Accessing Camera Device

Purpose:

This document purpose is to make the readers to understand, how to access the camera device of the mobile using PhoneGap (Appache Cordova) frame work.

Opening the Camera Gallery

**Step 1:**

Do the below configuration

App/config.xml

<feature name=”Camera”>

<param name=”android-package” value=”org.apache.cordova.camera.CameraLauncher”/>

</feature>

App/AndroidManiefest

<user-permission android:name=”android.permission.WRITE\_EXTERNAL\_STORAGE”/>

Note: The configuration will differ with respect to the Mobile OS using.

**Step 2:**

<script type="text/javascript" charset="utf-8" src="cordova.js"></script>

Load cordova javascript file into your page to access the phogap plugins

**Step 3:**

And the below code is used to access the camera photo gallery

//wait for the Phoegap to connect with device

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

Above bolded function will be called when the Phonegap device is ready to be used.

//A button will call this function to get the photo from the photo library

function getPhoto(){

//This method itself returns the camerapop

navigator.camera.getPicture(uploadPhoto, onFail, {quality: 50, detinationType:destination.FILE\_URI, sourceType: source});

}

sourceType :

1. Camera.PictureSourceType.PHOTOLIBRARY
2. Camea.PictureSourceType.SAVEDPHOTOALBUM

The return value sent to the **uploadPhoto** function. Here you can perform some action to get the photo and process it.

// Called when photo is successfully received

function uploadPhoto(imageURI){

var options = new FileUploadOptions();

options.fileKey = "file";

options.fileName = imageURI.substr(imageURI.lastIndexOf('/')+1);//to get the image name alone

options.mimeType = "image/jpeg";

pictureSource = imageURI.substr(imageURI.lastIndexOf('/')+1);

var params = new Object();

params.value1 = "test";

params.value2 = "param";

options.params = params;

options.chunkedMode = false;// If it is not set the PHP server won't able to read this image'

var ft = new FileTransfer();

ft.upload(imageURI,”filetobesent.php",win,fail,options);

}

**win** function will be called on upload success and the **fail** function will be called on the upload failiure.

Below is the html page

|  |
| --- |
| **<!DOCTYPE html>**  **<html>**  **<head>**  **<title>Capture Photo</title>**  **<meta name="viewport" content="width=device-width,height=device-height,initial-scale=1"/>**  **<script type="text/javascript" charset="utf-8" src="cordova.js"></script>**  **<script type="text/javascript" charset="utf-8">**    **var pictureSource; // picture source**  **// Wait for PhoneGap to connect with the device**  **//**  **document.addEventListener("deviceready",onDeviceReady,false);**  **// PhoneGap is ready to be used!**  **//**  **function onDeviceReady() {alert('ondeviceready');**  **}**  **// Called when a photo is successfully retrieved**  **//**  **function onPhotoDataSuccess(imageData) {**  **// Get image handle**  **//**  **var smallImage = document.getElementById('smallImage');**  **// Unhide image elements**  **//**  **smallImage.style.display = 'block';**  **// Show the captured photo**  **// The inline CSS rules are used to resize the image**  **//**  **smallImage.src = "data:image/jpeg;base64," + imageData;**  **}**    **// Called when a photo is successfully retrieved**  **//**  **function onPhotoFileSuccess(imageURI) {alert('file');**  **// Get image handle**  **console.log(JSON.stringify(imageData));**    **// Get image handle**  **//**  **var smallImage = document.getElementById('smallImage');**  **// Unhide image elements**  **//**  **smallImage.style.display = 'block';**  **// Show the captured photo**  **// The inline CSS rules are used to resize the image**  **//**  **smallImage.src = imageData;**  **}**  **// Called when a photo is successfully retrieved**  **//**  **function uploadPhoto(imageURI) {**  **var options = new FileUploadOptions();**  **options.fileKey = "file";**  **options.fileName = imageURI.substr(imageURI.lastIndexOf('/')+1);**  **options.mimeType = "image/jpeg";**    **pictureSource = imageURI.substr(imageURI.lastIndexOf('/')+1);**  **alert("Name:"+pictureSource);**  **alert(imageURI);**  **var params = new Object();**  **params.value1 = "test";**  **params.value2 = "param";**    **options.params = params;**  **options.chunkedMode = false;// If it is not set the PHP server won't able to read this image'**  **var ft = new FileTransfer();**  **ft.upload(imageURI,getBaseURL()+"?rquest=uploadImage",win,fail,options);**    **alert('completed uploading');**  **}**  **function win(r){**  **alert('success');**  **alert("Response = " + r.response);**  **alert("Sent = " + r.bytesSent);**  **var largeImage = document.getElementById('largeImage');**  **// Unhide image elements**  **//**  **largeImage.style.display = 'block';**  **// Show the captured photo**  **// The inline CSS rules are used to resize the image**  **//**  **largeImage.src = getSiteURL()+"/rest/uploads/images/"+pictureSource;**    **}**  **function fail(error){**  **alert('Failed');**  **}**  **// A button will call this function**  **//**  **function capturePhotoWithData() {**  **// Take picture using device camera and retrieve image as base64-encoded string**  **navigator.camera.getPicture(onPhotoDataSuccess, onFail, { quality: 50 });**  **}**  **function capturePhotoWithFile() {**  **navigator.camera.getPicture(onPhotoFileSuccess, onFail, { quality: 50, destinationType: Camera.DestinationType.FILE\_URI });**  **}**    **// A button will call this function**  **//**  **function getPhoto(source) {alert('getPhoto');**  **// Retrieve image file location from specified source**  **navigator.camera.getPicture(uploadPhoto, onFail, { quality: 50,**  **destinationType: destinationType.FILE\_URI,**  **sourceType: source });**  **}**  **// Called if something bad happens.**  **//**  **function onFail(message) {**  **alert('Failed because: ' + message);**  **}**  **</script>**  **</head>**  **<body>**  **<button onclick="capturePhotoWithData();">Capture Photo With Image Data</button> <br>**  **<button onclick="capturePhotoWithFile();">Capture Photo With Image File URI</button> <br>**  **<button onclick="getPhoto(pictureSource.PHOTOLIBRARY);">From Photo Library</button><br>**  **<button onclick="getPhoto(pictureSource.SAVEDPHOTOALBUM);">From Photo Album</button><br>**  **<img style="display:none;width:60px;height:60px;" id="smallImage" src="" />**  **<img style="display:none;" id="largeImage" src="" />**  **</body>**  **</html>** |

Recording Audio and Uploading to the remote server

**Step 1:**

Do the below configuration

App/config.xml

<feature name="Media">

<param name="android-package" value="org.apache.cordova.media.AudioHandler" />

</feature>

<feature name="http://api.phonegap.com/1.0/file"/>

App/AndroidManiefest

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.RECORD\_AUDIO" />

<uses-permission android:name="android.permission.MODIFY\_AUDIO\_SETTINGS" />

Note: The configuration will differ with respect to the Mobile OS using.

**Step 2:**

<script type="text/javascript" charset="utf-8" src="cordova.js"></script>

Load cordova javascript file into your page to access the phogap plugins

**Step 3:**

And the below code is used to record the audio.

|  |
| --- |
| var src;  var mediaRec;  //wait for the Phoegap to connect with device  document.addEventListener(“deviceready”,**onDeviceReady**,false);  Above bolded function will be called when the Phonegap device is ready to be used.  // device APIs are available  function onDeviceReady() {  //recordAudio();  src = "file:///storage/emulated/0/"+userId+"\_"+"myrecording.amr";  mediaRec = new Media(src, onSuccess, onError);  }  // onError Callback  function onError(error) {  alert('code: ' + error.code + '\n' +  'message: ' + error.message + '\n');  }  // Record audio  function recordAudio() {  // Start Recording audio  mediaRec.startRecord();  }  //Stop Recording  // A button will call this function  function stopRecording(){  mediaRec.stopRecord(); alert('record stopped');  var options = new FileUploadOptions();  var params = new Object();  params.value1 = "test";  params.value2 = "param";    options.params = params;  options.chunkedMode = false;// If it is not set the PHP server won't able to read this image'  var ft = new FileTransfer();  ft.upload(src,getBaseURL()+"?rquest=uploadAudio",win,fail,options);  }  //Call back function on file upload success  function win(r){  alert('success');  alert("Response = " + r.response);  alert("Sent = " + r.bytesSent);  }  //Call back function on file upload failed  function fail(error){  alert('Failed'+error);  }  <button value="Record Audio" onclick="recordAudio()">Record Audio</button>  <button value="Record Audio" onclick="stopRecording()">Stop Recording</button> |

**Recording Video & Uploading to the Remote Server**

**Step 1:**

Do the below configuration

App/config.xml

<feature name="File">

<param name="android-package" value="org.apache.cordova.file.FileUtils" />

</feature>

<feature name="Capture">

<param name="android-package" value="org.apache.cordova.mediacapture.Capture" />

</feature>

App/AndroidManiefest.xml

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

<uses-permission android:name="android.permission.RECORD\_AUDIO" />

<uses-permission android:name="android.permission. RECORD\_AUDIO " />

Note: The configuration will differ with respect to the Mobile OS using.

**Step 2:**

<script type="text/javascript" charset="utf-8" src="cordova.js"></script>

Load cordova javascript file into your page to access the phogap plugins

**Step 3:**

And the below code is used to record the video and upload to the remote server.

|  |
| --- |
| // Wait for PhoneGap to connect with the device  document.addEventListener("deviceready",onDeviceReady,false);  function onDeviceReady(){}  // A button will call this function  function captureVideo() {  // Launch device video recording application,  // allowing user to capture up to 2 video clips  navigator.device.capture.captureVideo(captureSuccess, captureError, {limit: 1});  }  // Upload files to server  function uploadFile(mediaFile) {  var ft = new FileTransfer(),  path = mediaFile.fullPath,  name = mediaFile.name;  ft.upload(path,getBaseURL()+"?rquest=uploadVideo",  function(result) {  alert('Upload success: ' + result.responseCode);  alert(result.bytesSent + ' bytes sent');  },  function(error) {  alert('Error uploading file ' + path + ': ' + error.code);  },  { fileName: name });  }  // Called when capture operation is finished  function captureSuccess(mediaFiles) {  uploadFile(mediaFiles[0]); // To access the first file which is recorded  }  // Called if the video is not recorded.  function captureError(error) {  var msg = 'An error occurred during capture: ' + error.code;  navigator.notification.alert(msg, null, 'Uh oh!');  }  <button value="Record Video" onclick="captureVideo()">Record Video</button> |