**API tiện ích mở rộng**

Trong bài viết này, chúng ta sẽ xem xét các yếu tố hữu ích nhất trong **API tiện ích mở rộng** và cách khai thác khung phía máy chủ và **[Javadoc đi kèm](http://docs2x.smartfoxserver.com/api-docs/javadoc/server/" \t "_blank)** .

**» Cấp khu / Cấp phòng**

Tiện ích mở rộng máy chủ có thể được cắm vào một Vùng (chúng tôi gọi là **Tiện ích mở rộng cấp vùng** ) để quản lý toàn bộ ứng dụng hoặc vào một Phòng duy nhất ( **Tiện ích mở rộng cấp phòng** ) để chỉ quản lý Phòng cụ thể đó.

Sự khác biệt giữa hai cách tiếp cận là độc quyền trong phạm vi. **Tiện ích mở rộng cấp Vùng** có thể lắng nghe bất kỳ sự kiện nào trong Vùng và kiểm soát tất cả các Phòng và người dùng mà nó quản lý. Mặt khác **, Tiện ích mở rộng cấp Phòng** chỉ có thể lắng nghe các sự kiện trong Phòng đó và quản lý người dùng có trong đó.

Một cách sử dụng điển hình của Tiện ích mở rộng cấp Phòng là để quản lý logic của trò chơi đang chạy trong Phòng đó.

**» Làm việc với API máy chủ**

Khung SFS2X chủ yếu bao gồm ba yếu tố khác nhau.

* **Các lớp API** : chúng cung cấp một cách có tổ chức để truy cập vào nhiều chức năng của máy chủ, chẳng hạn như tạo Phòng, đặt Biến phòng/Người dùng, xử lý các thử thách trò chơi, ghép người chơi, v.v.
* **Dịch vụ** : họ quản lý các yếu tố cụ thể của máy chủ. Ví dụ: Trình quản lý phòng cung cấp quyền truy cập vào các Phòng được tạo trong Vùng, cho phép thực hiện tìm kiếm, lọc, v.v.
* **Các lớp dữ liệu** : chúng có số lượng nhiều nhất trong khung và chúng cung cấp các phần trừu tượng cho dữ liệu SmartFoxServer (ví dụ: Người dùng, Phòng, Bạn bè, Lời mời, Sự kiện, v.v.).

Bây giờ chúng ta sẽ xem xét kỹ hơn từng vấn đề và thảo luận ngắn gọn về chúng.

**» Lớp API**

Cách đơn giản nhất để truy cập API là thông qua phương thức **getApi()** trong mã Tiện ích mở rộng của bạn, phương thức này cung cấp tham chiếu đến API lõi máy chủ. Bạn có thể tìm hiểu tất cả các phương thức của nó bằng cách kiểm tra **[lớp SFSApi javadoc](http://docs2x.smartfoxserver.com/api-docs/javadoc/server/com/smartfoxserver/v2/api/SFSApi.html" \t "_blank)** .   
Dưới đây là một ví dụ nhanh về cách tạo Phòng:

Java

[**?**](http://docs2x.smartfoxserver.com/ExtensionsJava/extension-api)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | **void** makeRoom() **throws** SFSCreateRoomException  {      CreateRoomSettings settings = **new** CreateRoomSettings();      settings.setName("The music room");      settings.setMaxUsers(20);        getApi().createRoom(getParentZone(), settings, **null**);  } |

SFS2X cũng cung cấp API chuyên dụng cho các tính năng Danh sách bạn bè và Kết hợp trò chơi. Trong đoạn sau, chúng tôi giải thích cách bạn có thể truy cập chúng.

**» Lớp SmartFoxServer**

Yếu tố quan trọng tiếp theo trong khung là **[lớp SmartFoxServer](http://docs2x.smartfoxserver.com/api-docs/javadoc/com/smartfoxserver/v2/SmartFoxServer.html" \t "_blank)** cung cấp quyền truy cập vào nhiều dịch vụ đang chạy trong hệ thống. Mặc dù làm việc với các dịch vụ là một chủ đề khá nâng cao, nhưng chúng ta cũng có thể sử dụng đối tượng này để truy cập API chuyên biệt.

Đoạn mã sau cho biết cách truy cập **[API trò chơi](http://docs2x.smartfoxserver.com/AdvancedTopics/game-api)** , API **[danh sách bạn bè](http://docs2x.smartfoxserver.com/AdvancedTopics/buddy-list-api)** và API **[MMO](http://docs2x.smartfoxserver.com/AdvancedTopics/mmo-rooms)** :

Java

[**?**](http://docs2x.smartfoxserver.com/ExtensionsJava/extension-api)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | **void** getMoreApi(User owner) **throws** SFSCreateRoomException  {      ISFSGameApi gameApi = SmartFoxServer.getInstance().getAPIManager().getGameApi();      ISFSBuddyApi buddyApi = SmartFoxServer.getInstance().getAPIManager().getBuddyApi();      IMMOApi mmoApi = SmartFoxServer.getInstance().getApiManager().getMMOApi();        CreateSFSGameSettings settings = **new** CreateSFSGameSettings();      settings.setName("PongRoom");      settings.setMaxUsers(2);      settings.setMaxSpectators(8);        // ...more game settings here...        gameApi.createGame(getParentZone(), settings, owner);  } |

**» Lớp dữ liệu**

Có nhiều lớp dữ liệu khác nhau trong khung cung cấp sự trừu tượng cho các phần tử khác nhau của API: Vùng, Phòng, Người dùng, Phiên, v.v. Chúng chủ yếu được tìm thấy trong **[gói com.smartfoxserver.v2.entities.\*](http://docs2x.smartfoxserver.com/api-docs/javadoc/server/com/smartfoxserver/v2/entities/package-summary.html" \t "_blank)** .

Một khía cạnh quan trọng của sự phát triển phía máy chủ là các lớp dữ liệu phải luôn được sử dụng để đọc các thuộc tính nhưng hiếm khi để ghi chúng. Ví dụ: thực hiện các thay đổi trực tiếp đối với các giá trị của Vùng hoặc Phòng **sẽ chỉ gây ra thay đổi cục bộ chứ không có bản cập nhật nào được gửi tới máy khách** .

**LƯU Ý**   
Các lớp dữ liệu hiển thị rất nhiều getters và setters nhưng để phát triển Tiện ích mở rộng, **chỉ getters mới thực sự quan trọng** . Thay vào đó, các thay đổi phải được áp dụng thông qua API cung cấp cơ chế cập nhật cho các khách hàng có liên quan về các cập nhật trạng thái.

Một ví dụ thực tế là đặt Biến người dùng cho khách hàng. Nếu bạn tham khảo javadoc, bạn sẽ nhận thấy rằng đối tượng Người dùng hiển thị một phương thức **setVariable** mà bạn có thể muốn sử dụng. Không có gì sai với nó, nhưng thay đổi sẽ chỉ có hiệu lực ở phía máy chủ và sẽ không có bản cập nhật nào được gửi tới máy khách (điều này cần thiết để duy trì đồng bộ hóa chính xác).

Cách **thích hợp để tiếp tục** là sử dụng phương pháp API **setUserVariables** , cho phép tự động thông báo cho tất cả các máy khách yêu cầu cập nhật và thậm chí kích hoạt một sự kiện máy chủ, nếu cần.

Để tóm tắt **không bao giờ làm điều này** :

Java

[**?**](http://docs2x.smartfoxserver.com/ExtensionsJava/extension-api)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | User kermit = getParentZone().getUserByName("Kermit");    List<UserVariable> vars = **new** ArrayList<>();  vars.add(**new** SFSUserVariable("nickname", "The Frog"));    kermit.setVariables(vars); |

**Thay vào đó làm điều này** :

Java

[**?**](http://docs2x.smartfoxserver.com/ExtensionsJava/extension-api)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | User kermit = getParentZone().getUserByName("Kermit");    List<UserVariable> vars = **new** ArrayList<>();  vars.add(**new** SFSUserVariable("nickname", "The Frog"));    getApi().setUserVariables(kermit, vars); |

**» Nhiêu tai nguyên hơn**

Bây giờ bạn đã học được những kiến thức cơ bản về phát triển Tiện ích mở rộng, đã đến lúc dành cho những thứ thú vị, viết của riêng bạn và thử nghiệm với khuôn khổ.

Ngoài ra, chúng tôi thực sự khuyên bạn nên sử dụng các tài nguyên sau để tìm hiểu thêm:

* [**Công thức phía máy chủ**](http://docs2x.smartfoxserver.com/ExtensionsJava/recipes) : một danh sách dài "Hướng dẫn" và các ví dụ về mã hóa phía máy chủ.
* Tài liệu [**khái niệm tiện ích mở rộng nâng cao**](http://docs2x.smartfoxserver.com/ExtensionsJava/advanced-concepts)
* Tài liệu [**API phía máy chủ Java , đặc biệt là lớp**](http://docs2x.smartfoxserver.com/api-docs/javadoc/server/) [**SFSApi**](http://docs2x.smartfoxserver.com/api-docs/javadoc/server/com/smartfoxserver/v2/api/SFSApi.html) đại diện cho điểm vào để tương tác với máy chủ
* các trò chơi [**Tris**](http://docs2x.smartfoxserver.com/ExamplesFlash/tris) và [**SpaceWar**](http://docs2x.smartfoxserver.com/ExamplesFlash/tris) , có sẵn cho tất cả các nền tảng được hỗ trợ, bao gồm mã nguồn phía máy chủ
* các trò chơi [**BattleFarm**](http://docs2x.smartfoxserver.com/ExamplesAS3/battlefarm) và [**Space Race**](http://docs2x.smartfoxserver.com/ExamplesAS3/space-race) ActionScript 3, bao gồm cả mã nguồn phía máy khách và phía máy chủ