**Tham gia và tạo phòng**

Một trong những khối xây dựng cơ bản trong khung SFS2X là đối tượng **Phòng** . Các phòng cho phép sắp xếp người chơi để họ có thể "nhìn thấy" nhau và tương tác với nhau. Cách sử dụng điển hình của Phòng là tạo các danh mục khác nhau trong ứng dụng trò chuyện, các địa điểm gặp mặt khác nhau trong thế giới ảo hoặc các địa điểm khác nhau để thách thức những người bạn khác trong tất cả các loại trò chơi.

Các phòng được tạo theo hai cách khác nhau.

* **Statically** : thông qua mô-đun Bộ **[cấu hình vùng](http://docs2x.smartfoxserver.com/GettingStarted/admintool-ZoneConfigurator)** trong AdminTool. Đây là một cách hữu ích để tạo các Phòng liên tục trong Khu vực của bạn ( ví dụ: Sảnh) được khởi tạo ngay khi phiên bản SmartFoxServer được khởi chạy.
* **Dynamically** : các phòng có thể được tạo và hủy trong thời gian chạy từ máy khách hoặc máy chủ. Không có sự khác biệt giữa các Phòng được tạo từ bên này hay bên kia, tuy nhiên, ở phía máy chủ, có thể tinh chỉnh một số khía cạnh của Phòng mà khách hàng không thể truy cập vì lý do bảo mật.

Ví dụ: trong *Vùng cơ bản của chúng tôi* (mà chúng tôi cung cấp theo mặc định) có một Phòng tĩnh được gọi là "Sảnh chờ" mà bất kỳ khách hàng nào cũng có thể tham gia sau khi họ đã đăng nhập thành công vào Vùng.

Việc tham gia Phòng từ API ứng dụng khách yêu cầu một dòng mã, trong đó **Yêu cầu Tham gia Phòng** được gửi đến Máy chủ. Đến lượt máy chủ sẽ phản hồi bằng một trong các sự kiện sau:

* **SFSEvent.ROOM\_JOIN** , nếu hoạt động thành công
* **SFSEvent.ROOM\_JOIN\_ERROR** , nếu xảy ra lỗi

Như thường lệ, chúng ta sẽ cần đăng ký các sự kiện trong thể hiện lớp *SmartFox chính của* mình để được thông báo về kết quả của thao tác nối. Đây là mã đầy đủ cần thiết để xử lý cả hai trường hợp ( sfs là thể hiện của lớp *SmartFox* ).

C# JS AS3

[**?**](http://docs2x.smartfoxserver.com/DevelopmentBasics/join-and-create-rooms)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | ...    // Add room-related event listeners during the SmartFox instance setup  sfs.AddEventListener(SFSEvent.ROOM\_JOIN, OnRoomJoin);  sfs.AddEventListener(SFSEvent.ROOM\_JOIN\_ERROR, OnRoomJoinError);    ... |

[**?**](http://docs2x.smartfoxserver.com/DevelopmentBasics/join-and-create-rooms)

|  |  |
| --- | --- |
| 1  2  3  4 | ...    // After the successful login, send the join Room request  sfs.Send(**new** Sfs2X.Requests.JoinRoomRequest("The Lobby")); |

[**?**](http://docs2x.smartfoxserver.com/DevelopmentBasics/join-and-create-rooms)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | **private** **void** OnRoomJoin(BaseEvent evt)  {      Room room = (Sfs2X.Entities.Room)evt.Params["room"];      Console.WriteLine("Room joined: " + room.name);  }    **private** **void** OnRoomJoinError(BaseEvent evt)  {      Console.WriteLine("Room join failed: " + (**string**)evt.Params["errorMessage"]);  } |

**Điều gì có thể đi sai?**

Có một số điều có thể làm mất hiệu lực nỗ lực tham gia Phòng.

* **Phòng đầy** : mỗi Phòng có một sức chứa nhất định. Nếu đạt đến số lượng người dùng tối đa, sẽ không có thêm khách hàng nào có thể tham gia cho đến khi có sẵn một vài vị trí.
* **Phòng không tồn tại** : id Phòng được cung cấp trong yêu cầu tham gia không hợp lệ. Không tìm thấy Phòng nào có tên hoặc Mã phòng đó (một số Phòng duy nhất có thể được sử dụng thay thế cho tên).
* **Mật khẩu sai** : Các phòng có thể được đặt ở chế độ riêng tư thông qua việc sử dụng mật khẩu. Nếu bạn không biết hoặc quên mật khẩu này, bạn sẽ không thể truy cập Phòng.

**Tham gia một hoặc nhiều phòng cùng một lúc**

Trong hầu hết các trường hợp, khách hàng có thể sẽ chuyển từ Phòng này sang Phòng khác, rời khỏi Phòng trước đó sau khi tham gia Phòng mới. Đây là chế độ mặc định trong đó SFS2X và API của nó hoạt động theo mặc định.

Tuy nhiên, có những tình huống mà chúng tôi cần khách hàng tiếp tục tham gia Phòng (thường là Sảnh chờ) trong khi vào Phòng khác (có thể là trò chơi hoặc Phòng trò chuyện). Hãy kiểm tra hàm tạo **JoinRoomRequest** từ API ứng dụng khách. Các đối số sau đây có sẵn.

* **id** : id phòng có thể là tên của nó (chuỗi) hoặc id số (int).
* **mật khẩu** : (tùy chọn) mật khẩu nếu Phòng ở chế độ riêng tư.
* **roomIdToLeave** : (tùy chọn) cho biết Phòng nào sẽ được để lại khi phòng mới được nối thành công. Theo mặc định , Phòng tham gia cuối cùng được sử dụng. Tuy nhiên , nếu Người dùng đã tham gia vào một số Phòng khác, tại đây chúng tôi có thể chỉ định Phòng sẽ được để lại. Giá trị -1 sẽ cho biết rằng không nên để lại Phòng nào.
* **asSpectator** : (tùy chọn) tùy chọn này chỉ khả dụng cho Phòng trò chơi và nó cho phép người dùng tham gia với tư cách là "khán giả" trong trò chơi (sẽ nói thêm về điều này sau).

Để theo dõi các Phòng đã tham gia ở phía máy khách, API cung cấp một số công cụ hữu ích:

* thuộc tính **lastJoinedRoom** của thể hiện lớp *SmartFox* cung cấp tham chiếu đến Phòng cuối cùng đã được nối; khi giá trị này được đặt thành null , chưa có Phòng nào được tham gia;
* thuộc tính **joinRooms** của cùng một lớp sẽ cung cấp danh sách tất cả các Phòng mà khách hàng hiện đang tham gia.

Cuối cùng , đối tượng **roomManager trong lớp** *SmartFox* cho phép truy vấn dữ liệu Danh sách phòng cục bộ.

**Tạo phòng động**

Phòng có thể được tạo từ mã bất kỳ lúc nào (sau khi đăng nhập) ở cả phía máy khách và phía máy chủ. Đây là một ví dụ nhanh với cách tiếp cận thông thường: thêm người nghe trong quá trình thiết lập lớp *SmartFox* và gửi yêu cầu thích hợp ( **CreateRoomRequest** trong trường hợp này).

C# JS AS3

[**?**](http://docs2x.smartfoxserver.com/DevelopmentBasics/join-and-create-rooms)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | ...    // Add room-related event listeners during the SmartFox instance setup  sfs.AddEventListener(SFSEvent.ROOM\_ADD, OnRoomCreate);  sfs.AddEventListener(SFSEvent.ROOM\_CREATION\_ERROR, OnRoomCreateError);    ... |

[**?**](http://docs2x.smartfoxserver.com/DevelopmentBasics/join-and-create-rooms)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | // Set Room configuration  RoomSettings settings = **new** RoomSettings("Piggy's Chat Room");  settings.MaxUsers = 40;  settings.GroupId = "ChatGroup";    // Send the create Room request  sfs.Send(**new** Sfs2X.Requests.CreateRoomRequest(settings)); |

[**?**](http://docs2x.smartfoxserver.com/DevelopmentBasics/join-and-create-rooms)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | **private** **void** OnRoomCreate(BaseEvent evt)  {      Room room = (Sfs2X.Entities.Room)evt.Params["room"];      Console.WriteLine("Room added: " + room.name);  }    **private** **void** OnRoomCreateError(BaseEvent evt)  {      Console.WriteLine("Room creation failed: " + (**string**)evt.Params["errorMessage"]);  } |

Lớp **RoomSetting** cho phép chỉ định một số lượng lớn các tham số để tinh chỉnh tất cả các khía cạnh chính và phụ của Phòng. Nếu quá trình tạo thành công, sự kiện **SFSEvent.ROOM\_ADD** sẽ được gửi lại cho máy khách, nếu không, một thông báo lỗi sẽ được thông báo trong sự kiện **SFSEvent.ROOM\_CREATION\_ERROR** . Có thể có một số lý do khiến Phòng không thể được tạo.

* **Thiếu quyền của người dùng** : tùy thuộc vào cấu hình [**Trình quản lý đặc quyền của bạn,**](http://docs2x.smartfoxserver.com/AdvancedTopics/privilege-manager) một Người dùng có hồ sơ quyền nhất định có thể không được phép tạo Phòng mới trong hệ thống.
* **Tên phòng trùng lặp** : hai Phòng không thể có cùng tên trong cùng một Khu vực. Kiểm tra được áp dụng trong chế độ phân biệt chữ hoa chữ thường.
* **từ xấu trong Tên phòng** : nếu Bộ lọc từ được định cấu hình trong Vùng và được áp dụng cho các tên Phòng (xem Trình [**cấu hình vùng**](http://docs2x.smartfoxserver.com/GettingStarted/admintool-ZoneConfigurator) trong Công cụ quản trị), việc tạo có thể bị từ chối vì các từ chửi thề được phát hiện trong tên Phòng.
* **đạt đến số lượng Phòng tối đa do người dùng tạo** : vì lý do bảo mật, Vùng được định cấu hình với số lượng Phòng tối đa có thể định cấu hình mà mỗi người dùng có thể tạo cùng một lúc. Nếu khách hàng đã tạo ba Phòng (tất cả đều đang hoạt động) và giới hạn được đặt thành ba, nỗ lực tạo Phòng mới sẽ không thành công cho đến khi một trong các Phòng trước đó bị xóa.
* **đạt đến số Phòng tối đa trong Vùng** : Có thể định cấu hình một Vùng với số Phòng tối đa. Khi đạt đến giá trị này, không thể tạo thêm Phòng cho đến khi một số Phòng cũ bị xóa.

Bạn có thể tìm hiểu tất cả thông tin chi tiết về các tham số **RoomSettings** trong tài liệu API (cả máy khách và máy chủ).

**Khái niệm phòng cơ bản**

Trong khuôn khổ SFS2X, Phòng là một công cụ khá phức tạp có thể được định cấu hình ở mọi chi tiết nhỏ. Bài viết này không nhằm mục đích phân tích tất cả các khía cạnh này. Nếu bạn muốn tìm hiểu tất cả các chi tiết , chúng tôi khuyên bạn nên tham khảo [**hướng dẫn này về Kiến trúc phòng**](http://docs2x.smartfoxserver.com/DevelopmentBasics/room-architecture) .

Trong phần này, chúng tôi sẽ đề cập đến hai loại Phòng khác nhau: Phòng trò chơi và Phòng không phải trò chơi (hoặc Phòng "thông thường"). Loại thứ ba cũng tồn tại (Phòng MMO), nhưng chúng tôi sẽ bỏ qua chúng ngay bây giờ.

* **Phòng thông thường** : theo mặc định , một Phòng mới được tạo với cờ **isGame** được đặt thành false. Bạn có thể sử dụng các Phòng này cho mọi mục đích như tạo Sảnh đợi, Phòng trò chuyện, Phòng hội nghị, v.v.
* **Phòng trò chơi** : khi cờ **isGame** được đặt thành true, Phòng sẽ cung cấp một số chức năng bổ sung cần thiết cho hầu hết các trò chơi.
  + **Tự động gán ID người chơi** : mỗi người dùng được tự động gán một giá trị *id người chơi duy nhất* từ 1 đến N, cung cấp cho nhà phát triển một cách đơn giản để nhận ra từng người chơi. ID người chơi được máy chủ quản lý và chỉ định một cách minh bạch.
  + **Hỗ trợ cho người dùng không phải người chơi (khán giả)** : Phòng có thể được tạo với một số vị trí người chơi và vị trí khán giả nhất định ( ví dụ: 2 người chơi và 8 khán giả). Điều này sẽ cho phép những người dùng khác tham gia Phòng với tư cách không phải người chơi và xem trò chơi. Nếu một hoặc nhiều người chơi rời khỏi trò chơi, nhà phát triển có thể cho phép khán giả tham gia trò chơi (xem yêu cầu **SpectatorToPlayerRequest** và **PlayerToSpectatorRequest** ).

Khi tạo Phòng trò chơi, hãy luôn nhớ đặt cờ **isGame** và định cấu hình số lượng người chơi và khán giả tối đa được phép trong Phòng đó.

**Phòng trò chơi nâng cao và API trò chơi**

Nền tảng SFS2X cung cấp một bộ API máy khách và máy chủ được thiết kế đặc biệt để tạo và quản lý trò chơi, bao gồm trò chơi công khai và riêng tư, lời mời tham gia trò chơi, kết hợp Người dùng và Phòng, v.v.

Lớp **SFSGame** , được tìm thấy ở cả phía máy khách và máy chủ của API, mở rộng đối tượng **Phòng** cung cấp hàng tá chức năng mới cho phép nhà phát triển tạo các lời mời chơi trò chơi, thử thách và so sánh người chơi nâng cao trong tích tắc.

Chúng tôi thực sự khuyên bạn nên xem [**bài viết API trò chơi**](http://docs2x.smartfoxserver.com/AdvancedTopics/game-api) để biết thêm thông tin.