# Ajax Push

[1. Khái niệm 2](#_Toc293998135)

[2. Sự thay đổi của API 2](#_Toc293998136)

[3. Hướng dẫn dùng Ajax Push 3](#_Toc293998137)

[3.1. Group Management 3](#_Toc293998138)

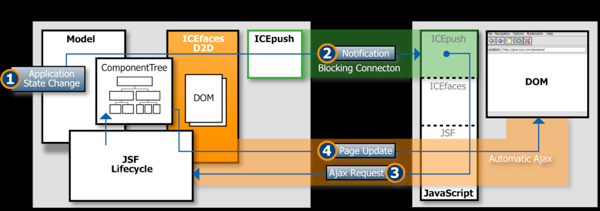
[3.2. Push Rendering 3](#_Toc293998139)

[3.3. Pushing from outside the JSF context 4](#_Toc293998140)

[4. Ví dụ: 4](#_Toc293998141)

## Khái niệm

* Ajax Push cho phép ứng dụng cập nhật bất cứ phần nào của trang web một cách hiệu quả. Đây là một sự thay đổi nhỏ về mặt kỹ thuật trong các ứng dụng Ajax nhưng nó lại là một thay đổi mang tính cách mạng của những ứng dụng cung cấp cho người dùng. Ajax Push là chìa khoá cho các ứng dụng web tương tác đa người dùng.
* Mô hình xử lý của Ajax Push:



* Khi có sự thay đổi trạng thái trên server thì server sẽ tạo ra một Ajax Push event.
* Event này được thông báo đến client thông qua ICEpush connection.
* Quá trình thông báo này sẽ tạo nên một JSF Ajax request.
* Server sẽ chuyển “Page Update” về client.

## Sự thay đổi của API

* ICEfaces 1.8 sử dung RenderManager API để thực hiện Ajax Push. RenderManager API gồm các lớp sau:

com.icesoft.faces.async.render.RenderManager

com.icesoft.faces.async.render.Renderable

com.icesoft.faces.async.render.OnDemandRenderer

com.icesoft.faces.async.render.IntervalRenderer

com.icesoft.faces.webapp.xmlhttp.PersistentFacesState

* Để đơn giản hoá việc thực hiện Ajax Push ICEfaces 2.0 sử dụng PushRenderer API để thay thế.

## Hướng dẫn dùng Ajax Push

* Để cập nhật Ajax Push cho các trang web thì ta phải đăng ký các trang đó vào một nhóm nào đó. Sau khi server thay đổi trạng thái ta sẽ gọi render cho nhóm đã đăng ký.

1. Group Management

Thêm một session với tất cả các trang liên quan vào một nhóm:

PushRenderer.addCurrentSession(String groupName);

Chỉ thêm một trang vào một nhóm:

PushRenderer.addCurrentView(String groupName);

Bạn cũng có thể xoá session hay trang ra khỏi nhóm:

PushRenderer.removeCurrentView(String groupName);

PushRenderer.removeCurrentSession(String groupName);

Chú ý rằng PushRenderer sẽ quản lý việc xoá các thành viên không còn sử dụng ra khỏi nhóm cho nên việc tự lập trình để xoá các thành viên ra khỏi nhóm không phải lúc nào cũng cần dùng.

1. Push Rendering

Khi có một sự thay đổi nào đó trong ứng dụng bạn có thể đẩy sự thay đổi đó vào một nhóm bằng code:

PushRenderer.render(groupName);

1. Pushing from outside the JSF context

Tất cả các API bên trên đều phải gọi bên trong một JSF Context. Đây là quá trình tương tác chính bởi vì cập nhật thường được thực hiện bởi một người dùng trong hệ thống do đó nó thường diễn ra bên trong một JSF Context.

Một số điểm cập nhật có thể diễn ra bên ngoài JSF Context như gọi web service, messaging services, hay gọi cơ sở dữ liệu. ICEfaces 2.0 cung cấp một phương thức để lấy PushRender từ bên ngoài JSF Context:

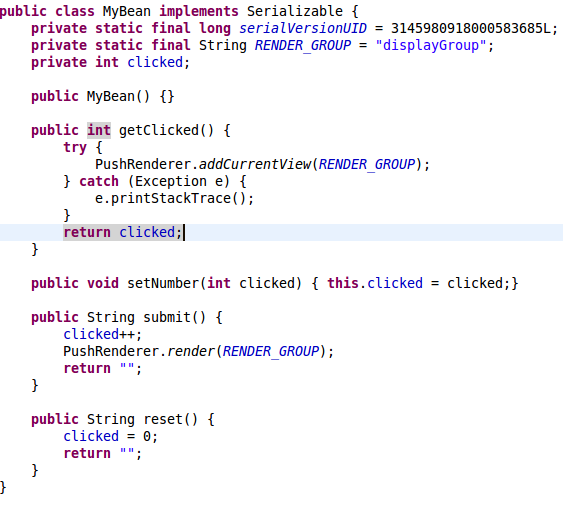
PortableRenderer PushRenderer.getPortableRenderer();

Để gọi cập nhật bên ngoài JSF Context ta dùng:

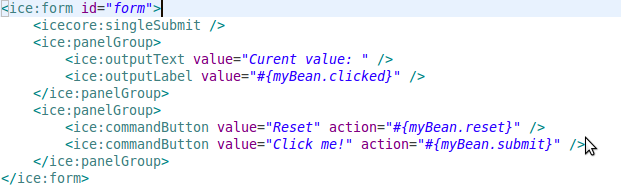
PortableRenderer.render(String groupName);

## Ví dụ:

* Viết một Web application đơn giản hiển thị số lần click chuột của người dùng.
* Lớp bean:



* Biến “clicked” sẽ lưu số lần click của người dùng
* Chúng ta đặt tên một nhóm các trang cần hiển thị thông qua Ajax Push là “displayGroup” và lưu trong biến “group”.
* Bất kỳ trang nào gọi biến “clicked” để hiển thị thì ta đăng ký trang đó vào nhóm “displayGroup”.
* Khi người dùng gọi phương thức submit để tăng số lần click lên thì ta gọi phương thức “render” để cập nhật lại các trang đã đăng ký.
* File: index.xhtml



* File “index.xhtml” sẽ hiển thị số lần click của người nhập thông qua component OutputText.
* Button “Click me!” sẽ gọi lại phương thức submit mỗi khi người dùng click.