Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

**Bakalářská práce**

**jQuery – plugin pro manipulaci se stromem štítků**

Plzeň, 2013 Libor Vohanka

1 Úvod 3

2 Podobné práce 3

3 Použité technologie 3

3.1 Technologie pro klientskou část 3

3.1.1 HTML a jiné značkovací jazyky 3

3.1.2 CSS 4

3.1.3 Javascript 4

3.1.4 AJAX 5

3.2 Serverové technologie 5

3.2.1 PHP 5

3.2.2 MySQL 5

3.3 Záloha dat 5

3.4 JSON 6

3.5 Vývojová prostředí 6

4 Použité frameworky 7

4.1 jQuery 7

4.1.1 jQuery UI 7

4.2 Twitter Bootstrap 8

4.3 CSS Framework 8

4.3.1 jQuery pluginy 8

5 Databáze 8

5.1 Návrh použité databáze 8

5.2 Záměna za vlastní 8

5.3 Práce s databází 8

6 Tvorba jQuery pluginu 8

7 Použité techniky 11

7.1 Vykreslování stromu 11

7.2 Kontextové menu 11

7.3 Změny položek stromu 11

7.3.1 Přidávání 11

7.3.2 Mazání 11

7.3.3 Úprava 11

7.3.4 Zobrazení predikátů 11

8 Tvorba ukázkové webové aplikace 11

9 Závěr 11

# Úvod

Framework jQuery je velice oblíbený hlavně kvůli jednoduchosti použití a jeho rozšiřitelnosti. Framework je z obecného hlediska jakési rozšíření stávajícího jazyka o funkce a proměnné, které programátorům ulehčují práci v daném jazyce. jQuery je nadstavbou jazyka Javascript.

V současné době je velké množství programátorů, kteří používají Javascript. Ještě více je však používáno jQuery. Proto lze na internetu nalézt nepřeberné množství zásuvných modulů, neboli pluginů právě do jQuery.

# Podobné práce

# Použité technologie

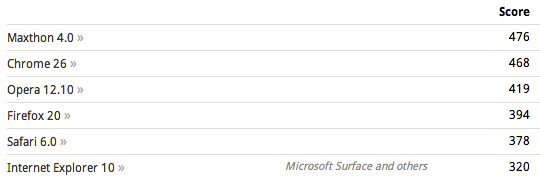
## Technologie pro klientskou část

### HTML a jiné značkovací jazyky

HTML je nejrozšířenější značkovací jazyk, který je používán pro tvorbu webových prezentací. Stavebními kameny jsou značky neboli tagy. Většinou párové. Do těchto značek jsou uzavírány části dokumentu, které mají být nějakým způsobem odlišeny. Každá značka má svůj speciální význam, kterým pozměňuje celkový vzhled prezentace.

Vše ale prochází určitým vývojem a s příchodem čtvrté verze jazyka HTML byl uveden jazyk XML. Později byl uveden jazyk XHTML syntaxí vycházející z XML. Je však jen úpravou HTML. Mezi nejzásadnější změny XHTML oproti HTML je párovost všech tagů.

Dalším vývojovým stupněm je HTML 5. Sestává z mnoha částí a přesto, že ještě není jeho definice ještě zcela hotova, prohlížeče již HTML5 více či méně podporují. Na stránkách [http://html5test.com](http://html5test.com/) lze otestovat kterýkoli prohlížeč. V závislosti na tom, kolik funkcí HTML 5 podporují, dostane prohlížeč bodové ohodnocení do celkového počtu 500 bodů. Na obrázku 3.1 je pořadí a skóre nejpoužívanějších prohlížečů dle výsledků z výše jmenované webové adresy.



Obrázek . : Žebříček prhlížečů seřazený podle počtu bodů

Mezi nejzajímavější rozšíření nové verze patří třeba přidání nového atributu data, který může pomoci například při odesílání formulářů, nebo při výběru tagu Javascriptem. Další rozšíření oproti starší verzi HTML je přidání několika typů vstupních polí u formulářů. To je výhodné převážně pro mobilní zařízení s dotykovými obrazovkami. Pokud je typ vstupního pole například email, může prohlížeč uzpůsobit klávesnici pro vkládání emailové adresy.

### CSS

Kaskádové styly[[1]](#footnote-1) slouží k definici vzhledu stránek vytvořených v HTML, XHTML, nebo XML. Díky CSS mohou být elementy po webové stránce posouvány, upravit měnit jejich velikost, nebo barvu. U text lze změnit font, barvu, či velikost. To je samozřejmě jen výběr toho, co kaskádové styly umí. Dokáží ale také výrazným způsobem zlepšit přehlednost webu. Existují tři způsoby jak vložit styly do dokumentu. Přímo do HTML tagu, mezi tagy <style></style> a třetí nejpoužívanější možnost je vložit vlastnosti do externího souboru, na který se odkáže v hlavičce HTML souboru.

Specifikace třetí verze kaskádových stylů také ještě není zcela hotova. O podpoře v jednotlivých prohlížečích informuje server W3Schools[[2]](#footnote-2). Mezi nejužitečnější vlastnosti pro designery internetových stránek patří možnost přidat elementům stíny, zaoblit rohy, nebo třeba možnost definice průhlednosti elementů.

### Javascript

Javascript[[3]](#footnote-3) je skriptovací jazyk běžící na straně klienta. Existují také řešení spouštěné na straně serveru jako například Node.js. Slovo java, je v názvu jazyka jen z marketingových důvodů a s programovacím jazykem Java má společnou pouze syntaxi. Javascript stránkám přidává na dyna. Například při validaci formulářů, které mají některá data v nekorektním formátu z řetězu vynechá odesílání dat na server a rovnou zobrazí varování. Javasript si ale uživatel v prohlížeči může vypnout, a proto je nutné odeslaná data validovat ještě na straně serveru. Způsob interpretace výstupů Javascriptu se bohužel liší prohlížeč od prohlížeče a je nutno tento problém brát při použití v úvahu. Jelikož nemá smysl objevovat objevené, tak nemá smysl ošetřovat ošetřené a je výhodné pro práci použít některý z Javascriptových frameworků, který nejen řeší problémy s interpretací výsledků, ale také přidává funkce usnadňující práci s dokumentem.

### AJAX

Asynchronní Javascript a xml technologie je používána pro dynamické načítání obsahu stránek bez nutnosti znovu načítat celou internetovou stránku. AJAX lze používat pro pouhé načítání statických HTML stránek, (po kliknutí na odkaz načíst příslušný formulář), nebo lze takto volat PHP kód, který může například vybrat určitá data z databáze, vložit je do pole a předat dále Javascriptu, který je dále zpracovává.

## Serverové technologie

### PHP

Skriptovací jazyk PHP je zpracováván na straně serveru a je tak hlavním článkem při validaci formulářů odesílaných uživatelem. Od čtvrté verze lze v PHP programovat objektově. Do páté verze byl však objektový model PHP přepracován. PHP je pravděpodobně nejrozšířenější skriptovací jazyk pro tvorbu webu a díky tomu pro něj existuje mnoho knihoven a frameworků. Syntaxí spadá do rodiny C/Java.

### MySQL

MySQL je databázový systém vycházející z normy jazyka SQL. Tuto normu však striktně nerespektuje. MySQL je používaný na většině webových hostingů. Od převzetí firmy MySQL firmou Oracle na podzim roku 2010 je nabízen jak pod bezplatnou, tak pod placenou licencí. Od verze 5.0 podporuje například i uložené procedury, trigery a pohledy.

## Záloha dat

Před samotným vývojem pluginu bylo nezbytné vyřešit problematiku zálohování zdrojových kódů a textu bakalářské práce. Způsobů jak zálohovat je hned několik. Od vypalování záloh na CD přes kopírování dat na externí disk, zrcadlení disků, nahrávání dat na internet do externích úložišť až po verzovací systémy. Pro mé potřeby a potřeby zadavatele, je nejvhodnější poslední jmenovaná možnost. Druhů systému pro správu verzí je více. Pro příklad uvedu čtyři nejpoužívanější:

* CVS
* SVN
* GIT
* Mercurial

Jejich hlavní dělení je na centralizované a decentralizované. CVS, stejně jako SVN spadají do kategorie centralizovaných verzovacích systémů, což v principu znamená, že mají jeden server, kde jsou nahrány veškeré zdrojové kódy. Jestliže vše funguje jak má, tak s tím není problém. Pokud se ale stane, že server spadne, tak uživatelé ztrácejí k repositáři přístup a často to znamená nemožnost další práce. Je-li repositář decentralizovaný, jako v případě Mercurial a GITu, tak každý kdo s ním pracuje, má staženou jeho verzi u sebe v počítači a lze s ním pracovat i bez připojení k internetu. Samozřejmě zde figuruje i server, kam každý uživatel odesílá svoje změny.

Díky službě http://github.com, kde lze umístit a verzovat jakýkoli kód zdarma, pokud je repositář veden jako veřejný, jsem na základě domluvy s vedoucím zvolil tuto variantu. Jelikož je mou povinností mít bakalářskou práci veřejně dostupnou je služba github zcela ideální. Další výhodou je pak jednoduchá kontrola stavu práce vedoucím.

## Přenos dat klient - server

Nejpoužívanější formáty pro přenos dat mezi klientem a serverem jsou JSON a XML. V současné době je při přenosu dat více využíván spíše JSON. Je tomu tak převážně kvůli objemu dat, který je potřeba přenést. Formát XML vychází ze syntaxe HTML. To znamená, že veškerá informace o datech je uložena mezi značkami. V kódu 3.1 je uveden příklad dat zapsaných pomocí XML. Značka osoba s atributy jmeno a prijmeni. Kód 3.2 dává za příklad jak vypadají data odesílána pomocí JSON.

<osoba jmeno="Petr" prijmeni="Novak">

</osoba>

Kód .: Příklad XML kódu

“jmeno“: “Petr“, “prijmeni“: “Novak“

Kód . : Příklad JSON kódu

Pokud porovnáme kód 3.1 a kód 3.2, lze si všimnout, v tuto chvíli nepatrných, rozdílů ve velikosti jednotlivých kódů. Pokud se přenášejí větší data, tento rozdíl bude znatelnější a JSON může ušetřit velké množství času.

Další výhodou JSONu je jednoduchost interpretace přijatých dat. PHP dokonce obsahuje funkci pro převod mezi polem a JSON strukturou. Tuto funkci má v sobě taé framework jQuery pro Javasript. Z těchto důvodů jsem si pro přenos dat vybral právě JSON.

Oba zmiňované formáty lze použít jak v internetových aplikacích, tak v offline aplikacích.

## Vývojová prostředí

Je dobrým zvykem psát kód v programu, který alespoň zvýrazňuje syntaxi jazyka, ve kterém programátor píše. Zástupcem nejjednodušších prostředí je pro platformu Windows například program Notepad++. Pro Mac OS X je to pak TextMate. Obě tato prostředí rozpoznají syntaxi velkého množství skriptovacích, značkovacích a programovacích jazyků. Mimo jiné také Javasript, HTML, CSS a PHP, které se většinou používají při tvorbě jQuery pluginů. Já pro práci na svých projektech používám Netbeans. Ten umí krom zvýrazňování klíčových slov, také doplňovat názvy proměnných, funkcí a podobně. Má také správu projektů a další funkce, které usnadňují vývoj.

# Použité frameworky

## jQuery

jQuery je Javascriptový framework, který si klade za úkol zjednodušit programování v Javascriptu a sjednotit jeho zobrazování v prohlížečích. Od jeho založení v roce 2006 jeho popularita raketově vzrostla. Nyní je používán nejnavštěvovanějšími webovými servery[[4]](#footnote-4) jako například <http://amazon.com>, <http://microsoft.com>, <http://twitter.com>. Více než 62% nejpopulárnějších internetových stránek[[5]](#footnote-5) používá jQuery.

Jednou z nejpoužívanějších funkcí jQuery, je výběr elementu na stránce. Cokoli je potřeba v dokumentu změnit, musí to být nejdříve vybráno. Zde je porovnání, jak vypadá vybrání prvku s id = menu.

document.getElementById('menu'); //Javascript

$('menu') //jQuery

Kód . : Porovnání jQuery a Javascriptu

Další funkce poskytované jQuery :

* Změna DOM (Document Object Model) elementů na stránce – přidávání či mazání html tagů ze stránky.
* Úprava kaskádových stylů po výběru elementu
* Funkce pro práci s AJAXem
* Systém událostí
* Animace a efekty
* Snadné přidávání pluginů

### jQuery UI

Framework jQuery je postaven tak, aby vývojářům co možná nejvíce zjednodušil tvorbu pluginů. Nejslavnějším pluginem pro jQuery je jQuery UI vytvořený samotnou společností jQuery. Jedná se o sadu Javascriptových funkcí pro tvorbu uživatelského rozhraní. Mezi nejzajímavější patří funkce pro posuv elementů po stránce, dále obsahuje různé doplňky jako tooltip, datepicker, nebo autocomplete a v neposlední řadě přidává oproti jQuery řadu efektů zobrazování a skrývání elementů. jQuery UI je stejně jako jQuery poskytován pod MIT licencí[[6]](#footnote-6).

## Twitter Bootstrap

Twitter Bootstrap[[7]](#footnote-7) je kolekce nástrojů pro tvorbu webových aplikací. Během necelého půl roku od vydání se stal nejpopulárnějším projektem na GitHubu. Jednou z nejzásadnějších věcí, které Bootstrap řeší, je nekompatibilita zobrazení různých prvků na stránce v různých prohlížečích. Pokud programátor použije k tvorbě designu svého webu výhradně stylů Bootstrapu, nemusí se nekompatibility bát.

Bootstrap je knihovna obsahující naprostou většinu stylů, které jsou zapotřebí ke tvorbě jednoduchých internetových prezentací. Mezi základní styly, díky kterým stojí zato Twitter Bootstrap použít, patří:

* Rozložení prvků na stránce
* Základní šablony pro web
* Responsivní design
* Typografii

Dále obsahuje styly pro menu, tlačítka, stránkování, štítky, nebo náhledy obrázků. Součástí Bootsrapu je i několik jQuery pluginů. Bootstrap se během necelých dvou let stal velmi oblíbený. Svědčí o tom nejen stránky <http://builtwithbootstrap.com/> které shromaždují stránky postavené na základu Bootstrapu, ale i web <https://wrapbootstrap.com/> který nabízí k prodeji šablony založené opět na Bootstrapu.

# Tvorba jednoduchého jQuery pluginu

Základní kostra dokumentu v HTML 5, kterou budeme dále rozšiřovat, vypadá následovně.

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type"

content="text/html; charset=UTF-8">

</head>

<body>

</body>

</html>

Kód .: Šablona HTML dokumentu

Vytváření pluginu do jQuery je zjednodušeno, jak nejvíce to jde. Abychom mohli používat jQuery, stačí do hlavičky dokumentu přidat následující řádek.

<script src="http://code.jquery.com/jquery-latest.min.js"

type="text/Javascript"></script>

Kód . : Voložení Javascriptu.

Ten načte zdrojové kódy jQuery frameworku ze stránek jquery.com a vloží jej na naše. Má to tu výhodu, že někteří uživatelé přicházející z jiných stránek, kde jQuery také používají a tudíž tyto kódy mohou mít staženy tedy, je nebudou stahovat znova. Nyní můžeme všechny funkce jQuery používat dle libosti. Řekněme, že budeme chtít vytvořit plugin, který nám jednou narychlo ukáže nějaký text. Mohlo by to být třeba upozornění,že jsme se na stránky přihlásili.

Nejdříve budeme potřebovat prvek, který budeme zobrazovat a skrývat. V našem případě div. To je jen jednoduché ohraničení nějakého textu, či bloku kódu. Následně ho budeme pomocí Javascriptu vybírat. Náš div bude vypadat následovně.

<div>Jste přihlášen/a.</div>

Tímto jsme s HTML hotovi a můžeme se pustit do tvorby pluginu. Budem ho vkládat do hlavičky hned za načtení jQuery. Veškerý kód Javascriptu začíná párovým tagem <script> a končí </script>.

Dále bude následovat znak dolaru “$”, který odpovídá volání funkce jQuery. Má pouze funkci zkrácení zápisu. Za ním v kulatých závorkách bude argument funkce, neboli takzvaný selektor, kterým vybereme potřebný element na stránce. Nejprve je potřeba počkat, než se celá stránka stáhne k uživateli a připraví k použití. Výběrem dokumentu a zavolání funkce ready to můžeme považovat za hotové. Takové čekání je nezbytné, protože jinak bychom mohli skrývat něco, co tam není, či obráceně. Dále budeme vše psát do anonymní funkce vytvořené v argumentu funkce ready.

$(document).ready(function(){});

Plugin budeme psát do jmenného prostoru k tomu vyhrazenému frameworkem jQuery, a to Jquery.fn.jmenoPluginu, do kterého uložíme anonymní funkci s kódem pluginu. Je dobré, kvůli bezpečnosti, uzavřít ho ještě do anonymní funkce, díky které se nemusíme mít strach, že by náš plugin kolidoval s nějakou jinou knihovnou nebo pluginem.

(function($){

/\*kód našeho pluginu v anonymní funkci\*/

})(jQuery);

Upozornění uživatele pomocí jednoduchého zobrazení a skrytí textu provedeme za pomoci tří funkcí jQuery. Ze všeho nejdříve prvek skryjeme funkcí hide(), jako že tam není. Pak ho pomalu zobrazíme, slideDown(1000) a nakonec zase shováme slideUp(1000). Argumentem funkcí slide je čas v milisekundách, jak dlouho má skrývání, popřípadě zobrazování trvat. Tedy jednu vteřinu každé. Tak a plugin máme hotov. Pokud bychom v tuto chvíli zkusili stránku načíst, nic by se nestalo. Musíme ho totiž zavolat, aby věděl, že může pracovat.

$('div').helloWorldPlugin();

Následuje celý zdrojový kód, který po uložení do souboru s příponou HTML a spuštěn v prohlížeči zobrazí a zase skryje řádku textu.

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html;

charset=UTF-8">

<script src=<http://code.jquery.com/jquery-latest.min.js>

type="text/Javascript"></script>

<script>

$(document).ready(function(){

(function($){

$.fn.helloWorldPlugin = function(){

this.hide();

this.slideDown(1000);

this.delay(1000);

this.slideUp(1000);

};

})(jQuery);

$('div').helloWorldPlugin();

});

</script>

</head>

<body>

<div>Jste přihlášen/a.</div>

</body>

</html>

# Adresářová struktura pluginu

Umístění jednotlivých souborů je klíčové pro přehlednost jak moji, tak potenciálních uživatelů mého pluginu, kteří ho budou chtít implementovat do svých webových stránek.



Obrázek . : Adresářová struktura pluginu

Kořenový adresář obsahuje čtyři složky a jeden soubor. Index.php musí být vždy hned v první úrovni struktury. Nad ním je složka php, kam spadá veškerý kód v jazyce PHP. V mém případě je to pouze třída pracující s databází a soubory volané AJAXem při zobrazování, nebo uprávě stromu. Ve složce libs se nacházejí knihovny použité v pluginu. Jedná se o jQuery, jQuery UI, Twitter Bootstrap. Dále jsem použil sadu funkcí s názvem Bootstrap-Select[[8]](#footnote-8) pro úpravu vzhledu selectboxu ve stylu Twitter Bootstrap. Složka config obsahuje nastavení konstant, které se týkají PHP a přístupů do databáze. Nakonci je složka assets, do které patří kaskádové styly, javascriptové soubory, obrázky. Je tam také umístěn kód mého pluginu s názvem jqTagTree.js. Jak adresářová struktura vypadá, je vidět na obrázku 6.1.

# Databáze

## Návrh použité databáze

Jeden z požadavků programu, bylo zachování stávající struktury databáze. Ta obsahuje dvě tabulky. První v sobě udržuje informace o štítku, a to jedinečné id, j viz tabulka 5.1 a v druhé jsou uloženy informace o vazbách mezi štítky, viz tabulka 5.2.

|  |  |  |
| --- | --- | --- |
| Id | Name | Uri |
| Int(11) | Text | Text |

Tabulka . : struktura tabulky se štítky

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Subject\_id | Object\_id | Predicate\_id |
| Int(11) | Int(11) | Int(11) | Int(11) |

Tabulka . : struktura tabulky s vazbami mezi štítky

Dalším požadavkem byla správa predikátů. K tomu bylo nezbytné přidat ještě jednu databázovou tabulku, viz tabulka 5.3.

|  |  |
| --- | --- |
| Id | Name |
| Int(11) | text |

Tabulka . : struktura tabulky s predikáty

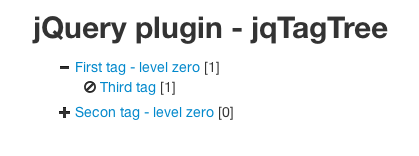
## Záměna za vlastní

## Práce s databází

Pro práci s databází Soubory se zdrojovými kódy obsahují

# Plugin jqTagTree

Testovacími daty naplněný plugin, je vidět na obrázku 8.1.



Obrázek . : Plugin jqTagTree s testovacími daty

## Před vykreslením stromu

Ihned po načtení stránky, kde má plugin jqTagTree běžet, se provede změna nastavení proměnných, které uživatel zadal při volání pluginu. Lze takto změnit použité ikony, cesty k PHP souborům a barvy, nebo doby zvýraznění právě přidaného prvku. Následuje zjištění kořenového tagu, vložení potřebného HTML a zapnutí kontextového menu.

## Vykreslování stromu

Najít způsob jak vykreslovat strom, který je zadaný v databázi nebylo jednoduché. Pokud bych chtěl načíst celou databází do prohlížeče najednou, uživatel by dlouhou dobu viděl pouze informaci o načítání stromu a po nějaké době závislé na velikosti databáze a rychlosti připojení k ní, by se mu zobrazil celý strom. Tento způsob nemusí být vždy špatný. Pro tento případ to je ale krajně nevhodné. Mnohem lepší způsob je dynamické načítání stromu po jednotlivých úrovních. Kliknutí na nějaký prvek se z databáze načtou jeho přímí potomci.

Inspiraci ke tvorbě stromu jsem získal ve knize jQuery novice to ninja. Funguje na principu duplikace šablony. Potřebné HTML vložené před začátkem vykreslování, je právě tato šablona, která bude později duplikována. Viz kód 8.1.

<**li** style='display: none' class='tagNameTemplate liId'>

<**div** style='position: relative'>

<**a** href='#' class='showUls name' ></**a**>

<**span** class='jqttNumOfPredic'></**span**>

<**img** src='"+JQTT.globUserVar.iconLoadingPath+"'

class='hidden'>

</**div**>

</**li**>

Kód . : Šablona prvku stromu

Po zkopírování šablony do pomocné proměnné se musí upravit pro zobrazení ve stromu. Nejprve je položce nečíslovaného seznamu odebrána třída tagNameTemplate a přidán atribut id s jedinečným identifikátorem id, který odpovídá záznamu v databázi. Dále je cíl odkazu nasměrován na adresu, kam má odkazovat. Mezi tagy <span> a </span> je vloženo číslo s počtem predikátů navázaných na daný prvek stromu. Značka img je většinu času skryta. Jedná se o obrázek načítání potomků objektu, na který bylo kliknuto. Je zobrazen pouze při čtení z databáze a ihned po načtení je skryt. Jelikož je tento vzor vkládán pomocí Javascriptu samotným pluginem, může zde být i řetězec JQTT.globUserVar.iconLoadingPath, zastupující proměnnou nastavitelnou při spouštění pluginu. Je v ní uložena cesta k souboru formátu \*.gif.

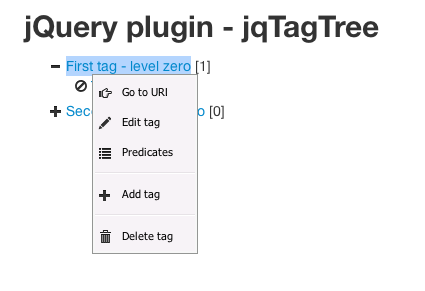
Součástí dotazu na databázi při načítání jedno či více prvků stromu, je i zjistit počet potomků každého z nich. Tento údaj se použije k výběru ikony vkládané do šablony ještě před odkaz. Je-li počet větší než 0, bude vybráno zobrazení ikony značící možnost otevření větve. V případě prvků, u kterých je vybrána druhá možnost, je uživatel po kliknutí na tento prvek přesměrován na adresu v odkazu.

Stejný postup dynamického načítání objektů z databáze a kopírování jedné šablony je použit i pro načtení vůbec prvního elementu stromu. Při načítání jednotlivých úrovní stromu je funkce, která to zajišťuje volána s parametrem představující unikátní identifikátor prvku. Načítá-li se první úroveň stromu, je tento parametr roven nule.

## Kontextové menu

S každým prvkem stromu je při jeho tvorbě svázána událost čekající na klinutí myši pravým tlačítkem. To vede z pravidla k vyvolání kontextového menu prohlížeče. Tato událost je však volána ještě před spuštění výchozího menu. Díky tomu lze volat vlastní funkce, menu lze nahradit vlastním a zabránit zobrazení přednastaveného. Všude jinde na stránce ovšem funguje výchozí menu beze změn.

Kontextová nabídka se obsahuje čtyři položky. V případě že element obsahuje predikáty, je přidána pátá. V pluginu jsou odkazy v nabídce rozděleny do třech polí. Jedno obsahuje přechod na adresu URI a úpravu štítku, druhé položku predikátů a třetí přidávání a mazání. Těsně před vykreslením se skládá v závislosti na počtu predikátů konkrétního prvku. Na obrázku 8.2 je vidět menu včetně odkazu na predikáty.



Obrázek . : Kontextová nabídka s predikáty

Jakmile je nabídka zobrazena, je zbytek stránky pokryt neviditelným tagem <div>. Je-li kliknuto na nějakou položku menu, provede se příslušná akce. V opačném případě se zavolá funkce, který skrývá kontextovou nabídku.

## Změny položek stromu

Zmiňovaná kontextová nabídka slouží k úpravě položek zobrazeného stromu. Po kliknutí na libovolnou položku, se zobrazí modální nabídka nabízená knihovnou Twitter Bootstrap. Tato nabídka je druhou šablonou vkládanou před vykreslením stromu. Oproti předchozí šabloně z tvorby stromu, která se duplikovala, tato je pouze jedna. Její data se pozměňují tak, aby vyhovovala příslušnému účelu. Kód šablony viz kód 8.2. Všechny změny stromové struktury probíhají jak na straně klienta v prohlížeči pomocí Javascriptu, tak na straně serveru pomocí technologie AJAX a jazyka PHP. Každá změna je interně rozdělena do tří menších funkcí, z nichž každá má na starosti jinou činnost. První upravuje šablonu modálního okna pro potřeby přidávání, úpravy, mazání, nebo zobrazení predikátů. Druhá provádí validaci a výměnu zadaných dat se serverem. A nakonec třetí část mění data ve stromu.

<div id='jqttModal' class='modal hide fade' tabindex='-1'

role='dialog' aria-labelledby='myModalLabel' aria-

hidden='true'>

<div class='modal-header'>

<button type='button' class='close' data-

dismiss='modal' aria-hidden='true'>×

</button><h3 id='modalLabel'></h3>

</div>

<div class='modal-body'></div>

<div class='modal-footer'>

<img src='"+

JQTT.globUserVar.iconLoadingPath +"'

class='hidden'>

<button class='btn' id='jqttModalCancel' data-

dismiss='modal' aria-hidden='true'>Cancel

</button>

<button class='btn btn-success'

id='jqttModalSave'>

Save changes

</button>

</div>

</div>"

Kód . : Šablona modálního okna

### Přidávání

Při vkládání nových štítků do databáze je nezbytné znát jeho jméno, URI adresu a rozhodnout o jeho vazbě na daný prvek. Jméno i adresu zadává uživatel z klávesnice. Vazbu však vybírá jako jednu z řádek tabulky predikátů v databázi. Tento výběr probíhá pomocí select boxu, který je zobrazen u jména štítku.

Při úpravě okna pro přidávání před zobraze

### Mazání

### Úprava

### Zobrazení predikátů

# Tvorba ukázkové webové aplikace

# Závěr

1. [www.w3.org/Style/**CSS**/](http://www.w3.org/Style/CSS/)   [↑](#footnote-ref-1)
2. <http://www.w3schools.com/cssref/css3_browsersupport.asp> [↑](#footnote-ref-2)
3. <https://developer.mozilla.org/en-US/docs/JavaScript> [↑](#footnote-ref-3)
4. <http://trends.builtwith.com/websitelist/jQuery> [↑](#footnote-ref-4)
5. <http://trends.builtwith.com/javascript/JQuery> [↑](#footnote-ref-5)
6. <http://opensource.org/licenses/MIT> [↑](#footnote-ref-6)
7. <http://twitter.github.io/bootstrap/> [↑](#footnote-ref-7)
8. <https://github.com/silviomoreto/bootstrap-select> [↑](#footnote-ref-8)