# PhoneGap APIs

## Làm việc với Device, Network, và Notifications

### Lấy thông tin từ thiết bị

PhoneGap có một *runtime object*được gọi là đối tượng *device,* để lưu giữ những thông tin hữu ích về thiết bị. Sau đây là những thuộc tính của đối tược *device*:

* device.name: tên của thiết bị (ví dụ như *my iPhone*)
* device.phonegap: phiên bản của PhoneGap
* device.platform: loại của thiết bị (ví dụ như *iPhone*)
* device.uuid: số UUID của thiết bị
* device.Version: phiên bản của hệ điều hành đang chạy

Một lưu ý quan trọng rằng, *device* được gán thành đối tượng *windows* (windows object), do vậy nó có phạm vi toàn cục. Để làm rõ, ta có khai báo hai biến khác nhau như sau và chúng đều tham chiếu tới *device*:

// Cả hai biến đều tham chiếu tới device như nhau

var myPhoneName = window.device.name;

var myPhoneName = device.name;

Trong mục này, ta sẽ học cách lấy các thông tin như: tên thiết bị, phiên bản của PhoneGap, số UUID của thiết bị và phiên bản của hệ điều hành đang dùng.

<ChuwxNTC.On-going>

## Accelerometer

### Accelerometer là gì?

Accelerometer (có nghĩa là gia tốc kế) là một API cho phép PhoneGap theo dỏi sự thay đổi các hướng của thiết bị trong không gian ba chiều (theo ba hướng X, Y và Z).

Ví dụ, với thiết bị nằm song song trên mặt bàn, mặt thiết bị hướng lên trên thì kết quả sẽ là X:0; Y:0; Z:10 và nếu mặt thiết bị hướng xuống thì X:0; Y:0; Z:-10. Khi thiết bị được dựng đứng đúng chiều X:0; Y:10; Z:0 và ngược chiều X:0; Y:-10; Z:0. Khi thiết bị nằm quay về bên trái X:10; Y:0; Z:0, bên phải X:-10; Y:0; Z:0. Như vậy, từ đó ta có thể hình dung sự thay đổi hướng X, Y và Z của thiết bị phụ thuộc vào sự thay đổi chiều của thiết bị khi xét đến như là một khối hộp 6 mặt. Nên nhớ rằng, khả năng của gia tốc kế có thể là xác định chuyển động, độ quay và gia tốc, giúp bạn trong việc tạo ra những ứng dụng sáng tạo và thú vị.

### Sử dụng đối tượng Acceleration

Đối tượng Acceleration là một đối tượng chỉ có thể đọc và nó chứa dữ liệu gia tốc kế được chụp tại một thời điểm nhất định trong một thời điểm. Nó gồm các thuộc tính:

* X – đại lượng chuyển động theo trục X.
* Y – đại lượng chuyển động theo trục Y.
* Z – đại lượng chuyển động theo trục Z.
* Timestamp – mốc thời gian lúc tạo ra, được trình bày trong mili giây.

Đối tượng này được tạo ra và cư trú trong PhoneGap, cách xác định giá trị được minh họa qua ví dụ sau:

navigator.accelerometer.getCurrentAcceleration(onSuccess, onError);

// Phương thức trên khai báo và lấy đối tượng Acceleration

// Sau đây ta sẽ lấy dữ liệu

function onSuccess(acceleration) {

alert(‘Acceleration X: ‘ + acceleration.x + ‘\n’ +

‘Acceleration Y: ‘ + acceleration.y + ‘\n’ +

‘Acceleration Z: ‘ + acceleration.z + ‘\n’ +

‘Timestamp: ‘ + acceleration.timestamp + ‘\n’);

};

function onError() {

alert(‘Sorry! Error!’);

};

### Cách sử dụng các phương thức trong Acceleration API

**Phương thức *getCurrentAcceleration***

Để lấy các giá trị của gia tốc kế X, Y và Z, ta sử dụng accelerometer .getCurrentAcceleration cụ thể như sau:

navigator.accelerometer.getCurrentAcceleration(accelerometerSuccess,

accelerometerError);

Dữ liệu sẽ trả thông qua hàm accelerometerSuccess:

function onSuccess(acceleration) {

alert(‘Acceleration X: ‘ + acceleration.x + ‘\n’ +

‘Acceleration Y: ‘ + acceleration.y + ‘\n’ +

‘Acceleration Z: ‘ + acceleration.z + ‘\n’ +

‘Timestamp: ‘ + acceleration.timestamp + ‘\n’);

};

function onError() {

alert(‘ooooops!’);

};

navigator.accelerometer.getCurrentAcceleration(onSuccess, onError);

Chú ý: trong các thiết bị iOS, phương thức này không hoạt động như ta nghĩ. Nó sẽ lấy và báo cáo giá trị lại của lần cuối mà Accelerometer API của PhoneGap gọi. Do vậy, trong iOS ta phải dùng phương thức watchAcceleration thay thế.

**Phương thức *watchAcceleration***

Phương thức watchAcceleration cho phép chúng ta lấy giá trị dữ liệu của gia tốc kế tại một khoảng thời gian cụ thể. Chúng ta sẽ thiết lập một biến chứa tần suất cập nhật và sẽ được một chu kỳ theo dỏi trong khoảng thời gian thiết lập đó. Theo dỏi ví dụ dưới đây:

function onSuccess(acceleration) {

alert(‘Acceleration X: ‘ + acceleration.x + ‘\n’ +

‘Acceleration Y: ‘ + acceleration.y + ‘\n’ +

‘Acceleration Z: ‘ + acceleration.z + ‘\n’ +

‘Timestamp: ‘ + acceleration.timestamp + ‘\n’);

};

function onError() {

alert(‘onError!’);

};

var options = { frequency: 1000 }; // Update every second

var watchID = navigator.accelerometer.watchAcceleration(onSuccess, onError, options);

Đoạn mã trên sẽ thông báo theo chu kỳ một giây.

Chú ý: trong các thiết bị iOS, gia tốc kết được quan tâm đặt biệt. Do đó, PhoneGap sẽ bị giới hạn mức đăng ký theo dỏi bởi chu kỳ từ 40 mili giây đến 1000 mili giây. Và khi bạn sử dụng một khoảng thời gian lớn hơn 1000 mili giây thì PhoneGap sử dụng trong iOS sẽ tự động cập nhật thành công trong 1000 mili giây.

**Phương thức *clearWatch***

Để dừng việc xem thay đỏi các giá trị của gia tốc kế được đăng ký ở phương thức watchAcceleration, ta dùng clearWatch như sau:

navigator.accelerometer.clearWatch(watchID);

Thông thường ta gắn đoạn mã này trong sự kiện của một nút lệnh Button hoặc cũng có thể tự động qua đoạn mã Javascript trong một khoảng thời gian chờ định sẳn.

## Camera

PhoneGap Camera API cung cấp cho một ứng dụng khả năng làm việc với một hình ảnh, khả năng chụp ảnh thông qua cameara hay nhận tệp ảnh từ kho ảnh trong thiết bị. Khi lấy về một ảnh, thì API có thể trả về một URL chỉ tới ảnh trên thiết bị hoặc có thể là một chuổi base64-encoded biểu diễn nội dung bức ảnh.

API cung cấp một phương thức đơn là navigator.camera.getPicture, dùng để lấy về một ảnh hoặc đối tượng cameraOptions, dùng để định nghĩa các tham số như ảnh nhận về như thế nào, định dạng,…

Các ứng dụng cũng có thể sử dụng PhoneGap Capture API để chụp ảnh. Tuy nhiên, PhoneGap Capture API và Camera API là khác nhau. Do vậy, cần phải đánh giá trước khi sử dụng các nào cho phù hợp với mục đích ứng dụng.

### Truy cập sử dụng ảnh

Để nhận truy xuất một ảnh, ứng dụng nên thực thi các chức năng sau:

navigator.camera.getPicture(onCameraSuccess, onCameraError, cameraOptions);

Giống như các thư viện API khác của PhoneGap, hàm getPicture đòi hỏi phải đưa vào hai hàm để thực thi khi lệnh chạy thành công hay thất bại. Hàm onCameraError gọi khi người dùng hủy bỏ lệnh hoặc có bất cứ lỗi nào xãy ra trong tiến trình. Hàm onCameraSuccess thực thi với ảnh được chụp cùng những tham số cài đặt trong hàm cameraOptions.

### Cấu hình tùy chỉnh Camera

PhoneGap Camera API cũng hỗ trợ những cài đặt cho Camera để tiến hành chụp ảnh theo ý muốn như sau:

* quality: chất lượng của ảnh.
* destinationType
* sourceType
* allowEdit: cho phép chỉnh sửa.
* encodingType: loại mã hóa tệp ảnh.
* targetWidth: kích thước chiều rộng.
* targetHeight: kích thước chiều cao.
* mediaType