# Sự kiện trong PhoneGap

## Hiểu thế nào là một sự kiện trong PhoneGap

Sự kiện trong PhoneGap tương đối giống sự kiện trong Javascript. Hiểu theo cách khác, sự kiện là những hành động, những thay đổi mà PhoneGap có thể nhận biết được. Ví dụ, sự kiện Document Object Model (DOM) đã được tải xong, và do đó thiết bị đang ở trạng thái sẵn sàng để PhoneGap có thể thực hiện một vài thứ được lập trình. Tóm lại, sự kiện có thể là một Click chuột, một hình ảnh đang được đọc, việc lựa chọn một trường, gửi trong biểu mẫu hay những đối tượng khác như DOM. Về đa số là các sự kiện trong Javascript đều hỗ trợ trong PhoneGap. Tuy nhiên, được định nghĩa bên trong PhoneGap thì có những sự kiện sau:

* backbutton
* deviceready
* menubutton
* pause
* resume
* searchbutton
* online
* offline

Trong các sự kiện trên thì sự kiện *deviceready* là sự kiện quan trọng nhất mà ta cần phải cân nhắc tới. Không có nó, ta sẽ không biết khi nào ứng dụng PhoneGap khởi tạo hoàn toàn. Khi nó được kích hoạt, ta có thể gọi các một cách an toàn các hàm API mà PhoneGap hỗ trợ (vì khi *deviceready* khởi tạo có nghĩa là DOM đã được đọc lên và PhoneGap API cũng đã được nạp hoàn toàn).

## Sử dụng Events Listener

Để sử dụng các sự kiện, ta phải dùng Events Listener để đăng ký theo dỏi diễn biến các thay đỗi của sự kiện đó. Ví dụ dưới đây sẻ minh họa việc sử dụng sự kiện *deviceready.*

<!DOCTYPE html>

<html>

<head>

<title>PhoneGap Device Ready Example</title>

<script type=”text/javascript” charset=”utf-8” src=”phonegap.js”></script>

<script type=”text/javascript” charset=”utf-8”>

document.addEventListener(“deviceready”, onDeviceReady, false);

function onDeviceReady() {

// call the phonegap api

}

</script>

</head>

<body>

</body>

</html>

Ví dụ trên cho thấy sự kiện *deviceready* là sự kiện được khai báo trong ứng dụng. Tuy nhiên, nó không phải là một sự kiện theo qui chuẩn của trình duyệt, mà nó được định nghĩa và kích hoạt bởi PhoneGap.

## Chi tiết thêm các loại sự kiện cơ bản

### Sự kiện deviceready

Sự kiện *deviceready* là sự kiện quan trọng nhất mà bất kỳ ứng dụng PhoneGap nào cũng cần phải khai báo và đăng ký sự kiện này đầu tiên. Bởi khi *deviceready* được kích hoạt thì DOM của ứng dụng đã được đọc lên thành công. Và ta khai báo sự kiện này như sau:

document.addEventListener(“deviceready”, onDeviceReady, false);

function onDeviceReady(){

//ready!

}

Riêng đối với các nhà phát triển trên BlackBerry OS 4.6, do RIM Browserfield không hỗ trợ nên PhoneGap đã cung cấp thêm PhoneGap.available, nhằm thay thế cho sự kiện này. Và ta dùng như sau:

function onLoad() {

var intervalID = window.setInterval(

function() {

if (PhoneGap.available) {

window.clearInterval(intervalID);

onDeviceReady();

}

},500);

}

function onDeviceReady() {

// use the phonegap api!

}

### Sự kiện backbutton

Sự kiện backbutton được kích hoạt khi người dùng bấm vào phím back (quay lại) trên Android. Để thực sự xác nhận sự kiện này, chúng ta cần phải đăng ký sự kiện như sau:

document.addEventListener(“backbutton”, onBackButton, false);

function onBackButton(){

//handle the back button

}

Lưu ý rằng, với sự kiện backbutton ta không thể đăng ký thành công nếu thành phần DOM của ứng dụng chưa được khởi động hoàn toàn. Chính vì vậy, chúng ta cần phải khai báo nó khi sự kiện *deviceready* đã được kích hoạt. Cụ thể ta làm như sau:

<!DOCTYPE html>

<html>

<head>

<title>PhoneGap backbutton Example</title>

<script type=”text/javascript” charset=”utf-8” src=”phonegap.js”></script>

<script type=”text/javascript” charset=”utf-8”>

document.addEventListener(“deviceready”, onDeviceReady, false);

function onDeviceReady() {

**// Register the event listener**

**document.addEventListener(“backbutton”, onBackButton, false);**

**}**

**// Handle the back button**

**//**

**function onBackButton() {**

**}**

</script>

</head>

<body>

</body>

</html>

### Sự kiện menubutton

Sự kiện menubutton được kích hoạt khi người dùng bấm vào phím menu trên Android. Ta đăng ký sự kiện này như sau:

document.addEventListener(“menubutton”, onMenuButton, false);

function onMenuButton(){

//handle the menu button

}

Cũng giống như các sự kiện khác, ta không nên đăng ký như thế này cho đến khi xác định được sự kiện *deviceready*.

<!DOCTYPE html>

<html>

<head>

<title>PhoneGap menubutton Example</title>

<script type=”text/javascript” charset=”utf-8” src=”phonegap.js”></script>

<script type=”text/javascript” charset=”utf-8”>

document.addEventListener(“deviceready”, onDeviceReady, false);

function onDeviceReady() {

// Register the event listener

document.addEventListener(“menubutton”, onMenuButton, false);

}

// Handle the menu button

//

function onMenuButton() {

}

</script>

</head>

<body>

</body>

</html>

### Sự kiện pause

Sự kiện pause được kích hoạt khi ứng dụng chuyển sang chế độ chạy nền. Ta khai báo:

document.addEventListener(“pause”, onPause, false);

function onPause(){

//handle the pause event

}

Sự kiện này ta cũng nên đăng ký theo dỏi khi sự kiện *deviceready* được kích hoạt:

<!DOCTYPE html>

<html>

<head>

<title>PhoneGap pause Example</title>

<script type=”text/javascript” charset=”utf-8” src=”phonegap.js”></script>

<script type=”text/javascript” charset=”utf-8”>

document.addEventListener(“deviceready”, onDeviceReady, false);

function onDeviceReady() {

// Register the event listener

document.addEventListener(“pause”, onPause, false);

}

// Handle the pause

//

function onPause() {

}

</script>

</head>

<body>

</body>

</html>

### Sự kiện resume

Sự kiện resume được kích hoạt khi một ứng dụng đang ở trạng thái chạy nền (tạm dừng) được kích hoạt trở lại làm việc. Ta đăng ký:

document.addEventListener(“resume”, onResume, false);

function onResume(){

//handle the resume event

}

Ta cần đăng ký khi sự kiện *deviceready* được kích hoạt:

<!DOCTYPE html>

<html>

<head>

<title>PhoneGap resume Example</title>

<script type=”text/javascript” charset=”utf-8” src=”phonegap.js”></script>

<script type=”text/javascript” charset=”utf-8”>

document.addEventListener(“deviceready”, onDeviceReady, false);

function onDeviceReady() {

// Register the event listener

document.addEventListener(“resume”, onResume, false);

}

// Handle the resume

//

function onResume() {

}

</script>

</head>

<body>

</body>

</html>

### Sự kiện searchbutton

Được kích hoạt khi người dùng sử dụng nút search trên thiết bị Android. Ta khai báo như sau:

document.addEventListener(“searchbutton”, onSearchButton, false);

function onSearchButton(){

//handle the search button

}

Ta cũng cần đăng ký sau sự kiện *deviceready*:

<!DOCTYPE html>

<html>

<head>

<title>PhoneGap searchbutton Example</title>

<script type=”text/javascript” charset=”utf-8” src=”phonegap.js”></script>

<script type=”text/javascript” charset=”utf-8”>

document.addEventListener(“deviceready”, onDeviceReady, false);

function onDeviceReady() {

// Register the event listener

document.addEventListener(“searchbutton”, onSearchButton, false);

}

// Handle the search button

//

function onSearchButton() {

}

</script>

</head>

<body>

</body>

</html>

### Sự kiện online

Sự kiện online được kích hoạt khi ứng dụng PhoneGap đang có kết nối tới Internet. Ta khai báo như sau:

document.addEventListener(“online”, isOnline, false);

function isOnline(){

//handle the online event

}

Ta cũng cần đăng ký sau sự kiện *deviceready*:

<!DOCTYPE html>

<html>

<head>

<title>PhoneGap online Example</title>

<script type=”text/javascript” charset=”utf-8” src=”phonegap.js”></script>

<script type=”text/javascript” charset=”utf-8”>

document.addEventListener(“deviceready”, onDeviceReady, false);

function onDeviceReady() {

// Register the event listener

document.addEventListener(“online”, isOnline, false);

}

// Handle the online event

//

function isOnline() {

}

</script>

</head>

<body>

</body>

</html>

### Sự kiện offline

Trái ngược với sự kiện online, sự kiện offline được kích hoạt khi ứng dụng chuyển sang chế độ offline (có nghĩa là không có kết nối tới Internet). Ta khai báo như sau:

document.addEventListener(“offline”, isOffline, false);

function isOffline(){

//handle the offline event

}

Ta cần khai báo sau sự kiện *deviceready*:

<!DOCTYPE html>

<html>

<head>

<title>PhoneGap offline Example</title>

<script type=”text/javascript” charset=”utf-8” src=”phonegap.js”></script>

<script type=”text/javascript” charset=”utf-8”>

document.addEventListener(“deviceready”, onDeviceReady, false);

function onDeviceReady() {

// Register the event listener

document.addEventListener(“offline”, isOffline, false);

}

// Handle the offline event

//

function isOffline() {

}

</script>

</head>

<body>

</body>

</html>

## Kết luận về event listener

Event listener là một từ khóa quan trọng. Nó dùng để xác định khi nào sự kiện *deviceready* được kích hoạt và cũng như đăng ký theo dỏi các sự kiện khác của PhoneGap một cách an toàn.