

## Technologie pro tvorbu webových aplikací - 3. díl (PHP, PERL, ASP.NET)

Vávra David · Informačné technológie

01.06.2012



Tento seriál je rozdelen na tri články, ktoré spoločne poskytujú pomerne široký náhľad do možností a technológií pro tvorbu webových aplikací. Z technológií byly vybrány asi nejčastěji používané - (X)HTML, CSS, JavaScript a PHP. Pomocí těchto technologií je dnes možné vytvářet velmi kvalitní dynamické webové aplikace. Vše začíná obecným rozdělením tvorby těchto aplikací.

První díl a druhý díl nesoucí podtitulky „rozdělení, HTML, XHTML,“ a „CSS, JavaScript, šablonovací systémy“ se zabývají současnými technologiemi používanými pro vývoj webových aplikací zpracovávanými na straně klienta. Poslední třetí díl s podtitulkem „PHP, PERL, ASP.NET“ se pak zabývá částmi, které jsou zpracovávány na straně serveru.

### Úvod

Mezi skriptovací jazyky prováděné na straně serveru patří například PHP a ASP. PHP je dnes nejrozšířenějším skriptovacím jazykem pro webové aplikace. Ve spojení s databází, např. MySQL, se jedná o velmi mocný nástroj programátorů webových aplikací.

PHP je skriptovací programovací jazyk nezávislý na platformě, určený především pro programování dynamických internetových stránek. PHP podporuje téměř každý hosting a to včetně mnoha freehostingů.

I Perl je interpretovaný programovací, který se s rozvojem internetu stal velmi populárním nástrojem pro tvorbu CGI skriptů (v praxi používaný zejména na WWW serverech). Umožňuje psát krátké programy jednoduše a rychle a přitom nebrání v psaní těch složitých. ASP.NET je součástí .NET Frameworku pro tvorbu webových aplikací. Ačkoliv je nástupcem technologie ASP, obě technologie jsou velmi odlišné. Koncept ASP.NET WebForms ulehčuje programátorům přechod od programování klasických aplikací pro Windows do prostředí webu.

### 1. PHP

PHP (rekurzivní zkratka PHP: Hypertext Preprocessor, „PHP: Hypertextový preprocesor“, původně Personal Home Page) je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek. Nejčastěji se

začleňuje přímo do struktury jazyka HTML, XHTML či WML, což lze využít při tvorbě webových aplikací. PHP lze použít i k tvorbě konzolových a desktopových aplikací.

PHP skripty jsou prováděny na straně serveru, k uživateli je přenášen až výsledek jejich činnosti (interpret PHP skriptu je možné volat pomocí příkazové řádky). Syntaxe jazyka je inspirována několika programovacími jazyky (Perl, C, Pascal a Java). PHP je nezávislý na platformě, skripty fungují bez větších úprav na mnoha různých operačních systémech. Podporuje mnoho knihoven pro různé účely - např. zpracování textu, grafiky, práci se soubory, přístup k většině databázových systémů (mj. MySQL, ODBC, Oracle, PostgreSQL, MSSQL), podporu celé řady internetových protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, LDAP...) [1],[2],[3]

Vzhledem k tomu, že téměř každý hosting podporuje PHP, včetně mnoha freehostingů, je volba tohoto jazyka skvělým řešením pro malé až středně velké webové aplikace. Navíc samotné PHP, i díky mnoha knihovnám usnadňujícím práci, umožňuje vytvořit velkou většinu běžných webových aplikací.

## 2. Výhody PHP

- PHP je specializované na webové stránky.
- Rozsáhlý soubor funkcí v základní knihovně PHP, další funkce v PECL.
- Nativní podpora mnoha databázových systémů.
- Multiplatformost (zejména Linux a Microsoft Windows).
- Lze využít nativní funkce operačního systému (možná nekompatibilita s jiným OS).
- Strmá křivka učení.
- Obrovská podpora na hostingových službách - PHP je fakticky standardem.
- Obrovské množství projektů a kódů, které lze zdarma stáhnout.
- Poměrně slušná dokumentace.
- Velmi svobodná licence - svobodnější než GPL.

## 3. Nevýhody PHP

- Mírně nekonzistentní vývoj v minulosti, který si s sebou PHP nese dosud (někdy často měnící se příkazy či jejich parametry atd.).
  - Nejednotné pojmenování funkcí, např. strpos, strchr, ale str\_replace, str\_pad.
  - Nejednotné pořadí parametrů.
  - Ač jazyk podporuje výjimky, jeho knihovna je používá jen zřídka.
  - Slabší podpora Unicode, pouze přes PHP knihovnu (v PHP 6 má být Unicode řetězec jako základní typ).
  - Ve standardní distribuci chybí ladící (debugovací) nástroj.
  - Po zpracování požadavku neudrží kontext aplikace, vytváří jej vždy znovu.
  - Některá bezpečnostní opatření jako safe\_mode, open\_basedir jsou implementována ne příliš elegantně (pro PHP6 se plánuje jejich zrušení).
- [4]

## 4. Historie PHP

### PHP/FI

PHP je nástupcem staršího produktu, nazvaného PHP/FI. PHP/FI vytvořil Rasmus

Lerdorf v roce 1995, na počátku jako jednoduchou sadu skriptů v jazyce Perl pro zpracování záznamů o přístupech k jeho webu. Tuto sadu nazval 'Personal Home Page Tools'. Protože byla třeba větší funkčnost, napsal Rasmus mnohem rozsáhlejší implementaci v C, která byla schopna komunikovat s databázemi a umožňovala uživatelům vyvíjet jednoduché dynamické aplikace pro Web. Rasmus se rozhodl uvolnit zdrojový kód PHP/FI pro všechny, takže kdokoli ho může používat, stejně jako opravovat chyby a vylepšovat kód.

PHP/FI, což znamená Personal Home Page / Forms Interpreter, obsahovalo něco ze základní funkcionality PHP, jak ho známe dnes. Mělo proměnné perlovského typu, automatickou interpretaci formulářových proměnných a syntaxi vloženou do HTML. Syntaxe samotná byla podobná jazyku Perl, přestože mnohem omezenější, jednodušší a v něčem nekonzistentní. V roce 1997 se PHP/FI 2.0, druhá implementace psaná v C, stala kultovní záležitostí pro (odhadem) tisíce uživatelů po celém světě, a s přibližně 50.000 doménami oznamujícími nainstalované PHP/FI, což čítalo zhruba 1 % všech domén na Internetu. I když do projektu začalo svými kusy kódu přispívat více lidí, stále to byl velký projekt jednoho muže.

### **PHP 3**

PHP 3.0 byla první verze, která se velmi blížila takovému PHP, jak ho známe dnes. Vytvořili ho Andi Gutmans a Zeev Suraski v roce 1997 jako kompletně přepsaný celek, poté co shledali PHP/FI 2.0 výrazně "poddimenzované" pro vývoj svých aplikací pro e-komerci. Ve snaze spolupracovat a zahájit budování nad existující uživatelskou základnou PHP/FI, rozhodli se Andi, Rasmus a Zeev pracovat společně a prohlásit PHP 3.0 za oficiálního nástupce PHP/FI 2.0, a vývoj PHP/FI 2.0 byl v podstatě zastaven.

Jednou z nejsilnějších zbraní PHP 3.0 byly jeho obrovské možnosti rozšíření. K poskytnutí pevné infrastruktury pro mnoho různých databází, protokolů a API koncovým uživatelům, přilákaly možnosti rozšíření PHP 3.0 také tucty vývojářů, kteří se připojili a vytvořili nové rozšiřující moduly. Toto byl nesporně klíč k obrovskému úspěchu PHP 3.0. Jiným klíčovým prvkem v PHP 3.0 byla podpora objektově orientované syntaxe a mnohem silnější a konzistentnější syntaxe jazyka.

Nový jazyk byl uvolněn pod novým názvem, který odstranil implikaci omezeného osobního použití, kterou neslo označení PHP/FI 2.0. Byl nazván pouze 'PHP', což je rekurzivní akronym - PHP: Hypertext Preprocessor. Na konci roku 1998 vyrostlo PHP do rozsahu instalací v řádu (odhadem) desítek tisíc uživatelů a stovek tisíc Webů. V době svého vrcholu bylo PHP 3.0 instalováno na přibližně 10 % všech WWW serverů na Internetu. PHP 3.0 bylo oficiálně uvolněno v červnu 1998, poté co strávilo cca 9 měsíců ve veřejném testování.

### **PHP 4**

V zimě 1998, krátce po oficiálním uvolnění PHP 3.0, začali Andi Gutmans a Zeev Suraski pracovat na přespání jádra PHP. Cílem návrhu bylo zvýšit výkon pro složité aplikace a zlepšit modularitu kódové báze PHP. Takové aplikace byly schopny pracovat s PHP 3.0 (díky novým možnostem a podpoře široké škály databází a API od jiných tvůrců), ale PHP 3.0 nebylo navrženo pro efektivní práci tak náročných aplikací.

Nový engine, nazvaný 'Zend Engine' (sestaven z jejich křestních jmen, Zeev a Andi), úspěšně splnil cíle návrhu a byl uveden v polovině roku 1999. PHP 4.0, založené na tomto enginu a doplněné širokou škálou nových prvků, bylo oficiálně uvolněno v květnu 2000, necelé dva roky po svém předchůdci, PHP 3.0. K podstatně zvýšenému výkonu této verze, přidává PHP 4.0 další klíčové prvky, jako je podpora pro mnoho WWW serverů, HTTP sessions, buffering výstupu, bezpečnější způsoby zpracování vstupů uživatele a mnoho nových jazykových konstruktů.

Dnes používají PHP (odhadem) stovky tisíc vývojářů a nainstalované PHP hlásí několik milionů serverů - tj. přes 20 % domén na Internetu. Vývojový tým PHP zahrnuje tucty vývojářů, stejně tak jako tucty dalších lidí, kteří pracují na projektech spojených s PHP, jako je PEAR a dokumentační projekt.

## PHP 5

Bylo vydáno v červenci 2004 po dlouhém vývoji a řadě předverzí. Je řízeno hlavně jádrem, Zend Engine 2.0, s novým objektovým modelem a tucty nových vlastností. [3],[5]

## 5. PERL

Perl je interpretovaný programovací jazyk vytvořený Larry Wallem v roce 1987. S rozvojem internetu se Perl stal velmi populárním nástrojem pro tvorbu CGI skriptů (v praxi používaný zejména na WWW serverech). Perl zahájil svou éru jako skriptovací jazyk, náhrada jazyka AWK a interpretru SH. Největšího rozšíření dosáhl ve verzi 4 z roku 1991. Verze 5 přinesla četná vylepšení, především výkonné datové struktury a možnost objektového programování. V poslední době získal Perl oblibu mimo jiné v bioinformatice.

Umožňuje psát krátké programy jednoduše a rychle a přitom nebrání v psaní těch složitých. Jeden ze způsobů je přitom obvykle velmi stručný, takže Perl získal nezaslouženou pověst jazyka, ve kterém se tvoří nesrozumitelný a neudržovatelný kód. Tato kritika ale není oprávněná, Perl je vhodný k řešení malých i velkých problémů. Schopnosti a nástroje, které se používají u velkých projektů, lze použít i v krátkých skriptech.

## 6. Výhody PERL

- Interpretovaný jazyk - rychlý vývoj bez nutnosti kompilace a linkování.
- Přes 18 000 volně dostupných modulů třetích stran v CPAN (Comprehensive Perl Archive Network).
- Pojmenování, kategorizace, dokumentace, testování a instalace modulů jsou standardizovány a zpřístupňují prakticky veškerá dostupná rozhraní a knihovny.
- Efektivita programování.
- Automatická práce s pamětí (není třeba explicitně alokovat a uvolňovat paměť).
- Pokročilé datové typy např. asociativní pole neboli hash.
- Svobodný software, licencován pod Artistic License či GNU.
- Snadné spojování již hotových komponent (modulů).
- Reference na statické, dynamické i anonymní datové struktury.
- Umožňuje procedurální, funkcionální i objektově orientované programování.

- Snadná práce s textem a značkovacími jazyky (XML, HTML...).
- Perl podporuje znakovou sadu Unicode a je (byl) Y2K kompatibilní.
- Stabilita – mnoho let vyvíjený programovací jazyk.
- Perl umí zacházet se zakódovanými webovými daty, mezi něž patří např. transakce u elektronického obchodování.
- Perl může být součástí web serverů, čímž může dojít ke zrychlení až o 2 000 %.
- Modul umožňuje web serveru Apache vložení Perlu s výhodami, jako u PHP.
- Rozsáhlá dokumentace a literatura, komunita kolem Perlu, konference, ...

## 7. Nevýhody PERL

- Nedisiplinovaný programátor může velmi snadno vytvářet nesrozumitelný kód.
- Při některých aplikacích se může projevit neefektivnost dynamicky typovaného jazyka ve srovnání se staticky typovanými jazyky, zejména spotřeba paměti.
- Kruhové odkazy a problematika jejich destrukce.
- Mnozí tvrdí, že je to jazyk nevhodný pro výuku programování. [6]

## 8. ASP.NET

ASP.NET je součástí .NET Frameworku pro tvorbu webových aplikací. Je nástupcem technologie ASP (Active Server Pages) a přímým konkurentem JSP (Java Server Pages). Ačkoliv název ASP.NET je odvozen od starší technologie pro vývoj webů ASP, obě technologie jsou velmi odlišné. ASP.NET je založen na CLR (Common Language Runtime), který je sdílen všemi aplikacemi postavenými na .NET Frameworku. Programátoři tak mohou realizovat své projekty v jakémkoliv jazyce podporujícím CLR, např. Visual Basic.NET, JScript.NET, C#, Managed C++, ale i mutace Perlu, Pythonu a další. Aplikace založené na ASP.NET jsou také rychlejší, neboť jsou předkompilovány do jednoho či několika málo DLL souborů, na rozdíl od ryze skriptovacích jazyků, kde jsou stránky při každém přístupu znovu a znovu parsovány.

Koncept ASP.NET WebForms ulehčuje programátorům přechod od programování klasických aplikací pro Windows do prostředí webu: stránky jsou poskládány z objektů, ovládacích prvků (Controls), které jsou protějškem ovládacích prvků ve Windows. Při tvorbě webových stránek je možné používat ovládací prvky jako tlačítko (Button), nápis (Label) a další. Těmto prvkům lze přiřazovat určité vlastnosti, zachytávat na nich události, atd. Tak, jako se ovládací prvky pro Windows samy kreslí do formulářů na obrazovku, webové ovládací prvky produkují HTML kód, který tvoří část výsledné stránky. ASP.NET MVC je další oficiální framework postavený na technologii ASP.NET. Tento framework umožňuje snadněji vyvíjet aplikace podle architektury Model-View-Controller.

## 9. Výhody ASP.NET

- Vyšší rychlost aplikace (díky kompilovanému kódu).
- Jednodušší odladění chyb při samotném vývoji.
- Uživatelsky definované ovládací prvky lze použít jako šablony (redukce kódu).
- Velké množství ovládacích prvků a knihoven tříd (rychlejší vývoj aplikací).
- Možnost využít mnoha programovacích jazyků.
- Schopnost cachovat celou stránku nebo její část.

- Od verze 2 generuje ASP.NET validní HTML 4.0 / XHTML 1.0/1.1 a JavaScript.

## 10. Nevýhody ASP.NET

- Špatná rozšířenost mezi hostingy a vysoká cena za hostování.
- Uzavřený kód ve vlastnictví Microsoftu.
- Špatná, případně žádná podpora pod jiným operačním systémem než MS Windows.
- Session jednotlivých verzí není kompatibilní.
- Vysoká závislost na JavaScriptu.

## 11. ASP.NET WebForms - stavové prostředí, bezstavový protokol

Ačkoliv webový protokol HTTP je sám o sobě bezstavový, událostmi řízené programování zachování stavu (uchování kontextu mezi jednotlivými požadavky) vyžaduje. ASP.NET tento problém řeší kombinací HTML a JavaScriptu pomocí dvou základních technik. ViewState uchovává informace mezi postbacky (opakovaným odesláním formuláře na server) v zakódovaném tvaru ve skrytých formulářových polích. Jeho výhodou je, že využívá pouze HTML a nevyžaduje žádnou speciální podporu na straně serveru ani klienta. Nevýhodou je, že se mezi serverem a klientem přenáší větší objem dat, zejména je-li ViewState využíváno nesprávně.

Session State oproti tomu ukládá veškeré informace na straně serveru a předává (typicky jako cookie nebo součást URL) pouze jednoznačný identifikátor. To sice zmenšuje objem přenášených dat, ale klade vyšší nároky na výkon serveru. Pokud se sessions používají nesprávně, může být server náchylný i k Denial of Service útokům. Oproti ASP umožňuje ASP.NET ukládání session state do samostatného procesu nebo na SQL server. To zjednodušuje použití session ve webových farmách, zvyšuje výkon a umožňuje stav zachovat i při restartu serveru.

## 12. ASP.NET MVC

Na přelomu roku 2007 a 2008 ohlásila firma Microsoft plán na vývoj ASP.NET MVC frameworku. Tento framework umožňuje tvorbu webových aplikací podle softwarové architektury Model-View-Controller. ASP.NET MVC má představovat alternativu oproti WebForms. Narozdíl od WebForms aplikace vytvořené pomocí ASP.MVC nevyžadují ViewState a dají se snadněji testovat. V současné době se ASP.NET MVC nachází ve verzi 3 Beta (5. února 2011). Microsoft ujistil komunitu, že vydáním ASP.NET MVC nekončí vývoj WebForms. [7]

## Zdroje

1. Wikipedie: otevřená encyklopedie [online]. [cit. 2009-04-18]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Php>
2. Kolektiv autorů. PHP5, MySQL, Apache - vytváříme webové aplikace. Computer Press, 2006, ISBN 80-251-1073-7.
3. VÁVRA, David. WWW prezentace a návrh e-obchodu soukromého subjektu - živnosti [bakalářská práce]. FAI UTB Zlín, 2009.
4. Wikipedie: otevřená encyklopedie [online]. [cit. 2012-06-26]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Php>
5. PHP: Historie PHP [online]. [cit. 2012-06-26]. Dostupný z WWW:

---

<http://www.php.net/manual/cs/history.php>

6. Wikipedie: otevřená encyklopedie [online]. [cit. 2012-06-26]. Dostupný z WWW:  
<http://cs.wikipedia.org/wiki/Perl>
  7. Wikipedie: otevřená encyklopedie [online]. [cit. 2012-06-26]. Dostupný z WWW:  
<http://cs.wikipedia.org/wiki/ASP.NET>
  8. VÁVRA, David. Tvorba e-obchodu a vývoj linuxového skriptu pro automatické generování nových e-obchodů [diplomová práce]. FAI UTB Zlín, 2011.
-