**Biến máy chủ**

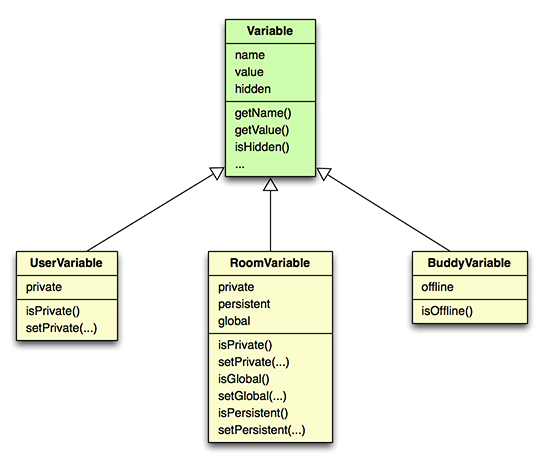
Khung SmartFoxServer 2X cung cấp các loại **Biến máy chủ khác nhau** có thể được sử dụng để duy trì trạng thái được đồng bộ hóa giữa người dùng, Phòng và bạn bè trong Danh sách bạn bè.

Nói chung, Biến máy chủ cho phép logic máy khách hoặc máy chủ tạo các giá trị tùy chỉnh được cập nhật tự động trong phạm vi của chúng. Có ba loại Biến máy chủ chính với phạm vi khác nhau:

* **Biến người dùng:** chúng đại diện cho dữ liệu tùy chỉnh được đính kèm với một người dùng. Tất cả các bản cập nhật được nhận bởi người dùng trong cùng (các) Phòng nơi chủ sở hữu được tham gia. Nói cách khác, người dùng có tên Piggy sẽ có thể truy cập và được cập nhật về tất cả các Biến của người dùng Kermit khi cả hai cùng tham gia trong cùng một Phòng. Trường hợp sử dụng phổ biến là lưu trữ hồ sơ Người dùng tùy chỉnh hoặc dữ liệu trò chơi (điểm số, thứ hạng, tuổi, sở thích, hình đại diện, thành tích trò chơi, v.v.).
* **Biến phòng:** chúng đại diện cho dữ liệu tùy chỉnh được đính kèm với đối tượng Phòng. Phạm vi của chúng là chính Phòng để người dùng có thể truy cập Biến phòng sau khi họ đã tham gia một Phòng cụ thể. Cũng có nhiều cách để hiển thị các biến này và nhận thông tin cập nhật từ bên ngoài Phòng. Trường hợp sử dụng phổ biến là giữ nguyên trạng thái và thuộc tính của trò chơi (cấp độ kỹ năng, điểm số, điểm số tốt nhất, trạng thái trò chơi hiện tại, v.v.).
* **Biến bạn bè:** chúng đại diện cho dữ liệu tùy chỉnh được đính kèm với bạn bè bên trong một hoặc nhiều Danh sách bạn bè. Phạm vi chính là Danh sách bạn bè, vì vậy bất kỳ ai đã thêm người dùng có tên Gonzo vào danh sách của họ sẽ thấy và nhận các bản cập nhật cho Biến của anh ấy. Ứng dụng hữu ích nhất là giữ cho mọi người bạn được cập nhật về trạng thái của chính họ trong thời gian thực.

**» Kiến trúc chung**

Sơ đồ sau đây minh họa một hệ thống phân cấp đơn giản hóa của các loại biến máy chủ khác nhau. Ở trên cùng, chúng ta có giao diện Biến xác định các thuộc tính cơ bản như *tên* và *giá trị* . Dưới đây, chúng tôi tìm thấy các lớp con khác nhau mở rộng các tính năng cơ bản và thêm các thuộc tính mới, chẳng hạn như *riêng tư* , *liên tục* , v.v.



Chúng ta hãy xem xét sâu hơn về từng thuộc tính:

* **tên** : có thể là bất kỳ chuỗi hợp lệ nào nhưng hãy nhớ rằng tên ngắn sẽ sử dụng ít byte hơn và do đó ít băng thông hơn. Thường thì 2-5 ký tự là đủ để xác định các biến của bạn.
* **giá trị** : tất cả các Biến máy chủ hỗ trợ một số loại cơ bản:
  + Null (chỉ dùng để xóa một biến)
  + Boolean
  + Số nguyên (32 bit đã ký)
  + Nhân đôi (nhân đôi 64 bit)
  + Chuỗi (tối đa 32768 ký tự)
  + SFSObject
  + SFArray
* **isHidden** : theo mặc định, tất cả các biến không bị ẩn. Bạn có thể bật cờ này cho Biến người dùng và biến phòng để chỉ giữ các giá trị này ở phía máy chủ. Khách hàng sẽ không bao giờ nhận được bất kỳ cập nhật nào về các biến ẩn. Cờ này chỉ khả dụng ở phía máy chủ.
* **isPrivate** : cờ này có các hiệu ứng khác nhau khi được kích hoạt trên Biến người dùng hoặc Biến phòng. Biến người dùng riêng chỉ có thể truy cập được đối với chủ sở hữu tương ứng của họ. Biến Phòng riêng hiển thị với tất cả khách hàng trong Phòng nhưng chỉ chủ sở hữu mới có thể sửa đổi.
* **isPersistent** : theo mặc định, Biến phòng do người dùng tạo sẽ bị xóa khỏi Phòng khi người dùng rời khỏi Phòng. Nếu cờ này được bật, Biến phòng sẽ tồn tại cho đến khi người dùng đăng xuất hoặc ngắt kết nối (mặc định=false).
* **isGlobal** : Biến phòng chung có thể nhìn thấy từ bên ngoài Phòng. Điều này có nghĩa là bất kỳ người dùng nào cũng có thể truy cập và nhận các bản cập nhật ngay cả khi họ không tham gia Phòng. Chạy quá nhiều Biến toàn cầu có thể sử dụng băng thông đáng kể trong máy chủ có lưu lượng truy cập cao. Sử dụng có kiểm duyệt (mặc định == false).
* **isStorable** : *(kể từ 2.14)* chỉ định thời tiết hay không, một biến có nên được lưu trữ bởi một dịch vụ lưu trữ liên tục chẳng hạn như **[Room Persistence API](http://docs2x.smartfoxserver.com/AdvancedTopics/room-persistence-api)** , cho phép lưu trữ các định nghĩa và biến của Phòng vào một tệp hoặc cơ sở dữ liệu (mặc định == true).

**Biến phòng tĩnh**   
Trong khi Biến người dùng và Biến bạn bè chỉ có thể được tạo động thông qua mã, **Biến phòng** cũng có thể được tạo tĩnh tại thời điểm cấu hình. Bạn có thể tìm hiểu thêm về điều này bằng cách tham khảo tài liệu về Bộ cấu hình Vùng của AdminTool.

**» Ví dụ về cách sử dụng**

Dưới đây, bạn sẽ tìm thấy một số ví dụ về cách sử dụng Biến phòng và người dùng từ API ứng dụng khách ( sfs là thể hiện của lớp *SmartFox* ). Biến bạn bè sẽ được xử lý riêng trong ngữ cảnh lớn nhất của **[API danh sách bạn bè](http://docs2x.smartfoxserver.com/AdvancedTopics/buddy-list-api)** .

**Biến người dùng**

Ví dụ sau đây cho thấy cách người dùng có thể lưu trữ hồ sơ của mình bằng cách sử dụng Biến và tự động cập nhật tất cả các máy khách khác trong cùng (các) Phòng:

C# JS AS3

[**?**](http://docs2x.smartfoxserver.com/DevelopmentBasics/server-variables)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | // Create a list of User Variables  List<UserVariable> listOfVars = **new** List<UserVariable>();    // Add a few User Variables  listOfVars.Add(**new** SFSUserVariable("avt", "MissPiggy.png"));  listOfVars.Add(**new** SFSUserVariable("occ", "Acting and singing"));  listOfVars.Add(**new** SFSUserVariable("loc", "Muppet's Show"));    // Send the request to set Room Variables  sfs.Send(**new** Sfs2X.Requests.SetUserVariablesRequest(listOfVars)); |

Để nhận được bản cập nhật, khách hàng phải lắng nghe sự kiện *SFSEvent.USER\_VARIABLES\_UPDATE* và xử lý nó như sau:

C# JS AS3

[**?**](http://docs2x.smartfoxserver.com/DevelopmentBasics/server-variables)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | ...    // Add variable-related event listener during the SmartFox instance setup  sfs.AddEventListener(SFSEvent.USER\_VARIABLES\_UPDATE, OnUserVarsUpdate);    ... |

[**?**](http://docs2x.smartfoxserver.com/DevelopmentBasics/server-variables)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **private** **void** OnUserVarsUpdate(BaseEvent evt)  {      // Get the user that changed his variables      User user = (User)evt.Params["user"];        // Do something with the values      Console.WriteLine("Location: " + user.GetVariable("loc").GetStringValue());      Console.WriteLine("Occupation: " + user.GetVariable("occ").GetStringValue());  } |

**biến phòng**

Sau đây là một ví dụ nhanh về cách đặt Biến phòng riêng từ phía máy khách:

C# JS AS3

[**?**](http://docs2x.smartfoxserver.com/DevelopmentBasics/server-variables)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | // Create a list of Room Variables  List<UserVariable> listOfVars = **new** List<UserVariable>();    // Add a Room Variable  listOfVars.Add(**new** SFSUserVariable("topic", "Multiplayer Game Development"));    // Send the request to set Room Variables  sfs.Send(**new** Sfs2X.Requests.SetRoomVariablesRequest(listOfVars)); |

Để nhận được bản cập nhật, khách hàng phải lắng nghe sự kiện *SFSEvent.ROOM\_VARIABLES\_UPDATE* và xử lý sự kiện như sau:

C# JS AS3

[**?**](http://docs2x.smartfoxserver.com/DevelopmentBasics/server-variables)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | ...    // Add variable-related event listener during the SmartFox instance setup  sfs.AddEventListener(SFSEvent.ROOM\_VARIABLES\_UPDATE, OnRoomVarsUpdate);    ... |

[**?**](http://docs2x.smartfoxserver.com/DevelopmentBasics/server-variables)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | **private** **void** OnRoomVarsUpdate(BaseEvent evt)  {      // Get the names of variables that were updated for the Room      List<String> changedVars = (List<String>)evt.Params["changedVars"];        // Obtain the variable and do something with it  **if** (changedVars.Contains("topic"))      {          RoomVariable topicRV = sfs.LastJoinedRoom.GetVariable("topic"));            Console.WriteLine("The room topic is now set to: " + topicRV.GetStringValue())      }  } |

**» Các biến ở phía máy chủ**

Việc sử dụng Biến ở phía máy chủ cũng đơn giản như nhau và sử dụng cùng một giao diện và logic. Ngay cả khi bạn vẫn phải học cách phát triển Tiện ích mở rộng phía máy chủ tùy chỉnh của riêng mình (xem phần **Phát triển tiện ích mở rộng** ), ở đây bạn sẽ tìm thấy một vài ví dụ cho thấy các nguyên tắc làm việc với Biến người dùng và Biến phòng trong Tiện ích mở rộng Java.

**Biến người dùng**

Hãy đặt một vài Biến người dùng. Một trong số chúng sẽ dành riêng cho phía máy chủ, sử dụng thuộc tính **ẩn** .

Java

[**?**](http://docs2x.smartfoxserver.com/DevelopmentBasics/server-variables)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | **private** **void** setUserProfile(User user, **int** databaseId)  {      // Define a public User Variable      UserVariable avatarPic = **new** SFSUserVariable("pic", "GonzoTheGreat.jpg");        // Define a private User Variable      UserVariable dbId = **new** SFSUserVariable("dbId", databaseId);      dbId.setHidden(**true**);        // Set User Variables via the server side API      getApi().setUserVariables(user, Arrays.asList(avatarPic, dbId));  } |

**biến phòng**

Bây giờ chúng ta hãy xem Biến phòng. Điều thú vị khi tạo chúng ở phía máy chủ là "người tạo" chúng sẽ là chính **máy chủ** . Cũng giống như phía khách hàng, người tạo là người dùng gửi yêu cầu.

Điều này có nghĩa là các Biến phòng do máy chủ tạo sẽ không bao giờ hết hạn, ngược lại với các Biến được tạo bằng API máy khách. Điều này có thể được sử dụng để duy trì các giá trị hoàn toàn ổn định trong bất kỳ Phòng nào yêu cầu nó.

Java

[**?**](http://docs2x.smartfoxserver.com/DevelopmentBasics/server-variables)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | **private** **void** setupRoomVariables(Room room)  {      // Define a private, global Room Variable; no one will be able to overwrite this value      // Being a global Variable, its value will be visible from outside the Room      RoomVariable chatTopic = **new** SFSRoomVariable("topic", "Multiplayer Game Development");      chatTopic.setPrivate(**true**);      chatTopic.setGlobal(**true**);        // Define an hidden Room Variable, only accessible from server side      RoomVariable isModerated = **new** SFSRoomVariable("isMod", **true**);      isModerated.setHidden(**true**);        // Set Room Variables via the server side API      // Passing null as the User parameter sets the ownership of the variables to the Server itself      getApi().setRoomVariables(**null**, room, Arrays.asList(chatTopic, isModerated));  } |

**» Học nhiều hơn nữa**

Để tìm hiểu thêm về Biến máy chủ, chúng tôi khuyên bạn nên kiểm tra API phía máy chủ và máy khách cụ thể cho nền tảng bạn chọn. Bạn tìm thấy tất cả các liên kết trên menu bên trái trong phần **Tài liệu API máy khách** và máy chủ .