa người dùng. Chào mừng bạn đến SignalR!