Devdactic

The Complete Ionic Images Guide (Capture, Store & Upload)

NOVEMBER 24, 2016 BY SIMON (HTTPS://DEVDACTIC.COM/AUTHOR/SIMON-REIMLER/)

In this guide to Ionic images we will learn to capture images from library & camera, copy files on your device and build a simple PHP upload to a server.

This Tutorial was updated for Ionic 3.2!

In this article we will build a simple app to capture images (camera/library), store it in our apps folder and finally build a simple PHP server to upload our captured images. You don't need PHP knowledge, this is just an example for a backend and a very simple one as well.

example for a backend and a very simple one as well.

Prerequisite

Learning Ionic can become overwhelming, I know that feeling. Is learning from tutorials and videos sometimes *not enough for you*? Then I got something for you.

If you want to learn Ionic with step-by-step video **courses**, hands-on training **projects** and a helpful **community** who has your back, then take a look at the Ionic Academy.

JOIN THE IONIC ACADEMY (HTTPS://IONICACADEMY.COM/?UTM_SOURCE=DEVDACTIC_POST_INTRO&UTM_CONTENT=TEXT)

Starting our Ionic Images App

First of all we create a new blank Ionic app. Additional we install a bunch of Cordova plugins, so go ahead and run:

Start our image upload app ZSH

```
ionic start devdactic-imageupload-2 blank
cd devdactic-imageupload-2
npm install --save @ionic-native/camera @ionic-native/file @ionic-native/file-p
ionic plugin add cordova-plugin-camera --save
ionic plugin add cordova-plugin-file --save
ionic plugin add cordova-plugin-file-transfer --save
ionic plugin add cordova-plugin-filepath --save
```

In detail these plugins are used for:

- **cordova-plugin-camera**: Needed for using the Ionic native camera (http://ionicframework.com/docs/v2/native/camera/)
- **cordova-plugin-file**: Needed for copying the result of the camera into our filesystem (http://ionicframework.com/docs/v2/native/file/)
- **cordova-plugin-file-transfer**: Needed for uploading files (http://ionicframework.com/docs/v2/native/transfer/)

(http://ionicframework.com/docs/v2/native/transfer/)

- **cordova-plugin-filepath**: Utility for fixing some Android paths to files (http://ionicframework.com/docs/v2/native/filepath/)

To load all of our plugins we have to add them to our **src/app/app.module.ts**, so go ahead and change it to:

Import plugins to providers array JavaScript

```javascript
import { BrowserModule } from '@angular/platform-browser';
import { ErrorHandler, NgModule } from '@angular/core';
import { IonicApp, IonicErrorHandler, IonicModule } from 'ionic-angular';
import { SplashScreen } from '@ionic-native/splash-screen';
import { StatusBar } from '@ionic-native/status-bar';

import { MyApp } from './app.component';
import { HomePage } from '../pages/home/home';

import { File } from '@ionic-native/file';
import { Transfer } from '@ionic-native/transfer';
import { FilePath } from '@ionic-native/file-path';
import { Camera } from '@ionic-native/camera';

@NgModule({
  declarations: [
    MyApp,
    HomePage
  ],
  imports: [
    BrowserModule,
    IonicModule.forRoot(MyApp)
  ],
  bootstrap: [IonicApp],
  entryComponents: [
    MyApp,
    HomePage
  ],
  providers: [
    StatusBar,
    SplashScreen,
    File,
    Transfer,
    Camera,
    FilePath,
    {provide: ErrorHandler, useClass: IonicErrorHandler}
  ]
})
export class AppModule {}
```

Now we are ready to use them all with the latest version of Ionic Native!

Building the Ionic images capturing class

To start the capturing process we need to distinguish between camera and library, therefore we will display the user a simple action sheet with those options. Also, we already import all the stuff we need later so don’t worry if they are marked as unused at this point.

Make sure to add all your plugins to the constructor as well, as this is needed since Ionic Native 3!

We work with the already generated home page, so open the

We work with the already generated home page, so open the **src/pages/home/home.ts** and replace everything inside with:

The base class for our image app JavaScript

```
import { Component } from '@angular/core';
import { NavController, ActionSheetController, ToastController, Platform, Load

import { File } from '@ionic-native/file';
import { Transfer, TransferObject } from '@ionic-native/transfer';
import { FilePath } from '@ionic-native/file-path';
import { FilePath } from '@ionic-native/file-path';
import { Camera } from '@ionic-native/camera';

declare var cordova: any;

@Component({
  selector: 'page-home',
  templateUrl: 'home.html'
})
export class HomePage {
  lastImage: string = null;
  loading: Loading;

  constructor(public navCtrl: NavController, private camera: Camera, private t

  public presentActionSheet() {
    let actionSheet = this.actionSheetCtrl.create({
      title: 'Select Image Source',
      buttons: [
        {
          text: 'Load from Library',
          handler: () => {
            this.takePicture(this.camera.PictureSourceType.PHOTOLIBRARY);
          }
        },
        {
          text: 'Use Camera',
          handler: () => {
            this.takePicture(this.camera.PictureSourceType.CAMERA);
          }
        },
        {
          text: 'Cancel',
          role: 'cancel'
        }
      ]
    });
    actionSheet.present();
  }
}
```

The user can select between these 2 options inside the action sheet, and for each we can pass the specific **sourceType** to our `takePicture` function which is not yet implemented. Let's change that, and prepare yourself as this is going to be a bit tricky.

In general we want to call the native camera dialog with some options, and get a path to the image back inside the then block. Once we got the image path we want to copy that file into our own apps directory, just because we might want to store it and make sure the user won't delete it.

might want to store it and make sure the user won't delete it.

As this is not really working on the combination Android + library we need an additional function that resolves that special image path. But after we got the image right, we do the same stuff as in the regular block:

1. **currentName**: Grab the current name of the image from the path
2. **correctPath**: Get only the path to the image without the name
3. **copyFileToLocalDir**: Copy from the current path to our app and use new name from `createFileName`

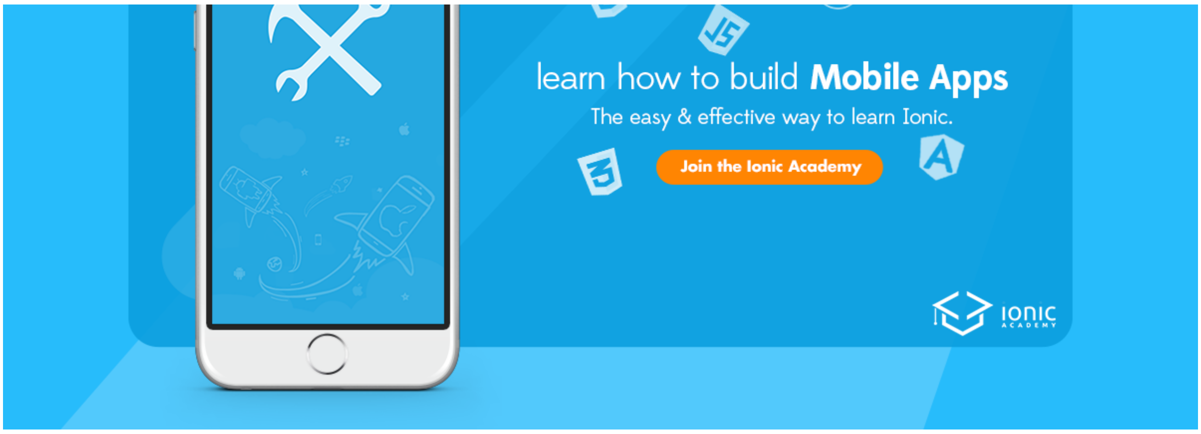
All of that is inside the function below, so add that to your **HomePage** now:

The function to get the image data JavaScript

```javascript
public takePicture(sourceType) {
  // Create options for the Camera Dialog
  var options = {
    quality: 100,
    sourceType: sourceType,
    saveToPhotoAlbum: false,
    saveToPhotoAlbum: false,
    correctOrientation: true
  };

  // Get the data of an image
  this.camera.getPicture(options).then((imagePath) => {
    // Special handling for Android library
    if (this.platform.is('android') && sourceType === this.camera.PictureSource
      this.filePath.resolveNativePath(imagePath)
        .then(filePath => {
          let correctPath = filePath.substr(0, filePath.lastIndexOf('/') + 1);
          let currentName = imagePath.substring(imagePath.lastIndexOf('/') + 1
          this.copyFileToLocalDir(correctPath, currentName, this.createFileName
        });
    } else {
      var currentName = imagePath.substr(imagePath.lastIndexOf('/') + 1);
      var correctPath = imagePath.substr(0, imagePath.lastIndexOf('/') + 1);
      this.copyFileToLocalDir(correctPath, currentName, this.createFileName())
    }
  }, (err) => {
    this.presentToast('Error while selecting image.');
  });
}
```

After that hard stuff we need some more helper functions we already used inside our previous functions. The 3 functions below are used to create a new image name simply from the current timestamp, to copy a file from one location to our app, to present a message for errors and success and finaly to resolve the path of an image to our current apps folder.

The path to our app can change, so whenever you plan to store these images somehow make sure to **save the name of the image and not the complete path** as this can change from time to time!


learn how to build Mobile Apps
The easy & effective way to learn Ionic.
Join the Ionic Academy
IONIC ACADEMY

Anyway, for now just add these functions to our **HomePage**:

Helper methods for our image capturing process JavaScript

```javascript
// Create a new name for the image
private createFileName() {
  var d = new Date(),
  n = d.getTime(),
  newFileName =  n + ".jpg";
  return newFileName;
}

// Copy the image to a local folder
private copyFileToLocalDir(namePath, currentName, newFileName) {
  this.file.copyFile(namePath, currentName, cordova.file.dataDirectory, newFile
    this.lastImage = newFileName;
  }, error => {
    this.presentToast('Error while storing file.');
  });
}

private presentToast(text) {
  let toast = this.toastCtrl.create({
    message: text,
    duration: 3000,
    position: 'top'
  });
  toast.present();
}

// Always get the accurate path to your apps folder
public pathForImage(img) {
  if (img === null) {
    return '';
  } else {
    return cordova.file.dataDirectory + img;
  }
}
```

We are now done with the process of capturing and storing images. Actually it wasn't that long and quite easy to achieve using the Ionic native wrapper. Let's continue with the last missing function for our images app before we hook up everything inside the view.

Building the Ionic image upload function

After we have captured the image and moved it into our apps folder, we want to give the user to upload this image. Upload to a server can look very different depending on the endpoint and architecture.

For this example we will make a simple upload to a PHP backend as this is quite usual and lots of you have requests to see it with PHP. If you want to see something different (like POST to a REST API) please let me know below in the comments!

below in the comments!

To upload our file we use the **FileTransfer** plugin we initially installed. We have to create an object of different options which need to be passed to the transfer, you can experiment with them if you need different values or more information that needs to reach the server.

We are also showing a progress indicator as it can sometimes take a few second and it’s a good idea to inform your user about the stuff going on (unless you want to perform the upload hidden in the background).

Now append our last function to the **HomePage**:

Upload the image to a remote server JavaScript

```javascript
public uploadImage() {
  // Destination URL
  var url = "http://yoururl/upload.php (http://yoururl/upload.php)";

  // File for Upload
  var targetPath = this.pathForImage(this.lastImage);

  // File name only
  var filename = this.lastImage;

  var options = {
    fileKey: "file",
    fileName: filename,
    chunkedMode: false,
    mimeType: "multipart/form-data",
    params : {'fileName': filename}
  };

  const fileTransfer: TransferObject = this.transfer.create();

  this.loading = this.loadingCtrl.create({
    content: 'Uploading...',
  });
  this.loading.present();

  // Use the FileTransfer to upload the image
  fileTransfer.upload(targetPath, url, options).then(data => {
    this.loading.dismissAll()
    this.presentToast('Image succesful uploaded.');
  }, err => {
    this.loading.dismissAll()
    this.presentToast('Error while uploading file.');
  });
}
```

Right now we are done with the logic but can’t actually use it because the view is still missing, so let’s craft something really simple.

Building the view for our app

Actually we only need an image where we can display the currently captured image and two buttons for capturing and uploading images. We are not storing all of the images, but you can very easily store the names as you can always simply add the rest of the path. This is even recommend as **your apps path might change**!

Additional we use the `[hidden]` or `[disabled]` attribute on some of our elements to hide/disable those elements if we don’t have an image selected yet. Now go ahead and put everything below into your **src/pages/home/home.html**:

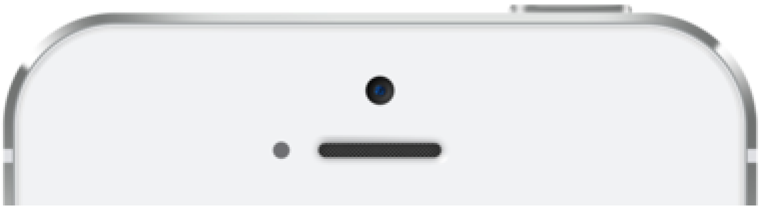
src/pages/home/home.html:

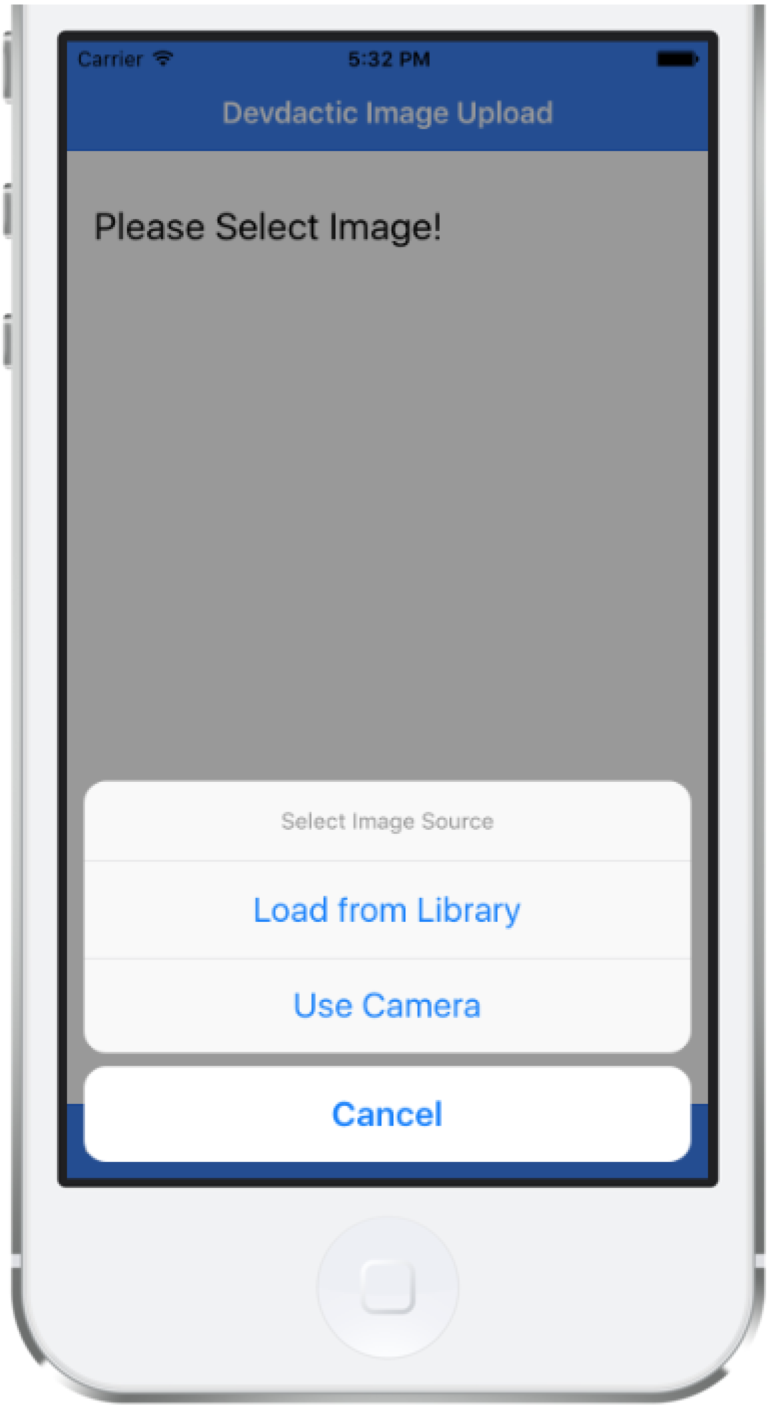
The view for our Ionic image app XHTML

```
<ion-header>
  <ion-navbar color="primary">
    <ion-title>
      Devdactic Image Upload
    </ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>
  <img src="{{pathForImage(lastImage)}}" style="width: 100%" [hidden]="lastImag
  <h3 [hidden]="lastImage !== null">Please Select Image!</h3>
</ion-content>

<ion-footer>
  <ion-toolbar color="primary">
    <ion-buttons>
      <button ion-button icon-left (click)="presentActionSheet()">
        <ion-icon name="camera"></ion-icon>Select Image
      </button>
      <button ion-button icon-left (click)="uploadImage()" [disabled]="lastImag
        <ion-icon name="cloud-upload"></ion-icon>Upload
      </button>
    </ion-buttons>
  </ion-toolbar>
</ion-footer>
```

Your app is now ready to run! But keep in mind: **We are using Cordova plugins, so this won't work inside the browser!**

To test the app you need to run it on a device and use those functions (or use the simulator and chose the library for images). Also, our backend is not yet existing so the upload won't work at this point. Anyway, you app should look like below at this point.





Creating our simple image upload PHP server

As this is the same as the last tutorial about Ionic images

As this is the same as the last tutorial about Ionic images (https://devdactic.com/ionic-image-upload-php/) I'm not really making any changes to the code.

I am not using PHP frequently, but many of you asked for a PHP solution and I found it quite easy to setup this littler server to accept files. I am mostly more a fan of Firebase solutions as a Backend (https://devdactic.com/ionic-2-firebase/), but for this example PHP works perfectly.

If you have a server you can use that one, otherwise I simply recommend to download XAMPP (https://www.apachefriends.org/de/index.html) and install it local.

I'm not going to cover that process since this is about Ionic image upload and not how to configure PHP. If you have set it up, you can first of all create a **upload.php** to accept uploads:

A simple PHP server that accepts files PHP

```php
<?php
header('Access-Control-Allow-Origin: *');
$target_path = "uploads/";

$target_path = $target_path . basename( $_FILES['file']['name']);

if (move_uploaded_file($_FILES['file']['tmp_name'], $target_path)) {
    echo "Upload and move success";
} else {
echo $target_path;
    echo "There was an error uploading the file, please try again!";
}
?>
```

Also, make sure to create a **uploads** folder next to this file, as it will copy the images into that folder.

Additionally, to see the results of our hard work, I created a little HTML file that will scan the uploads folder and show them so we can directly see if our upload worked, Create this as **index.php** next to the previous file and insert:

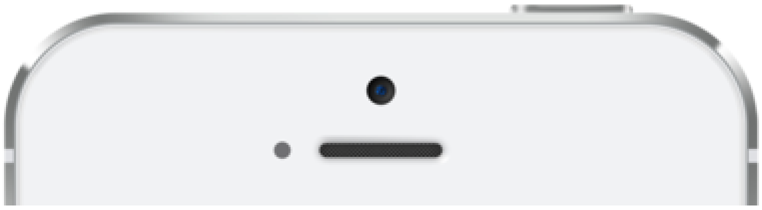
insert:

Simple HTML overview page to see our images XHTML

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalab
  <title>Devdactic Image Upload</title>
</head>
<body>
<h1>Ionic Image Upload</h1>
  <?php
  $scan = scandir('uploads');
  foreach($scan as $file)
  {
    if (!is_dir($file))
    {
        echo '<h3>'.$file.'</h3>';
      echo '<img src="uploads/'.$file.'" style="width: 400px;"/><br />';
    }
  }
  ?>
</body>
</html>
```

Now you got everything in place and can start your Ionic image upload adventures!

If you now pick an image and upload it, you should see this dialog as a result (after finished upload).



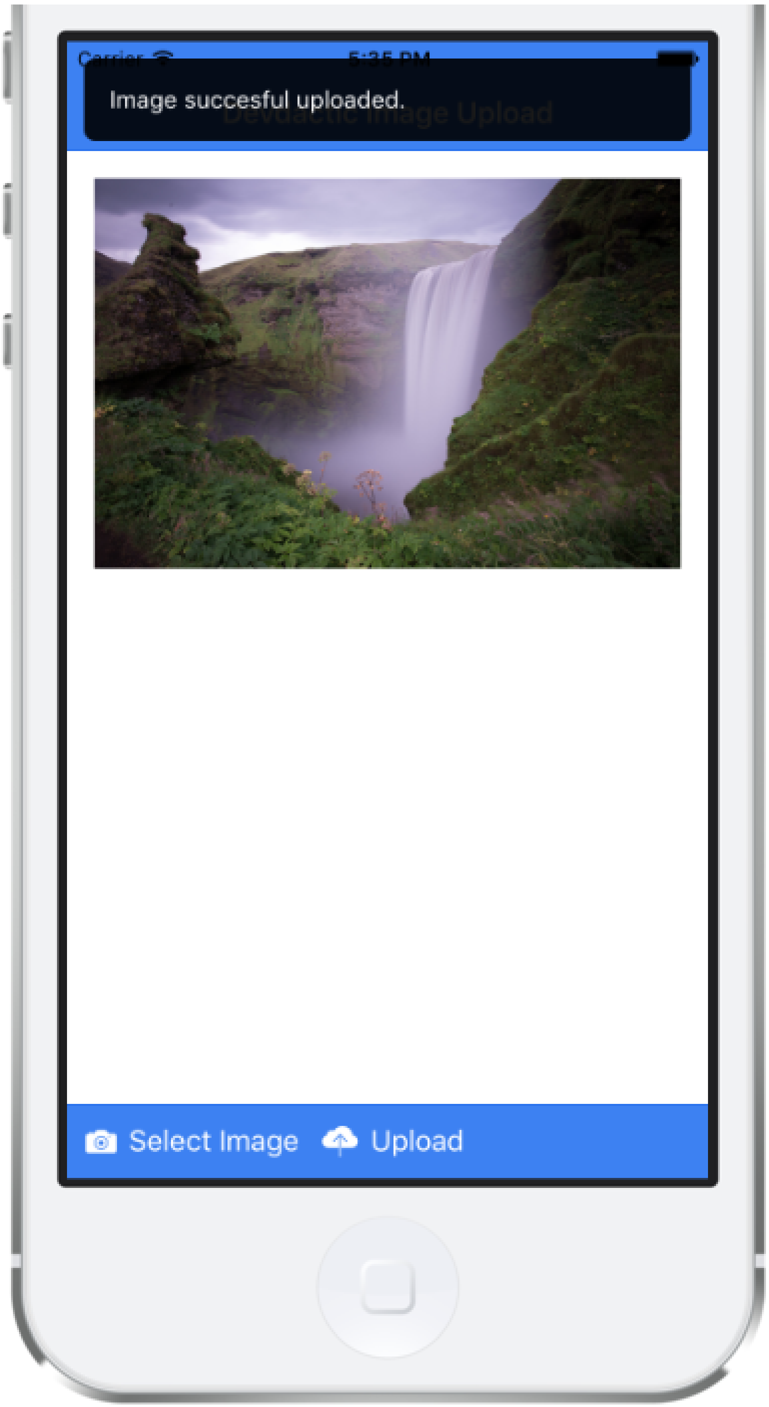


Image succesful uploaded.

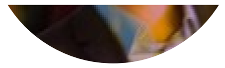
Conclusion

Just like Ionic 1 it’s quite easy to work with images in Ionic 2+. I hope this article helps you to overcome any issues you had while working with Ionic images and files.

Happy Coding,
Simon

About Simon

Ionic Framework Expert • Educator, Consultant & Speaker • Creator of the Ionic Academy



172 Comments **Devdactic** **Arjun Gupta**

♡ **Recommend** 31 **Share** **Sort by Best**

Join the discussion…

Sreekanth karini • 5 months ago

Android build failed, because of cordova plugin file-path any help plz?

11 ^ | v • Reply • Share ›

Rafa Salazar → Sreekanth karini • 21 days ago

Remove platform android previously with ionic cordova platform remove android, after build your android project.

^ | v • Reply • Share ›

Udaysinh Vikramsinh Shinde → Sreekanth karini • 3 months ago

i am also facing the same issue.. Did you get any solution to solve it?

^ | v • Reply • Share ›

Michael Schöller → Sreekanth karini • 5 months ago

i have the same issue, did you find a way to solve it yet?

^ | v • Reply • Share ›

Jorge Cotrina • a year ago

With REST API ?

10 ^ | v • Reply • Share ›

Syed Danish Ali Shah → Jorge Cotrina • 7 months ago

Did you find anything for uploading image on ios & andriod with REST API ? Please share it.

6 ^ | v • Reply • Share ›

b__d055 → Jorge Cotrina • 4 months ago

@s**@Syed Danish Ali Shah** I was able to get this example to work with my REST API by simply specifying the POST verb in the FileUploadOptions. Here is my code:

```
var options = {
fileKey: "file",
fileName: filename,
chunkedMode: false,
```

mimeType: "multipart/form-data",
params : {'fileName': filename},
httpMethod: 'POST'
};

https://ionicframework.com/...

^ | ⌄ • Reply • Share ›

Amelie Perrin • 3 months ago

For ios , if you don't see the picture add this:
import { normalizeURL } from 'ionic-angular';
and change this:
return cordova.file.dataDirectory + img;
with this:
return normalizeURL(cordova.file.dataDirectory + img);

4 ^ | ⌄ • Reply • Share ›

Yuriy Tsoy ➦ Amelie Perrin • 3 months ago

Thank you very much.

^ | ⌄ • Reply • Share ›

Lauren • a year ago

I'm just getting a white screen when trying to test on iOS (iPhone 7). It builds properly but when I load on my device via XCode I see the splash screen and then a white screen. Any ideas? I purchased the book but I was hoping this would work...

4 ^ | ⌄ • Reply • Share ›

Smart Code Creator ➦ Lauren • a year ago

Dear Lauren, First I had white screen. This command solved the problem:
npm install ionic-native --save

7 ^ | ⌄ • Reply • Share ›

خالد الصيفي ➦ Smart Code Creator • a year ago

Thank You Very much Dear Smart Code Creator
you helped me to solve this problem
could you tell me what is the meaning of
--save ?

^ | ⌄ • Reply • Share ›

fureszk ➦ خالد الصيفي • a year ago

--save flag makes sure the dependency (ionic-native) is added to the package.json file, so when nothing is installed and you run "npm install" it gets installed.

1 ^ | ⌄ • Reply • Share ›

sami aji ➦ Smart Code Creator • 7 months ago

its not worked for me, help me :(

^ | v • Reply • Share ›

Shubham Sharma ➦ sami aji • 7 months ago

same

^ | v • Reply • Share ›

Neonic, Inc ➦ Lauren • a year ago

Woah, a female coder! Woah!

^ | v • Reply • Share ›

Finz Aiko • a year ago

I like your great article but I got a problem with toas "Error while storing file." when I pick a picture load from library, I am using genymotion API-22

4 ^ | v • Reply • Share ›

Matthew Steven ➦ Finz Aiko • a year ago

I'm also having the same issue. I wonder if the source of this problem is the FilePath plugin. I get the following error in debugger: FileError {code: 1, message: "NOT_FOUND_ERR"}

I'll post back if I find the solution.

^ | v • Reply • Share ›

Felipe Albuquerque de Almeida ➦ Matthew Steven • a year ago

Try: https://devdactic.com/ionic...

1 ^ | v • Reply • Share ›

sandi ramadhan ➦ Finz Aiko • a year ago

is it solved yet ?

^ | v • Reply • Share ›

Arne • 10 months ago

The selected image will not be visible if you use the iOS Emulator (Not allowed to load local resource) if you start it with "ionic cordova run ios --livereload" but it will if you remove the "--livereload". Thanks for the tutorial btw.

2 ^ | v • Reply • Share ›

detalhar ➦ Arne • 3 months ago

Thanks!!! Works for me too.

^ | v • Reply • Share ›

Agustin Aliaga ➦ Arne • 7 months ago

Worked for me

⌃ ⌄ • Reply • Share ›

Mohammad Mahmoudi • 10 months ago

hi very thanks for this tutorial
how can i select multiple image for upload ?

3 ⌃ ⌄ • Reply • Share ›

Pooja ➦ Mohammad Mahmoudi • 3 months ago

Hello Mohammad
Can you send the php file which you use to upload image in php Server.

⌃ ⌄ • Reply • Share ›

RaviKumar Thuta • a year ago

```
this.file.copyFile(namePath, currentName, cordova.file.dataDirectory,
newFileName).then(success => {
this.lastImage = newFileName;
}, error => {
this.presentToast('Error while storing file.');
});
```

i got the error 'copyFile' does not exist on the type of the File. could you please any one help me how to resolve this error

4 ⌃ ⌄ • Reply • Share ›

sandi ramadhan ➦ RaviKumar Thuta • a year ago

is it solved yet ?

⌃ ⌄ • Reply • Share ›

Jens Nachtigal ➦ sandi ramadhan • 10 months ago

import { File } from "@ionic-native/file";

2 ⌃ ⌄ • Reply • Share ›

Shafiq Essani • 7 months ago

is This plugin "File Chooser" gonna work with iOS ?

ionic cordova plugin add cordova-plugin-filechooser

2 ⌃ ⌄ • Reply • Share ›

Diana Itzel Rodriguez Villalva • 2 months ago

I followed the tutorial step by step, but when I run the application my images are not uploaded to the server, it sends me the message that the image has been uploaded correctly, but in reality the image is never uploaded.

1 ⌃ ⌄ • Reply • Share ›

Victor Carvalho • 3 months ago

Hi, the file transfer plugin is now deprecated (https://github.com/apache/c... Is there a way to do the upload with an @angular/http post request?

1 ⌃ | ⌄ • Reply • Share ›

Hubert Gudan • a year ago

Great article! Like it.
To launch cordova plugins inside browser you can simply use:

$ ionic run browser ☺

1 ⌃ | ⌄ • Reply • Share ›

iotriado ➦ Hubert Gudan • a year ago

hi , I have tried to run this tutorial using browser , and I am get the runtime javasript error below.

Uncaught TypeError: Cannot assign to read only property 'message' of object '[object DOMException]'
at Function.File.fillErrorMessage (file.js:567)

This error is not reproduced on Android , though.
the error is produced when File.copyFile of copyFileToLocalDir is executed.
Any ideas?

⌃ | ⌄ • Reply • Share ›

iotriado ➦ iotriado • a year ago

Ok,
I manage to solve the above by upgrading ionic-native to 2.8.1.

Although File.copyFile function of copyFileToLocalDir throws this exeption

Error while storing file EncodingError: A URI supplied to the API was malformed, or the resulting Data URL has exceeded the URL length limitations for Data URLs.

Can you help on this ? (I have already tried the proposals below regarding filepath and no luck!)
Has anyone manage to run this exampls in browser ,though???

⌃ | ⌄ • Reply • Share ›

cookienawer • 6 months ago

Your update to ionic 3 is incorrect.

The plugin file-transfer change name to FileTransfer, not Transfer, and even if you fix it, other error apear: when you pass lastImage to upload, return code 1 in android, code 3 in ios android, code 0 in ios

1 ^ | ⌄ • Reply • Share ›

leegee ➦ cookienawer • 5 months ago

Nope. https://github.com/dsgriffi...

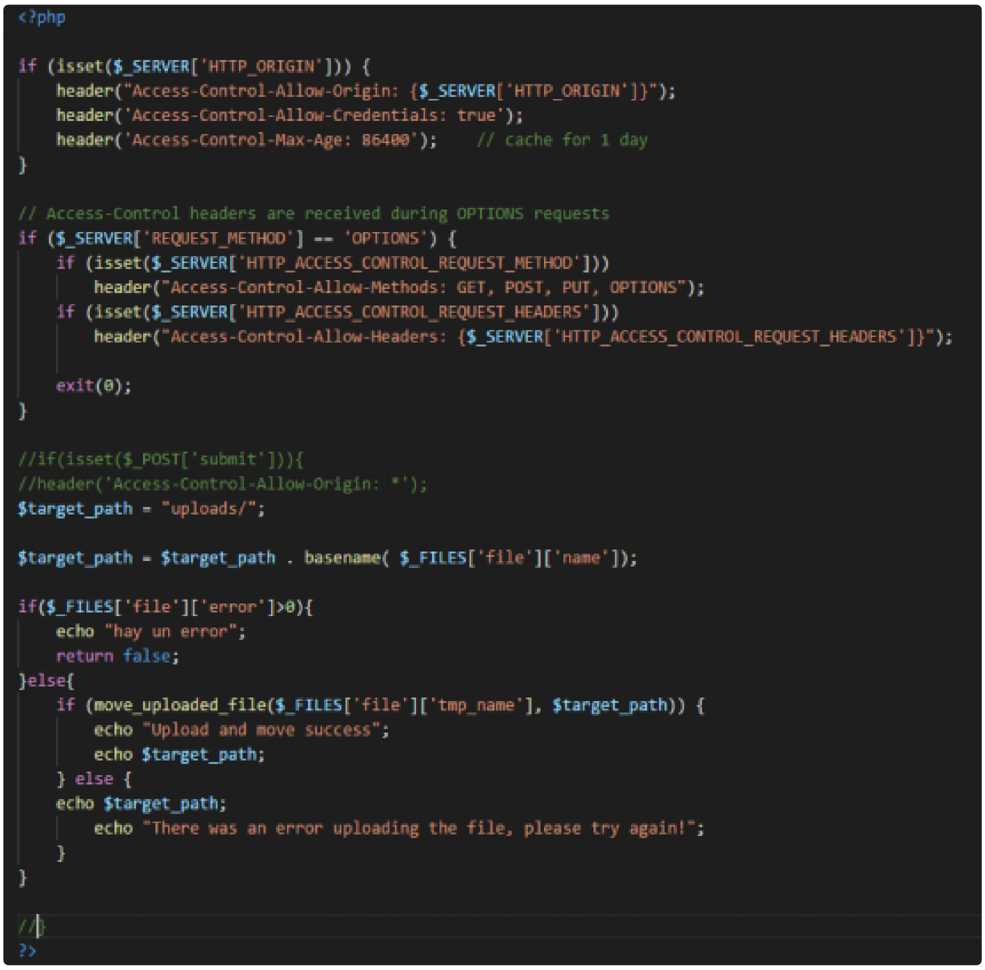
^ | ⌄ • Reply • Share ›

Bruno Fasolin Oliveira • a year ago

I did everything exactly as written in the tutorial however my images are not uploaded to the server. Anyone else with this problem?

1 ^ | ⌄ • Reply • Share ›

Diana Itzel Rodriguez Villalva ➦ Bruno Fasolin Oliveira • 2 months ago



```php
<?php

if (isset($_SERVER['HTTP_ORIGIN'])) {
    header("Access-Control-Allow-Origin: {$_SERVER['HTTP_ORIGIN']}");
    header('Access-Control-Allow-Credentials: true');
    header('Access-Control-Max-Age: 86400');    // cache for 1 day
}

// Access-Control headers are received during OPTIONS requests
if ($_SERVER['REQUEST_METHOD'] == 'OPTIONS') {
    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_METHOD']))
        header("Access-Control-Allow-Methods: GET, POST, PUT, OPTIONS");
    if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']))
        header("Access-Control-Allow-Headers: {$_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']}");

    exit(0);
}

//if(isset($_POST['submit'])){
//header('Access-Control-Allow-Origin: *');
$target_path = "uploads/";

$target_path = $target_path . basename( $_FILES['file']['name']);

if($_FILES['file']['error']>0){
    echo "hay un error";
    return false;
}else{
    if (move_uploaded_file($_FILES['file']['tmp_name'], $target_path)) {
        echo "Upload and move success";
        echo $target_path;
    } else {
    echo $target_path;
        echo "There was an error uploading the file, please try again!";
    }
}

//}
?>
```

I solved it, just add some lines to the php server

^ | ⌄ • Reply • Share ›

Diana Itzel Rodriguez Villalva ➦ Bruno Fasolin Oliveira • 2 months ago

the same problem

^ | ⌄ • Reply • Share ›

Ved Prakash • a year ago

HI,

I want to upload image as base64 using rest API help me

1 ∧ | ∨ • Reply • Share ›

vinu ➦ Ved Prakash • 9 months ago

```
this.camera.getPicture({
sourceType: this.camera.PictureSourceType.SAVEDPHOTOALBUM,
destinationType: this.camera.DestinationType.DATA_URL
}).then((imageData) => {
this.base64Image = 'data:image/jpeg;base64,' + imageData;
return this.base64Image;
}, (err) => {
console.log(err);
});
```

∧ | ∨ • Reply • Share ›

Frank Paepens • a year ago

Thanks for this really clear tutorial.
But I'm having a problem with the FilePath plugin.
Just noticed somebody else already has a solution for this... Thx!

1 ∧ | ∨ • Reply • Share ›

Sajal Suraj ➦ Frank Paepens • 3 months ago

What's the solution ?

∧ | ∨ • Reply • Share ›

Hanan Ur Rehman • 3 days ago

SOLVED!
If you're having issues with the photo not appearing after snapping it from the camera
and maybe in some cases if says in the console
"net::ERR_FILE_NOT_FOUND"
OR
"GET file:///data/user/0/io.ionic.starter/files/1525117528240.jpg 0 ()"
In var options Try changing
saveToPhotoAlbum: false,
to
saveToPhotoAlbum: true,

∧ | ∨ • Reply • Share ›

Himesh • 7 days ago

Amazing tutorial!!!! Works like a charm in android. However in ios, it shows Error Domain=NSCocoaErrorDomain Code=260 "The file “1524723056472.jpg” couldn’t be opened because there is no such file. Any help would be appreciated.

∧ | ∨ • Reply • Share ›

Manoj • 23 days ago

I want to upload multiple files at once, is it possible to do that?

⌃ | ⌄ • Reply • Share ›

walid willy • a month ago

Thanks it works ♥

⌃ | ⌄ • Reply • Share ›

mohamedfarjallah • 2 months ago

Hi developers could this work with multiple select for both images and videos !

⌃ | ⌄ • Reply • Share ›

Vidit Goyal • 2 months ago

Hi, I have done exactly as mentioned on the article but after my plugin is active and i have selected the image, the ionic application restarts instead or returning to the page with the image

⌃ | ⌄ • Reply • Share ›

FOLLOW ME ON TWITTER

Build a Tab Bar navigation

Make HTTP calls to a REST API

Store Data inside your app

Use Cordova plugins

Style your Ionic app

The course is free, so there's nothing you can lose!

Tweets by @schlimmson

Simon Grimm
@schlimmson

Optimising the performance of an Ionic PWA - Part 2 buff.ly/2KtzMCr

Optimising the performan…
In a previous post, I showed you how to add Angular's Service Worker (NGSW)
robferguson.org

4h

Simon Grimm
@schlimmson

Create a cool Drawing app using @Ionicframework and the #Canvas to export Image Files buff.ly/2ETo6oq @JavaScriptDaily

Embed View on Twitter

FEATURED ON: