Středoškolská technika 2013

Setkání a prezentace prací středoškolských studentů na ČVUT

AJAXIZOVANÝ HUDEBNÍ PORTÁL METALISTA.CZ

David Holada

Gymnázium Vlašim

Tylova 271, Vlašim

METALISTA

Ajaxizovaný hudební portál metalista.cz

Středoškolská odborná činnost

Obor: 18. Informatika

Autor: David Holada (4. ročník SŠ)

Škola: Gymnázium Vlašim, Tylova 271, 258 01 Vlašim

Kraj: Středočeský kraj

Hulice 2013

Prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně, použil jsem pouze podklady (literaturu, SW atd.) uvedené v přiloženém seznamu a postup při zpracování a dalším nakládání s prací je v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V Hulicích dne 17. 5. 2013 David Holada .....

Poděkování

Chtěl bych poděkovat diskusi serveru www.jakpsatweb.cz jakožto mému jedinému „konzultantovi“, dále Marku Lalošákovi za příkladné vedení webu metalista.cz a v neposlední řadě také Ing. Martě Bechyňové za možnost přihlásit se do SOČ.

Anotace

Tato práce pojednává o tvorbě ajaxizovaného hudebního portálu metalista.cz.

V první části jsou shrnuty výhody a nevýhody Ajaxu jako technologie pro tvoření rychlejších webových aplikací, využití jeho výhod a vypořádání se s jeho nevýhodami.

Druhá část je zaměřena na web samotný, tzn. popsání jeho důležitých prvků a součástí, funkcionality, zajímavostí a v neposlední řadě administrace.

Závěr se věnuje přínosu pro tým lidí kolem portálu metalista.cz a přínosu webu jako průkopníka relativně nové technologie.

Tato práce by také měla posloužit tvůrcům webu jako příručka proti vypořádání se s úskalím ajaxizace webových stránek.

Klíčová slova: web, internet, hudební portál, Ajax, ajaxizace, JavaScript, HTML, PHP, internet, SEO

Obsah

Užité zkratky

zkratka	význam
AJAX	AJAX (Asynchronous JavaScript and XML) je obecné označení pro technologie vývoje interaktivních webových aplikací, které mění obsah svých stránek bez nutnosti jejich znovunačítání.
<body>	Tělo dokumentu. Obsahuje veškerý zobrazovaný obsah stránky.
	HTML tag, který obklopuje určitou logickou oblast stránky.
FB	Facebook (FB) je rozsáhlý společenský webový systém sloužící hlavně k tvorbě sociálních sítí, komunikaci mezi uživateli, sdílení multimediálních dat, udržování vztahů a zábavě.
GET	GET je v informatice jedna z dotazovacích metod HTTP protokolu, kterou webový prohlížeč (klient) získává webovou stránku (nebo jiný objekt – například obrázek) z webového serveru.
hash	Křížek (mřížka, hash, významově také označení čísla) je označení pro znak #.
hashbang	Hashbang je označení pro soubor dvou po sobě jdoucích znaků, z nichž první je znak # a druhý !.
<head>	Hlavička dokumentu, která se nezobrazuje. Obsahuje nepovinně další tagy (title, meta, link, style, script aj.).
HTML	HyperText Markup Language, označovaný zkratkou HTML, je značkovací jazyk pro hypertext. Je hlavním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu.
<input>	Angl. input = "vstup" (ve smyslu vstupní údaj), označovaný jako "vstupní pole". Nepárový tag <input> je součástí formuláře (tag <form>).
JS, JavaScript	JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk na straně klienta.
Kb	= 1,024 bitů -> Bit (z anglického binary digit - dvojková číslice; angl. bit = drobek, kousek) je základní a současně nejmenší jednotkou informace, používanou především v číslicové a výpočetní technice. Značí se malým písmenem b, např. 16 b, ale současně se může také objevit i označení bit, např. 16 bit.
Mb/s	= 1 048 576 b/s -> Bit za sekundu (značka bit/s, někdy také b/s, nebo bps z anglického bit per

	second) je jednotka přenosové rychlosti. Jednotka udává, kolik bitů informace je přeneseno za jednu sekundu.přenesených bajtů (byte). Zpravidla platí, že 1 B/s = 8 bit/s.
onClick	Angl. onclick = "při kliknutí" Javascriptová událost, která se spouští při kliknutí levým tlačítkem myši na určitý element.
PHP	PHP je skriptovací programovací jazyk. Je určený především pro programování dynamických internetových stránek a webových aplikací například ve formátu HTML, XHTML či WML.
SEO	Search Engine Optimization (zkratka SEO, optimalizace pro vyhledávače) je metodika vytváření a upravování webových stránek takovým způsobem, aby jejich forma a obsah byly vhodné pro automatizované zpracování v internetových vyhledávačích.
SW	Software (též programové vybavení) je v informatice sada všech počítačových programů používaných v počítači, které provádějí nějakou činnost.
tag	Značka mající určitý význam ve zdrojovém kódu webové stránky.
<title>	Tag <title> se nachází v hlavičce dokumentu a obsahuje text s titulkem, názvem stránky. Např. <title>Název stránky</title>.
URL	URL, celým názvem Uniform Resource Locator („jednotný lokátor zdrojů“) je řetězec znaků s definovanou strukturou, který slouží k přesné specifikaci umístění zdrojů informací (ve smyslu dokument nebo služba) na Internetu.

Tabulka 1 Užité zkratky

Předmluva

Zhruba před 2 lety jsem byl členem týmu jednoho internetového rádia. Tenkrát jsme zde řešili, jak udělat to, aby si posluchači mohli prohlížet různé sekce stránek bez přerušování přehrávání přehrávače rádia v bočním panelu. Programátor, který web tehdy dělal, přišel s myšlenkou použít Ajax. To byl vlastně moment, kdy jsem se prvně potkal s touto technologií.

Přišlo mi to jako docela dobrý nápad, protože způsob načítání obsahu webu Ajaxem vylučoval naprosto nekomfortní stavy, když jste narazili na webu na nějaký přehrávač, třeba s videoklipem, který mohl trvat např. kolem 5 minut. Abyste ho zhlédli plynule, nesměli jste mezitím přejít na jiný odkaz, protože se Vám video v tom okamžiku přerušilo a museli jste ho pouštět od začátku. Ti odvážnější se pak maximálně mohli pokusit najít, kde předtím skončili a spustit ho znova.

A tak, když jsem se dostal zhruba po roce k tomu, že sám naprogramuji webový hudební portál, sáhl jsem okamžitě po Ajaxu, jako po technologii pro větší komfort uživatele.

Na začátku tvorby webu jsem však vlastně ani netušil, do čeho se to pouštím. Měl jsem sice naprogramováno už spoustu webů a aplikací, ale přeci jenom toto bylo moje první setkání s Ajaxem. Postupně jsem tedy začal zjišťovat, o co jde a řešit všechny vyvstávající problémy. Ovšem nebylo moc odkud čerpat informace. Pokud se podíváme na webovou scénu, zjistíme, že hledat ajaxový web je jako *„hledat jehlu v kupce sena“*. Když pominu web rádia, kde jsem Ajax objevil a giganty jako Google nebo Facebook, našel jsem vlastně pouze jeden ajaxový web a to byl ještě značně nedoladěný. Prakticky se tedy nebylo od čeho odrazit. Naštěstí to mělo i světlou stránku, posloužilo mi to jako motivace, abych web dodělal až do konce a uvedl tak v činnost něco, co jen tak někde neměli.

Na různých diskusích jsem střádal informace a rady a web nakonec úspěšně dokončil. Doteď mi však leží v hlavě, proč se na webové scéně nesetkáváme s větším množstvím ajaxizovaných webů. Spousta lidí vidí v Ajaxu *seo-unfriendly* technologii, a proto se jí vyhýbá. Dle mě jde však i ajaxový web optimalizovat pro vyhledávače, jen to chce trochu práce navíc. Navíc SEO není všechno. Použiji citaci specialisty na internetové podnikání a výkonnostní internetový marketing Zbyňka Hyráka o této problematice: *„Nejčastější chybou, se kterou se ve své odborné praxi setkávám, je* **přílišná optimalizace pro vyhledávače (SEO)** *a minimální snaha měření výkonnosti a efektivity veškerých marketingových aktivit s minimální optimalizací webu pro návštěvníky a uživatelskou použitelnost (UX).“*[1]

[1] http://www.ipodnikatel.cz/Poradci-iPoradny/pet-otazek-na-telo-nasim-poradcum-specialista-na-internetove-podnikani-a-vykonnostni-internetovy-marketing-zbynek-hyrak.html

Pokud však lidé tyto argumenty nepřijímají, je jasné, že je Ajax stále jen okrajovou záležitostí. Dalším důvodem pro jeho nízký výskyt by také mohla být vyšší cena, za kterou se nechají ajaxové stránky vytvořit, což je docela pochopitelné, protože vývoj ajaxového webu samozřejmě zabere více času.

Záměrně jsem si udělal průzkum cen webů a když mi jedna firma odpověděla, že web jako metalista.cz by byla schopna udělat za 150 000 Kč, „protočily se mi panenky“. Tudíž se pak ani nedivím, že majitelé stránek takovéto plusové technologie nepožadují.

Hlavním důvodem pro nízký výskyt ajaxizovaných webů však zřejmě bude samotná nerozšířenost Ajaxu mezi lidmi, co si nechávají stránky vytvářet. Když nevíte, že něco existuje, nemůžete to logicky ani požadovat.

Proto je jedním z mých záměrů této práce a hlavně webu jako takového zasloužit se o zvýšení povědomí o Ajaxu.

Úvod

V dnešní době dosahuje průměrná rychlost internetu v České republice kolem 7 Mb/s. Tato rychlost hravě stačí na stahování současných webových stránek, jejichž průměrná velikost dosahovala v prosinci 2012 hodnoty 1 286 kB[2]. Jak je tedy možné, že se všechny stránky nenačítají stejně rychle, když by je rychlost internetového připojení měla hravě zvládat?

Problém je ve způsobu, jakým se data webové stránky načítají. Většina dnešních stránek totiž funguje na principu: *pošli požadavek (klikni na odkaz) → změň současnou URL na URL odeslaného požadavku (odkazu) → reloadni stránku → → zobraz obsah nového požadavku (odkazu).*

Nyní si představte, že existuje technologie, kterou lze jeden z těchto kroků vynechat. Tím krokem je reloadování stránky a onou technologií je Ajax. V reálu to pak znamená, že se průběh načítání stránky změní na princip: *pošli požadavek (klikni na odkaz) → změň současnou URL na URL odeslaného požadavku (odkazu) → zobraz obsah nového požadavku a díky nenutnosti reloadu, načti jen skutečně změněný obsah (příklad: při zobrazení nového článku nech hlavičku, menu, postranní panely, patičku a změň pouze obsah středního panelu s článkem).*

Náhle se tak dostáváme zhruba na poloviční dobu načtení článku než u prvního způsobu, což může znamenat změnu například z 0,88 s na 0,44 s, což se povedlo konkrétně u mého webu metalista.cz, který jsem kompletně postavil na Ajaxu a dále na něm budu demonstrovat vše potřebné.

Abych to ještě více rozebral, tak bez Ajaxu se za 0,88 s načte pouze článek jako takový. Ovšem bez Ajaxu se také načítají i všechny ostatní části stránky. Z těch největších např. právě na metalista.cz zmíním reklamní slider, Facebook like box či YouTube přehrávač, které však důmyslný Ajax nenačítá. Pokud tedy budeme počítat do celkového času i načtení ostatních elementů než jen samostatného článku, což je správná úvaha, dostaneme se dokonce na 3,32 s a to už je oproti 0,44 s hodně velký rozdíl.

Cílem moderních webů by měla být bezpochyby plynulost webu, tedy jeho rychlá odezva. Hranice vnímání plynulosti je u člověka 0,1 s[3]. Do této hranice není člověk schopen zaregistrovat prodlevu při načítání. Pokud tedy s Ajaxem tuto hranici atakujeme, zdá se mi dobrým nástrojem pro budoucnost webů.

[2] http://www.cnews.cz/prumerna-webstranka-roce-2012-narostla-o-tretinu-na-disketu-se-jeste-vejde
[3] http://www.sitelab.cz/rady/definice.html#HR01

Ajax PRO vs. PROTI

Ajax však přináší i některá úskalí. Z jeho principu fungování totiž vyplývají kromě těch kladných vlastností i určitá omezení, která se musí dále v aplikaci ošetřit. V tabulce níže (Tabulka 2) demostruji základní z nich.

+	-
Rychlost načítání.	Nutnost zapnutého JavaScriptu.
Při přechodu na jinou stránku se nenačítá obsah neměněnných částí webu. Nepřeruší se tak například přehrávání videa či zvuku v postranním panelu.	Nemění se URL adresa, nevytváří se tak historie procházení a nejde se tak vracet na předchozí navštívené stránky.
	Krkolomná SEO optimalizace.
	Problematická změna titulku.
	Nelehké zprovoznění FB like tlačítka.
	Stránka se při přechodu na jiný odkaz neposouvá nahoru na její začátek.

Tabulka 2 PRO a PROTI Ajaxu

Eliminace „PROTI“

1) Nutnost zapnutého JavaScriptu

Tento problém lze vyřešit zhotovením verze webu, která bude fungovat bez JavaScriptu, což zároveň pomůže k SEO optimalizaci - viz bod 3).

Jelikož se všechny ajaxové odkazy mění přes JavaScript, je nutné udělat alternativní odkazy pro uživatele bez zapnutého JavaScriptu. Sice je takových uživatelů dle průzkumu maximálně 2 %[4], ale i s nimi se při optimalizaci aplikace musí počítat nehledě na bod 5). Těmto uživatelům samozřejmě web bude fungovat běžně neajaxově.

```
<div id=“menuHlavni“>
...
<a href="?stranka=novinky&amp;cisloStranky=1&amp;rubrika=obecne">Obecné</a>
...
</div>
```

Tento kód se načte všem uživatelům. Pokud je však zapnutý JavaScript, proběhne níže uvedený skript, který do *menuHlavni* načte soubor *menuHlavni.htm*, jehož obsahem jsou JavaScriptové odkazy. Pokud tedy není zapnutý JavaScript, zůstanou na stránce normální neJavaScriptové odkazy.

```
<script>
ajaxLoader('includy/menuHlavni.htm','menuHlavni');
</script>
```

menuHlavni.htm

```
<a onclick="odkaz('obecne'); return false;">Obecné</a>
```

JavaScriptová funkce *odkaz pak vypadá takto:*

[4] http://www.searchenginepeople.com/blog/stats-no-javascript.html

```
function  odkaz(adresa) {

window.location.hash='stranka=novinky&cisloStranky='+ cisloStranky
+'&rubrika=' + adresa;

}
```

Proměnná *cisloStranky* se pak získává za pomoci souborů *get_hash.js* a *get_hash_promenne.js*, jejichž cílem je vydolovat z URL adresy za pomoci práce s řetězcem obsah proměnných, jelikož v adrese za hashem (#), respektive hashbangem (#!), nelze používat žádné obdobné funkce jako *GET* v PHP.

get_hash.js:

```
function  get_hash() {

hash = document.location.hash.replace (new  RegExp ('^(.*)#'),'');

get = {}

    if (hash) {

        hash = hash.split ('&');

        for  (i = 0; i < hash.length ; i++) {

            hash[i] = (hash[i]).split ('=');

            get[hash[i][0]] = (hash[i][1] ? hash[i][1] : null);}

            }

 return  get;

 }

 _get = get_hash();[5]
```

get_hash_promenne.js:

```
var  cisloStranky = _get.cisloStranky;  //načítám obsah proměnné
stranka z url ...
```

[5] http://mike.treba.cz/javascript-prace-s-getem-jako-v-php/

2) Nemění se URL adresa, nevytváří se tak historie procházení a nejde se tak vracet na předchozí navštívené stránky

Udělat měnící se URL bez reloadu, tedy bez „PHPéčkového“ znaku *„?“*, jde za pomoci JavaScriptové funkce *window.location.hash*, která přidává určitý řetězec za znak *„#“*, což už nevyžaduje reload stránky. Tímto je základní problém vyřešen, protože se vytváří historie procházení a při kliknutí na tlačítko *„Zpět“* ve vašem prohlížeči se už dostanete na předchozí adresu. Problém je však v tom, že při přechodu na předchozí adresu nedostanete požadovaný obsah, protože se nijak nespustí základní funkce *AjaxLoader.js*, která se stará o načítání veškerého obsahu na webu. Toto lze naštěstí také opatřit, a sice funkcí, která bude každou chvíli zjišťovat, zda se nezměnil obsah za *„#“* a pokud ano, zavolá funkce, které se postarají o zobrazení správného obsahu vůči aktuální adrese. Níže uvádím část skriptu, který toto zajišťuje.

```
function(){

var l = location, h = l.hash;

var t = setInterval(function(){if(h !== l.hash) h = l.hash, f();},
50);

window.onhashchange = function(){clearInterval(t); f();
window.onhashchange = f; };

function f(){

var  clanek = _get.clanek;

...

if((!clanek) && (document.getElementById('prostredek')))

ajaxLoader('includy/novinky.php?cisloStranky='+ cisloStranky
+'&rubrika=' + rubrika,'prostredek');

...
```

Pozn.: Soubor s celou funkcí pojmenovaný interval_hash.js je obsahem přiloženého archivu soubory.rar.

3) Krkolomná SEO optimalizace

Princip fungování vyhledávacích robotů, jako je např. Googlebot, komplikuje SEO optimalizaci ajaxových webů. Vyhledávací robot totiž funguje tak, že prochází obsah webu a pokud na něm narazí na nějaký odkaz, tak ho zaindexuje a dále prochází i obsah tohoto odkazu. Problém je však v tom, že ignoruje veškerý JavaScript, takže odkazy za hashem (#), které jsou JavaScriptové, prostě nevidí. Tento problém lze vyřešit dvěma způsoby:

a) Pokud budeme dávat proměnné v adrese za hashbang (#!) místo za samotný hash (#), vyhledávací robot nám už takový odkaz zaindexuje. Problém je však v tom, že když se nějaký návštěvník dostane na tento odkaz skrz vyhledávač a nemá zapnutý JavaScript, tak se mu stejně obsah, na který odkaz odkazuje, nezobrazí správně. I toto lze ošetřit různými přesměrováními v závislosti na zapnutém JavaScriptu. V bodě b) podobný postup uvedu.

b) Nezapomínejme na to, že máme udělanou verzi pro uživatele s vypnutým JavaScriptem - viz bod 1). Právě tyto odkazy v nonJavaScriptovém webu vyhledávací robot bez problému zaindexuje, protože jsou normální serverové za znakem „?“, což vyhledávačům nevadí. Ovšem pokud by návštěvník klikl na tento odkaz, tak se mu nezobrazí ajaxový web, nýbrž ten neajaxový. Toto lze vyřešit přesměrováním na začátku stránky. Podobný postup lze praktikovat i u bodu a).

```
<?php

$clanek = (int)$_GET['cl'];

$rubrika1=mysql_query("select rubrika from news where id=$clanek");

$rubrika=mysql_result($rubrika1, 0);

if($_GET['cl'])

echo
"<script>window.location.href='http://metalista.cz/#!stranka=novinky
&cisloStranky=1&rubrika=$rubrika&clanek=$clanek'</script>";

?>
```

4) Problematická změna titulku

Ajax sice umí měnit určité části webu, nicméně jenom ty v tagu *<body>*. Pokud bychom tedy chtěli změnit titulek, narážíme na problém. Tag *<title>* totiž není součástí tagu *<body>* nýbrž tagu *<head>*. Z toho vyplývá, že musíme použít nějaký alternativní postup, který by nám požadovanou změnu titulku umožnil.

Naštěstí existuje JavaScriptová konstrukce *window.document.title*, skrz níž už můžeme titulek měnit. Pokud chceme změnit titulek nějakých statických částí webu, není tedy nic jednoduššího, než napsat následující kód.

```
<a onClick="window.document.title='Nový titulek';">Změň titulek</a>
```

Potíž ovšem nastává, pokud máme nějaký dynamický obsah. Například pokud chceme měnit titulek pokaždé dle názvu článku. V tomto případě musíme nějakým způsobem propojit JavaScript s PHP, abychom mohli v JavaScriptu pracovat s názvem článku z databáze. Vždy když potřebuji propojit klienta a server, používám metodu *„input type hidden"*, která lehce vyřeší tento problém.

Tato metoda je založena na tom, že k tagu *<input>* může přistupovat jak JavaScript, tak PHP. Pokud tedy vypisujeme novinky, tak si někde vedle titulku umístíme skrytý input (type=hidden), jemuž přiřadíme hodnotu aktuálního článku. Tento *<input>* však musíme ještě udělat unikátním, aby si pak následně JavaScript mohl vybírat konkrétní název titulku pro každou novinku zvlášť. Výše popisované v kódu pak vypadá následovně:

```
<input id='input$i' type='hidden' value='$titulek'>
```

Pozn.: $i značí pořadí článku ve výpisu novinek v rámci jedné stránky a $titulek značí titulek konkrétní novinky.

Přes PHP si poté můžeme vygenerovat JavaScriptovou *onClick* událost, která se už bude dynamicky měnit, vždy pro potřeby daného článku, díky čemuž se při kliknutí na odkaz k přečtení určitého článku změní i titulek stránky dle titulku článku. V kódu to pak vypadá následovně:

```
<a
onClick="window.document.title=document.getElementById('input$i').va
lue;">Čti konkrétní článek</a>
```

5) Nelehké zprovoznění FB like tlačítka

Hlavní potíž je v tom, že Facebook opět funguje na podobném principu jako vyhledávač a pokud jsou na webu adresy za hashem (#), tak je bere všechny jako jednu jedinou a nelze tak diferencovat jednotlivé články, které mají být „olajkovány“. Je tedy nutností použít hashbang (#!), který facebook už dokáže zaregistrovat, a tudíž diferencovat od sebe jednotlivé články. Ale tím problém nekončí.

Facebook předělává všechny odkazy za hashbangem (#!) následujícím způsobem:

Z:http://metalista.cz/#!stranka=novinky&cisloStranky=1&rubrika=recenzealb&clanek=9

Na:http://metalista.cz/?_escaped_fragment_=stranka%3Dnovinky%26cisloStranky%3D1%26rubrika%3Drecenzealb%26clanek%3D6&fb_action_ids=472244816143949&fb_action_types=og.likes&fb_source=aggregation&fb_aggregation_id=246965925417366

Pokud tedy na stránce s článkem s URL, v níž je hashbang (#!), klikneme na FB like tlačítko, vytvoří se nám v našem facebookovém účtu v sekci „To se mi líbí“ odkaz na článek, který jsme „olajkovali“. Tento odkaz však převede *„#!“* na *„?_escaped_fragment=“* a zároveň si k tomu přidá kromě očekávaných proměnných z naší stránky spoustu svých atributů. To vyvolává problém. Pokud se totiž přemístíme na adresu s *„?_escaped_fragment=“* a pak pokračujeme v dalším brouzdání po webu, tvoří se nám duplicitní adresy a to je nepřípustné, proto je třeba přesměrovávat všechy adresy z Facebooku tedy z *„?_escaped_fragment=“* opět na původní adresy s *„#!“*. K tomu jsem vytvořil následující PHP skript, který toto zajišťuje.

```
<?php

if($_GET['fb_action_ids'])

{

$fragment=$_GET['_escaped_fragment_'];

$dilky = explode("=", $fragment);

$cisloClanku=$dilky[4];

$rubrika2=$dilky[3];

$dilky2 = explode("&", $rubrika2);

$rubrika=$dilky2[0];

header("HTTP/1.1 301 Moved Permanently");
```

```
header("Location:http://metalista.cz/#!stranka=novinky&cisloStr

anky=1&rubrika=$rubrika&clanek=$cisloClanku ");

header("Connection: close");}

?>
```

V průběhu testování webu metalista.cz jsem však přišel ještě na jednu věc, a sice že někdo zadává adresu jako *www.metalista.cz* a někdo jako *metalista.cz*. Obě vedou na stejný obsah, ale Facebook je bere jako dvě odlišné adresy. Proto je potřeba všechny adresy sjednotit na *metalista.cz* následujícím přesměrováním.

```
<?php

$server = $_SERVER['SERVER_NAME'];

$www = substr($server,0,3);

if($www=='www')

{header("location:http://metalista.cz");}

?>
```

Pozn.: Lze přesměrovat i na www.metalista.cz. Výsledek bude stejný, jen budou adresy sjednocené v jiném tvaru. Záleží na každém, co se mu víc líbí.

A konečně poslední poznatek ze zkoumání facebookovských principů, který jsem objevil. Tento problém se týká všech Facebook like tlačítek a nejenom těch v ajaxových aplikacích. Jde o to, že si Facebook přidává jeho atributy i do těch neajaxových adres. Tím pádem, pokud „olajkujeme“ nějakou stránku třeba na adrese *metalista.cz/322-ten-masked-men-x-eiffel-65-blue-da-da-bee* a pak si ji otevřeme skrz odkaz v našem facebookovském účtu v sekci „To se mi líbí“, dostaneme se na adresu *www.metalista.cz/index.php/322-ten-masked-men-x-eiffel-65-blue-da-da-bee?fb_action_ids=472254296143001&fb_action_types=og.likes&fb_source=aggregation&fb_aggregation_id=246965925417366* a jelikož to jsou dvě odlišné URL, Facebook like tlačítko náhle funguje neracionálně, protože ač jsme na stejném obsahu, tak URL je jiná a Facebook tlačítko se na téže stránce zobrazuje jako ještě „neolajkované“. Skript výše tomuto však zabraňuje, takže na metalista.cz funguje Facebook like tlačítko racionálně, což se, bohužel, nedá říct o drtivé většině ostatních Joomla webů.

Pozn.: Iracionální a racionální chování FB like tlačítek na starém a na novém webu je také zaznamenáno jako videosoubory s názvem fbiracionalni.avi a fbracionalni.avi v přiloženém archivu ajaxMetalistaSoc.zip.

6) Stránka se při přechodu na jiný odkaz neposouvá nahoru na její začátek

Tento problém lze vyřešit pomocí JavaScriptové funkce *window.scrollTo*, která dokáže návštěvníka vrátit zpět nahoru na začátek stránky, respektive článku. Opět ukážu, jak mám například řešené odkazy na metlista.cz.

```
<a onclick=\"window.scrollTo(485,485);
window.location.href='#!stranka=novinky&cisloStranky='+ cisloStranky
+'&rubrika=".$row['rubrika']."&clanek=".$row['id']."';\";>".$row['ti
tulek']."</a>
```

Ajaxizace v praxi → metalista.cz

Jak již jsem zmínil v úvodu své práce, celou problematiku ajaxizace webu jsem řešil na webu www.metalista.cz, který považuji za správné představit jako důkaz toho, že výše uvedené řádky nejsou jen pouhou teorií.

Metalista.cz je hudební portál o metalové hudbě. Jeho možnosti se pokusím shrnout v několika níže uvedených bodech.

A) Články

Základem každého webu je podat jeho návštěvníkům požadované informace. Pokud je těchto informací více a s časem přibývají, není nic efektivnějšího, než je řádně strukturovat, aby se v nich uživatel dokázal orientovat. Na metalista.cz jsou těmito informacemi články.

Články jsou rozděleny do následujících kategorií „Recenze alb“, „Rozhovory“, „Koncerty“ atd. Na úvodní stránce webu, tedy na adrese metalista.cz, se nachází všechny vydané články, tedy články s časem vydání nižším než současnost. Datum a čas vydání se nastavují jako jeden z parametrů při přidávání novinky v administraci. V reálu pak vydávání článků funguje následovně: *napsání článku → zobrazení článku v sekci „aktivace a editace článků“, kam má přístup jen šéfredaktor → → schválení článku, korektura a nastavení data vydání šéfredaktorem → jakmile se potká současnost s datem vydání, článek se zobrazí na webu*.

Tento princip zajišťuje plynulost vycházení článků, což je velmi důležité. Tímto způsobem se totiž nikdy nestane, že by se najednou vydalo třeba 5 článků a druhý den by se nevydal ani jeden. V praxi se totiž dost často stává, že redaktoři přes víkend sepíšou „hromadu“ článků a přes týden pak skoro nikdo nepíše. Kdyby vyšly všechny články najednou, v průběhu týdne by pak nevyšel žádný a to je špatně, proto tento praktický systém.

Co se týče samotného rozhraní přidávání článku, tak stojí za zmínku poukázat na implementaci JavaScriptového CKEditoru, pomocí kterého se dá formátovat text článku podobným způsobem jako třeba v Microsoft Word, což je velká výhoda oproti běžné <textarea>.

B) Galerie

Ke článkům je často potřeba přiložit obrazovou dokumentaci. K těmto účelům je na metalista.cz zhotovena galerie. Tuto galerii je možno přiložit ke každému článku opět skrz administrační rozhraní.

Základem galerie je JavaScriptový skript pro zobrazování obrázků LightBox. Zobrazení obrázku pomocí LightBoxu má tento základní tvar:

```
<a href='velkyObrazek.png' rel='lightbox'><img
src='miniaturaObrazku.png' width='99' height='99'></a>
```

Ten způsobuje zobrazení miniatury obrázku. Po kliknutí na tuto miniaturu se zobrazí jeho zvětšenina v graficky pěkně zpracovaném prostředí.

Obrázky pak jdou také řadit do galerie. To znamená, že se lze mezi jednotlivými obrázky posouvat šipkami doprava a doleva. V kódu LightBoxu se tohoto docílí pomoci přidání názvu pole do atributu odkazu *rel.*

```
<a href='velkyObrazek.png' rel='lightbox[galerie]'>
```

Z výše uvedených předpokladů pak lze naprogramovat přidávání galerie k jednotlivým článkům. Na metalista.cz je toto realizováno skrz dvě tabulky v databázi, z nichž jedna nese název *galerie* a druhá *galerie_odsazeni*. Právě druhá z nich umožňuje rozdělovat galerii v rámci jednoho článku na více subgalerií a k nim také přidávat jednotlivé titulky. Z následující struktury obou tabulek pak lze vybírat potřebné údaje ke správnému zobrazení každé galerie.

galerie → id, src, clanek

galerie_odsazeni → id_clanku, cislo_odsazeni, nadpis

Snad jen pro upřesnění uvedu, že *cislo_odsazeni* uchovává hodnotu, po kolikátém obrázku se má začít vypisovat další subgalerie.

Jak funguje přidávání obrázků v samotné administraci, popisuje obrázek níže.

Obr. 1 Přidávání galerie v administraci

C) Reklamní slider

Pokud chce web inzerovat, je dobré mít k tomu také nějaký prostředek. Na metalista.cz je k tomuto účelu využíván, kromě postranních bannerů, také hlavní středový reklamní slider.

Tento slider je postaven opět na JavaScriptovém podkladu, a sice TinySlideru. Slider v podstatě jen mění obrázky v jednom * u* s doprovodnými efekty v určitém intervalu. V administraci se pak dá měnit zdrojový soubor obrázku a url, na kterou obrázek odkazuje.

D) Poslední videoklip

Další věc, co stojí za zmínku na metalista.cz, je systém posledního videoklipu. Tento systém funguje následovně.

Pokud někdo přidá do článku video ze serveru YouTube, což lze pomocí YouTube rozšíření pro CKEditor, tak se adresa, na které se video nachází na YouTube např. http://www.youtube.com/watch?v=Q7C90sLh5Ok přepíše na adresu, která je pořeba k tomu, aby mohla být obsahem atributu *src* tagu *<iframe>*, na které se nachází pouze video jako takové bez komentářů a dalších rušivých elementů např. http://www.youtube.com/v/Q7C90sLh5Ok, což zapříčiní zobrazení dobře známého FlashPlayerového YouTube okénka v novince. Toto okénko je však potřeba zobrazit také v sekci „Poslední videoklip“.

Aby se dostalo i do sekce „Poslední videoklip“, je potřeba si do databáze uložit následující adresu http://www.youtube.com/v/Q7C90sLh5Ok. Tuto adresu lze získat, pokud se upraví skript, který zajišťuje chod CKEditoru tak, aby přeměněnou adresu CKEditor neposílal jenom do textového editoru jako takového, ale taktéž např. do nějakého skrytého *<input>u*, který se pak přečte při odesílání celé novinky a uloží se do tabulky novinek do sloupce videoklip. Datumově nejmladší videoklip se pak na webu používá jako atribut *src* tagu <iframe>, který zajišťuje právě zobrazení FlashPlayeru s daným YouTube videem v sekci „Poslední videoklip“.

E) Kalendář vycházejících alb

Asi poslední, co stojí za zmínku na metalista.cz, je důmyslný „Kalendář vycházejících alb“, což je vlastně taková aplikace v aplikaci.

Při vývoji této aplikace byly hlavními požadavky proporcionalita spolu s úsporou použitého prostoru. Oba dva požadavky se podařilo skloubit dohromady a vznikl tak koncept, který se pokusím popsat níže.

Pozn.: Pro pochopení celého problému je nutností ukázat celý zdrojový kód, jelikož by zde však zabíral spoustu místa, je přiložen v souboru kalendarAlb.php.

Jak už jsem se zmínil, bylo důležité dodržet proporcionalitu. Proporcionality je dosáhnuto tak, že se vypočítá výška měsíce podle měsíce s největším výskytem akcí. V rámci měsíce se pak vypočítává poloha akce jako:

poloha akce = den akce / počet dnů v měsíci * výška měsíce - půl velikosti písma.

Poloha využitého místa v kalendáři se zapisuje do pole a poté se každý termín další akce porovnává s polohami již využitého místa, a když je poloha obsazená, odsouvá se dolů, dokud nenarazí na volné místo. Tím je docíleno toho, že se písma nikdy nepřekrývají, což je díky relativnímu pozicování klidně možné. Na druhou stranu bez relativního pozicování by nebylo možné dosáhnout oné proporcionality. Pro dokonalé využití prostoru je ve skriptu zřízeno také střídání levé a pravé polohy umístění textu.

Za přítomnosti výše uvedených postupů pak kalendář může fungovat jako dobrý nástroj pro grafické znázornění vycházení jednotlivých alb v rámci daného roku.

F) Administrace

V administraci lze přidávat, editovat a mazat novinky, přidávat postřehy na vylepšení do sekce „plány“, sledovat statistiky návštěvnosti webu, měnit obsah reklamních ploch a v neposlední řadě upravovat kalendář alb. Takže vlastně vše, co je potřeba ke správnému chodu celého webu.

Kvůli bezpečnosti nemohu poskytovat údaje k administraci. Na přehlídce však bude náležitě předvedena.

Závěr

Cílem nového webu pro doménu metalista.cz bylo zmodernizovat dosavadní web, který byl postaven na redakčním systému Joomla. To způsobovalo mimo jiné jeho pomalost. Použitím Ajaxu se web posunul o několik příček výše, zejména rychlostí. Navíc byla vytvořena administrace „ušitá“ na míru, kterou může obsluhovat každý člověk bez nutnosti znalosti jakéhokoliv redakčního systému. Pro portál metalista.cz to tak znamenalo značné ušetření času nutného pro zaškolování lidí v práci s redakčním systémem. Také bylo odstraněno nelogické chování FB-like tlačítka.

Do budoucna je v plánu web nadále rozvíjet, a pokud vše vyjde, tak ho i propojit s internetovým rádiem.

Doufám, že metalista.cz bude přínosem k dalšímu rozvoji webů a že Ajax nezůstane jenom záležitostí gigantů, jako je FaceBook nebo Google.

Zdroje

Vzhledem k povaze této práce není možné sepsat seznam všech zdrojů užitých pro její vypracování.

Využité znalosti jsou výsledkem zhruba čtyřletého studování problematiky tvorby webu, přičemž je nemyslitelné sepisovat si zdroj každého nového poznatku.

Z těch největších zdrojů však mohu jmenovat webové stránky: jpw.cz a tvorba-webu.cz a následující odborné diskuse: diskuse.jakpsatweb.cz a stackoverflow.com.

Pro sekci zkratek byla z velké části také použita webová encyklopedie wikipedia.org.

Co se týče literatury, uvádím knihy, které mi pomohly propracovat se až k Ajaxu a vlastně i s Ajaxem.

1) PHP nejen pro začátačníky, Martin Pokorný, 2005

2) Velký průvodce JavaScriptem, Dave Thau, 2009

Použitý software: Photoshop, PSPad, FileZilla, IETester, Google Chrome, Mozilla Firefox, Internet Explorer, Opera, phpMyAdmin

Poznámka o designu: Upozorňuji, že grafika webu není moje práce. Pro design stránek byla použita jako předloha volně dostupná Wordpressová šablona „MusicLife“ stažená z fthemes.com. Tvůrcem loga je Michal Novák a grafiku administrace navrhl Tomáš Matucha.

Přílohy

ajaxMetalistaSoc.zip:

1) interval_hash.js

2) fbiracionalni.avi

3) fbracionalni.avi

4) kalendarAlb.php

- pozn: přílohy najdete také ke stažení na www.metalista.cz/ajaxMetalistaSoc.zip

Seznam citací

1) http://www.ipodnikatel.cz/Poradci-iPoradny/pet-otazek-na-telo-nasim-poradcum-specialista-na-internetove-podnikani-a-vykonnostni-internetovy-marketing-zbynek-hyrak.html

2) http://www.cnews.cz/prumerna-webstranka-roce-2012-narostla-o-tretinu-na-disketu-se-jeste-vejde

3) http://www.sitelab.cz/rady/definice.html#HR01

4) http://www.searchenginepeople.com/blog/stats-no-javascript.html

5) http://mike.treba.cz/javascript-prace-s-getem-jako-v-php/