



Osobní údaje

Jméno / Příjmení

Adresa

Telefon

E-mail

Státní příslušnost

Datum narození

Pohlaví

Michal Novák

Charvatská 25, 612 00, Brno, Česká republika

(+420) 777 995 408

michal.novak@wheee.cz

Česká

17.5.1987

muž



Pracovní zkušenosti

Období

Pracovní pozice

Pracovní náplň

červen 2009 - současnost

Junior Programmer

Ve firmě LifeWeb s.r.o. působím primárně jako PHP programátor a podílím se na vývoji redakčního systému. Pro některé úkony využívám také znalosti JavaScriptu a XHTML, případně CSS. Moji hlavní pracovní činností je vývoj modulů do zmíněného systému.

Zaměstnavatel

Obor činnosti

LifeWeb s.r.o., U Vodárny 3032/2a, 616 00, Brno, Česká republika

internetové technologie, tvorba webových aplikací

Období

Pracovní pozice

Pracovní náplň

duben 2009 - současnost

podnikání na živnostenský list

Jako živnostník se specializuji na realizaci webových prezentací klientům na míru. Vzniklé webové stránky splňují webové standardy. Stránky programuji ve skriptovacím jazyce PHP, kóduji v XHTML, styluji CSS 2.1. Při programování využívám vhodných frameworků (Nette, Zend Framework, dibi).

Obor činnosti

internetové technologie, tvorba webových aplikací

Období

Pracovní pozice

Pracovní náplň

leden 2007 - červen 2009

externista (jako grafik, kodér a programátor)

Zde jsem byl zaměstnán na univerzální pozici webového vývojáře. Zajišťoval jsem celý vývoj drobných webových stránek (dynamické i statické).

Zaměstnavatel

Obor činnosti

webové studio WEBDESIGN Tomáš Petříček

internetové technologie, tvorba webových aplikací

Dosažené vzdělání

Období

Vysoká škola

2007 - současnost

Fakulta informatiky Masarykovy univerzity

obor Počítačová grafika a zpracování obrazu

Období

Střední škola

1999 – 2007

Gymnázium Brno, třída Kapitána Jaroše 14

Jazyky

Mateřský jazyk(y)

*Sebehodnocení
Evropská úroveň^(*)*

angličtina

němčina

Schopnosti, znalosti a dovednosti

Programovací jazyky

Značkovací jazyky

Dotazovací jazyky

Řidičský průkaz

Doporučení

Dodatečné informace

Zájmy

čeština

Porozumění				Mluvení				Psaní	
Poslech		Čtení		Ústní interakce		Samostatný ústní projev			
B2	Samostatný uživatel	C1	Zkušený uživatel	B1	Samostatný uživatel	B1	Samostatný uživatel	B2	Samostatný uživatel
A1	Uživatel základů jazyka	A1	Uživatel základů jazyka	A1	Uživatel základů jazyka	A1	Uživatel základů jazyka	A1	Uživatel základů jazyka

^(*) Společný evropský referenční rámec pro jazyky

C, C++, C#, PHP, Java, JavaScript

XHTML, LaTeX, XML, UML, CSS

SQL

skupina B

Ondřej Bouda, Bořetická 11, 628 00, Brno, Česká republika

Ing. Tomáš Petříček, U Hrobečku 42a, 610 00, Brno, Česká republika

cyklistika, turistika, lyžování, cestování

Kapitola 1

PHP framework Nette

1.1 Úvod

Zkratka PHP (z anglického PHP: Hypertext Preprocessor) označuje populární skriptovací jazyk primárně navržený pro vývoj webových aplikací. Jeho oblíbenost vyplývá především z jednoduché syntaxe a pozvolné křivky učení. PHP je open-source a je vydáno pod PHP licenci v3.01[1]. PHP je jazyk interpretovaný a to na straně serveru.

Původním záměrem vzniku jazyka bylo umožnit programátorům jednoduše a rychle vyvíjet dynamické webové stránky. Tato skutečnost vedla k tomu, že se hlavní hnací silou stala rozsáhlá databáze funkcí. Ukázalo se, že PHP je dostatečně výkonné na to, aby v něm mohly být napsány i rozsáhlé projekty a komplexní webové aplikace. Jeho původní procedurální přístup ale nebyl pro takové řešení optimální. Proto PHP od své čtvrté verze obsahuje podporu pro objektově orientované programování.

Nové možnosti jazyka nenabízí pro takový koncept programování dostatečné zázemí. Vzniká proto množství frameworků, které se snaží tento nedostatek řešit. Jedním z nich je i framework Nette.

1.2 Architektura Nette

Nette je objektově orientovaný balíček tříd, které zefektivňují vývojářovu práci při vývoji webové aplikace. Framework je založen na MVC¹ přístupu. Knihovna hojně využívá návrhového vzoru convention-over-configuration². Framework je rozdělen do několika částí a ty jsou na sobě nezávislé.

⁰Framework je softwarová struktura, která slouží jako podpora při programování a vývoji a organizaci jiných softwarových projektů[2].

¹Model-View-Controller je softwarová architektura, která rozděluje datový model aplikace, uživatelské rozhraní a řídicí logiku do tří nezávislých komponent tak, že modifikace některé z nich má minimální vliv na ostatní[3].

²Princip convention-over-configuration snižuje množství práce, kterou by bylo jinak nutné vynaložit pro správné nastavení a propojení aplikace.

1.2.1 Komponenty frameworku

Model

Model tvoří rozhraní pro práci s daty. Různé aplikace mají rozdílné datové modely a každý takový model je specifický pro jinou aplikaci. Proto není rozhraní modelu ničím zastřešeno a způsob implementace je volný. Nette neobsahuje žádné podpůrné třídy pro vývoj této komponenty.

View

Pohled prezentuje stav aplikace uživateli. V případě webových stránek to jsou HTML dokumenty. Pohled by měl být nezávislý na modelu a převážně i na řadiči. Obsah jeho zpracování by měly být pouze jednoduché iterace a větvení. Pohled nemá vyvolávat žádnou změnu v aplikaci, proto by měl být oproštěn od logiky aplikace samotné. Díky tomu lze pohled upravit či zaměnit, aniž by to vyvolávalo nutnou změnu jiné části aplikace.

U rozsáhlých aplikací není možné spravovat každou stránku samostatně. Takové řešení by bylo příliš časově náročné, nepřívětivé pro případnou změnu a náchylné k nekonzistenci. Nette proto poskytuje v této oblasti značné zjednodušení v podobě šablon. Šablony jsou dokumenty, jejichž zpracováním je sestavena výsledná HTML stránka. Správnou volbou hierarchie šablon je docíleno DRY³ konceptu v pohledových komponentách. Tím se značně snižují původní rizika a navíc se kód stává snadno spravovatelným.

Šablony zpracovává třída `Template`. Při vytváření instance třídy je jako parametr předána cesta k šabloně. Při požadavku o vykreslení šablony je tato interpretována jako PHP skript. K dispozici jsou proměnné, které v předchozím životním cyklu aplikace nastavil řadič (viz níže).

Návrhář vzhledu aplikace není programátor a psaní šablon v PHP tvoří při výkonu jeho práce překážku. Proto lze šablony zaregistrovat tzv. filtry. Filtry fungují jako preprocesory; šablonu před samotným vyhodnocením upraví. Tím můžeme sestavit vlastní syntaxi pro šablony a udělat je tak uživatelsky příjemné i pro neprogramátory. Nette pro tento účel poskytuje jako jeden ze základních filtrů `CurlyBracketsFilter`.

Controller

Řadič je výkonné jádro aplikace. Obsahuje veškerou logiku aplikace a tedy se jedná o primární komponentu. Jako jediná ze všech MVC komponent není nahraditelná. Řadič zprostředkovává samotné vykonávání operací v aplikaci. Komunikuje s datovým modelem a zpracovaná data z něj přeposílá pohledu, který je následně prezentuje uživateli.

V Nette frameworku funkci řadiče zastávají potomci odvození od třídy `Presenter`. Tato třída prochází při zpracování třemi fázemi – výkonnou, vy-

³Don't repeat yourself. Vychází z principu slučování stejných částí aplikace.

hodnocovací a vykreslovací. Ve fázi první se vyhodnocuje požadavek obdržený od uživatele (HTTP požadavky) – tzv. akce. Ve fázi druhé se vyhodnocuje výsledek a vybírá pohled. Na stejnou akci tak lze odpovědět různými pohledy. Ve fázi poslední se předávají potřebné parametry výslednému pohledu (šabloně). Z těchto parametrů potom pohled čerpá a prezentuje podle nich výsledek.

1.2.2 URL abstrakce

Jedním z požadavků, které jsou na moderní webové prezentace kladeny, jsou tzv. SEO⁴ URLs. To jsou adresy obsahující klíčová slova stránky (zbavena diakritiky a interpunkce) a svou kompozicí odrážejí skutečnou logickou strukturu webu. Stránky splňující výše uvedené kritérium získávají lepší pozici ve vyhledávačích.

Nette framework funguje na bázi URL abstrakce. Tvar a vzhled adres jednotlivých stránek lze kdykoliv změnit. K dosažení této vlastnosti slouží technika zvaná routování. Routa je obecný předpis webové adresy, který má být aplikován. Rozhraní `IRouter` definuje povinné metody, které musí routa obsahovat. Třída `Router` toto rozhraní implementuje a je základní součástí frameworku. S využitím této třídy můžeme vytvořit vlastní vzor adres, pomocí nichž budou tvořeny odkazy na samostatné stránky aplikace. Routy lze seskupovat do množin, jejichž namapování zpracovává třída `MultiRouter`, která také implementuje rozhraní `IRouter`.

⁴Search engine optimization.

Rejstřík

controller, 2

framework, 1, 2

model, 2

PHP, 1

URL, 3

view, 2

Literatura

- [1] T. P. Group. Php: License information, 2001. [Online; dostupné z <http://www.php.net/license/>; navštíveno 23. 4. 2009].
- [2] Wikipedie. Framework — wikipedie: Otevřená encyklopedie, 2009. [Online; dostupné z <http://cs.wikipedia.org/w/index.php?title=Framework&oldid=3574665>; navštíveno 5. 20. 2009].
- [3] Wikipedie. Model-view-controller — wikipedie: Otevřená encyklopedie, 2009. [Online; dostupné z <http://cs.wikipedia.org/w/index.php?title=Model-view-controller&oldid=3911881>; navštíveno 5. 20. 2009].