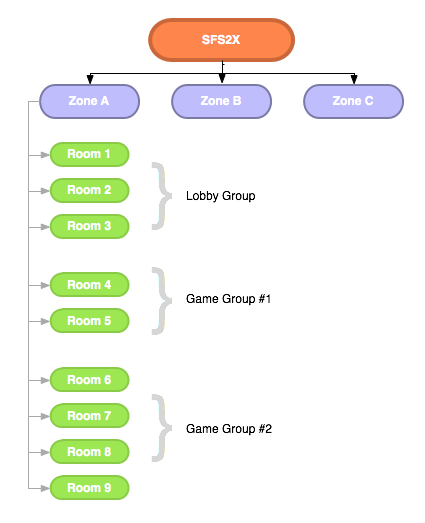
**kiến trúc phòng**

Đối **tượng Phòng** là một trong những khối xây dựng quan trọng nhất trong cấu trúc của bất kỳ ứng dụng nhiều người chơi nào. Nó chịu trách nhiệm nhóm và kết nối người dùng lại với nhau và khiến họ tương tác với nhau. Từ quan điểm của nhà phát triển, nó cung cấp các sự kiện hữu ích giúp xây dựng logic ứng dụng và mỗi Phòng có thể được quản lý bởi một Tiện ích mở rộng phía máy chủ khác nhau.

Trong chương này, chúng ta sẽ phân tích nhiều tính năng được cung cấp bởi đối tượng *Room* và các chức năng mạnh mẽ mới được cung cấp trong khung SFS2X.

**» Kiến trúc chung**

Chúng tôi đã giới thiệu trong các bài viết khác của tài liệu này một yếu tố chính trong SFS2X, **Nhóm phòng** :



Sơ đồ trên sẽ làm rõ tất cả những gì về các Nhóm phòng này. Như chúng ta có thể thấy, tất cả các Phòng vẫn được chứa trong Khu vực và chúng trông giống như ở cùng một cấp độ. Đây thực sự là cách tốt nhất để thể hiện cấu trúc chung của **Nhóm phòng** : trên thực tế, Nhóm chỉ là một ID bổ sung được áp dụng cho mỗi Phòng hoạt động như một "danh mục", do đó cho phép lọc và phân tách.

Tính năng này cho phép khách hàng giới hạn chế độ xem cục bộ của toàn bộ cấu trúc Vùng chỉ trong một hoặc nhiều nhóm, thay vì toàn bộ danh mục. Các trường hợp sử dụng là vô số: tách các Phòng trò chơi thuộc các loại khác nhau, tạo các khu vực khác nhau của ứng dụng, ẩn một số Phòng nhất định cho một số loại người dùng, v.v.

Ưu điểm của việc nhóm Phòng là chúng ta có thể giảm thiểu dữ liệu Phòng được truyền đến máy khách và giảm số lượng sự kiện cập nhật, giúp ứng dụng hoạt động hiệu quả hơn.

**" Loại phòng**

SFS2X cung cấp ba loại Phòng: **Thông thường** , **Trò chơi** và **MMO** , mỗi loại có mức độ kiểm soát rất cao. Với khả năng thiết lập chi tiết từng phút của một Phòng, các nhà phát triển có thể nhanh chóng tạo ra nhiều hành vi khác nhau, các tương tác tinh vi và cấu trúc phức tạp cho tất cả các loại trò chơi.

Trong số các tính năng của Phòng trong SFS2X, chúng tôi có khả năng đổi tên, thay đổi kích thước và khóa/mở khóa từng Phòng một cách linh hoạt khi chạy. Ngoài ra, chúng tôi có thể chọn những sự kiện mà Phòng sẽ gửi cho khách hàng và kiểm soát chính xác vòng đời của Phòng.

**Biến phòng** có sẵn để lưu trữ dữ liệu tùy chỉnh và API phía máy chủ cung cấp các tính năng nâng cao để tìm kiếm Phòng theo tiêu chí tùy chỉnh, tạo trận đấu, thử thách, lời mời, v.v.

**» Phòng trò chơi**

Phòng trò chơi là một loại Phòng chuyên biệt với các tính năng bổ sung:

* Tự động gán ID người chơi cho mỗi người chơi (đặc biệt hữu ích trong các trò chơi theo lượt)
* Hỗ trợ cho Người chơi và Khán giả, được tính riêng; khán giả không được chỉ định ID người chơi
* Theo mặc định, vòng đời của Phòng trò chơi kết thúc khi tất cả người chơi rời khỏi Phòng

**» Phòng MMO**

Phòng MMO mở rộng chức năng của Phòng thông thường bằng cách thêm **Khu vực quan tâm** (viết tắt là AoI) để xác định phạm vi sự kiện mà người dùng sẽ nhận được. Tham số **AoI** đại diện cho khu vực mà người dùng sẽ ảnh hưởng lẫn nhau, chẳng hạn như khi gửi tin nhắn công khai, cập nhật Biến người dùng, v.v. Phòng MMO đặc biệt phù hợp với thế giới ảo rộng lớn, nơi hàng nghìn người chơi đang khám phá và tương tác trên một bản đồ rộng lớn với chi phí tối thiểu chi phí mạng.

**» Giới thiệu Cài đặt phòng**

Để tạo Phòng như được mô tả trong **[bài viết này](http://docs2x.smartfoxserver.com/DevelopmentBasics/join-and-create-rooms)** , chỉ có một số tham số bắt buộc, tất cả các tham số khác có thể để nguyên giá trị mặc định nếu bạn không quan tâm đến các tùy chọn nâng cao hơn.

Hãy xem tất cả các cài đặt có sẵn, tất cả đều có trong lớp **Cài đặt** phòng (kiểm tra tài liệu API). Tất cả các cài đặt này đều khả dụng ở cả phía máy khách và máy chủ, mặc dù ở phía máy khách, một số cài đặt trong số đó bị hạn chế để ngăn mã độc hại làm hại máy chủ bằng thư rác hoặc các cuộc tấn công khó chịu khác.

**» Cài đặt cơ bản**

Sau đây là danh sách tất cả các cài đặt cơ bản.

* **name** : tên của Phòng, phải là duy nhất
* **mật khẩu** : nếu mật khẩu được chỉ định, Phòng sẽ trở thành riêng tư (tùy chọn)
* **maxUsers** : số lượng người dùng tối đa được phép trong Phòng (tùy chọn, mặc định là 20)
* **maxVariablesAllowed** : số lượng Biến phòng tối đa được phép trong Phòng (tùy chọn, mặc định là 5)
* **isGame** : nếu Phòng sẽ được sử dụng cho các trò chơi thì cờ này phải được đặt thành đúng (tùy chọn)
* **maxSpectators** : nếu đây là Phòng trò chơi, bạn nên định cấu hình số lượng khán giả được phép vào bên trong (tùy chọn, mặc định là 0)
* **groupId** : tên của Nhóm để chỉ định Phòng (tùy chọn, mặc định là "mặc định"); kiểm tra tài liệu **[kiến trúc Vùng và Phòng](http://docs2x.smartfoxserver.com/Overview/zones-room-architecture)** để biết thêm thông tin về Nhóm

**" Cài đặt nâng cao**

Sau đây là danh sách các cài đặt nâng cao hơn. Thêm chi tiết được cung cấp thêm trên.

* **autoRemoveMode** : quyết định vòng đời của Phòng, nói cách khác là khi nào Phòng sẽ bị hủy (tùy chọn)
* **roomVariables** : cung cấp danh sách các Biến phòng sẽ được thêm vào Phòng khi tạo (tùy chọn)
* **tiện ích mở rộng** : tự động đính kèm Tiện ích mở rộng phía máy chủ vào Phòng, thêm logic tùy chỉnh cho các trò chơi của bạn (tùy chọn)

**» Tinh chỉnh cài đặt**

Sau đây là danh sách một số cài đặt bổ sung. Thêm chi tiết được cung cấp thêm trên.

* **roomSettings** : cho phép chỉ định quyền và sự kiện được Phòng sử dụng (tùy chọn)
* **badWordsFilter** : định cấu hình các tùy chọn lọc từ xấu (tùy chọn)
* **customPlayerIdGeneratorClass** : cho phép cung cấp một lớp tùy chỉnh xử lý việc tạo (các) playerId cho Phòng trò chơi (tùy chọn)

**»Cài đặt nâng cao của Phòng**

**» Chế độ tự động gỡ bỏ**

chọn **AutoRemoveMode** cho phép bạn chọn 4 cách khác nhau để xử lý vòng đời của Phòng.

* **DEFAULT** : điều này được áp dụng theo mặc định nếu cài đặt không được cung cấp. Ở chế độ này, các Phòng thông thường sẽ bị xóa khi Phòng trống và người tạo Phòng đó không còn đăng nhập vào Vùng. Các phòng trò chơi sẽ bị xóa ngay khi chúng trống.
* **WHEN\_EMPTY** : Phòng bị xóa khi trống.
* **WHEN\_EMPTY\_AND\_CREATOR\_IS\_GONE** : Phòng bị xóa khi Phòng trống và người tạo Phòng đã đăng xuất.
* **KHÔNG BAO GIỜ** : Phòng không bao giờ bị xóa.

**LƯU Ý #1**   
Để tính năng tự động xóa hoạt động, cờ **isDynamic** của Phòng phải được đặt thành true.  **LƯU Ý #2**Cài đặt **AutoRemoveMode** chỉ khả dụng trong API phía máy chủ. Ở phía máy khách, chế độ MẶC ĐỊNH luôn được sử dụng.

**» Biến phòng**

**Biến phòng** là một công cụ hữu ích để lưu trữ dữ liệu tùy chỉnh trong Phòng để duy trì trạng thái ứng dụng. Các biến này được tự động đồng bộ hóa giữa tất cả người dùng trong Phòng, khiến nó trở thành một công cụ mạnh mẽ để nhanh chóng thay đổi trạng thái trò chơi, điểm số và các giá trị khác được truyền tới tất cả người chơi.

API SFS2X cũng cung cấp một số cài đặt để kiểm soát quyền truy cập, vòng đời và khả năng hiển thị của từng Biến phòng. Hãy xem chi tiết từng thuộc tính của RoomVariable.

* **name** : tên duy nhất của biến
* **isPrivate** : nếu được đặt, RV chỉ có thể được sửa đổi hoặc xóa độc quyền bởi người tạo ra nó (tùy chọn, mặc định là sai)
* **isPersistent** : theo mặc định, RV do người dùng tạo sẽ bị xóa khỏi Phòng khi người dùng rời khỏi Phòng. Nếu cờ này được bật, RV sẽ tồn tại cho đến khi người dùng đăng xuất hoặc ngắt kết nối (tùy chọn, mặc định là sai)
* **isGlobal** : theo mặc định, các bản cập nhật RV được gửi tới tất cả người dùng trong Phòng. RV chung cũng được cập nhật bên ngoài Phòng cho tất cả người dùng trong cùng một Nhóm phòng (tùy chọn, mặc định là sai)
* **isHidden** : nếu được đặt, RV sẽ chỉ khả dụng ở phía máy chủ và không bao giờ được truyền ở phía máy khách (tùy chọn, mặc định là sai)
* **value** : giá trị của RV Các loại sau đây được hỗ trợ:
  + Boolean
  + số nguyên
  + Gấp đôi
  + Chuỗi
  + SFSObject
  + SFArray
  + Null (đây không phải là một loại khóa học, nhưng nó đáng được nhắc đến vì nó được sử dụng để xóa một RV; trên thực tế, khi một RV được đặt thành null, nó sẽ bị xóa khỏi Phòng)

Về phía khách hàng, việc thiết lập Biến phòng trong Phòng hiện được tham gia rất đơn giản:

C# JS AS3

[**?**](http://docs2x.smartfoxserver.com/DevelopmentBasics/room-architecture)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | // Create a list of Room Variables  List<RoomVariable> listOfVars = **new** List<RoomVariable>();    // Add some basic values  listOfVars.Add(**new** SFSRoomVariable("bgImage", "coolBackground.jpg"));  listOfVars.Add(**new** SFSRoomVariable("stars", 4));    // Add a nested object  ISFSObject chatSettings = **new** SFSObject();  chatSettings.PutUtfStringArray("allowedFonts", **new** **string**[] {"Verdana", "Arial", "Times New Roman"});  chatSettings.PutIntArray("allowedFontSizes", **new** **int**[] {10, 11, 12, 13});  chatSettings.PutBool("isBoldAllowed", **true**);  chatSettings.PutBool("isItalicAllowed", **true**);  chatSettings.PutBool("isUnderlineAllowed", **false**);    listOfVars.Add(**new** SFSRoomVariable("settings", chatSettings));    // Send the request to set Room Variables  sfs.Send(**new** Sfs2X.Requests.SetRoomVariablesRequest(listOfVars)); |

Về phía máy chủ, có thể tạo các Biến phòng do chính phiên bản SmartFoxServer sở hữu. Điều này giống như một "người dùng ảo" không bao giờ rời khỏi hệ thống, do đó cho phép tạo các Biến phòng không bao giờ bị xóa. Để làm như vậy, bạn cần chuyển một **giá trị rỗng** với tư cách là chủ sở hữu của biến.

Đây là một ví dụ nhanh cho thấy cách đặt Biến phòng trong Tiện ích mở rộng phía máy chủ:

Java

[**?**](http://docs2x.smartfoxserver.com/DevelopmentBasics/room-architecture)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | **private** **void** setSomeVariables(Room room)  {      // Create a list of Room Variables      List<RoomVariable> listOfVars = **new** ArrayList<RoomVariable>();        // Add some basic values      listOfVars.add(**new** SFSRoomVariable("bgImage", "coolBackground.jpg"));      listOfVars.add(**new** SFSRoomVariable("stars", 4));        // Add an hidden Room Variable      RoomVariable hiddenVar = **new** SFSRoomVariable("isPremiumUsersOnly", **true**);      hiddenVar.setHidden(**true**);      listOfVars.add(hiddenVar);        // Set the Room Variables      getApi().setRoomVariables(**null**, room, listOfVars);  } |

Như bạn có thể thấy Biến phòng có tên *hiddenVar* được đánh dấu là **ẩn** và nó sẽ không được truyền tới máy khách. Mã phía máy chủ sẽ có thể kiểm tra và xem liệu Phòng chỉ *dành cho người dùng cao cấp hay không* .

**» Tiện ích mở rộng phía máy chủ**

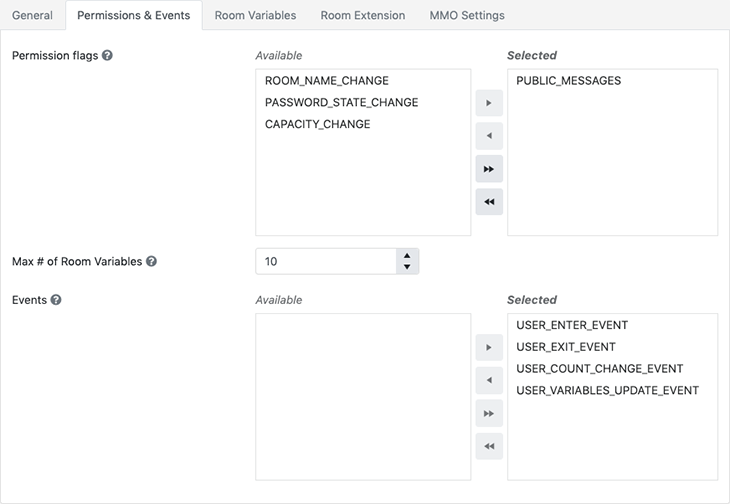
**Tiện ích mở rộng** phía máy chủ có thể được tự động cắm vào Phòng khi tạo. Điều này cho phép nhà phát triển đính kèm logic tùy chỉnh vào Phòng và kiểm soát việc sử dụng nó. Thông thường, điều này được sử dụng để chạy trò chơi, cho phép kiểm soát kiểm duyệt nâng cao, lọc tin nhắn, v.v.

Để có cuộc thảo luận đầy đủ về Tiện ích mở rộng, vui lòng đảm bảo tham khảo tài liệu Tiện ích mở rộng phía máy chủ dành cho [**Java**](http://docs2x.smartfoxserver.com/ExtensionsJava/overview) hoặc [**JavaScript**](http://docs2x.smartfoxserver.com/ExtensionsJS/quick-start) .

**» Cài đặt tinh chỉnh Phòng**

**» Quyền và Sự kiện**

Trong SmartFoxServer 2X, bạn có toàn quyền kiểm soát các sự kiện được kích hoạt bởi mỗi Phòng.   
Đây là một ví dụ từ cấu hình Phòng trong **Công cụ quản trị** :



Giao diện ở trên cho thấy cách bạn có thể tinh chỉnh từng quyền và sự kiện ở cấp Phòng khi tạo Phòng mới. Theo cấu hình trên, người dùng sẽ có thể gửi tin nhắn công khai, trong khi thay đổi kích thước Phòng (thay đổi dung lượng) và thay đổi Tên phòng và mật khẩu sẽ bị từ chối. Tất cả các sự kiện có sẵn cũng sẽ được thông báo. Trong một Phòng khác, bạn có thể muốn định cấu hình các cài đặt này theo cách khác, loại bỏ một số sự kiện và cung cấp một nhóm quyền khác.

phòng cung cấp hai loại cài đặt: **Quyền** và **Sự kiện** .

**Quyền** cho biết các thao tác liên quan đến Phòng nào có thể được thực thi trong thời gian chạy:

* **ROOM\_NAME\_CHANGE** : chuyển đổi khả năng của người tạo Phòng thay đổi tên Phòng khi chạy
* **PASSWORD\_STATE\_CHANGE** : chuyển đổi khả năng thêm, xóa hoặc sửa đổi mật khẩu Phòng của người tạo Phòng
* **PUBLIC\_MESSAGES** : chuyển đổi khả năng gửi tin nhắn công khai
* **CAPACITY\_CHANGE** : chuyển đổi khả năng của người tạo Phòng thay đổi kích thước Phòng khi chạy. Có thể "thu nhỏ" Dung lượng phòng sao cho số người dùng *tối đa* < số người dùng: trong trường hợp này sẽ không có gì xảy ra với người dùng "phụ". Ngay sau khi khách hàng rời khỏi Phòng, số lượng người dùng sẽ giảm xuống giá trị *maxUsers mới*

**Các sự kiện** cho biết những sự kiện nào mà Phòng sẽ thông báo cho các khách hàng đã tham gia:

* **USER\_ENTER\_EVENT** : bật/tắt thông báo khi có người dùng mới tham gia Phòng (mặc định là true)
* **USER\_EXIT\_EVENT** : bật/tắt thông báo khi người dùng rời khỏi Phòng (mặc định là đúng)
* **USER\_COUNT\_CHANGE\_EVENT** : bật/tắt thông báo về số lượng người dùng và khán giả cho tất cả các Phòng trong nhóm đã đăng ký (mặc định là đúng)
* **USER\_VARIABLES\_UPDATE\_EVENT** : bật/tắt thông báo cập nhật Biến người dùng (mặc định là true)

**LƯU Ý #1**   
Sự kiện *USER\_COUNT\_CHANGE\_EVENT* là một trong những thông báo ngốn nhiều băng thông nhất trong điều kiện lưu lượng truy cập lớn. SmartFoxServer cung cấp một số tối ưu hóa ở cấp độ động cơ giúp giảm đáng kể tác động của nó.

* **ưu tiên của gói** : công cụ SmartFoxServer 2X cho phép chỉ định mức độ ưu tiên cho từng gói, cho phép loại bỏ các phản hồi có mức độ ưu tiên thấp, chẳng hạn như sự kiện thay đổi số lượng người dùng nếu hàng đợi của máy khách quá bận rộn. Điều này làm giảm đáng kể sự cố tin nhắn bị rớt khi máy khách đang sử dụng kết nối chậm.
* **Điều chỉnh sự kiện theo số lượng người dùng** : để tăng thêm mức độ kiểm soát đối với sự kiện này, SmartFoxServer 2X có thể được định cấu hình để "làm chậm" tốc độ gửi các bản cập nhật này bằng cách tổng hợp các tin nhắn và chỉ gửi những tin nhắn quan trọng. Nói cách khác, bạn có thể chỉ định khoảng thời gian sau đó sự kiện sẽ được kích hoạt. Tất cả các bản cập nhật giữa hai khoảng thời gian sẽ được tổng hợp loại bỏ những bản cũ.   
  Mặc định là thời gian thực; để tiết kiệm băng thông, bạn có thể muốn giảm các cập nhật này xuống 1 lần mỗi giây hoặc ít hơn. Kiểm tra Trình [**cấu hình vùng**](http://docs2x.smartfoxserver.com/GettingStarted/admintool-ZoneConfigurator) trong Công cụ quản trị để biết thêm về điều này.

**LƯU Ý #2**   
Tắt các sự kiện *USER\_ENTER* và *USER\_EXIT* sẽ khiến Tin nhắn Công khai / Riêng tư không hoạt động bình thường. Bạn vẫn có thể giao tiếp qua tin nhắn Buddy.   
  
**LƯU Ý #3**   
Các sự kiện *USER\_ENTER* và *USER\_EXIT* phải luôn được sử dụng kết hợp. Nếu bạn quyết định tắt một trong số chúng, thì cái còn lại cũng nên được đặt theo cách tương tự vì chúng là hai mặt của cùng một thông báo.

**» Chế độ xóa**

phòng có một cách xác định *chế độ xóa của chúng* , chế độ này cho biết liệu và khi nào Phòng sẽ được tự động xóa (hủy). Các cài đặt sau có sẵn:

* **Mặc định** : *Phòng thông thường* bị xóa khi trống và người tạo ngoại tuyến; *Phòng trò chơi* sẽ bị xóa khi trống
* **When\_Empty** : Phòng bị xóa khi trống
* **When\_Empty\_And\_Creator\_Is\_Gone** : Phòng bị xóa khi trống và người tạo sẽ ngoại tuyến
* **Never\_Remove** : Phòng không bao giờ bị xóa tự động

**» Lọc từ xấu**

Nếu **Bộ lọc từ** được định cấu hình ở cấp Vùng, thì nhà phát triển có thể áp dụng bộ lọc đó cho các tin nhắn công khai được trao đổi trong Phòng.

**» Trình tạo ID người chơi**

Cuối cùng, đối với logic trò chơi rất nâng cao, các nhà phát triển có thể cung cấp một lớp **Trình tạo PlayerId thay thế** sử dụng mã tùy chỉnh để xử lý và phân phối ID người chơi trong Phòng. Lớp này được yêu cầu để triển khai lớp [**com.smartfoxserver.v2.util.IPlayerIdGenerator**](http://docs2x.smartfoxserver.com/api-docs/javadoc/server/com/smartfoxserver/v2/util/IPlayerIdGenerator.html) được cung cấp trong API phía máy chủ SFS2X.

**" Nhiêu tai nguyên hơn**

Nếu bạn muốn tìm hiểu thêm chi tiết về nhiều khía cạnh mà chúng tôi đã trình bày trong phần tổng quan này, chúng tôi đề xuất các bài viết sau:

* [**biến phòng**](http://docs2x.smartfoxserver.com/DevelopmentBasics/server-variables)
* [**API trò chơi SFS**](http://docs2x.smartfoxserver.com/AdvancedTopics/game-api)
* [**Phòng MMO**](http://docs2x.smartfoxserver.com/AdvancedTopics/mmo-rooms)