

Ce TP va vous permettre gérer les votes des internautes. L'échange d'information entre le navigateur et le serveur se fera par AJAX.

Sur le net, vous trouverez facilement beaucoup de modules en JavaScript qui permettent d'implanter un système de vote.

Arbitrairement, j'ai choisi **Rateit** disponible sur : [rateit.codeplex.com](http://rateit.codeplex.com)

Ce plugin est très compact (taille: 4.14ko en en version compactée).

Il dispose de plusieurs paramètres (sous forme d'attributs data-\*) qui permettent d'ajuster ses caractéristiques à nos besoins; voici les plus utiles:

Nom du paramètre en HTML	Nom du paramètre en JavaScript	Description
data-rateit-value	<b>value</b>	valeur initiale (identique à min par défaut)
data-rateit-min	<b>min</b>	valeur minimale pour le vote (0 par défaut)
data-rateit-max	<b>max</b>	valeur maximale pour le vote (5 par défaut)
data-rateit-step	<b>step</b>	pas (valeur par défaut: 0.5)
data-rateit-readonly	<b>readonly</b>	indique si le système de vote est en lecture seule (false par défaut)
data-rateit-resetable	<b>resetable</b>	indique s'il faut afficher le bouton de reset (true par défaut)

En consultant les pages d'exemples, vous découvrirez qu'il existe plusieurs façons d'opérer. Ainsi, les options peuvent être mises en place par 2 méthodes :

- en utilisant les attributs **data-\*** introduits par le HTML5 :

```
<div class="rateit" data-rateit-value="2" data-rateit-resetable="true" ></div>
```

- en paramétrant le plugging dans le code JavaScript :

```
$('#rateit6').rateit({ value: 2, resetable: true});
```

### CONSEIL:

Consultez la page <http://www.radioactivethinking.com/rateit/example/example.htm> pour voir (et comprendre) les exemples d'exploitation.

### Activité 1 (30 min)

/ apps / templates / filmo.tpl

Vous devez ajouter dans la page `filmo.php` un affichage du système de vote pour l'internaute.

- Téléchargez le fichier **rateit.zip** à partir de la page <http://www.radioactivethinking.com/rateit/example/example.htm>
- dézippez ce fichier
- copiez `jquery.rateit.min.js` vers `/ assets / js / vendor`
- copiez `rateit.css` vers `/ assets / css`
- dans le dossier `/ assets / img`, ajouter un nouveau dossier `/ rates`
- copiez `delete.gif` et `star.gif` vers `/ assets / img / rates`
- modifiez dans `rateit.css` tous les chemins vers les 2 fichiers image:
  - `url(star.gif) → url(../img/rates/star.gif)`
  - `url(delete.gif) → url(../img/rates/delete.gif)`

Mettez en place (inspirez-vous des exemples du site [radioactivethinking.com](http://radioactivethinking.com)), sous forme de balise HTML, le système de vote avec :

- 5 étoiles,
- par défaut aucune étoile n'est affichée,
- le bouton de reset est activé
- un bouton **Je vote**

Le visuel doit ressembler à ceci:



### CONSEIL:

Vous allez utiliser la technique "du champ caché".

La valeur du vote est placée dans une balise `input` de type `text`, mais, **grâce au traitement automatiquement** mis en place par la librairie `jquery.rateit.js`, la zone de texte est rendue invisible et le visuel avec les 5 étoiles apparaît.

Inspirez-vous du code présenté dans l'exemple 2 de la page  
adioactivethinking.com/rateit/example/example.htm

suivant pour mettre en place les étoiles :

```
<form action="{Smarty.const.ROOT}rate">
  <input type="range" min="0" max="5" value="0" step="1" id="rate{$film.id}" name="r" />
  <input type="hidden" name="idmovie" value="{ $film.id}" />
  <div class="rateit" data-rateit-backingfld="#rate{$film.id}"></div>
  <button class="rate btn btn-primary btn-xs">Je vote</button>
</form>
```

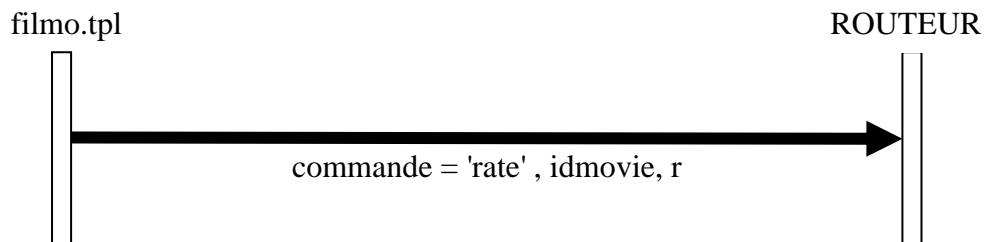
Ce qui donne dans le navigateur :



## Activité 2 ( 20 min)

Dans le routeur, il faut traiter la commande **rate** pour enregistrer le vote.

Voici le principe d'échange:



Dans la page routeur, utilisez l'instruction **debug** pour afficher le tableau `$_GET` et observer les valeurs des paramètres transmis.

### CONSEIL:

Il faut ajouter dans le ROUTEUR un traitement pour la commande **rate**.

Les votes sont enregistrés dans les champs `rates` et `crates` de la table `movies`.

- `rates` contient la moyenne des votes internautes
- `crates` contient le nombre de votes exprimés

Pour calculer la nouvelle valeur de la moyenne des votes, vous pouvez utiliser la formule suivante:

Nouvelle valeur =  $\text{rates} \cdot (1 - 1 / \text{crates}) + \text{value} / \text{crates}$

ici `value` est la valeur du vote de l'internaute

```
elseif ( 'rate' == $command )
{
  debug( $_GET, 'le vote');
  $rate = (int) $_GET['r'];
  $idmovie = (int) $_GET['idmovie'];
  // lire les informations sur le films ($rates, $crates)
  $oMovie = new Movie( $fpdo );
  extract( $oMovie->get($idmovie) );
  // mettre à jour le vote du lien
  $crates = (int)$crates + 1;
  $rates = $rates * (1 - 1 / $crates) + $rate / $crates;
  $oMovie->edit($idmovie, array('crates' => $crates, 'rates' => $rates));
}
```

Quand le traitement est terminé, une redirection doit ramener vers la page `filmo`.

## Activité 3 ( 3 min)

La page principale doit aussi afficher la **valeur moyenne courante** des votes sur un lien.

Modifiez le traitement pour obtenir le visuel suivant :

[3.67/5]



#### **Activité 4 ( 8 min)**

Il est possible d'améliorer l'ergonomie pour l'internaute en réagissant **directement** à un clic sur les étoiles afin d'enregistrer le vote sans utiliser le bouton **Je vote**

C'est une autre approche que celle mise en place à l'activité 1. Elle est basée sur les traitements JavaScript.

Pour enregistrer les votes des internautes, il faut être capable de réagir aux actions sur les étoiles.

Le plugin **rateit** met en place 3 événements que l'on peut traiter en JavaScript:

- `rated` est un événement déclenché quand l'utilisateur clique sur une étoile,
- `reset` est un événement déclenché quand l'utilisateur clique sur le bouton de reset,
- `hover` est un événement déclenché quand l'utilisateur survole une étoile

Effacez le bouton **Je vote** (en fait, le formulaire ne sera plus utilisé).

*exemple:* pour afficher un message d'alerte au clic sur une étoile, on écrit:

```
// clic sur une étoile sur toutes les div de classe .rateit
$(".rateit").on('rated', function(event){
    var $vote = $(this),
        valeur = $vote.rateit('value');
    alert('la valeur du vote actuel est: ' + valeur);
});
```

Quelques explications complémentaires:

dans la fonction, `$(this)` représente l'élément sur lequel on vient de cliquer.

`$vote.rateit()` permet de lire l'un des attributs définis dans la balise HTML (confère le tableau présenté au début du TP); par exemple, pour lire l'attribut `data-rateit-max`, on écrit `$vote.rateit('max')`.

Pour modifier un attribut, on doit écrire: `$vote.rateit( nom_de_attribut, valeur )`.

Modifiez le JavaScript de manière à ce que, quand un vote est effectué, l'internaute ne puisse plus recliquer sur les étoiles.

#### **CONSEIL:**

Utiliser l'attribut `readonly`.

Pour chaque clic, il faut connaître:

- la valeur du vote,
- l'identifiant du film qui vient d'être cliqué.

Il faut ajouter un attribut `data-rateit-idmovie` dans chaque div de vote de la page `filmo.php`.

#### **CONSEIL:**

Pour relire la valeur de l'identifiant du film en JavaScript, il faut écrire `$vote.rateit('idmovie')`

Testez.

## Activité 5 ( 5 min)

Vous devez maintenant transférer ces informations vers le serveur pour qu'elles soient enregistrées.

Vous le ferez en AJAX (*Asynchronous Javascript And XML*).

Cette technique (très utilisée aujourd'hui) permet au JavaScript d'envoyer des demandes au serveur web; les réponses du serveur (en XML ou JSON ou en texte simple) permettront de modifier le contenu de la page web.

Incontestablement, cette technique apporte un vrai confort pour l'internaute puisqu'il n'y a pas de rechargement complet de la page.

jQuery va vous simplifier la vie pour faire de l'AJAX; un objet `$.ajax()` est à disposition.

Saisir le code JS suivant:

```
$.ajax({
  url: 'index.php?ajx-rate', // la page appelée avec la commande spécifique
  data: { idmovie: idmovie, rate: valeur },
  type: 'POST', // la méthode
  // data contient les données retournées par le serveur
  success: function ( data ) {
    // ... à compléter ultérieurement
  }
});
```

Remarques :

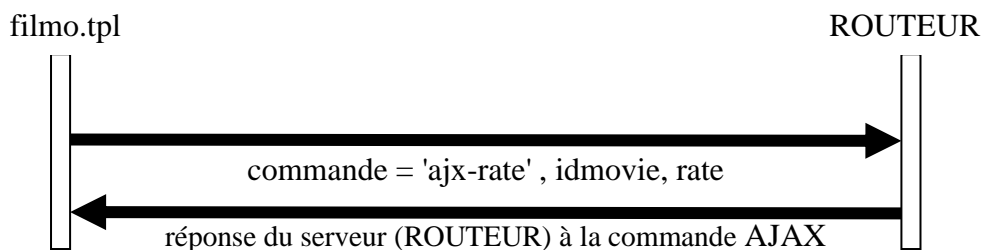
1 c'est une bonne pratique que de préfixer les commandes (ici avec l'écriture `ajx-`); de cette manière, on peut facilement distinguer les commandes en AJAX des autres.

2 il faut modifier le code de la fonction `get_command()` du fichier `/apps/lib/functions.php`; cette modification permet de prendre en compte les commandes de type AJAX.

```
/**
 * Retourne la commande de l'URL courante.
 * @return string
 */
function get_command() {

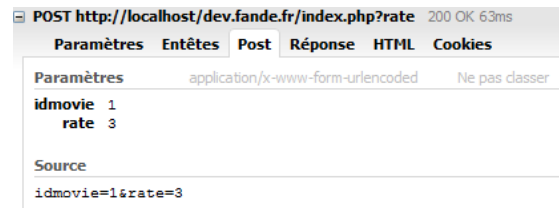
  $url = parse_url( $_SERVER['REQUEST_URI'] );
  // traitement ajax?
  if ( isset( $_SERVER['HTTP_X_REQUESTED_WITH'] ) ) {
    return $url['query'];
  } else {
    $url = str_replace( ROOT_URL, '', $url['path'] );
    // enlever le / éventuel au début
    $url = str_replace( '/', '', $url );
    return $url;
  }
}
```

Voici le principe d'échange:



Observez, avec la console sous Firefox, l'échange entre le navigateur et le serveur :

La page `index.php` est appelée avec la commande **rate** Les paramètres **idmovie** et **rate** sont transmis en méthode POST



## Activité 6 ( 5 min)

Dans le routeur `index.php`, enregistrez le vote du lien. Il faut reprendre complètement le traitement écrit à l'activité 2.

La seule différence, c'est que maintenant le serveur doit répondre à la requête AJAX en retournant la nouvelle valeur moyenne du vote :

```
elseif ( 'ajx-rate' == $command )
{
    // à compléter ...
    echo number_format( $rates, 1 );
}
```

Observez, avec la console sous Firefox, la réponse du serveur :



Remarque : la valeur dépend des différents tests que vous avez effectués.

## Activité 7 ( 15 min)

Quand le navigateur client reçoit la réponse du serveur, la fonction de traitement **succes** (callback en anglais) est déclenchée et les valeurs transmises par le serveur sont consultables dans la variable `data`.

Saisissez le code suivant et observez :

```
$.ajax({
    url: 'index.php?ajx-rate', // la page appelée
    data: { idmovie: idmovie, rate: valeur },
    type: 'POST', // la méthode
    // data contient les données retournées par le serveur
    success: function ( data ) {
        console.log( data );
    }
});
```

### CONSEIL:

Il reste maintenant à placer la valeur reçue au bon endroit sur la page (on parle du DOM du navigateur).

```
success: function ( data ) {
    $vote.parent().prev().find('span').text(data);
}
```

### Prolongement:

ATTENTION, pour le moment, vous pouvez vérifier vous-même qu'un internaute peut voter plusieurs fois pour le même lien; ceci fausse l'équité des votes.

Réfléchissez à un principe qui empêcherait cette situation (cela implique peut être une nouvelle gestion dans les tables)

---

Bon travail et ... courage !!!

---