**Giai đoạn kết nối**

Kết nối với SFS2X được thực hiện theo hai bước.

1. Một kết nối TCP vật lý được mở tới máy chủ.
2. Một "bắt tay" được thực hiện giữa máy khách và máy chủ trao đổi một số tham số nhất định.

Trong quá trình bắt tay, máy chủ xác minh rằng phiên bản API máy khách được hỗ trợ và gửi lại một số cài đặt mà máy khách lưu trữ cục bộ.

Đoạn mã sau được lấy từ ví dụ **Trình kết nối** có sẵn trong hầu hết các gói ví dụ cho các ngôn ngữ được hỗ trợ. Xin lưu ý rằng **[thiết lập API ứng dụng khách](http://docs2x.smartfoxserver.com/GettingStarted/client-api-setup)** là bước bắt buộc trước khi tiếp tục.

C# (Unity) JS AS3

[**?**](http://docs2x.smartfoxserver.com/DevelopmentBasics/connection-phase)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100 | **public** **class** Connector : MonoBehaviour  {  **public** Button button;  **private** SmartFox sfs;        //----------------------------------------------------------      // Unity callback methods ( Các phương thức gọi lại Unity )      //----------------------------------------------------------    **void** Start()      {          // Thêm trình nghe nhấp chuột vào nút giao diện người dùng          button.onClick.AddListener(OnButtonClick);      }    **void** Update()      {          // Vì Unity không phải là luồng an toàn nên chúng tôi xử lý các cuộc gọi lại được xếp hàng đợi trên mọi khung hình  **if** (sfs != **null**)              sfs.ProcessEvents();      }    **void** OnApplicationQuit()      {          // Luôn ngắt kết nối trước khi thoát  **if** (sfs != **null** && sfs.IsConnected)              sfs.Disconnect ();      }        //----------------------------------------------------------      // Các phương thức công khai cho giao diện người dùng      //----------------------------------------------------------    **public** **void** OnButtonClick()      {          // Tắt nút          button.interactable = **false**;            Debug.Log("Now connecting...");            // Initialize SFS2X client          sfs = **new** SmartFox();            // Add event listeners          sfs.AddEventListener(SFSEvent.CONNECTION, OnConnection);          sfs.AddEventListener(SFSEvent.CONNECTION\_LOST, OnConnectionLost);            // Set connection parameters          ConfigData cfg = **new** ConfigData();          cfg.Host = "127.0.0.1";          cfg.Port = 9933;            // Kết nối với SFS2X          sfs.Connect(cfg);      }        //----------------------------------------------------------      // SmartFoxServer event listeners      //----------------------------------------------------------    **private** **void** OnConnection(BaseEvent evt)      {  **if** ((**bool**)evt.Params["success"])          {              Debug.Log("Connection established successfully");              Debug.Log("SFS2X API version: " + sfs.Version);          }  **else**          {              Debug.LogError("Connection failed; is the server running at all?");                // Xóa trình nghe SFS2X và bật lại nút              reset();          }      }    **private** **void** OnConnectionLost(BaseEvent evt)      {          Debug.LogWarning("Connection was lost; reason is: " + (**string**)evt.Params["reason"]);            // Remove SFS2X listeners and re-enable button          reset();      }        //----------------------------------------------------------      // Private helper methods      //----------------------------------------------------------    **private** **void** reset()      {          // Remove SFS2X listeners          sfs.RemoveEventListener(SFSEvent.CONNECTION, OnConnection);          sfs.RemoveEventListener(SFSEvent.CONNECTION\_LOST, OnConnectionLost);            sfs = **null**;            // Enable button          button.interactable = **true**;      }  } |
|  |  |

Ngay khi người dùng nhấp vào nút "kết nối", đối tượng **SmartFox** sẽ được tạo: đây là lớp API chính. Sau đó, chúng tôi tiến hành bằng cách đăng ký các sự kiện máy chủ mà chúng tôi muốn lắng nghe bằng cách sử dụng **SFSEvent**lớp.

Bước tiếp theo là tạo một đối tượng cấu hình chứa các chi tiết về kết nối của chúng ta và chuyển nó tới phương thức *connect* (điều này hơi khác một chút trong JavaScript, trong đó đối tượng cấu hình được chuyển trực tiếp tới hàm tạo của lớp SmartFox ).

Cuối cùng, một trong hai trình lắng nghe kết nối được gọi bởi API, báo hiệu kết nối thành công hoặc thất bại với máy chủ. Để biết thêm thông tin và chi tiết bổ sung, vui lòng xem hướng dẫn **Trình kết nối** cho nền tảng bạn chọn ( **[Unity](http://docs2x.smartfoxserver.com/ExamplesUnity/connector)** , **[HTML5](http://docs2x.smartfoxserver.com/ExamplesJS/connector)** , **[iOS](http://docs2x.smartfoxserver.com/ExamplesIOS/connector)** , **[Android](http://docs2x.smartfoxserver.com/ExamplesAndroid/android-connector)** , **[Flash](http://docs2x.smartfoxserver.com/ExamplesFlash/connector)** ).

**» Cần làm gì sau khi kết nối được thiết lập?**

Bắt đầu kết nối với máy chủ đảm bảo rằng máy khách có thể "nói" giao thức SFS2X, nhưng tại thời điểm này, máy khách không được hệ thống công nhận là **người dùng thực** . Vẫn còn một bước bắt buộc phải thực hiện trước khi máy khách có thể bắt đầu tự do tương tác với máy chủ và những người dùng khác: đăng nhập.

Giai [**đoạn đăng nhập**](http://docs2x.smartfoxserver.com/DevelopmentBasics/login-phase) thúc đẩy khách hàng được kết nối thành người dùng thực sự và tham gia cùng anh ta vào một trong các Vùng có sẵn. Nhưng trước khi chúng ta tiếp tục với giai đoạn thứ hai này, vẫn còn một số điều về kết nối mà chúng ta cần kiểm tra.

**» Điều gì có thể đi sai?**

Trong môi trường cục bộ, rất khó có khả năng xảy ra bất kỳ vấn đề nào và bạn sẽ có thể kết nối nhanh chóng, tuy nhiên, trong một máy chủ sản xuất có "máy khách thực" kết nối qua internet, một số vấn đề có thể xuất hiện.

Thông thường , bạn có thể thấy người dùng phàn nàn về lỗi kết nối vì một trong những lý do sau.

* **cố tường lửa:** máy khách nằm sau tường lửa không cho phép kết nối trên cổng máy chủ mặc định (TCP 9933). Nếu khách hàng đang chạy tường lửa cục bộ , anh ta có thể được khuyên nên cấp quyền cho cổng SFS2X. Mặt khác , nếu máy khách đứng sau tường lửa của công ty, điều này có thể được giải quyết theo cách khác bằng cách sử dụng kết nối BlueBox (sẽ nói thêm về vấn đề này sau).
* **cố proxy:** máy chủ Proxy có thể đứng giữa máy khách và SFS2X khiến không thể thiết lập kết nối ổ cắm trực tiếp. Một lần nữa , đây là điển hình hơn của các mạng công ty và BlueBox cũng sẽ đến giải cứu ở đây.
* **vấn đề về chính sách tên miền** chéo của **Flash Player hoặc Unity Player** : các plugin dựa trên trình duyệt cũ hơn như Flash và Unity **webplayer** yêu cầu quyền đặc biệt để truy cập tài nguyên bên ngoài miền của chúng. Vì lý do này, bạn có thể cần định cấu hình tệp chính sách tên miền chéo cho phép kết nối ổ cắm với SFS2X .

Bạn có thể tìm thông tin khác về cách khắc phục lỗi kết nối máy khách **[trong hướng dẫn này](http://docs2x.smartfoxserver.com/GettingStarted/troubleshooting" \t "_blank)** .

**» Hộp xanh**

Công nghệ BlueBox cung cấp một công cụ tạo **đường hầm HTTP** hiệu suất cao cho phép người dùng sử dụng proxy và các điều kiện mạng bị hạn chế khác để tận hưởng các ứng dụng và trò chơi nhiều người chơi ở tốc độ tối đa.

Bạn có thể đọc tất cả thông tin về BlueBox trong **[tài liệu chuyên dụng này](http://docs2x.smartfoxserver.com/GettingStarted/bluebox)** .

**» Hệ thống kết nối đàn hồi cao (HRC)**

HRC là một cơ chế cấp thấp cho phép tiếp tục ngắt kết nối đột ngột theo cách hoàn toàn minh bạch cho nhà phát triển. Khi HRC được bật trong Vùng của bạn, API máy khách sẽ phát hiện các lần ngắt kết nối không mong muốn và cố gắng kết nối lại với máy chủ trong một khoảng thời gian nhất định được định cấu hình ở cấp Vùng.

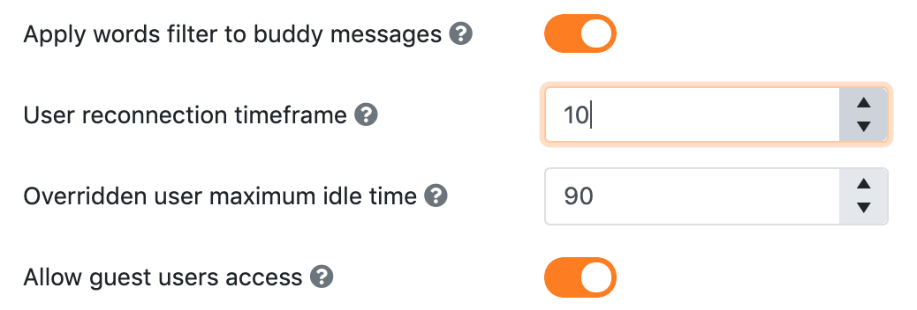
**LƯU Ý** : HRC không khả dụng cho máy khách HTML5/WebGL.

Toàn bộ quá trình diễn ra ở hậu trường và nhà phát triển được thông báo về các sự kiện khác nhau.

* **CONNECTION\_RETRY** : lệnh này được gửi đi khi API phát hiện mất kết nối đột ngột và quá trình kết nối lại với máy chủ được tiếp tục.
* **CONNECTION\_RESUME** : sự kiện này được thông báo khi kết nối lại thành công. Trong trường hợp kết nối lại không thành công , thay vào đó, sự kiện *SFSEvent.CONNECTION\_LOST* sẽ được gửi.

Về phía máy chủ, nhà phát triển tiện ích mở rộng có thể xử lý tình huống này theo cách tương tự, bằng cách lắng nghe các sự kiện *SFSEventType.USER\_RECONNECTION\_TRY* và *SFSEEventType.USER\_RECONNECTION\_SUCCESS* .

Để bật tính năng này, bạn cần định cấu hình tham số *khung thời gian kết nối lại Người dùng* trong [**tab Chung**](http://docs2x.smartfoxserver.com/GettingStarted/admintool-ZoneConfigurator-zone) của mô-đun Bộ [**cấu hình vùng từ**](http://docs2x.smartfoxserver.com/GettingStarted/admintool-ZoneConfigurator) AdminTool (được tô sáng trong hình ảnh sau).



Giá trị được biểu thị bằng số giây và bất kỳ giá trị nào lớn hơn 0 sẽ kích hoạt hệ thống. Các cài đặt phổ biến nằm trong khoảng từ 30 đến 60 giây tùy thuộc vào yêu cầu ứng dụng của bạn và mức độ nhạy cảm của logic máy chủ khi chờ trình phát bị thiếu.

Để kiểm tra chức năng Kết nối lại, bạn có thể mô phỏng việc ngắt kết nối đột ngột bằng cách gọi phương thức API phía máy khách **killConnection ( )** được hiển thị bởi lớp **SmartFox chính** .

**LƯU Ý**   
Không thử kiểm tra hệ thống kết nối lại bằng cách kéo cáp ethernet hoặc tắt kết nối WIFI. Cả hai hoạt động **sẽ không chấm dứt** các kết nối ổ cắm hiện tại của bạn và do đó, sự kiện ngắt kết nối sẽ không bao giờ được kích hoạt.

Để biết tổng quan chi tiết hơn về **hệ thống HRC+** , [**vui lòng đọc tại đây**](http://docs2x.smartfoxserver.com/GettingStarted/reconnection-hrc-plus) .

Tài liệu tham khảo :

[0] : “ http://docs2x.smartfoxserver.com/DevelopmentBasics/connection-phase ”