What is the best way to upload and store pictures on the site?

Asked 7 years, 9 months ago Active 2 years, 4 months ago Viewed 42k times

54

45

I have no idea how the big websites save the pictures on their servers. Could any one tell me how do they save the pictures that are uploaded by the users in their database?

I was thinking, maybe they would just save the file(the picture) in some path and just save that path in the databse is that right?

But I want to do it this way. Is this right? For example, a website named `www.photos.com` . When a user uploads a picture I would create a folder of the user name and save those pictures in that folder.

I believe we can create a directory using `php file concepts` . So when a new user uploads his picture or file, I want to create a directory with his name.

Example: if user name is john, I would create a directory like this on photos.com `www.photos.com/john/` and then save all his pictures to this directory when he uploads a picture. Is this the right way to do this?

I have no one here that has good knowledge of saving the files to servers so please let me know how to do this? I want to do it the correct and secure way.

php file-upload

edited Jan 19 '12 at 13:40
Jason
8,162 21 75 105

asked Jan 19 '12 at 6:44
niko
5,574 22 69 119

2 @AdrianCornish Could you tell me what you mean by what have you tried? Im not asking you to write a code for me ,i just want the algorithm or the process the most people follow so that I could go ahead if i know what to do – niko Jan 19 '12 at 6:48

Again what have you tried? You can store a path to the file, you can store the binary data in the database. What is your actual question? – Adrian Cornish Jan 19 '12 at 6:51

Your question is badly phrased - you are asking how to upload pictures. Not how to store pictures upload to your site. Also what are you doing to protect users against upload malicious code, images with hidden payloads, stopping people just grabbing all your images from a index'able url – Adrian Cornish Jan 19 '12 at 6:55

1 the word 'database' has a special meaning. and you misled every one tried to answer your question. however, that's their bad as they didn't read the question actually – Your Common Sense Jan 19 '12 at 8:43

1 @Frankie sorry i dont know but i accepted your answer and thanks for letting me know that well i

4 Answers

173

✓

All big websites **don't save pictures to the database** they store them in the disk. They save a reference to the picture's position in a table. And then link from there.

Why? Performance.

Pulling heavy content from a database is a huge performance bottleneck. And databases don't scale horizontally that well, so it would mean even a bigger problem. All big sites use static content farms to deal with static content such as images. That's servers who won't care less about your identity.

How do they keep the pictures really private you might ask? They don't.

The picture's link is, in itself, the address and the password. Let's take Facebook, for example. If I store a private picture on my account you should not be able to open it. But, as long as you have the correct address you can.

This picture is private. Notice the filename
`10400121_87110566301_7482172_n.jpg`
(facebook changes the url from time to time so the link may be broken)

It's non sequential. The only way to get the picture is to know it's address.
Based on a previous user photo you can't guess the next one.
It has a huge entropy so even if you start taking random wild guesses you'll have an extensive amount of failures and, if you do get to a picture, you won't be able to, from there, realize the owners identity which, in itself, is protection in anonymity.

Edit (why you should not store images in a "username" folder:
After your edit it became clear that you do intent to put files on disk and not on the database. This edit covers the new scenario.

Even though your logic (create a folder per user) seams more organized it creates problems when you start having many users and many pictures. Imagine that your servers have 1T disk space. And lets also imagine that 1T is more or less accurate with the load the server can handle.

Now you have 11 users, assume they start uploading at the same time and each will upload more than 100GB of files. When they reach 91GB each the server is full and you must start storing images on a different server. If that user/folder structure is followed you would have to select one of the users and migrate all of his data to a different server. Also, it makes a hard-limit on a user who can't upload more than 1T in files.

Should I store all files in the same folder, then?
No, big-sites generally store files in sequential folders (/000001/, /000002/, etc) having an x defined number of files per folder. This is mainly for file-system performance issues.

More on **how many files in a directory is too many?**

edited Jun 1 '17 at 7:04

answered Jan 19 '12 at 6:48
Frankie
19.7k 10 65 105

1 @Fred absolutely true. This answer is specifically designed to set the correct mindset on someone who has no idea on how things work. As soon you start hosting a couple of million pictures on your site you'll probably wanna split the problem and/or outsource hosting those assets. The mindset, though, stays the same. On the database you only reference image location. Thanks for the comment. – Frankie Dec 15 '16 at 16:33

6

It is usually a bad idea to store images in your database (if your site is popular). Database is, traditionally, one of main bottlenecks in most any application out there. No need to load it more than necessary. If images are in the filesystem, many http servers (nginx, for example) will serve them most efficiently.

The biggest social network in Russia, Vkontakte does exactly this: store images in the filesystem.

Another big social network implemented a sophisticated scalable blob storage. But it's not available to the public, AFAIK.

Summary of this answer: don't store blobs in the database.

answered Jan 19 '12 at 6:50

Sergio Tulentsev

192k 32 309 324

3

Image management may best be achieved by physically uploading images to the server and then recording file location and image details in a database. Subsequently, a Search Form could be configured to permit the user to do a text search, part number search, or other queries. A PHP script could be written to produce a valid HTML image tag based on data found in the table.

uploading images into a MySQL™ BLOB field is such a bad idea such image data is generally problematic if the images are much larger than thumbnails. If the images are large, you can end up having to copy/paste one SQL INSERT statement at a time (into phpMyAdmin). If the images are large and the SQL INSERT statement is broken into two lines by your text editor, you'll never be able to restore the image.

answered Jan 19 '12 at 7:19

user319198

3

> is this the right way to do

Yes.
The only thing I'd suggest to use not name but id

Your Common Sense
137k 21 154 268