

PDA | O portálu | Podpořte nás | Ke stažení | Redakce | Inzerce | Partneři | Registrace | Login

Přidejte se

prográmujte

zaměřeno na informační technologie

Články | Diskuzní fórum | Kritika webů | Podcast | Kalendář akcí | Články na přání | E-shop | Login/Nový účet

Anketa

Jakou agregační službu používáte pro sledování nových článků?

Programujte » Webdesign » PHP

Přidat do vlastních článků

Generujeme náhledy s PHP objektově

09. 01. 2010 | 01:00 - Juda Kaleta (thco) - 6019× přečteno

Ve velmi volném pokračování předchozího článku o vytváření webové galerie si napíšeme třídu pro vytváření automatických náhledů, jejich zobrazování a použití.

Vytvářet náhledy je u webové galerie nutnost. I fotky z běžného digitálu mají minimálně 2 MB a načítat takto velký obrázek by se jistě nikomu nechtělo. A protože jsme programátoři myslící na uživatele, který chce mít co nejméně potíží a práce, musíme vytvářet zmenšeniny sami. A nejlépe automaticky.

I PHP umožňuje pracovat s obrázky. Pro vytvoření náhledu moc pokročilých funkcí nepotřebujete, jen vytvoříte zmenšenou kopii. Generovat náhled pokaždé by bylo výpočetně náročné, proto budeme chtít náhledy i ukládat.

Během čtení doporučuji přeskakovat na ukázky kódu. Lépe si tak uděláte představu, o čem je řeč. Někdy mám sklony moc se rozepisovat, proto neváhejte pasáže, kde popisují jasné a zbytečné, přeskočit. Pokud vám něco nebude jasné, můžete se kdykoli vrátit zpět. Dost už ale řečí, jdeme na to.

Co potřebujeme?

Budu předpokládat, že znáte PHP. Pokud už jste se setkali s objekty (třídami), budete z článku mít o něco víc. Budou se tu používat takové věci, jako jsou výjimky a podobně. Jestli nevíte, o čem je řeč, doporučuji prostudovat nějaký článek o objektivním programování (doporučuji článek Jakuba Kulhana - [OOP v PHP](#)). Dále se nebudu rozepisovat podrobně o jednotlivých funkcích - seznámím vás s jejich existencí a možným využitím, více informací můžete získat na php.net.

Možná se setkáte s tím, že vám bude v PHP chybět knihovna GD, která

Vyhledávání »

Hledat

richmond, datagrid view,
uncover, st232, playstation,
page, splinter cell,
validation, zdnet.co.uk

Přihlášení »

geniv

.....

Přihlásit se

Založit nový účet
Zapomenuté heslo
Výhody registrace

Reklama »
Reklama »

www.jobs.cz

SPOJENÍ S ELITOU

Práce v IT

TEAM LEADER, Programmer
- Documentum

Product Manager -
strukturovaná kabeláž a
optika

IT - programátor

Linux Administrátor

Programátor .NET

PL/SQL vývojář - junior

více nabídek práce »

Gopas & DAQUAS &
Microsoft si Vás dovolují
pozvat na největší IT
odbornou konferenci TechEd
2010 TECHNICAL
EDUCATION & DEVELOPER
DATABASE CONFERENCE
2010 ve dnech 26. - 29.
dubna 2010 v multiplexu
CineStar Černý Most.
www.teched.cz

Anketa »

Jakou agregační službu
používáte pro sledování
nových článků?

Desktopovou RSS čtečku - 10042 hl.

Google Reader - 9920 hl.

PraveDnes.cz - 9561 hl.

Weblogy.cz - 9616 hl.

ITzpravy.cz - 9872 hl.

práci s grafikou obsluhuje. Dnes je na většině hostingů dostupná, neměl by být tedy problém, na Linuxu byste ji měli nalézt v repozitáři své distribuce.

Řekněme, že obrázky budeme načítat ze složky `img`, ideální by bylo, kdybychom mysleli i na načítání z podsložek. Pro ukládání náhledů si vytvoříme složku `thumb`, u které nastavíme oprávnění pro zápis a čtení všem skupinám (`chmod 777`). Pokud tak neučiníte, poznáte to podle chybové hlášky. Soubor s třídou nazveme `class.thumb.php` a soubor, který nám bude obsluhovat zobrazení po zadání GET parametru, se bude jmenovat `thumb.php`.

Adresářová struktura

- » `./img/`
- » `./thumb/`
- » `./class.thumb.php`
- » `./thumb.php`

Soubor `class.thumb.php`

Na začátku souboru bychom si měli nadefinovat cesty složek, které budeme používat.

```
Ukázat zdrojový kód Zkopírovat do schránky Tisk
01. define('DIR_IMAGES', './img/');
02. define('DIR_THUMBS', './thumbs/');
```

Pokud už máme vytvořenou kostru nějaké galerie a tyto cesty máme nadefinované jinde, použijeme:

```
Ukázat zdrojový kód Zkopírovat do schránky Tisk
01. include 'soubor_s_definicemi.php';
```

Třída `Thumb`

Co bude vlastně třída `Thumb` dělat? Možná raději než třída bychom měli použít výraz objekt - bude nám vlastně zastupovat jakýsi abstraktní náhled. Když vytvoříme tento objekt, rozumí se tím, že vytváříme novou instanci náhledu nějakého obrázku. Budeme k tomu potřebovat nějaké vnitřní proměnné, určitě pro cestu k obrázku (`$img`), cestu k vytvořenému náhledu (`$thumb`) a časem se nám bude hodit i proměnná, kam uložíme formát obrázku (`$extension`). Do proměnných `$width` a `$height` si uložíme požadovanou velikost náhledu. Protože nechceme, aby nám něco jiného na proměnné sahalo, nastavíme je všechny jako privátní.

```
Ukázat zdrojový kód Zkopírovat do schránky Tisk
01. class Thumb {
02.     // paths and files
03.     private $img = '';
04.     private $thumb = '';
05.
06.     private $extension = '';
07.
08.     // thumb sizes
09.     private $width = 0;
10.     private $height = 0;
```

Konstruktor

Z těchto úvah pak budeme vycházet při tvorbě konstruktoru. Jeden parametr je jasný - bude jím název souboru, ze kterého chceme vytvářet náhled. A protože vytváříme zmenšeninu obrázku, určitě budeme požadovat alespoň jeden rozměr.

Počet hlasů: 49011

Newsletter »

Máte-li zájem přihlášení/odhlášení odběru nových článků na Programujte, můžete si jej zdarma nechat posílat 1x týdně na svoji emailovou adresu:

@

Projekty »

Diskuzní fórum
Diskuze o portále

Tkinter

Zpravodajství v oblasti IT NOVINKA

Partnerské portály »

Soom.cz - počítačová bezpečnost
ER.cz - přesměrování zdarma
Vyvojar.cz - vývojáři sobě

První co při vytváření nového objektu musíme udělat, je ověření, zda soubor vůbec existuje. V této fázi ještě nebudeme zjišťovat, zda ho i dokážeme zpracovat, to se lépe hodí do jiné funkce. Pomocí funkce `is_file` zjistíme, zda soubor existuje. (Zde výhodnější než použití funkce `file_exists` - ta by nám vrátila `true` i při existenci složky.) Správně řečeno, zjistíme, zda neexistuje. Pokud totiž neexistuje, budeme pracovat v duchu objektů a vytvoříme výjimku.

Výjimku umístíme na začátku souboru pod definice cest. Já ji nazval `ThumbExceptionNotFile`, vy si ji pojmenujte tak, abyste na první pohled věděli, proč výjimka existuje a k čemu je. Tím situaci, kdy soubor neexistuje, více řešit nemusíme.

Pokud soubor existuje, pokračujeme dále - zapíšeme si celou cestu k souboru do proměnné - včetně naší definice `DIR_IMAGES` na začátku skriptu. K získání koncovky obrázku použijeme funkci `pathinfo(string $path, PATHINFO_EXTENSION)`. Díky druhému parametru získáme pouze koncovku. Zapíšeme si i požadovanou šířku a výšku náhledu a při tom si vynutíme číselný typ zápisem `(int)`. Jako poslední uložíme cestu k náhledu - protože chceme umožnit více velikostí náhledu od jednoho obrázku, zakomponujeme do názvu i jeho velikost. Abychom se zbavili nepříjemných znaků, jednoduše název zakódujeme pomocí funkce `md5()`.

Poslední co v konstruktoru chceme mít, je vytvoření náhledu, pokud ještě neexistuje. Pro ověření nám v budoucnu bude sloužit metoda `thumbExists()` a pro vytvoření `createThumb()`.

Celá metoda `__construct()`:

```
Ukázat zdrojový kód Zkopírovat do schránky Tisk
01. public function __construct($img, $width, $height=0) {
02.     if(!is_file(DIR_IMAGES.$img)) {
03.         throw new ThumbExceptionNotFile('File doesnt exist');
04.     }
05.
06.     $this->img = DIR_IMAGES.$img;
07.     $this->extension = pathinfo($this->img, PATHINFO_EXTENSION);
08.
09.     $this->width = (int) $width;
10.     $this->height = (int) $height;
11.     $this->thumb = DIR_THUMBS.md5($img.$width.$height);
12.
13.     if(!$this->thumbExists()) {
14.         $this->createThumb();
15.     }
16. }
```

Metoda `thumbExists()`

Dobrým zvykem je definovat metody bezprostředně po volání, a to ve stejném pořadí. Pokud jsme tedy jako první volali v konstruktoru metodu `thumbExists()`, měla by pro dobrou čitelnost kódu následovat.

V této funkci budeme zjišťovat, zda náhled již existuje a zda není starší než originál. To můžeme smrsknout do jedné podmínky - pomocí funkce `filemtime()` ověříme datum poslední modifikace náhledu a originálu, s tím, že pro situaci, kdy náhled nebude ještě existovat, potlačíme chybové hlášení pomocí `@`. Obecně potlačovat chybová hlášení velice nedoporučuji, stejně jako používat zápis `or die()` při každé možné příležitosti. Zde si to dovolit můžeme, i tak si raději poznamenejme, že je to schválně.

Ukázat zdrojový kód Zkopírovat do schránky Tisk

```
01. private function thumbExists() {
02.     if (@filetime($this->thumb) > filetime($this->img)) { // intentionally @
03.         return true;
04.     }
05.
06.     return false;
07. }
```

Metoda createThumb()

Další metoda bude celý proces vytvoření nového náhledu. Nejdříve si zjistíme rozměry původního obrázku pomocí `getimagesize()`, čímž zároveň zjistíme, zda je to ve skutečnosti obrázek. Pokud není, budou rozměry nulové a my vyhodíme další výjimku, tentokrát `ThumbExceptionNotImage`.

Následuje dopočítávání velikostí stran náhledu, pokud byla zadána jen jedna. Protože je to celkem specifická operace, vytvoříme si pro to metodu `calcSize()`, které předáme poměr stran.

Následuje konečné tvoření obrázku. Pomocí funkce `imagecreatetruecolor()` si vytvoříme obrázek s rozměry náhledu. Pak musíme načíst původní obrázek - tam se nám možnosti větví na různé formáty, proto to vyřešíme novou metodou `createFromOriginal()`. Získaný originál pak pomocí `imagecopyresampled()` zmenšíme na velikost náhledu. Šlo by použít i funkci `imagecopyresized()`, ta oproti funkci, která i převzorkovává, dává divný, přestřehý obrázek.

Posledním krokem je uložení náhledu. Já zvolil formát PNG, z čehož vyplývá použití funkce `imagepng()` (pro JPEG je to `imagejpeg()`, použití `imagegif()` bych nedoporučoval).

Ukázat zdrojový kód Zkopírovat do schránky Tisk

```
01. private function createThumb() {
02.     list($img['width'], $img['height']) = getimagesize($this->img);
03.
04.     if (!$img['width'] or !$img['height']) {
05.         throw new ThumbExceptionNotImage('Size is zero. Maybe this is not image.');
```

Metoda calcSize()

Pomocí jednoduché matematiky chybějící velikost dopočítáme z poměru stran, kdy platí, že šířka originálu ku jeho výšce se rovná šířce náhledu ku jeho výšce.

Ukázat zdrojový kód Zkopírovat do schránky Tisk

```
01. private function calcSize($ratio) {
02.     if ($this->width==0) {
03.         $this->width = $this->height * $ratio;
04.     } else if ($this->height==0) {
05.         $this->height = $this->width / $ratio;
06.     }
07. }
```

Metoda createFromOriginal()

Jednoduchým použitím `switch` použijeme pro každý formát jinou funkci. Protože

jsme kontrolovali pouze příponu, ne skutečný obsah, může se nám stát, že skončíme na chybové hlášce. Nic horšího. Pro případ, že formát dodán bude a bude to i obrázek (projde funkcí `getimagesize()`), ale my ho nebudeme umět zpracovat, vytvoříme další výjimku: `ThumbExceptionUnknownFormat`.

Vypisovat, pro jaký formát je jaká funkce, nemá cenu, přečtěte si kód:

```
Ukázat zdrojový kód Zkopírovat do schránky Tisk
01. private function createFromOriginal() {
02.     switch(strtolower($this->extension)) {
03.         case 'jpg':
04.             return imagecreatefromjpeg($this->img);
05.         break;
06.         case 'jpeg':
07.             return imagecreatefromjpeg($this->img);
08.         break;
09.         case 'png':
10.            return imagecreatefrompng($this->img);
11.        break;
12.        case 'gif':
13.            return imagecreatefromgif($this->img);
14.        break;
15.        default:
16.            throw new ThumbExceptionUnknownFormat('Unknown file format');
17.        break;
18.    }
19. }
```

Metoda view()

Náhled se nám vytváří, teď ho ještě poslat na monitor. To nám zajistí veřejná metoda `view()`. V ní pošleme informace o typu souboru a jeho velikosti, obrázek vypíšeme funkcí `file_get_contents()` a pošleme vše k uživateli pomocí `flush()`, protože nechceme, aby nám k obrázku něco přibýlo - pak by byl k ničemu, nezobrazil by se. To samé bychom měli udělat i se všemi texty předtím. Vyprázdníme proto na začátku funkce paměť funkcí `ob_clean()`.

```
Ukázat zdrojový kód Zkopírovat do schránky Tisk
01. public function view() {
02.     ob_clean();
03.
04.     header('Content-type: image/png');
05.     header('Content-Length: ' . filesize($this->thumb));
06.
07.     echo file_get_contents($this->thumb);
08.     flush();
09. }
```

Tím máme třídu pro vytváření náhledů hotovou. Přímo nabízející se rozšíření funkčnosti je přidání vodotisku. Pokud jste článek pochopili, neměl by to pro vás být problém. Pokud vás téma zaujalo, berte to jako "domácí úkol". Šikovné funkce naleznete na php.net.

Ukázat zdrojový kód Zkopírovat do schránky Tisk

```

01. <?php
02.
03. define('DIR_IMAGES', './img/');
04. define('DIR_THUMBS', './thumbs/');
05.
06. class ThumbExceptionNotFile extends Exception {}
07. class ThumbExceptionNotImage extends Exception {}
08. class ThumbExceptionUnknownFormat extends Exception {}
09.
10. /**
11.  * Need PHP with GD libraries.
12.  */
13.
14. class Thumb {
15.     // paths and files
16.     private $img = '';
17.     private $thumb = '';
18.
19.     private $extension = '';
20.
21.     // thumb sizes
22.     private $width = 0;
23.     private $height = 0;
24.
25.     public function __construct($img, $width, $height=0) {
26.         if(!is_file(DIR_IMAGES.$img)) {
27.             throw new ThumbExceptionNotFile('File doesnt exist');
28.         }
29.
30.         $this->img = DIR_IMAGES.$img;
31.         $this->extension = pathinfo($this->img, PATHINFO_EXTENSION);
32.
33.         $this->width = (int) $width;
34.         $this->height = (int) $height;
35.         $this->thumb = DIR_THUMBS.md5($img.$width.$height);
36.
37.         if(!$this->thumbExists()) {
38.             $this->createThumb();
39.         }
40.     }
41.
42.     private function thumbExists() {
43.         if(@filetime($this->thumb) > filetime($this->img)) { // intentionally @
44.             return true;
45.         }
46.
47.         return false;
48.     }
49.
50.     private function createThumb() {
51.         list($img['width'], $img['height']) = getimagesize($this->img);
52.
53.         if(!$img['width'] or !$img['height']) {
54.             throw new ThumbExceptionNotImage('Size is zero. Maybe this is not image.');
```

Soubor thumb.php

Jakýmsi kontrolerem, který nám bude náhledy zobrazovat, bude soubor thumb.php. V něm budeme zpracovávat GET data - adresu obrázku, jeho velikost(i) - a pak obrázek pomocí třídy vypíšeme. Také ošetříme výjimky, které mohou nastat.

```

01. <?php
02. include 'class.thumb.php';
03.
04. if(isset($_GET['src']) and (isset($_GET['width']) or isset($_GET['height']))) {
05.     try {
06.         $thumb = new Thumb($_GET['src'], $_GET['width'], $_GET['height']);
07.         $thumb->view();
08.     } catch (exception $e) {
09.         # akce, která nastane, pokud se nepovede vytvorit nahled.
10.         # muze to byt napriklad zobrazení nejakého chybového obrázku.
11.         echo 'Error';
12.     }
13. }
```

A pro úplnost ještě jak zobrazit obrázek v HTML, i když je to podle mne jasné:

```

01. 
```

Tento článek je součástí kurzu či seriálu, ke kterému máte možnost odevzdávat úkoly. Máte-li zájem se zapojit, stačí se [přihlásit](#). Úkoly opravují lektori, kteří v případě potřeby poradí, jak lze řešení lépe zpracovat.

Účast ve všech kurzech je **bezplatná**.

Juda Kaleta

Autor programuje v PHP a Pythonu, má velmi rád Linux a svobodný software. Na svém desktopu upřednostňuje openSUSE, ale nemá žádné předsudky proti jiným distribucím. Aby mu neuniklo nic, co se ve světě IT děje, vytvořil si web <http://www.vseopc.eu>

Korektura: Martin Šimeček (DeedX)

[Tisk](#)
[Doporučit](#)
[RSS](#)
[PDF](#)
[VYBRÁLIŠME SK](#)
[DEL](#)
[1010.US](#)
[LINKUJ.CZ](#)
[DAGNI TO](#)

« Trojrozměrné logo pro webové stránky

Silverlight a práce s databází »

Přečtěte si také:

- [JSON: první kroky aneb používáme PHP a jQuery](#)
- [PHP a MySQL – Mysql](#)
- [OOP v PHP](#)

Povinné ručení

Spočítejte si nejlevnější **povinné ručení**, sazby pro rok 2009.

Kvalitní www stránky

intencio - tvorba **www stránek**, eshopů, řešení reklamy na stránky

Společenské hry

Nenuďte se a společně hrajte **společenské hry** z VeselyKlaun.cz

Hip hop shop

Hip hop shop a Skateshop obchod pro moderní lidi. Vše skladem.

Diskuze k článku (6)

Zobrazit všechny	Přidat nový	Sledovat diskuzi
Nepoužívejte private, kde stač..	RR	17. 01. 2010 21:19
Validita	Vašek	18. 01. 2010 16:44

HOME

→

RSS VYSKÚZEJTE

RSS

KOMENTÁŘE

0

10000

Tagováno:

php oop galerie fotogalerie fotografie image photogallery thumb nahledy

ZÁLOŽKY

★

© 2004-2010 Programujte by Lukáš Churý, ISSN 1801-1586

Tento server dodržuje právní předpisy o ochraně osobních údajů. Všechna práva vyhrazena. Bez svolení redakce není možno texty dále rozšiřovat!

Kontakt | Reklama | Redakce | Podmínky užívání obsahu | Podpořte Programujte.com | Ke stažení | O portálu | RSS exporty [80.78.156.4]

HOME
RSS DISKUSE
KOMENTÁŘE

ZÁLOŽKY

MENU