

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

RAD VÝVOJOVÉ PROSTŘEDÍ PRO DATABÁZOVÉ
APLIKACE VYUŽÍVAJÍCÍ JAZYK PHP

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

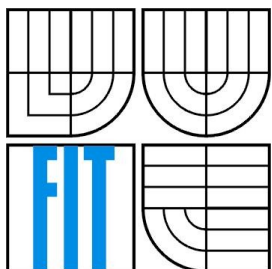
AUTOR PRÁCE
AUTHOR

TOMÁŠ NAIBRT

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

RAD VÝVOJOVÉ PROSTŘEDÍ PRO DATABÁZOVÉ APLIKACE VYUŽÍVAJÍCÍ JAZYK PHP

RAD DEVELOPMENT FRAMEWORK WITH PHP

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

TOMÁŠ NAIBRT

VEDOUCÍ PRÁCE
SUPERVISOR

ING. JAROSLAV RÁB

BRNO 2008

Zadání bakalářské práce

Řešitel: **Naibr Tomáš**

Obor: Informační technologie

Téma: **RAD vývojové prostředí pro databázové aplikace využívající jazyk PHP**

Kategorie: Web

Pokyny:

1. Seznamte se s problematikou tvorby webových informačních systémů v jazyce PHP, spolu s technologií JavaScript.
2. Seznamte se s prostředími Oracle Forms, JDeveloper a dalšími.
3. Navrhněte model vývojového prostředí pro rychlý vývoj aplikací pro WWW prostředí, které bude umožňovat automatickou práci nad daty z databáze. Inspirujte se prvky a vlastnostmi prostředí, se kterými jste se seznámili.
4. Navržený model implementujte v rozsahu prototypu pro tvorbu jednoduchých aplikací.
5. Proveďte test implementace, zhodnoťte navržený model prostředí, diskutujte další rozvoj prostředí.

Literatura:

- Oracle Forms Developer's Guide (<http://otn.oracle.com>)
- PHP Developer's Guide (<http://www.php.net>)

Při obhajobě semestrální části projektu je požadováno:

- Body 1.-3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese
<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Ráb Jaroslav, Ing., UIFS FIT VUT**

Datum zadání: 1. listopadu 2007

Datum odevzdání: 14. května 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

**LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO**

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Tomáš Naibrť**
Id studenta: 78841
Bytem: Dvořákova 879, 289 03 Městec Králové
Narozen: 01. 12. 1985, Městec Králové
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

.....
(dále jen "nabyvatel")

Článek 1

Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
bakalářská práce

Název VŠKP: RAD vývojové prostředí pro databázové aplikace využívající
jazyk PHP

Vedoucí/školitel VŠKP: Ráb Jaroslav, Ing.

Ústav: Ústav informačních systémů

Datum obhajoby VŠKP:

VŠKP odevzdal autor nabyvateli v:

| | |
|--------------------|-----------------------------------------------------|
| tištěné formě | počet exemplářů: 1 |
| elektronické formě | počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD) |

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2

Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
 - ihned po uzavření této smlouvy
 - 1 rok po uzavření této smlouvy
 - 3 roky po uzavření této smlouvy
 - 5 let po uzavření této smlouvy
 - 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3

Závěrečná ustanovení

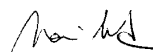
1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne:

.....

Nabyvatel

.....



Autor

Abstrakt

Cílem této práce je analyzovat aktuální stav na poli vývoje databázových aplikací v jazyce PHP a RAD nástroje k tomu určené. Popisuje několik PHP frameworků, které lze aktuálně označit jako nejpoužívanější. Druhá část se zabývá popisem implementace vlastního jednoduchého interaktivního RAD nástroje.

Klíčová slova

PHP, MySQL, databáze, Zend, Symfony, CakePHP, Delphi for PHP, framework, web

Abstract

The tendency of this work is analysing a current situation on the field of progress database application in the languages PHP and RAD tools. It describes some PHP frameworks, which we can specify actually as the most widely used. The second part describes a description implementation own simple interactive RAD tool.

Keywords

PHP, MySQL, database, Zend, Symfony, CakePHP, Delphi for PHP, framework, web

Citace

Tomáš Naibr: RAD vývojové prostředí pro databázové aplikace využívající jazyk PHP, bakalářská práce, Brno, FIT VUT v Brně, 2008

RAD vývojové prostředí pro databázové aplikace využívající jazyk PHP

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jaroslava Rába.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Tomáš Naibrť
15.05.2008

Poděkování

Tímto bych chtěl poděkovat svému vedoucímu Ing. Jaroslavu Rábovi za poskytnutou pomoc a konzultace při tvorbě bakalářské práce.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

| | |
|----------------------------------------------------------------------|----|
| Obsah..... | 1 |
| 1 Úvod..... | 2 |
| 2 Aktuální situace ve vývoji databázových aplikací v jazyce PHP..... | 3 |
| 2.1 Bez pomoci frameworku..... | 3 |
| 2.2 Použití framework..... | 4 |
| 2.2.1 Přehled frameworků..... | 5 |
| 2.3 Komplexní vývojové prostředí..... | 15 |
| 3 Návrh a implementace jednoduchého RAD nástroje..... | 17 |
| 3.1 Výběr PHP frameworku..... | 17 |
| 3.2 Vlastní aplikace napsaná v jazyce Java..... | 17 |
| 3.3 Vygenerovaná PHP aplikace..... | 19 |
| 4 Závěr..... | 21 |
| Literatura..... | 22 |
| Seznam příloh..... | 23 |

1 Úvod

Cílem této práce je analyzovat aktuální stav na poli vývoje databázových aplikací v jazyce PHP.

V kapitole 2. popisují jednotlivé metody a nástroje nejčastěji používané k vývoji těchto aplikací. Zmiňují jejich přednosti a upozorňují na jejich nevýhody. Dále pak zmiňují aplikace, které tyto nástroje využívají a licence, pod kterými jsou vydány.

V další kapitole popisují jednoduchý RAD nástroj, který jsem v rámci této práce implementoval.

Poslední kapitola je shrnutí dnešní situace na tomto poli a můj názor budoucí vývoj.

2 Aktuální situace ve vývoji databázových aplikací v jazyce PHP

Pokud se dnes rozhodneme vyvíjet aplikace v jazyce PHP¹ (<http://www.php.net>), pracující s databází, ve většině případů jde o databázi MySQL², máme několik možností, které si níže popíšeme. První z nich je napsat aplikaci takzvaně „na zelené louce“. Tedy pouze za pomoci jazyka PHP a jeho knihovných funkcí. Druhá z možností je využít některý PHP framework³. Třetí možností je vyvíjet aplikaci za pomoci komplexního vývojového prostředí. Jediný software na trhu, který se snaží být komplexním vývojovým prostředím pro jazyk PHP, je aktuálně CodeGear Delphi for PHP (<http://www.codegear.com/products/delphi/php>).

2.1 Bez pomoci frameworku

Můžeme aplikaci programovat pouze pomocí „čistého“ PHP a jeho standardních knihovných funkcí. Aplikace se tímto způsobem programovali hlavně v dobách PHP 4. Vlastní PHP kód bylo zvykem vkládat přímo mezi HTML tagy. Takto napsané programy je poměrně obtížné spravovat a případně rozšiřovat. Proto už v dobách PHP 4 vznikali projekty, které měli za cíl vytvořit základní knihovny, které usnadní programátorovi vývoj (nejen) webových aplikací. Například PEAR (<http://pear.php.net>), oficiálně podporovaný tvůrci PHP, který se postupně vyvinul ve velice obsáhlý framework. Nebo komunitní repositář PHP knihoven PHP Classes Repository (<http://www.phpclasses.org>).

Při vývoji větších projektů, kde byla vyžadována spolupráce několika pracovníků, zpravidla programátora a HTML kodéra, vznikla potřeba oddělit vlastní aplikační vrstvu od prezentční. Tím se zajistilo, že tvůrce HTML struktury webu mohl na projektu pracovat bez obavy, že poškodí aplikační kód. A naopak nemusel žádat o spolupráci programátora kvůli změnám, které se týkají prezentace dat.

1 PHP (rekurzivní zkratka PHP: Hypertext Preprocessor, původně Personal Home Page) je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek. PHP lze ovšem také použít i k tvorbě konzolových a desktopových aplikací. [1]

2 MySQL je databázový systém, vytvořený švédskou firmou MySQL AB. Jeho hlavními autory jsou Michael „Monty“ Widenius a David Axmark. Komunikace s ním probíhá pomocí jazyka SQL.[2]

3 Framework je softwarová struktura, která slouží jako podpora při programování a vývoji a organizaci jiných softwarových projektů. Může obsahovat podpůrné programy, knihovnu API, návrhové vzory nebo doporučené postupy při vývoji. Cílem frameworku je převzetí typických problémů dané oblasti, čímž se usnadní vývoj tak, aby se návrháři a vývojáři mohli soustředit pouze na své zadání. [3]

Pro oddělení aplikační (logické) a prezenční vrstvy aplikace vznikly šablonovací systémy. Aby byla práce efektivní, mělo by být cílem, aby mohl grafik nebo tvůrce HTML struktury webu pracovat na projektu bez obavy, že poškodí aplikační kód, a naopak aby grafik nemusel žádat o spolupráci programátora kvůli změnám, které se týkají prezentace dat. Tohoto oddělení můžeme dosáhnout, pokud použijeme vhodný šablonovací systém (template engine). [4] Typickým zástupcem šablonovacího systému jsou Smarty (<http://smarty.php.net>). Smarty je šablonovací systém vytvořený za použití skriptovacího jazyka PHP. Dnes se při programování s moderními frameworky většinou šablonovací systémy příliš nevyužívají a má se zato, že samo PHP je kvalitní šablonovací jazyk. Zvláště pak s množstvím dobře napsaných helperů⁴. Ovšem většina moderních frameworků je přizpůsobena k jednoduchému propojení s běžně používanými šablonovacími systémy.

2.2 Použít framework

Další možností je postavit aplikaci na základě vhodného PHP frameworku. Ať už na vlastním nebo na nějakém z podporovaných a vyvíjených komunitou nebo komerčními společnostmi. Ještě dnes, kdy je výběr kvalitních open-source⁵ PHP frameworků dostatečně široký, poměrně dost společností nebo jednotlivců, zabývajících se vývojem webových aplikací v PHP, si píše a udržuje framework svůj vlastní.

O důvodech, proč si nepsat vlastní framework, napsal článek Ing. Václav Stoupa (*Proč si nepsat vlastní framework*, <http://blog.php-group.cz/2008/04/18/proc-si-nepsat-vlastni-framework>). I přes důvody, zmíněné v článku se najdou jedinci, kteří se rozhodli takový vlastní framework vytvořit. Jedním z českých (a Brněnských) zástupců je David Grudl (<http://davidgrudl.com>) se svým frameworkem Nette (<http://nettephp.com>). Cesta, kterou si dnes vybírá stále více vývojářů, je použít některý z už vytvořených open-source PHP frameworků.

4 Helpery jsou sady funkcí, které zjednodušují psaní HTML tagů. Především prvků formulářů.

5 Open source nebo také open-source software (OSS) je počítačový software s otevřeným zdrojovým kódem. Otevřenost zde znamená jak technickou dostupnost kódu, tak legální dostupnost - licenci software, která umožňuje, při dodržení jistých podmínek, uživatelům zdrojový kód využívat, například prohlížet a upravovat. [5]

2.2.1 Přehled frameworků

Díky příchodu PHP verze 5, která značně vylepšila objektový model, podporu Unicode⁶, práce s XML⁷ a díky knihovně PDO přístup k databázím, započal vývoj větších, robustních webových frameworků. PHP tím mohlo směle konkurovat jiným objektově orientovaným interpretovaným jazykům, jako jsou především Python a Ruby. Pro ně existují výkonné webové frameworky, například:

- Python: Django (<http://www.djangoproject.com>)
- Ruby: Ruby on Rails (<http://www.rubyonrails.org>)

Po vzoru frameworku Ruby on Rails, který zaznamenal velký úspěch a zapříčinil raketový vzestup zájmu o tento programovací jazyk, začalo velké množství webových frameworků využívat architekturu Model-view-controller.

2.2.1.1 Model-view-controller

Model-view-controller [8] (dále jen MVC) je často chápán jako návrhový vzor, nicméně se týká architektury aplikací mnohem více, než klasický návrhový vzor. Obecně řečeno, vytváření aplikací s využitím architektury MVC vyžaduje vytvoření tří komponent, mezi které patří:

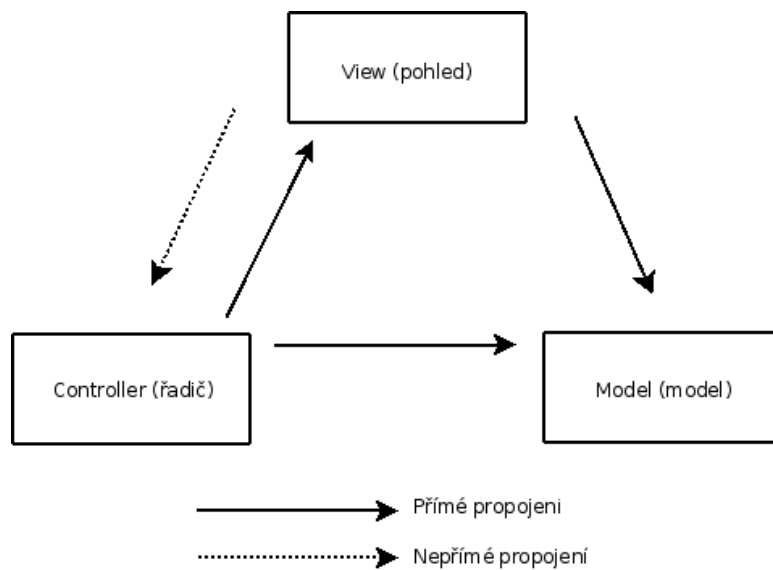
- **Model (model)**, což je doménově specifická reprezentace informací, s nimiž aplikace pracuje.
- **View (pohled)**, který převádí data reprezentovaná modelem do podoby vhodné k interaktivní prezentaci uživateli.
- **Controller (řadič)**, který reaguje na události (typicky pocházející od uživatele) a zajišťuje změny v modelu nebo v pohledu.

Komponenty řadič a pohled jsou ve standardním rozdělení vrstev na prezentační, doménovou a datovou obvykle zařazovány jako prezentační vrstva. V MVC je tato prezentační vrstva rozdělena

⁶ Unicode je původně šestnáctibitová tabulka znaků všech existujících abeced, později rozšířená na 31 bitů. Jeho autorem je Unicode Consortium. Nejpoužívanější variantou je UTF-8, popsáný v ISO 10646-1:2000 Annex D a také v RFC 3629. [6]

⁷ XML (eXtensible Markup Language, česky rozšiřitelný značkovací jazyk) je obecný značkovací jazyk, který byl vyvinut a standardizován konsorciem W3C. Umožňuje snadné vytváření konkrétních značkovacích jazyků pro různé účely a široké spektrum různých typů dat. [7]

mezi komponenty řadič a pohled, nicméně nejdůležitější rozdělení je mezi prezentací a doménovou vrstvou.



Ilustrace 1: Diagram závislostí komponent MVC

Ačkoliv může být koncept MVC realizován různým způsobem, obecně platí tento princip:

1. Uživatel provede nějakou akci v uživatelském rozhraní (např. stiskne tlačítko).
2. Řadič obdrží oznámení o této akci z objektu uživatelského rozhraní.
3. Řadič přistoupí k modelu a v případě potřeby ho zaktualizuje na základě provedené uživatelské akce (např. zaktualizuje nákupní košík uživatele).
4. Model je pouze jiný název pro doménovou vrstvu. Doménová logika zpracuje změněná data (např. přepočítá celkovou cenu, daně a expediční poplatky pro položky v košíku). Některé aplikace užívají mechanismus pro perzistentní uložení dat (např. databázi). To je však otázka vztahu mezi doménovou a datovou vrstvou, která není architekturou MVC pokryta.
5. Komponenta pohled použije zaktualizovaný model pro zobrazení zaktualizovaných dat uživateli (např. vypíše obsah košíku). Komponenta pohled získává data přímo z modelu, zatímco model nepotřebuje žádné informace o komponentě View (je na ní nezávislý). Nicméně je možné použít návrhový vzor pozorovatel, umožňující modelu informovat jakoukoliv komponentu o případných změnách dat. V tom případě se komponenta view zaregistruje u modelu jako příjemce těchto informací. Je důležité

podotknout, že řadič nepředává doménové objekty (model) komponentě pohledu, nicméně jí může poslat příkaz, aby svůj obsah podle modelu zaktualizovala.

6. Uživatelské rozhraní čeká na další akci uživatele, která celý cyklus zahájí znovu.

Některé dnešní frameworky (například PRADO nebo Nette) navíc přidávají možnost pracovat s komponentami a programování řízené událostmi (event-driven). Několik dnes nejpoužívanějších PHP frameworků si představíme.

2.2.1.2 Zend Framework

U nás jedním z nejznámějších a nejpoužívanějších PHP frameworků je Zend Framework (<http://framework.zend.com>), vytvořený a spravovaný firmou Zend (<http://www.zend.com>). Tato firma stojí za vývojem jádra PHP od verze 4. Vývoj frameworku sahá do roku 2005. Koncem léta 2007 byla představena verze 1.0, kterou si ihned oblíbilo mnoho uživatelů. Zend byl od začátku vyvíjen s důrazem na modularitu. Z toho vyplývá možnost použít jakoukoli jeho část (modul, balíček) samostatně. Tím se liší od invazivních frameworků, jako je například Symfony nebo PRADO.

Jednotlivé části Zend frameworku dodržují jasně definované pojmenování. Jak složek a souborů tak vlastních tříd. Toto strukturování se pak využívá ve funkci autoload, která pozná z názvu třídy, v jakém souboru je ta konkrétní třída umístěna. Uživatel pak nemusí při každém použití třídy linkovat její zdrojový soubor. Například funkcemi „require“ nebo „include“ a jejich obdobami. Podobným systémem pro automatické linkování zdrojového souboru používaných tříd, disponuje většina moderních PHP frameworků.

V základní instalaci zprostředkovávají přístup k databázi třídy `Zend_DB` a `Zend_DB_Table`, které implementují návrhový vzor Active record. Zend framework je ale snadno rozšiřitelný o ORM nástroje, jako je například Doctrine (<http://www.phpdoctrine.com>), který se poměrně snadno kombinuje s autoload systémem, který používá Zend.

Mezi další zajímavé vlastnosti frameworku patří integrace API významných webových služeb poskytovatelů jako jsou Google, Yahooo, Akismet, Amazon, Delicious, Flickr a další.

Při návrhu Zend Frameworku byl kladen velký důraz na dodržování návrhových vzorů a možnost uživatelského rozšíření většiny tříd. Jednou z věcí, která je Zend Frameworku vyčítána, je jeho rychlost. Pokud na serveru, kde je hostovaná vaše aplikace, není nainstalován Zend Optimizer (<http://www.zend.com/en/products/guard/optimizer>) nebo eAccelerator, (<http://eaccelerator.net>) bývá aplikace poměrně pomalá.

Základní komponenty

Při použití Zend frameworku má programátor k dispozici následující komponenty. Detailní popis komponent a jejich použití najdete na manuálových stránkách a v dokumentaci (<http://framework.zend.com/manual>).

Model-View-Controller (MVC)

- Zend_Controller, Zend_Controller_Action, Zend_Controller_Dispatcher, Zend_Controller_Plugin, Zend_Controller_RewriteRouter, Zend_View
- Zend_Http_Request, Zend_Http_Response

Databáze

- Zend_Db
- Zend_Db_Table

Internacionalizace (i18n) a lokalizace (l10n)

- Zend_Date
- Zend_Locale
- Zend_Measure
- Zend_Translate

Autentizace, autorizace a správa sezení (session)

- Zend_Acl
- Zend_Authentication
- Zend_Session

Web a webové služby (web services)

- Consuming services: Zend_Feed, Zend_Rest_Client, Zend_Service, Zend_XmlRpc_Client, Zend_Gdata, Zend_Http_Client
- Exposing services: Zend_Http_Server, Zend_Rest_Server, Zend_Server_Documentor, Zend_Server_Reflection, Zend_Soap_Server, Zend_XmlRpc_Server
- Zend_Uri

E-mail, formáty a vyhledávání

- Zend_Json, Zend_Pdf
- Zend_Mail, Zend_Mime
- Zend_Search_Lucene

Infrastruktura jádra

- Zend_Cache, Zend_Config, Zend_Console_Getopt, Zend_Log, Zend_Memory
- Zend_Debug, Zend_Environment, Zend_Loader, Zend_Registry, Zend_Version
- Zend_Filter, Zend_Validate

Začátkem roku vyšla verze 1.5, která přidává například následující komponenty:

- Zend_Layout
- Zend_Form
- Zend_TimeSync

Mezi další novinky verze 1.5 patří lepší práce s technologií AJAX⁸ a podpora nových webových služeb, jako jsou například Remember The Milk, SlideShare, Technorati a Gravatar.

Významné weby, které jsou vytvořeny za použití Zend Frameworku:

(<http://framework.zend.com/whyzf/casestudies>)

- Adooda (<http://adooda.co.uk>)
- bwin Games AB, PokerRoom.com (<http://www.pokerroom.com>)
- IBM Software Group, QEDWiki (<http://services.alphaworks.ibm.com/qedwiki>)
- Indianapolis Motor Speedway (<http://www.indy500.com>)
- Varien, Magento e-commerce (<http://www.magentocommerce.com>)

Zend Framework je vydáván pod BSD licenci (<http://framework.zend.com/license>).

2.2.1.3 Symfony

Autoři představují Symfony framework na webu (<http://www.symfony-project.org>) těmito slovy: „Symfony je kompletní framework navržený tak, aby optimalizoval vývoj webových aplikací pomocí několika klíčových vlastností. Pro začátečníky u webové aplikace odděluje business logiku, serverovou logiku a prezentační pohledy. Obsahuje četné nástroje a třídy, určené pro zkrácení času, potřebného pro vývoj komplexní webové aplikace. Navíc automatizuje obecné úlohy, takže vývojář se může zaměřit pouze na specifika dané aplikace. Tyto výhody v konečném výsledku znamenají, že není potřeba znovu vynalézat kolo pokaždé, když je vytvářena nová webová aplikace!

⁸ AJAX (Asynchronous JavaScript and XML) je obecné označení pro technologie vývoje interaktivních webových aplikací, které mění obsah svých stránek bez nutnosti jejich znovunačítání. Na rozdíl od klasických webových aplikací poskytují uživatelsky příjemnější prostředí, ale vyžadují použití moderních webových prohlížečů. [9]

Symfony bylo celé napsáno v PHP 5. Bylo důkladně otestováno v různých v praxi nasazených projektech a používá se pro e-business u hodně vytěžovaných webových sídel. Je kompatibilní s většinou dostupných databází, včetně MySQL, PostgreSQL, Oracle a Microsoft SQL Serveru. Běží na platformách *nix a Windows.“

Jak autoři zmiňují, framework je napsán a pro svůj běh vyžaduje PHP ve verzi 5 a vyšší. Poměrně zajímavé jsou způsoby distribuce frameworku [10]. Zde následuje jejich stručný přehled:

Instalace Sandboxu

Tento způsob distribuce je určen spíše pro testování a pokud chcete vyzkoušet, co Symfony framework umí. Sandbox je jednoduše archiv souborů. Obsahuje prázdný symfony projekt včetně všech potřebných knihoven (symfony, pake, lime, Creole, Propel a Phing), výchozí aplikaci a základní konfiguraci. Bude ihned fungovat, není potřeba konfigurovat server nebo použít nějaké dodatečné balíčky.

Instalace symfony jako balíčku PEAR

Symfony je také možno získat přes distribuční kanál PEAR. PEAR balíček symfony obsahuje knihovny symfony a všechny jejich závislosti. Také obsahuje skript, který rozšíří vaše CLI přidáním příkazu symfony. Pokud je zveřejněna nová verze Symfony je velmi jednoduché provést tímto způsobem update.

Instalace Symfony balíčku z repozitáře vaší Linuxové distribuce

Pokud používáte jako prostředí pro běh nebo vývoj PHP aplikací některou z distribucí Linuxu. A vaše distribuce disponuje jedním z mnoha balíčkovacích systému pro instalaci software. Je nejjednodušší možností přidat do vašeho systému repozitář symfony a provést instalaci přes něj. Případné aktualizace se vám provádějí automaticky – pokud to máte tak nastaveno.

Získání symfony z SVN repozitáře

Poslední možností jak Symfony získat je stažení aktuální verze z Subversion repozitáře. Tato možnost je zvláště výhodná pro situace, kde není možné použít PEAR.

První verze symfony byla vydána v říjnu 2005 zakladatelem projektu Fabienem Potencierem. Fabienem Potencierem je CEO společnosti Sensio (<http://www.sensio.com>), která za projektem stojí a financuje jeho vývoj.

Symfony je postaven na několika odděleně vyvíjených nástrojích, které Fabien poskládal dohromady a vydal jako plnohodnotný framework. Základem je MVC framework Mojavi

(www.mojavi.org). Další nedílnou součástí (do verze 1.1, od verze 1.1 je „pouze“ jako plugin) je nástroj pro objektově relačním mapování (ORM⁹) Propel (<http://propel.phpdb.org/trac>).

Fabien ještě přidal spoustu šablonovacích „pomocníků“ (helperů) inspirovaných v Ruby on Rails. Tyto pomocníci dělají ze Symfony nesmírně mocný nástroj. Díky nim je například vytvoření Drag-and-Drop aplikace využívající AJAX otázkou několika desítek řádků PHP kódu. Můžete psát aplikace s bohatou JavaScriptovou funkcionalitou bez toho abyste napsali jediný řádek kódu v jazyce Javascript. Za toto vděčí Symfony dobré integraci Javascriptového frameworku Prototype (<http://www.prototypejs.org>), který je výchozí součástí instalace. Pro ty, kteří preferují jiné Javascriptové frameworky, například výborné JQuery (<http://www.jquery.com>), nebo YUI (<http://developer.yahoo.com/yui>) od společnosti Yahoo, jsou připraveny rozšiřující pluginy.

Aplikace, napsané v Symfony frameworku, jsou kompletně konfigurovatelné pomocí YAML¹⁰ (<http://www.yaml.org>) souborů. Tyto soubory jsou kaskádově umístěny v jednotlivých složkách config podle hierarchie aplikace a dědí nastavení z nadřazeného souborů.

Tento systém konfigurace projektu většinu uživatelů začínajících se Symfony, dalo by se říci, skoro děsí. Ale po několika použitích si na práci s YAML soubory většina zvykne a nakonec si tuto možnost konfigurace i velmi chválí.

YAML soubory nezůstávají jen u konfigurace samotné aplikace, jako je připojení do databáze, nastavení i18n¹¹ a l10n¹² nebo třeba nastavení favicon. Symfony přes tyto soubory umožňuje například i validovat formuláře nebo spravovat možnosti přístupu jednotlivých uživatelů na konkrétní části aplikace (autentifikace).

Jak jsem se výše zmínil, Symfony v základní instalaci integruje ORM systém Propel. Pokud by někomu Propel nevyhovoval a dával by přednost jinému systému, například Doctrine, není problém Symfony rozšířit o příslušný plugin.

Symfony implementuje programátorské principy jako jsou KISS¹³, DRY¹⁴ nebo filosofii XP¹⁵. Díky tomu se vývojáři mohou zaměřit přímo na aplikační logiku a neutráčet čas psaním stéle těch stejných věcí dokola.

9 ORM (Object-relational mapping) je programovací technika, která typicky umožňuje mapovat data z relačních databází na objekty. Dosáhne toho vytvořením „virtuální“ objektové databáze, která je používána v objektovém programovacím jazyku. Změny v ní jsou replikovány do „skutečné“ relační databáze.

10 YAML je přímočarý, strojově zpracovatelný formát na serializaci dat, který je navržen jako čitelný pro člověka a schopný interakce se skriptovacími jazyky.

11 i18n (Internationalization) neboli internacionalizace

12 l10n (Localization) neboli lokalizace

13 KISS (Keep It Simple Stupid) neboli „Udržuj to jednoduché, hlupáku“ – psát kód co nejjednodušeji

14 DRY (Don't Repeat Yourself) neboli „Neopakuj sebe sama“

15 XP neboli Extreme Programming je metoda extrémního programování

Symfony je typický enterprise framework. Z čehož vyplývá poměrně dlouhá doba, která je potřeba k proniknutí do jeho vlastností a naučení se s ním zacházet. Proto je vhodný obzvláště na velké projekty, kde vývojáři ocení například i integrovanou podporu jednotkových testů (Unit tests). Pro nasazení nejen na větších projektech je určitě výhodou CLI¹⁶ rozhraní pro příkazovou řádku. Které je vhodné k založení projektu a aplikace, vygenerování modelu podle struktury databáze a další. Za zmínku určitě stojí, že tento nástroj se dá použít například i k vygenerování kompletní administrace k vyvíjené webové stránce. Více o admingenerátoru a screencast s jeho použitím naleznete na <http://www.symfony-project.org/screencast/admin-generator>.

Několik webových aplikací vytvořených za pomoci Symfony:

- Yahoo! Answers (<http://answers.yahoo.com>)
- Delicious (<http://del.icio.us>)
- Yahoo! Bookmarks (<http://bookmarks.yahoo.com>)
- Katedra kybernetiky ZCU (<http://www.kky.zcu.cz/cs>)

Symfony framework je uvolněn jako open-source pod MIT licenci (<http://www.symfony-project.org/content/license>).

2.2.1.4 PRADO

Prado [11] (<http://www.pradosoft.com>) je komponentový framework, který byl jako jeden z prvních PHP frameworků napsán pro kdysi nové PHP 5 a uveřejněn v soutěži Zend's PHP 5 Coding Contest, kde jednoznačně zvítězil jak u odborné veřejnosti, tak u porotců (<http://interval.cz/clanky/Zend-s-php-5-coding-contest-prado-framework>). V současné době je již v třetí verzi (3.1.2) a obsahuje více než 100.000 řádků kódu. Jeho duchovním otcem je Qiang Xue.

Framework je samozřejmě čistě objektový, využívá všechny vymoženosti PHP 5 (minimální vyžadovaná verze je PHP 5.1) a masivně používá XML konfiguraci. V názvech a konceptu tříd se framework zjevně inspiroval v Borland Delphi (nalezneme zde třídy TApplication, TPage, TForm, TComponent, TButton) a svým pojetím pro změnu ASP.NET (komponenty a jejich události, validační mechanismus, třídy TServerValidateEventParameter, TValidator a další).

PRADO tak nabízí zcela nový přístup k vývoji aplikací v PHP, vizuální, komponentový a událostmi řízený přístup, který ale nebude kompatibilní s existujícími systémy a existujícím PHP kódem. Ale je to velice zajímavá alternativa pro vývojáře, kteří jsou zvyklí pracovat například s

¹⁶ CLI (Command line interface) neboli příkazová řádka. Představuje uživatelské rozhraní, ve kterém uživatel s programem nebo operačním systémem komunikuje zapisováním řádků textových příkazů. [12]

ASP.NET. PRADO je velice náročné na paměť, což bývá u PHP aplikací problém. Na toto musí uživatel při vývoji myslet a na jednu stránku umístit jen přiměřený počet komponent.

Hlavní rysy frameworku PRADO jsou následující:

- oddělení prezentace od aplikační logiky
- událostmi řízený přístup
- vysoce použitelné webové komponenty, jako validace vstupu, tvorba průvodce a jiné
- objektově orientovaný a znovu použitelný kód
- přizpůsobitelné a lokalizovatelné hlášení chyb a výjimek
- podpora internacionalizace (i18n) a lokalizace (l10n)
- podpora AJAXu
- konfigurovatelná a modulární architektura
- bezpečnost: prevence cross-site skriptování (XSS), ochrana cookies a další
- bohatá dokumentace a silné uživatelské zázemí

PRADO framework patří také spíše do oblasti vývoje enterprise aplikací. Všeobecně se má zato, že na tvorbu malých aplikací typu firemní webová prezentace, se příliš nehodí.

Webové aplikace využívající PRADO:

- Discollective (<http://www.discollective.com>)
- FleaPit (<http://www.fleapit.com>)

Framework je uvolněn pod BDS licenci (<http://www.pradosoft.com/license>).

2.2.1.5 CakePHP

Začátky CakePHP (<http://www.cakephp.org>) sahají do roku 2005, kdy Michal Tatarynowicz vytvořil minimalistický framework nazvaný Cake. Tento framework vydal pod licenci MIT, čímž položil základ pro dnešní CakePHP framework.

Základní vlastnosti frameworku jsou:

- Model-View-Controller architektura
- pomocníci (helpers) pro HTML, formuláře, stránkování, AJAX, Javascript, XML, RSS a další
- Autentifikace a autentizace
- jednoduchá a přesto obsáhlá validace dat v modelu
- router pro mapování URL

- nástroje pro práci se soubory, složkami, poli a dalšími
- integrovaná podpora databází DB2, Firebird, MSSQL, MySQL (mysql i mysqli), ODBC, Oracle, PostgreSQL, SQLite, Sybase
- PHP 4.3.2 nebo vyšší. (jako jediný mnou představovaný PHP framework funguje i na PHP verze 4)

Model CakePHP frameworku využívá návrhový vzor Active Records. CakePHP se snaží být hlavně jednoduchý a přímočarý. Dodržuje principy DRY a KISS. Je vhodný zejména na menší a středně velké aplikace.

CakePHP framework je vydáván pod MIT licenci (<http://www.opensource.org/licenses/mit-license.php>).

2.2.1.6 Nette

PHP framework Nette (<http://nettephp.com>) zmiňuji spíše pro zajímavost. Nepatří mezi trojici nejpoužívanějších (Zend framework, Symfony, CakePHP). Dokonce se nachází zatím jen v alpha verzi (aktuálně 0.7 alpha). To ale neznamená, že nestojí za pozornost a že nepřináší nové myšlenky a nápady. Spíše naopak. Přístup Nette je v lecčems unikátní.

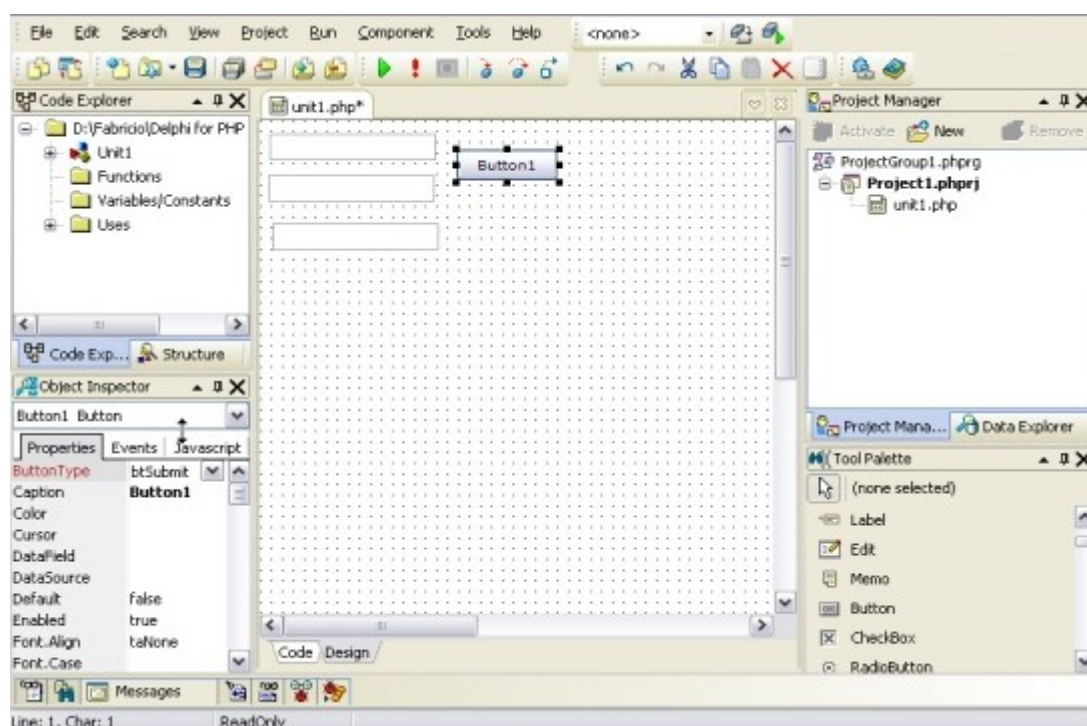
Autorem PHP frameworku Nette je Brňák, David Grudl (<http://davidgrudl.com>), autor open-source nástroje pro převod textů do HTML Taxy! (<http://taxy.info>).

Framework Nette je zajímavý především tím, že kombinuje výhody MVC přístupu s komponentami a programováním řízeným událostmi. MVC architekturu realizuje dvoucestný router (v ostatních PHP frameworkích je běžný jednocestný). Jako databázovou vrstvu využívá nástroj Dibi (<http://dibi.php.com>) od stejného autora. Dibi je k dispozici i samostatně.

Nette je vydáno pod Nette licenci (<http://nettephp.com/cs/license>).

2.3 Komplexní vývojové prostředí

Poslední ze zmiňovaných možností je použití komplexní vývojové prostředí pro PHP. Bohužel dnes je jediným takovým software na trhu Delphi for PHP od firmy CodeGear (<http://www.codegear.com>). Delphi for PHP je RAD nástroj pro tvorbu interaktivních webových aplikací s možností vizuálního vývoje metodou Drag-and-Drop (táhni-a-pusť) a výkonnou sadou komponent a šablon v jazyce PHP, postavený na knihovně vizuálních komponent VLC for PHP (<http://www.qadram.com/vcl4php>)



Ilustrace 2: Uživatelské rozhraní Delphi for PHP

Cílem Delphi for PHP bylo usnadnit a zrychlit vývoj webových aplikací. Bohužel první verzi tohoto nástroje se to příliš nepovedlo. Pokud chtěl vývojář pracovat s Delphi for PHP, musel projít poměrně složitou registrací, která ne vždy fungovala tak, jak měla. Pokud se Vám Delphi for PHP podaří nainstalovat a spustit, ještě nemáte vyhráno. Je poměrně pravděpodobné, že se Vám žádnou aplikaci nepodaří zkompilovat a spustit, alespoň takové jsou moje zkušenosti. Uživatelé, popisující svoji práci s Delphi for PHP na internetu, většinou docházejí k podobným závěrům. Namátkou zmíním recenzi od Davida Grudla (*Smějeme se s Delphi for PHP*, <http://latrine.dgx.cz/smejeme-se-s-delphi-for-php>).

Ve dnech psaní této práce vychází druhá verze, tedy Delphi for PHP 2.0. Autoři slibují opravu většiny chyb předchozí verze, vylepšený debugger a samozřejmě se zaměřují na ještě větší usnadnění

vytváření webových aplikací. Za zmínku stojí, že v nové verzi by mělo jít používat komponent ze Zend frameworku.

Delphi for PHP je komerční produkt.

3 Návrh a implementace jednoduchého RAD nástroje.

Zadáním bylo navrhnout a implementovat (v rozsahu prototypu) jednoduchý RAD nástroj s grafickým uživatelským rozhraním. Jako platformu pro tento nástroj a jeho grafické uživatelské rozhraní jsem vybral jazyk Java a sadu grafických komponent Swing. Mezi kandidáty byl i jazyk C# s frameworkem .NET. Platforma Java dostala nakonec přednost pro svoji vlastnost, že program vytvořený v tomto jazyce bez jakýchkoliv úprav funguje jak na operačních systémech společnosti Microsoft, tak na systémech unixového typu. Tato vlastnost je poměrně důležitá, protože jazyk PHP je oblíbený mezi uživateli Linuxu.

3.1 Výběr PHP frameworku

Aby PHP kód generovaný RAD nástrojem byl snadno rozšiřitelný a udržovatelný, rozhodl jsem se generovat aplikaci, která bude využívat některý z představených PHP frameworků.

Základní kritéria pro výběr byla:

- Neinvazivnost – pro zprovoznění aplikace se nemusí nic instalovat, stačí nakopírovat vygenerovaný projekt složky přístupné z webu.
- Modulárnost, snadná rozšiřitelnost
- Nástroj pro generování formulářů

Tyto kritéria vyloučila frameworky Symfony a PRADO. Tyto frameworky jsou invazivní. Samostatně se nedá použít jen jejich část. A k tomuto účelu jsou zbytečně „mohutné“.

Zbývají tedy CakePHP a Zend framework. CakePHP neobsahuje žádný vyspělý nástroj pro vytváření a validaci formulářů. Zend představil ve verzi 1.5 třídu `Zend_Form`, která se chová přesně tak, jak potřebuji. Zend klade velký důraz na striktní dodržování standardů a návrhových vzorů a je snadno rozšiřitelný. Tyto důvody udělaly ze Zend frameworku vítěze. A na jeho základech jsem implementoval kostru aplikace, kterou generuje můj jednoduchý RAD nástroj.

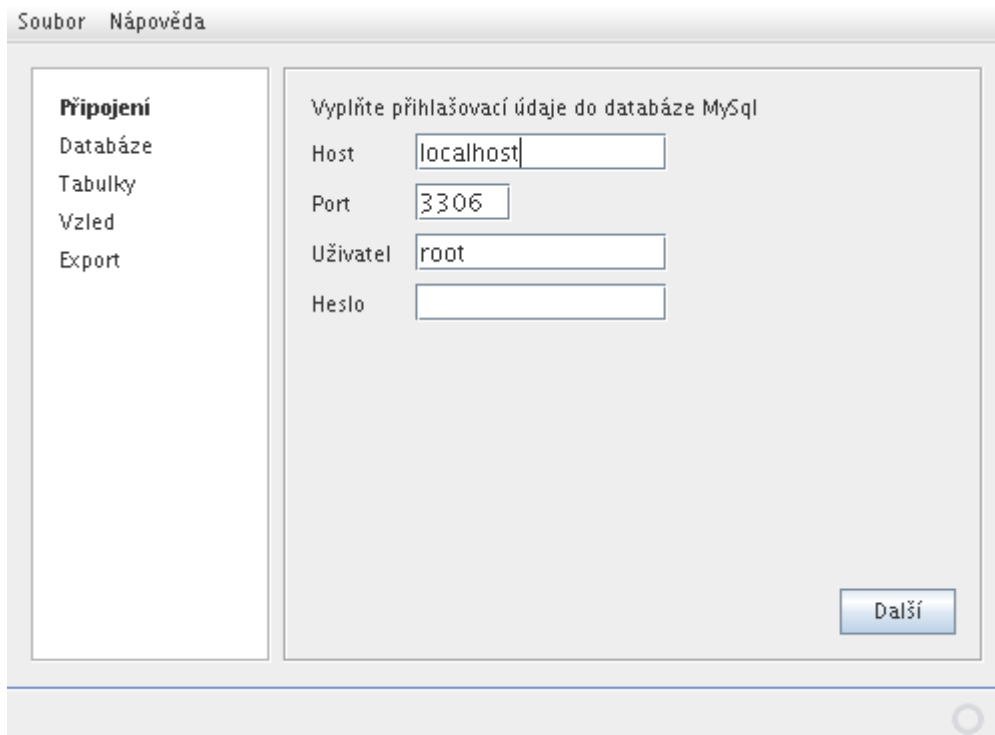
3.2 Vlastní aplikace napsaná v jazyce Java

Mnou vytvořený jednoduchý RAD nástroj umožňuje vygenerovat webovou administraci k jednotlivým tabulkám z databáze MySQL. Je napsán v jazyce Java (JDK verze 6) za použití

frameworku grafických komponent Swing. K vytvoření této aplikace jsem použil vývojové prostředí Netbeans IDE 6.

Popis kroků, které vedou k vygenerování PHP aplikace:

1. Uživatel vyplní údaje pro připojení k MySQL serveru a aplikace se připojí.
2. Aplikace vypíše všechny databáze na serveru, ke kterým má uživatel přístup a nabídne uživateli, aby některou z nich vybral.
3. Uživatel vybere databázi a v ní tabulky, ke kterým chce vygenerovat administraci.
4. Aplikace ke každé tabulce vypíše všechny její sloupce a nabídne uživateli, aby si vybral, které sloupce chce mít v základním přehledu a jak se mají jmenovat. Uživatel vyplní názvy a zaškrtně požadované sloupce a pokračuje.
5. Aplikace nabídne uživateli motiv vzhledu (aktuálně může uživatel vybírat pouze z jednoho, aplikace je přece jen prototyp)
6. Na poslední obrazovce aplikace uživatel stiskne tlačítko „Exportovat“ a vybere místo na disku, kam se má PHP aplikace vygenerovat.
7. Aplikace vygeneruje PHP kód a uloží ho na vybrané místo na disku.

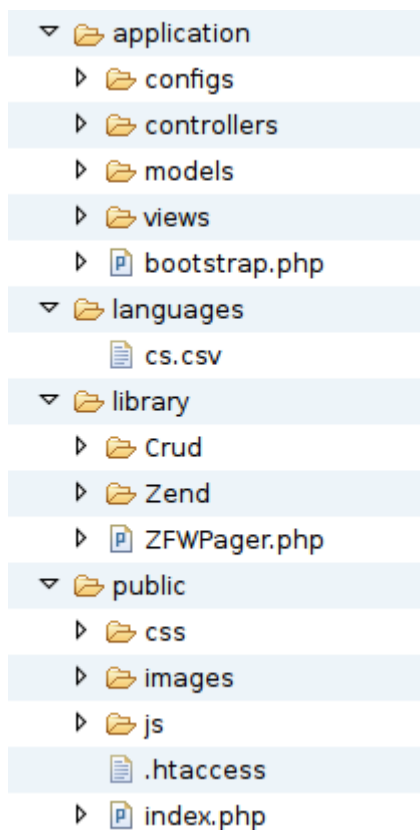


Ilustrace 3: Grafické rozhraní implementovaného RAD nástroje

Zajímavá věc na implementaci je, že připojení k databázovému serveru a generování samotné PHP aplikace je řešeno v jiném programovém vlákně, než v kterém běží samotná aplikace.

3.3 Vygenerovaná PHP aplikace

Jak jsem se výše zmínil, aplikace je postavená na Zend frameworku. Její souborová struktura je taková, jak doporučuje Zend. Je znázorněna na následujícím obrázku.



Ilustrace 4: souborová struktura PHP aplikace

Složka `public` je kořenový adresář webu a měla by být jediná přístupná z internetu. Pokud Váš hosting neumožňuje nastavit kořenovou složku webu, omezte alespoň přístup do ostatních složek pomocí souboru `.htaccess`.

Konfigurační soubory aplikace se nachází ve složce `application/configs` a jsou ve formátu INI¹⁷. Tato složka obsahuje vždy minimálně dva soubory. A to `db.ini` kde jsou přístupové hodnoty

¹⁷ INI formát je textový formát pro konfigurační soubory, znám například z operačních systémů Microsoft.

k MySQL serveru. Dále pak minimálně jeden soubor `nazevtabulkyTable.ini`, ve kterém jsou uloženy názvy jednotlivých sloupců a data pro vygenerování a validaci formuláře třídou `Zend_Form`.

Pokud bychom chtěli přidat další funkcionalitu, můžeme editovat soubory ve složce `application/controllers`, kde jsou jednotlivé `controllery` s názvy `nazevtabulkyController.php`. Tyto `controllery` jsou potomci třídy `Crud_Controller_Action_Abstract`, která dědí funkcionalitu ze `Zend_Controller_Action_Abstract`. Informace o syntaxi `Zend controllerů` získáte na manuálových stránkách (<http://framework.zend.com/manual/en/zend.controller.html>).

Aplikace umožňuje stránkování a řazení podle jednotlivých sloupců. Validuje vstup podle typu sloupce tabulky v databázi.



Students Učitelé

Přehled
Nový záznam

Záznam byl uložen.

Studenti - Přehled

| Jméno | Příjmení | Věk | Upravit | Smazat |
|-------|----------|-----|---------|--------|
| Jan | Vavřín | 14 | Upravit | Smazat |
| Jan | Neřpor | 27 | Upravit | Smazat |
| Josef | Nový | 25 | Upravit | Smazat |
| Karel | Löhnla | 17 | Upravit | Smazat |
| Matěj | Fuchs | 21 | Upravit | Smazat |

Jméno Příjmení Věk Upravit Smazat

1 / 3 5 Záznamů na stránku (11 celkem)

Ilustrace 5: grafické rozhraní vygenerované PHP aplikace

4 Závěr

RAD nástroje pro jazyk PHP dnes existují hlavně ve formě frameworků. Tyto frameworky jsou většinou založeny na architektuře MVC. Jedinou výjimkou je interaktivní vývojové prostředí Delphi for PHP, které je ale u odborné veřejnosti považováno za takzvaný propadák.

To mě utvrzuje v názoru, že interaktivní (s grafickým rozhraním, naklikání vzhledu webové aplikace metodou Drag-and-Drop) vývojová prostředí nejsou ta správná cesta. Tato prostředí vidím použitelná jen pro malé systémy, které se nebudou příliš rozšiřovat a měnit a u kterých příliš nezáleží například na vzhledu. Moje dosavadní zkušenosti jsou takové, že vytvořit rozsáhlejší webovou stránku, která se bude zobrazovat ve všech prohlížečích na všech platformách korektně. Bývá v nějakém WYSIWYG¹⁸ editoru obtížnější, nežli napsat tu samou stránku ručně v HTML. Toto platí dvojnásob, pokud do stránky přidáme Javascriptovou funkcionalitu.

Ideální nástroj si představuji jako kombinaci výkonného frameworku, textového editoru se zvýrazněním syntaxe a doplňováním kódu a sadou CLI nástrojů pro generování částí kódu. Pokud by se některému z vývojových týmů povedlo tyto části dobře propojit, měl by výsledný produkt, podle mého názoru, na trhu úspěch.

Tomuto ideálu se dnes blíží kombinace Zend framework + editor Zend Studio. Nebo framework Symfony + editor Eclipse s PDT pluginem.

Mnou implementovaný RAD nástroj nemá ambice konkurovat výše uvedeným programům. Je to spíše pomocník, který dovede rychle vygenerovat jednoduchou webovou správu databáze s přívětivým grafickým rozhraním.

Do budoucna je možné ho rozšířit o další motivy vzhledu. Vhodné by bylo i implementovat správu relací mezi tabulkami.

18 WYSIWYG je akronym anglické věty „What you see is what you get“, česky „co vidíš, to dostaneš“. Tato zkratka označuje způsob editace dokumentů v počítači, při kterém je verze zobrazená na obrazovce vzhledově totožná s výslednou verzí dokumentu. [13]

Literatura

- [1] Wikipedie: Otevřená encyklopedie: PHP [online]. c2008 [citováno 12. 05. 2008]. Dostupný z WWW: <<http://cs.wikipedia.org/w/index.php?title=PHP&oldid=2517587>>
- [2] Wikipedie: Otevřená encyklopedie: MySQL [online]. c2008 [citováno 12. 05. 2008]. Dostupný z WWW: <<http://cs.wikipedia.org/w/index.php?title=MySQL&oldid=2204581>>
- [3] Wikipedie: Otevřená encyklopedie: Framework [online]. c2008 [citováno 12. 05. 2008]. Dostupný z WWW: <<http://cs.wikipedia.org/w/index.php?title=Framework&oldid=2124221>>
- [4] Tomáš Kavalek: Smarty, šablonovací systém - 1 (úvod) [online]. [citováno 10. 05. 2008]. Dostupný z WWW: <<http://www.abclinuxu.cz/clanky/navody/smarty-sablonovaci-system-1-uvod>>
- [5] Wikipedie: Otevřená encyklopedie: Open source software [online]. c2008 [citováno 12. 05. 2008]. Dostupný z WWW: <http://cs.wikipedia.org/w/index.php?title=Open_source_software&oldid=2551392>
- [6] Wikipedie: Otevřená encyklopedie: Unicode [online]. c2008 [citováno 12. 05. 2008]. Dostupný z WWW: <<http://cs.wikipedia.org/w/index.php?title=Unicode&oldid=2561342>>
- [7] Wikipedie: Otevřená encyklopedie: Extensible Markup Language [online]. c2008 [citováno 12. 05. 2008]. Dostupný z WWW: <http://cs.wikipedia.org/w/index.php?title=Extensible_Markup_Language&oldid=2561278>
- [8] Wikipedie: Otevřená encyklopedie: Model-view-controller [online]. c2008 [citováno 10. 05. 2008]. Dostupný z WWW: <<http://cs.wikipedia.org/w/index.php?title=Model-view-controller&oldid=2506429>>
- [9] Wikipedie: Otevřená encyklopedie: Asynchronous JavaScript and XML [online]. c2008 [citováno 12. 05. 2008]. Dostupný z WWW: <http://cs.wikipedia.org/w/index.php?title=Asynchronous_JavaScript_and_XML&oldid=2438827>
- [10] Symfony: The Definitive Guide to symfony [online]. [citováno 10. 05. 2008]. Dostupný z WWW: <http://www.symfony-project.org/book/1_0>
- [11] Aichi: PHP framework PRADO [online]. C2008 [citováno 12. 05. 2008] Dostupný z WWW: <<http://www.czechdesign.cz/blogs/aichi/php-framework-prado>>
- [12] Wikipedie: Otevřená encyklopedie: Příkazový řádek [online]. c2008 [citováno 12. 05. 2008]. Dostupný z WWW: <http://cs.wikipedia.org/w/index.php?title=P%C5%99%C3%ADkazov%C3%BD_%C5%99%C3%A1dek&oldid=2531331>
- [13] Wikipedie: Otevřená encyklopedie: WYSIWYG [online]. c2008 [citováno 12. 05. 2008]. Dostupný z WWW: <<http://cs.wikipedia.org/w/index.php?title=WYSIWYG&oldid=2312450>>

Seznam příloh

Příloha 1. CD/DVD ...