

# Bakalářská práce

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

DOČASNÁ TITULNÍ STRANA

Jméno a příjmení: Libor Vohanka  
Osobní číslo: A09B0494P  
E-mail: [vohanka@students.zcu.cz](mailto:vohanka@students.zcu.cz)  
Akademický rok: 2012/2013

# Obsah

1. Úvod .....	4
2. Cíle projektu .....	4
3. Podobné práce .....	4
4. Použité technologie .....	4
4.1. HTML a jiné značkovací jazyky.....	4
4.2. CSS .....	4
4.3. Javascript .....	5
4.4. PHP .....	5
4.5. MySQL .....	5
4.6. Záloha dat .....	5
4.7. AJAX .....	6
4.8. JSON.....	6
4.9. Vývojová prostředí.....	6
5. Použité frameworky.....	7
5.1. jQuery .....	7
5.1.1. jQuery UI.....	7
5.2. Twitter Bootstrap .....	7
5.2.1. CSS Framework .....	7
5.2.2. Jquery pluginy .....	7
6. Databáze .....	7
6.1. Návrh použité databáze .....	7
6.2. Záměna za vlastní .....	7
6.3. Práce s databází.....	7
7. Tvorba jQuery pluginu .....	7
8. Použité techniky .....	10
8.1. Vykreslování stromu .....	10
8.2. Kontextové menu.....	10
8.3. Změny položek stromu .....	10
8.3.1. Přidávání.....	10
8.3.2. Mazání.....	10

8.3.3. Úprava .....	10
8.3.4. Zobrazení predikátů.....	10
9. Tvorba ukázkové webové aplikace .....	10
10. Závěr.....	10

# 1. 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 nepřehledné množství zásuvných modulů, neboli pluginů právě do jQuery.

## 2. Cíle projektu

## 3. Podobné práce

## 4. Použité technologie

### 4.1. 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, který se stal všeobecně velmi používaný díky možnosti definice vlastních značek. Používá se jak v offline aplikacích, tak v čistě internetových aplikacích. Stal se jedním z nejpoužívanějších jazyků pro přenos dat.

Jazyk XHTML syntaxí vychází z jazyka XML a je jen úpravou jazyka 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, většina jeho částí je již funkční a prohlížeči plně podporována. Mezi nejzajímavější rozšíření nové verze patří třeba přidání nového atributu data který může pomoci například při odesílání formulářů, nebo při výběru tagu javascriptem.

### 4.2. CSS

Kaskádové styly 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. Přímou do HTML tagu, mezi tagy `<style></style>` a třetí nepoužívanější možnost je vložit vlastnosti do externího souboru, na který se odkáže v hlavičce HTML souboru.

Podobně jako HTML 5 jsou na tom i CSS verze 3. Nejsou ještě zcela hotovy, ale velká část je již v prohlížečích podporována. Mezi nejužitečnější vlastnosti pro designery internetových stránek patří možnost přidat elementům stíny, zaoblit rohy, nebo třeba možnost definice průhlednosti elementů.

### 4.3. Javascript

Javascript 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 v určitých případech dodá na rychlosti. Například při validaci formulářů, které mají některá data v nekorektním formátu z řetězu vynechá odesílání dat na server a rovnou zobrazí varování. Javascript 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.

### 4.4. PHP

Skriptovací jazyk PHP je zpracováván na straně serveru a je tak vhodným doplňkem při validaci formulářů k javascriptu. Od čtvrté verze lze v PHP programovat objektově. 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.

### 4.5. MySQL

MySQL je databázový systém využívající syntaxe jazyka SQL a používaný na většině webových hostingů. Od převzetí firmy MySQL firmou Oracle na podzim roku 2010 je nabízen jak pod bezplatou, tak pod placenou licenci. Od verze 5.0 podporuje například i uložené procedury, trigery a pohledy.

### 4.6. 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. Druhý systému pro správu verzí je více: CSV, SVN, nebo GIT. Vzhledem k rozšířenosti GITu v poslední době a hlavně díky službě <http://github.com>, kde lze umístit a verzovat jakýkoli kód zdarma, pokud je veden jako open source, 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.

## 4.7. 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.

## 4.8. JSON

Komunikace mezi javascriptem a PHP může probíhat pomocí již zmiňovaného XML. Další možností je JSON, tomu lze předat v podstatě jakýkoli objekt, nebo pole objektů. Serializace probíhá způsobem.

```
"nazevPromenne": "hodnotaPromenne", "dalsiNazev": "dalsiHodnota"
```

Zřejmě největší výhoda oproti XML je menší velikost přenášených dat. Další výhodou je jednoduchost interpretace dat přijatých pomocí JSON. PHP dokonce obsahuje převaděč mezi polem a JSON strukturou. Tuto funkci má také v sobě obsaženu framework jQuery pro javascript. Z těchto důvodů jsem si pro přenos dat vybral právě JSON.

## 4.9. 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 například Textmate. Obě tato prostředí rozpoznají syntaxi velkého množství skriptovacích, značkovacích a programovacích jazyků. Mimo jiné také Javasript, HTML, CSS a PHP, které se z pravidla používají při tvorbě jQuery pluginů. Já pro práci na svých projektech používám Netbeans. Ten umí krom zvýrazňovat klíčových slov, také doplňovat názvy proměnných, funkcí a podobně. Má také správu projektů a další funkce, které usnadňují vývoj.

## 5. Použité frameworky

### 5.1. jQuery

#### 5.1.1. jQuery UI

### 5.2. Twitter Bootstrap

#### 5.2.1. CSS Framework

#### 5.2.2. JQuery pluginy

## 6. Databáze

### 6.1. Návrh použité databáze

### 6.2. Záměna za vlastní

### 6.3. Práce s databází

## 7. Tvorba jQuery pluginu

Základní šablona dokumentu HTML ve verzi 5 vypadá následovně.

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type"
          content="text/html; charset=UTF-8">
  </head>
  <body>
  </body>
</html>
```

Tu budeme dále rozšiřovat o další kód.

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>
```

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 znovu.

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();
```



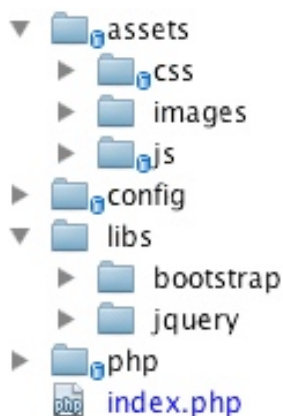
Následuje celý zdrojový kód, který po uložení do souboru s příponou HTML a spuštění v prohlížeči zobrazí a zase skryje řádku textu.

```
<!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>
```



Obrázek 4.1: Adresářová struktura

```
<body>
  <div>Jste přihlášen/a.</div>
</body>
```

</html>

## 8. Použité techniky

### 8.1. Vykreslování stromu

### 8.2. Kontextové menu

### 8.3. Změny položek stromu

#### 8.3.1. Přidávání

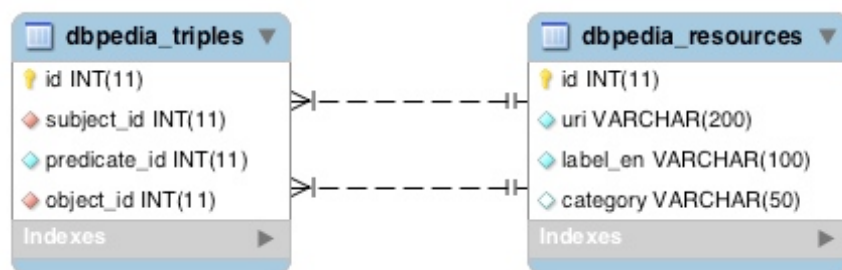
#### 8.3.2. Mazání

#### 8.3.3. Úprava

#### 8.3.4. Zobrazení predikátů

## 9. Tvorba ukázkové webové aplikace

## 10. Závěr



Obrázek 4.2: ERA model databáze

