Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

**Dokumentace k Projektu 5**

### jQuery - prototyp pluginu pro rezervační systém

Plzeň, 2012 Petr Kukrál

# Úvod

Zadání bylo vytvořit v jazyce JavaScript plugin pro knihovnu JQuery. Plugin má sloužit jako obecnější rezervační systém a má být odzkoušen na rezervaci v kině. Vstup má být ze souboru ve formátu cvs.

## Seznámení se s problémem

Po prostudování knihy JQuery (Jquery kuchařka programátora, vydavatelství CPRESS) jsem navrhl jednoduchý plugin pro rezervaci kina tak, aby se pomocí parametrů pluginu, css a html dal použít i v dalších rezervačních systémech, jako je třeba squash.

# Tvorbu jednoduchého pluginu.

## Popis nastavení pluginu:

### format

**Popis**: Formát vstupu. Musí být xls, cvs nebo 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)

[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

### input

**Popis**: Jde o samotný vstup.

**Typ**: řetězec

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

**Příklad vstupu** (pro xls):

„[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“

**Poznámka**:

Jak je vidět z příkladu výše, značky v souboru se rozdělují do čtyř skupin.

1 znázorňuje jeden element. Například u kina je to jednosedačka, 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.

### name\_class\_element

**Popis**: Název třídy elementu pro css. V kinu to bude jednosedačka.

**Typ**: řetězec

**Automaticky**: "sedadlo"

### name\_class\_without\_element

**Popis**: Název třídy elementu pro prázdné místo pro css. V kinu to bude tedy řada.

**Typ**: řetězec

**Automaticky**: "vynechane\_misto"

### mouse\_over

**Popis**: Funkce která se spustí po přejetí myši přes element.

**Typ**: funkce

**Automaticky**: mouseOverDefault

**Poznámka**:

Tato defaultní funkce vypadá takto.

function mouseOverDefault(e) {

$(e.target)

.stop(true)

.animate({opacity: 0}, 500);

}

### mouse\_up

**Popis**: Funkce která se spustí po klepnutí myši na element.

**Typ**: funkce

**Automaticky**: mouseUpDefault

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

### mouse\_out

**Popis**: Funkce která se spustí po opuštění myší element.

**Typ**: funkce

**Automaticky**: mouseOutDefault

**Poznámka**:

Tato funkce má automaticky tuto implementaci:

function mouseOutDefault(e) {

$(e.target)

.stop(true)

.animate({opacity: 1}, 200);

}

### img\_element

**Popis**: Cesta k obrázku elementu. V kině to budenapříklad obrázek jednosedačky.

**Typ**: řetězec

**Automaticky**: "images/modre\_jednosedadlo.png"

### img\_double\_element

**Popis**: Cesta k obrázek dvojelementu např. dvojsedačky.

**Typ**: řetězec

**Automaticky**: "images/modre\_dvojsedadlo.png"

### img\_reserved

**Popis**: Cesta k obrázku rezervovaného elementu např. rezervované jednosedačky.

**Typ**: řetězec

**Automaticky**: "images/tmavesede\_jednosedadlo.png"

### img\_double\_reserved

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

**Typ**: řetězec

**Automaticky**: "images/tmavesede\_dvojsedadlo.png"

### img\_selected

**Popis**: Cesta k obrázku vybraného elementu např. vybrané jednosedačky.

**Typ**: řetězec

**Automaticky**: "images/zelene\_jednosedadlo.png"

### img\_double\_selected

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

**Typ**: řetězec

**Automaticky**: "images/zelene\_dvojsedadlo.png"

### legend

**Popis**: Jestli se má vypsat legenda. Tato legenda jsou obrázky vybraného, rezervovaného a normálního elementu. Parametry se zadávájí do dalších nastavení.

**Typ**: řetězec - pravdivostní hodnota

Automaticky: "TRUE"

### legend\_target

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

**Typ**: řetězec

**Automaticky**: "#legend .row"

### legend\_title

**Popis**: Slouží jako popisek legendy.

**Typ**: řetězec

**Automaticky**: "Legenda"

### legend\_element

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

**Typ**: řetězec

**Automaticky**: "volná sedačka"

### legend\_ reserved

**Popis**: Popisek rezervovaného elementu.

**Typ**: řetězec

**Automaticky**: "rezervovaná sedačka"

### legend\_selected

**Popis**: Popisek vybraného elementu.

**Typ**: řetězec

**Automaticky**: "vybraná sedačka"

# Nastavující JavaScript může tedy vypadat např. takto:

$(document).ready(function () {

$("#cinema").booking\_system({

format: "json",

input: $("#json").text()

});

# HTML soubor musí obsahovat tuto strukturu

<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>

Kde na div s id cinema se spouští celý plugin (spouštíte ho ve scriptu, jde měnit id na který se plugin spustí). V divu reserved pak musí být seznam. Do tohoto seznamu si můžete vygenerovat obsazené elementy. V divu selected musí být form. Tento form může obsahovat tlačítko odeslat popřípadě id uživatele, ale není to podmínkou. Do formu pak script ukládá jednotlivá vybraná místa.

# Závěr

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. Proto si programátor který bude tento plugin implementovat nejen dát své obrázky, které se budou různě měnit při přejetí myší, ale dal jsem základní metody jako je právě přejetí myší, kliknutí na element a opuštění elementu programátorovi k překrytí. Může si tedy napsat své vlastní funkce a tak grafické funkce pluginu se mohou měnit s každou implementací a mohou tak snadno sladit s grafickým designem webové stránky.