

Php - 4. část

Ing. Martin Dostal, Ph.D.

madostal@kiv.zcu.cz
github.com/madostal

Obsah

- Opakování - MVC
- Co se jinam nevešlo:
 - Verzovací systémy - Git a Mercurial
 - Dependency management - Composer a Bower
 - Task runner - Grunt
 - Layout - tipy
- Příklady z praxe
- Diskuze - nechápu, nejde mi ...

Opakování - MVC

- MVC – Model, View, Controller
 - Model – provádí obchodní logiku (práce s DB)
 - View – část zajišťující formátování výstupu systému
 - Controller – zpracovává vstup a předává ho modelu, řídí aplikaci
- Hlavní cíl:
 - Oddělení aplikační logiky od zobrazování
 - Aplikace je pružnější, snadno modifikovatelná - např. snadná záměna databázové vrstvy
 - Kód je přehlednější
 - Roste opětovná použitelnost částí systému

MVC prakticky

index.php

- hlavní soubor aplikace, includeuje všechny ostatní objekty
- dle konfigurace vytvoří spojení s databází
- dle parametru nebo URL určí typ obsahu
- např. ze souboru obsah.inc.php se načte požadovaný obsah. S pomocí php wrapperu se uloží do proměnné
- všechny texty se odešlou do šablony k zobrazení

MVC - hlavní zásady

- je třeba dělat věci pořádně, ale není vhodné se “vyžívat” ve složité struktuře aplikace
- občas kostra aplikace složená ze 3 souborů může být naprosto dostatečná:
 - index.php
 - obsah.inc.php
 - sablona.tpl
 - + css apod.

Upload souboru - formulář

- Nastavit typ kódování polí formuláře:
`enctype="multipart/form-data"`
- Nastavit max. velikost souboru:
`<input type="hidden" name="MAX_FILE_SIZE" value="10000" />`
- Vložit pole pro vložení jména souboru:
`<input type="file" name="userfile" />`

Upload souboru – přístup k souboru

```
$tmp_name = $_FILES['userfile']['tmp_name'];
```

```
$file_name = $_FILES['userfile']['name'];
```

nebo size a type

Dále:

- načíst - přes fopen
- Nebo přesunout - např. move_uploaded_file

Přesun všech uploadovaných souborů

```
$uploads_dir = '/uploads';  
foreach ($_FILES["pictures"]["error"] as $key => $error) {  
    if ($error == UPLOAD_ERR_OK) {  
        $tmp_name = $_FILES["pictures"]["tmp_name"][$key];  
        $name = $_FILES["pictures"]["name"][$key];  
        move_uploaded_file($tmp_name, "$uploads_dir/$name");  
    }  
}
```


Zajímavé ukázky z github.com/madostal

- viz. [github](https://github.com)

Verzovací systémy

- Cíl:
 - uchování historie všech provedených změn
 - umožnit spolupráci více programátorů
- Typy:
 - centralizované - např. CVS, Apache Subversion
 - distribuované - např. GIT, Mercurial

Git - opakování

- distribuovaný systém = více repozitářů
- respektive jde o systém a sadu nástrojů
- každý vývojář má vlastní lokální repozitář s celou historií vývoje!
- rychlá tvorba větví a rychlé mergování
- git umožňuje emulaci CVS, takže je možný snadný převod “starých” programátorů na git

Git - hlavní operace

- clone - stažení repozitáře z Internetu
- commit - potvrzení lokální změny
- push - odeslání ven
- pull - stažení změn
- merge - spojení externích a interních změn
- branche - tvorba větve
- tagging - označení daného stavu sw - např. vývojové verze

Mercurial

- multiplatformní distribuovaný nástroj
- primárně pro příkazový řádek
- je monolitický = jedná se o jediný nástroj
 - oproti gitu, který se skládá ze sady nástrojů
- je relativně neflexibilní, dělá dobře to, k čemu byl primárně určen
- mercurial je jen o málo pomalejší oproti gitu při provádění základních operací

Kde hostovat git?

- github.com - zdarma pouze public repozitáře
- bitbucket.org - zdarma i private repozitáře

Git - nástroje

- SourceTree - výborný nástroj pro git i mercurial
- Git Extensions
- SmartGit - komerční sw, ale je zde i free verze

Dependency management

- správa závislostí - jeden plugin závisí na druhém
- můžeme dělat buď ručně, nebo využít specializovaný sw:
 - Bower
 - Composer

Composer

- getcomposer.com
- vhodný pro Php vývojáře
- používá se primárně pro backend (server)
- umožňuje automatickou instalaci a aktualizaci závislostí
- Vy zvolíte požadované knihovny a composer určí závislosti a stáhne i knihovny nutné pro provoz zvolených knihoven



Bower

- použití pro front-end package management a dependency injection
- vyžaduje Node.js a npm



Bower - příklad použití

- v projektu založím soubor bower.json:

```
{  
  "name": "MyProject",  
  "dependencies": {  
    "bootstrap": "*"   
  }  
}
```

Bower - použití

- instalaci závislostí provedeme příkazem: `bower install`
- automaticky stáhne bootstrap i jQuery a uloží typicky do složky *vendor*
- umožňuje stahovat existující balíky
- umožňuje i stažení projektů z gitu

Grunt JS - JavaScript task runner

- gruntjs.com
- Proč? Automatizace.
- Je možné ho použít k automatizaci opakujících se úkolů jako je:
 - minimalizace
 - kopírování a přesuny
 - spouštění unit testů atd.
 - konfigurace: package.json
 - příkazy: Gruntfile - obyčejný JavaScript



GruntJS - konfigurace

- vytvoříme soubor package.json:

```
{  
  "name": "my-project-name",  
  "version": "0.1.0",  
  "devDependencies": {  
    "grunt": "~0.4.5",  
    "grunt-contrib-jshint": "~0.10.0",  
    "grunt-contrib-nodeunit": "~0.4.1",  
    "grunt-contrib-uglify": "~0.5.0"  
  }  
}
```



Layout - tipy

- Modernizr - modernizr.com
 - zjistí podporu prohlížeče - HTML5, CSS a JavaScript
 - v případě absence dané funkce je pak možné použít specializovaný JS, který chybějící funkce nahradí
- Yeoman - yeoman.io
 - generátor pro nové projekty
 - podporován Googlem
 - připraví a nastaví nový projekt: yo, dependencies (Bower) a build system (Grunt)

Layout - tipy 2

- Bootstrap - getbootstrap.com
 - nově bude verze 4!
 - v4-alpha.getbootstrap.com
 - vyzkoušejte si kartu = Card, což je náhrada za několik component ve verzi 3

Příklady z praxe

- PMDP prodávalo roční předplatné na rok 2016:
 - Předplatné platí 365 dnů
 - Při zakoupení od 1.1.2016 skončí 30.12.2016
 - Proč? Co je tady špatně?

PMDP

- Cílem byla možnost zakoupit “roční předplatné” kdykoliv a docílit tak možnosti nákupu na období např. 1.3.2015 - 1.3.2016
- Město schválilo a programátoři naprogramovali období na přesně 365 dnů. Jinak to neuměli.
- Nikdo si ale nevzpomněl, že existuje něco jako přestupný rok, který má 366 dnů!
- PMDP to vyřešilo “administrativně” a tedy roční předplatné označilo jako “pracovní označení” období, které nemusí trvat celý rok. Je to správně?

PMDP - jaké je správné řešení?

- Diskuze

PMDP - možnosti

- umožnit nákup předplatného na 365 dnů
- umožnit nákup předplatného na celý kalendářní rok - od 1.1. do 31.12.
- vyhlásit, že na silvestra jezdí celá Plzeň zdarma a problému se tak úplně vyhnout a přilákat nové zákazníky

UTF-8 aneb silver bullet na jazyky

- v minulosti se na webu používalo mnoho různých kódování - ISO 8859-2, Windows-1250 a další
- s podporou UTF-8 u všech prohlížečů se situace změnila a většina lidí začala dělat weby pouze v UTF-8
- Výhody: mohu udělat web česky, anglicky, arabsky i čínsky a vždy to funguje! Ověřeno.

Diskuze

- Nechápu ...
 - MVC?
 - Šablony?
 - Něco jiného?
- Nejde mi ...
 - Wamp?
 - Návrh databáze - MySql workbench?
 - Něco s gitem?