

jQuery - prototyp pluginu pro rezervační systém

KIV/PRJ5 - Dokumentace

| | |
|-----------------|-------------------------------|
| student: | <i>Petr Kukrál</i> |
| studijní číslo: | <i>A10B0618P</i> |
| email: | <i>kukral@students.zcu.cz</i> |
| datum: | <i>9. 12. 2012</i> |

Obsah

| | | |
|-------|--|---|
| 1 | Úvod..... | 1 |
| 1.1 | Cíl práce | 1 |
| 2 | Použití pluginu | 1 |
| 2.1 | Konfigurace:..... | 1 |
| 2.1.1 | Vstupu:..... | 1 |
| 2.1.2 | Používaných tříd | 2 |
| 2.1.3 | Události | 2 |
| 2.1.4 | Nastavování obrázků..... | 3 |
| 2.1.5 | Legenda | 4 |
| 2.2 | Vložení do HTML stránky: | 5 |
| 2.2.1 | Vložení jQuery, pluginu a konfigurace do stránky | 5 |
| 2.2.2 | Potřebné prostředí v HTML stránce..... | 5 |
| 3 | Vzhled pluginu | 6 |
| 4 | Literatura | 7 |
| 5 | Příloha | 8 |
| 5.1 | Kód | 8 |
| 5.1.1 | Příklad připojení skriptů a jejich konfigurace | 8 |
| 5.1.2 | Ukázka generování rezervovaných elementů | 8 |
| 5.1.3 | Ukázka vytvoření potřebného prostředí v HTML stránce | 8 |
| 5.1.4 | Defaultí funkce pro přejetí myši nad elementem | 9 |
| 5.1.5 | Defaultí funkce po opuštění elementu myší..... | 9 |
| 5.2 | Konfigurace v cvs – ukázka vstupu..... | 9 |
| 6 | Závěr | 9 |

1 Úvod

Toto zadání jsem si vybral proto, že se zajímám o problematiku webových stránek. Neznal jsem knihovnu jQuery ani programovací jazyk javascript a lákalo mě si tuto problematiku prostudovat. V práci jsem se chtěl naučit vytvořit si složitější plugin v jazyce javascript s pomocí knihovny jQuery.

1.1 Cíl práce

Cílem je vytvořit plugin pro rezervační systém. Tento rezervační systém by měl být univerzální. Půjde použít například pro rezervační systém kina nebo rezervaci squashového kurtu. Systém bude disponovat velkým množstvím nastavení, aby bylo dosaženo co největší univerzálnosti. Vstupní data mají být ve formátu CSV (Comma-separated values) a mají se načítat ze souboru. Velký důraz bude kladen na vzhled celého pluginu.

2 Použití pluginu.

2.1 Konfigurace:

Zde je popis veškerého nastavení pluginu.

2.1.1 Vstupu:

2.1.1.1 input

Popis: Vstup informací o elementech a sériích do pluginu. Tento vstup zahrnuje nejdůležitější část konfigurace. Právě zde řeknete, kolik sedaček má být v jaké řadě (v případě že použijete plugin například pro kino), jak se má jaká řada jmenovat a kolik jich bude. Příklad jak může vstup vypadat naleznete v příloze v sekci 5.2.

Typ: řetězec

Automaticky (ve formátu cvs): "1,1,1,2,,1,1,1,1,,2,,,"

Příklad vstupu (pro xls): Naleznete v příloze 5.2.

Poznámka:

Značky v souboru se rozdělují do čtyř skupin.

1 znázorňuje jeden element. Například u kina je to jednosedáčka, u kurtu to může být jeden hráč.

2 znázorňuje dvojelement. V kině si můžeme představit jako dvojsedačku.

- (pomlčka) znázorňuje vynechané místo. Dá se využít například k vytvoření uličky.

[název řady] (v příkladu [A]) se vygeneruje jako název řady. Hranaté závorky slouží jen jako označení řady pro script, ve scriptu se pak odstraní.

2.1.1.2 format

Popis: Formát vstupu do pluginu. Podporované formáty jsou xls, cvs nebo JSON. Zde si nastavíte, jaký typ formátu budete používat v nastavení *input*. Chcete-li konfigurovat plugin ze souboru, doporučuji formát xls nebo cvs. V případě konfigurace z databáze či z objektu javascript doporučuji použít formát JSON.

Typ: řetězec

Automaticky: cvs

Příklad vstupu: "xls"

Poznámka:

Jde-li o formát xls, musíte si dát pozor v novodobých offisech na způsob uložení. Formát xls nesmí být komprimovaný jako formát.xlsx. Příklad vstupu xls (z Microsoft Office Excel) v nešifrovaném tvaru. (můžete si ověřit spuštěním souboru z poznámkového bloku v DEMU) Příklad vstupu naleznete v příloze 5.2.

2.1.2 Používaných tříd

Tyto nastavení se týkají nastavení názvu tříd u vygenerovaných elementů. Jednoduše přizpůsobíte generované třídy vaší stránce.

2.1.2.1 name_class_element

Popis: Název třídy elementu. Tento tato třída se váže přímo k elementu. Například v kině to bude jednosedačka, ve squashy to bude hráč.

Typ: řetězec

Automaticky: "sedadlo"

2.1.2.2 name_class_without_element

Popis: Název třídy elementu pro prázdné místo. Například v kině bude tento prvek představovat uličku mezi sedadly.

Typ: řetězec

Automaticky: "vynechane_misto"

2.1.3 Události

Jde o události, které odchytní plugin. Všechny níže zmíněné události obsluhují funkce, které je možno překrýt vlastními funkcemi. Tyto funkce slouží hlavně pro vylepšení vzhledu, protože může různě měnit element, ale může posloužit i při tvoření logiky stránky.

2.1.3.1 mouse_over

Popis: Funkce, která se spustí po přejetí myši přes element. Když ponecháte defaultní, tak pracuje následovně. Po najetí kurzoru myši zprůhledňuje obrázek elementu až do úplné průhlednosti.

V příkladu funguje tak, že změní modré sedadlo (obrázek, který je navrchu a zprůhlední se) do žlutého sedadla (obrázek který je naspodu a je vidět, když je vrchní zprůhledněný)

Typ: funkce

Automaticky: mouseOverDefault

Poznámka:

Přesnou implementaci této funkce naleznete v příloze 5.1.4.

2.1.3.2 mouse_up

Popis: Funkce, která se spustí po klepnutí myši na element. Předpokládám že tato funkce bude sloužit spíše pro logiku stránky. Změna obrázku elementu při označení sedačky je uvnitř pluginu a překrytím této funkce se neovlivní.

Typ: funkce

Automaticky: mouseUpDefault

Poznámka: Tato funkce zatím nebyla jako defaultní implementována, můžete ji ale překrýt svou funkcí.

2.1.3.3 mouse_out

Popis: Funkce, která se spustí po opuštění myši element. Tuto funkci je dobré využít k navrácení do původního stavu z funkce *mouse_over*. Když ponecháte defaultní, tak ta mění průhlednost vrchního obrázku na původní (neprůhledný) a tak se element po opuštění kurzorem myši vrátí do původního stavu.

Typ: funkce

Automaticky: mouseOutDefault

Poznámka:

Tato funkce má automaticky tuto implementaci:

2.1.4 Nastavování obrázků

Plugin dává možnost nastavit si obrázky se kterými pracuje.

2.1.4.1 img_element

Popis: Cesta k obrázku elementu. V kině to bude například obrázek sedačky ve squashovém kurtu to bude obrázek hráče hráče.

Typ: řetězec

Automaticky: "images/modre_jednosedadlo.png"

2.1.4.2 img_double_element

Popis: Cesta k obrázek dvojelementu např. dvojsedačky.

Typ: řetězec

Automaticky: "images/modre_dvojsedadlo.png"

2.1.4.3 img_reserved

Popis: Cesta k obrázku rezervovaného elementu např. rezervované sedačky.

Typ: řetězec

Automaticky: "images/tmavesede_jednosedadlo.png"

2.1.4.4 img_double_reserved

Popis: Cesta k obrázku rezervovaného dvojelementu např. rezervované dvojsedačky.

Typ: řetězec

Automaticky: "images/tmavesede_dvojsedadlo.png"

2.1.4.5 img_selected

Popis: Cesta k obrázku vybraného elementu např. vybrané jednosedáčky.

Typ: řetězec

Automaticky: "images/zelene_jednosedadlo.png"

2.1.4.6 **img_double_selected**

Popis: Cesta k obrázku vybraného dvojelementu např. vybrané dvojsedačky.

Typ: řetězec

Automaticky: "images/zelene_dvojsedadlo.png"

2.1.5 **Legenda**

Tato nastavení slouží pro vytvoření legendy všech používaných obrázků. Jde tedy o element, rezervovaný element a vybraný element. Jak legenda vypadá můžete vidět v kapitole 3. *Vzhled pluginu* na obrázku 3.

2.1.5.1 **legend**

Popis: Jestli se má vypsát legenda. Tento parametr slouží na to, jestli vůbec budete legendu používat. Jestli ne, tak stačí zadat řetězec "FALSE" a legenda se vůbec nezobrazí.

Typ: řetězec - pravdivostní hodnota

Automaticky: "TRUE"

2.1.5.2 **legend_target**

Popis: Id elementu nebo třída kam se má legenda vypsát. Tedy jednoduše si nadekladujete nějaký prvek (třeba div) s třídou .row a script tam legendu vypíše.

Typ: řetězec

Automaticky: "#legend .row"

2.1.5.3 **legend_title**

Popis: Slouží jako popis legendy. Jde tedy o text, který se vedle legendy zobrazí tedy například "Toto je moje vytvořená legenda".

Typ: řetězec

Automaticky: "Legenda"

2.1.5.4 **legend_element**

Popis: Popisek elementu. V kině by se např. jednalo o obrázek sedačky v normálním stavu.

Typ: řetězec

Automaticky: "volná sedačka"

2.1.5.5 **legend_reserved**

Popis: Popisek rezervovaného elementu.

Typ: řetězec

Automaticky: "rezervovaná sedačka"

2.1.5.6 **legend_selected**

Popis: Popisek vybraného elementu.

Typ: řetězec

Automaticky: "vybraná sedačka"

2.2 Vložení do HTML stránky:

Pro vložení do HTML stránky jsou potřeba čtyři věci. Vložit knihovnu jQuery, vložit samotný plugin, vložit konfiguraci pluginu a vytvořit potřebné prostředí v HTML stránce tak, aby plugin mohl správně fungovat.

2.2.1 Vložení jQuery, pluginu a konfigurace do stránky

První vkládáme knihovnu jQuery. Je tomu tak proto, abychom mohli načíst její knihovní funkce. Dále vkládáme plugin a nakonec konfiguraci. Pro popis jednotlivých prvků konfigurace nahlédněte do kapitoly Konfigurace. Nemusíte konfigurovat vše, stačí si vybrat potřebné prvky a zbytek se nakonfiguruje automaticky. Co ale doporučuji nastavit je formát vstupu (format) a vstup samotný (input).

Ukázku celé konfigurace najdete v příloze v oddílu 5.1.1.

2.2.2 Potřebné prostředí v HTML stránce

Vytvoření prostředí ve stránce je snadné. Nejdříve musíte do stránky vložit element na který spouštíte celý plugin. Já jsem v ukázce zvolil element DIV s ID. Nad ním se pak bude plugin spouštět (jak je vidět z ukázky v příloze 5.1.1). Tento element pak zůstane prázdný, plugin si ho sám naplní.

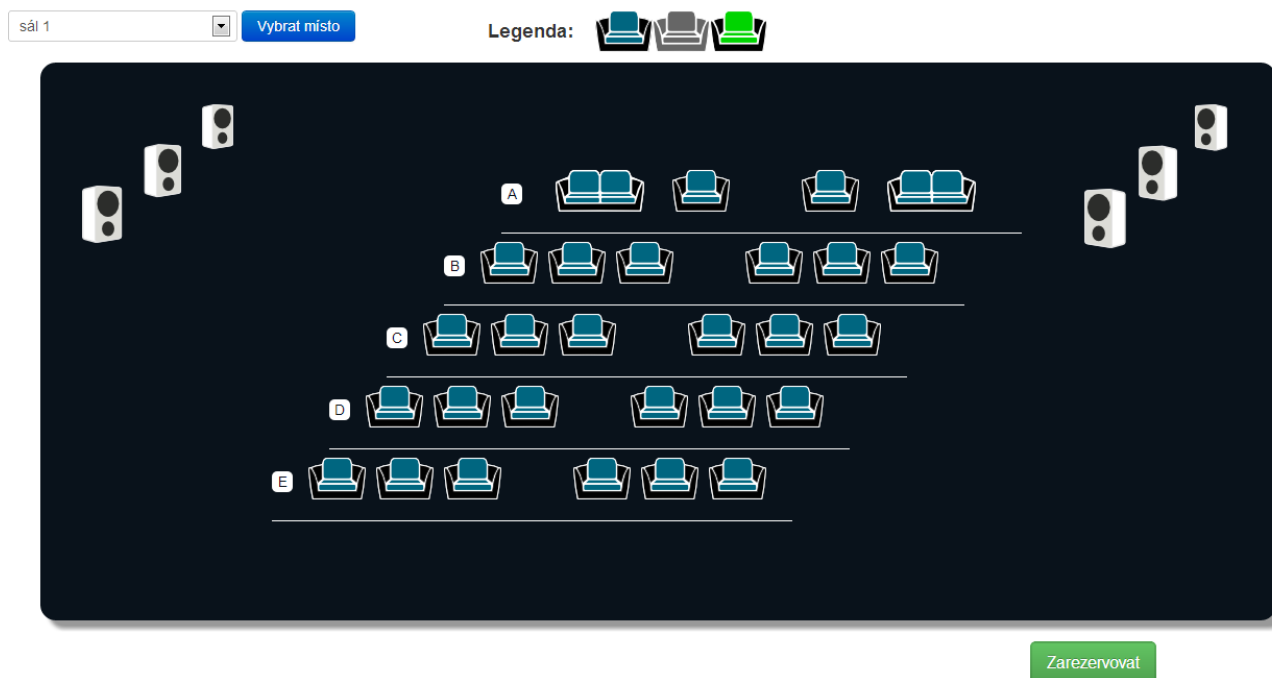
Dále je potřeba vytvořit element pro ukládání zarezervovaných sedadel. Zde je potřeba dodržet danou strukturu. Tedy vytvořit element DIV s ID reserved. Do tohoto elementu pak vložíte značky pro začátek a konec seznamu. Zde také budete generovat jednotlivé sedačky. Generují se jako prvky seznamu (značka) do kterého dáte číslo elementu, podtržítka a číslo řady. Ukázka generování elementů je v příloze 5.1.2.

Posledním důležitým prvkem je možnost, aby si plugin ukládal vybraná sedadla. Vytvořte tedy element DIV s ID selected. V něm jednoduchý formulář, který bude sloužit pro odeslání vybraných sedadel do databáze. Jak bude formulář vypadat je již na Vás. Důležité ale je, aby byl formulář přímým potomkem elementu DIV.

Ukázku vytvoření tohoto prostředí najdete v příloze v oddíle 5.1.3.

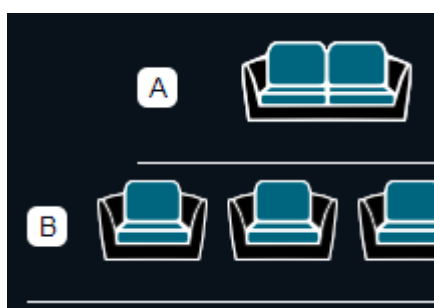
3 Vzhled pluginu

Vzhled ukáží na reservačním systému kina. Tento vzhled slouží pouze jako ukázka jedné z mnoha možností implementace pluginu.



Obrázek 1

Na obrázku jedna je vidět celkový vzhled kina. V náhledu sálu vidíme jednotlivé řady sedaček. Řady sedadel jsou posunuté do prava pro vytvoření jednoduchého prostorového efektu. Je na programátorovi, zda se rozhodne pro posun řad, jde pouze o grafický efekt. V levé a pravé části kina jsou umístěny dva obrázky s reprobednami. Toto je opět na programátorovy. Když bude někdo chtít rychle implementovat tento plugin, stačí si stáhnout demo a tyto obrázky vyměnit za jiné.



Obrázek 2

Na obrázku dva vidíme, že každá řada má svůj popisek. Tento popisek nejde v nastavení vypnout, protože se s ním plugin vypořádá sám. Stačí popisek nezadat do vstupu a plugin bude vytvářet řady bez popisku.



Obrázek 3

Obrázek 3 nám ukazuje používání legendy v pluginu. Výhodou používání legendy je, že plugin sám ukáže obrázky, které používá a programátorovi tak šetří čas. Popis nastavení legendy najdete v oddílu konfigurace.



Obrázek 4

Obrázek 4 znázorňuje, že při najetí kurzorem myši nad sedačku v legendě se ukáže její popis.



Obrázek 5

Obrázek 5 ukazuje, při najetí myší nad sedačku v kině (pomocí funkce mouse_over – více v oddílu konfigurace) sedačka postupně mění barvu z modré na žlutou. Žlutá pak zůstane po celou dobu, co je nad sedačkou kurzor myši. Dále se objeví pořadové číslo sedačky.

4 Literatura

<http://jquery.com/>

<http://knihy.cpress.cz/jquery.html>

5 Příloha

5.1 Kód

5.1.1 Příklad připojení skriptů a jejich konfigurace

```
<script type="text/JavaScript" src="js/jquery.js"></script>
<script type="text/JavaScript" src="js/jquery.cinemaPlugin.js"></script>
<script type="text/JavaScript">
$(document).ready(function () {
    $("#cinema").booking_system({
        format: "json",
        input: $("#json").text()
    });
});
</script>
```

5.1.2 Ukázka generování rezervovaných elementů

```
<div id="reserved"><ul>
    <?php $elements = Model_db::getInstance()->query("SELECT * FROM " .
$TABLE_RESERVED_ELEMENTS . " WHERE id_place=" . $place );
    while ($reserved = $elements->fetch())
    {
        echo "<li>" . $reserved->element_number . " _ " . $reserved->serie_number . "</li>";
    }
    ?>
</ul></div>
```

5.1.3 Ukázka vytvoření potřebného prostředí v HTML stránce

```
<div id="cinema"></div>

<div id="reserved"><ul></ul></div>

<div id="selected">
    <form action="save.php" method="post">
        <input type="hidden" name="id_user" value="" />
        <input type="submit" value="" />
    </form>
</div>
```

5.1.4 Defaultí funkce pro přejetí myší nad elementem

```
function mouseOverDefault(e) {  
    $(e.target)  
        .stop(true)  
        .animate({opacity: 0}, 500);  
}
```

5.1.5 Defaultí funkce po opuštění elementu myší

```
function mouseOutDefault(e) {  
    $(e.target)  
        .stop(true)  
        .animate({opacity: 1}, 200);  
}
```

5.2 Konfigurace v cvs – ukázka vstupu

| | | | | | | | |
|-----|---|---|---|---|---|---|---|
| [A] | 2 | 1 | | - | 1 | 2 | |
| [B] | 1 | 1 | 1 | - | 1 | 1 | 1 |
| [C] | 1 | 1 | 1 | - | 1 | 1 | 1 |
| [D] | 1 | 1 | 1 | - | 1 | 1 | 1 |
| [E] | 1 | 1 | 1 | - | 1 | 1 | 1 |

6 Závěr

Prací na této úloze jsem se naučil programovat v jazyce javascript a používat knihovnu jQuery. Dále jsem již schopen vytvořit si vlastní jQuery plugin.

Plugin byl navržen a implementován. K pluginu jsem též dodělal příklad, jak by se dal plugin použít. Jedná se o rezervační systém kina od výběru města po vybrání sedačky. Co se týče vývoje pluginu, kladl jsem důraz na vzhled a jeho možnost přizpůsobit ho webu. Proto si programátor, který bude tento plugin implementovat může plugin přizpůsobit svým potřebám, tedy libovolnému rezervačnímu systému. A to nejen tím, že si může vložit své obrázky (to je spíše samozřejmost), ale může si napsat své funkce pro chování pluginu po přejetí myší přes element a jednoduše je vložit v nastavení. Může se tak snadno sladit s logikou a grafickým designem webové stránky.