

Notions abordées :

- ✓ Création d'une chaîne *JSON*
- ✓ Extraction d'une chaîne *JSON*
- ✓ Encodage *JSON* en *PHP*
- ✓ Décodage *JSON* en *JavaScript*

Préambule

Le but de ce TP est de comprendre le format de données *JSON* à travers des exemples simples et concrets en *JavaScript* et *PHP*. Afin de ne pas perdre de temps sur l'aspect visuel, mais aussi pour respecter les standards actuels d'affichage sur téléphone et tablette (*Responsive Design*), nous utiliserons le framework *CSS* nommé « **Bootstrap** » : <https://getbootstrap.com/docs/>

Nous utiliserons aussi par moment la bibliothèque d'icônes « **Font Awesome** » pour plus de lisibilité : <https://fontawesome.com/>

Pour démarrer, la structure du TP vous est donnée dans les ressources du TP qui sont à télécharger sur l'ENT.

Exercice1 - *PHP*

Préambule

Le but de cet exercice est de créer un ensemble de données structurées en *PHP* et de l'encoder dans une chaîne *JSON* avant de l'échanger avec la partie cliente. Pour illustrer ce mécanisme, nous utiliserons un code *JavaScript* permettant de dessiner une horloge, voir figure 1.



FIGURE 1 – Capture d'écran de l'horloge *JavaScript*.

Afin de dessiner l'horloge, le code *JavaScript* effectue une requête *AJAX* sur la ressource suivante :

GET `php/time.php` Retourne l'heure en JSON

Afin de traiter correctement la réponse, le code *JavaScript* attend une chaîne *JSON* (où l'heure est variable) ayant le format suivant :

```
["Il est : 9:35:45",{"hours":9,"minutes":35,"seconds":45}]
```

1 Traitement de la requête en *PHP*

Dans cette partie, nous souhaitons traiter la requête côté serveur en *PHP*. Pour cela, créez un script « `time.php` » dans le dossier « `php` ». Dans celui-ci, ajoutez le code nécessaire à l'extraction de l'heure courante à l'aide de la fonction *PHP* `getdate`. Puis, créez la structure de données correspondant à la chaîne *JSON* attendue par le client et encodez là. Pour finir, envoyez la chaîne encodée avec les *header* (c.-à-d. en-têtes) adéquats.

Exercice2 - *JavaScript*

Préambule

Le but de cet exercice est de décoder côté client une chaîne *JSON* reçue du serveur dans le but d'en extraire des informations. Pour illustrer ce mécanisme, nous utiliserons un code *PHP* permettant de lire les résultats de sondages stockés dans une base de données. Le code que vous devez écrire devra permettre de traiter les chaînes *JSON* envoyées par le serveur au travers d'un script *PHP* afin d'afficher les sondages, voir figure 2.

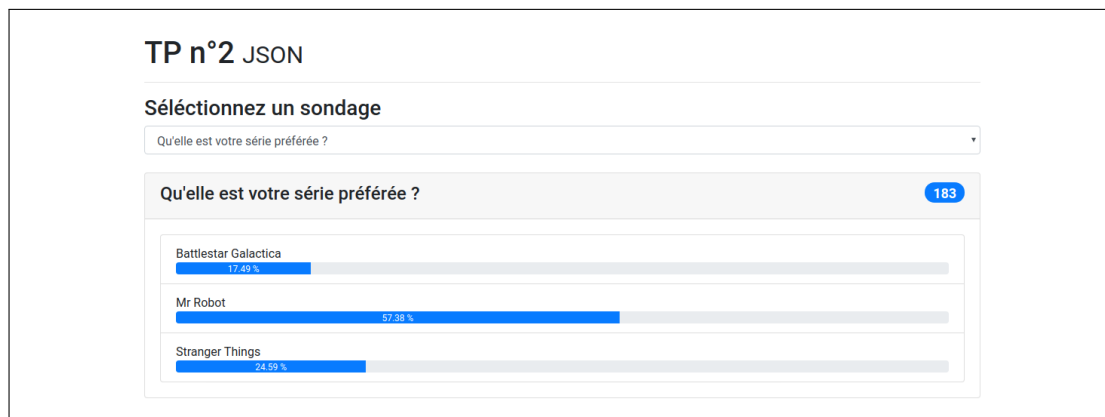


FIGURE 2 – Capture d'écran de du résultat d'un sondage.

Le script *PHP* principal disponible sur le serveur est capable de répondre à deux requêtes différentes permettant respectivement de récupérer la liste de sondages ainsi que les résultats d'un sondage en particulier.

Requête pour la liste des sondages :

GET `php/request.php/polls/` Retourne la liste des sondages

```
[{"id":"1","title":"titre 1"}, {"id":"2","title":"titre 2"}, {"id":"3","title":"titre 3"}]
```

Requête pour un sondage en particulier :

GET `php/request.php/polls/1` Retourne les informations du sondage n°1



```
{
  "id": "1",
  "title": "Titre 1",
  "option1": "Option1",
  "option2": "Option2",
  "option3": "Option3",
  "option1score": "32",
  "option2score": "105",
  "option3score": "45",
  "participants": "183"
}
```

1 Initialisation de la base de données *MySQL*

Avant toute chose, il est nécessaire de créer et d'initialiser la base de données des sondages. Pour cela, connectez-vous à l'interface *phpmyadmin* : <http://localhost/phpmyadmin>
Créez ensuite un nouvel utilisateur avec les informations suivantes :

- Login : comweb_tp
- Mot de passe : pt_bewmoc_isen29
- Client : local

Cochez la case : « créer une base portant son nom et donner à cet utilisateur tous les privilèges sur cette base. ».

Chargez ensuite le contenu du script *SQL* « sql.sql » disponible dans les ressources du TP. Les modèles conceptuel et physique de données sont fournis dans la figure 3.

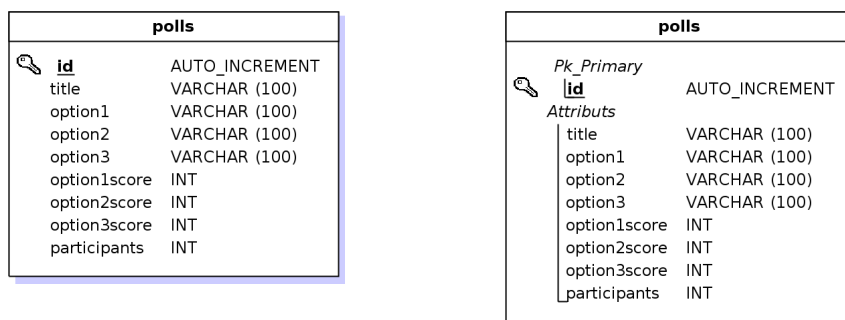


FIGURE 3 – Modèles conceptuel (à gauche) et physique (à droite) de données de la BDD mise en place.

2 Requête de la liste des sondages en *JavaScript*

Dans cette partie, nous souhaitons effectuer une requête *AJAX* permettant de récupérer la liste des sondages. Pour cela, créez un script « polls.js » dans le dossier « js » et ajoutez-le à votre page *HTML*. Vous définirez en haut de ce script le mode « strict » grâce à :

```
'use strict';
```

Pour commencer, écrivez une fonction *JavaScript* nommée `requestPolls` permettant d'effectuer la requête *AJAX* pour récupérer la liste des sondages. Si elle est valide, affichez la réponse du serveur dans votre console *JavaScript* sinon affichez le statut d'erreur. Enfin, appelez cette fonction au début de votre script. Si tout se passe bien vous devriez voir apparaître la liste des sondages dans votre console.

3 Affichage de la liste des sondages en *JavaScript*

Dans cette partie, nous souhaitons afficher la liste des sondages dans la liste déroulante d'id `polls-select` (définie dans le code *HTML*). Pour cela, écrivez la fonction suivante qui doit être appelée en cas de réponse valide dans la fonction `requestPolls` :

```
function displayPolls(polls)
```

Ajoutez à cette fonction les instructions suivantes :

- Création des options de la liste déroulante. Chaque choix aura la forme suivante :

```
<option value="??">??</option>
```

Où les premiers ?? sont à remplacer par l'id du sondage et les seconds ?? par son titre.

- Appel de la nouvelle fonction suivante pour récupérer les informations du premier sondage :

```
function requestPoll(id)
```

Celle-ci doit permettre de récupérer les informations du sondage, correspondant à l'id passé en paramètre, au travers d'une requête *AJAX*.

- Ajout d'une gestion de l'évènement `change` à la liste déroulante qui doit appeler la fonction `requestPoll` du point précédent avec l'id du sondage sélectionné. On obtient l'id de celui-ci grâce à :

```
document.getElementById('polls-select').value;
```

4 Affichage des résultats d'un sondage en *JavaScript*

Dans cette partie, nous souhaitons afficher les résultats du sondage sélectionné grâce à la fonction suivante :

```
function displayPoll(poll)
```

Elle sera appelée en cas de réponse valide dans la fonction `requestPoll`.

Ici, chaque sondage est composé de trois choix associées chacun à un score. Le nombre de participants à un sondage est aussi disponible pour calculer les pourcentages. Le code *HTML* définit les id suivants :

- `poll-title` : Emplacement du titre du sondage avec le nombre de participants.
- `poll-option1` : Emplacement pour les résultats du premier choix.
- `poll-option2` : Emplacement pour les résultats du second choix.
- `poll-option3` : Emplacement pour les résultats du troisième choix.

Vous pourrez obtenir une barre de progression avec le composant « **Progress** » de *Bootstrap* ([cliquez ici pour la documentation](#)). Pour le calcul du pourcentage, vous pouvez utiliser le code *JavaScript* suivant :

```
let score = (100*parseFloat(poll['option1score'])/parseFloat(poll.participants)).toFixed(2);
```