
TP3 : AJAX

Ayoub KARINE (ayoub.karine@isen-ouest.yncrea.fr)

Préparation :

Construction d'un lien symbolique entre le chemin du serveur local apache et un autre chemin au choix

1. Ouvrir la WSL sur VScode : Terminal > new Terminal
Puis dans le menu en bas à droite, cliquer sur "Select Default Shell" puis choisir WSL Bash
En cliquant sur "+" dans le menu en bas à droite, le terminal WSL va s'afficher
2. Dans la WSL intégré dans VS code, taper :

```
sudo ln -s /mnt/c/Users/ISEN/'Google Drive'/Enseignement/2019-2020/ComWeb_CIR2 /var/www/html
```

Suite à l'espace du dossier Google Drive, il est entre deux guillemets simples

Pour s'assurer que le lien est bien établi entre les deux chemins, taper :

```
sudo ls -l /mnt/c/Users/ISEN/'Google Drive'/Enseignement/2019-2020/ComWeb_CIR2 /var/www/html
```

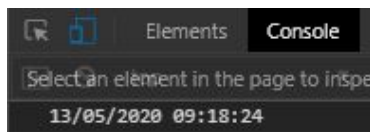
et vérifier que l'équivalent de cette ligne existe :

```
lrwxrwxrwx 1 root root 65 mai 12 18:43 ComWeb_CIR2 -> '/mnt/c/Users/ISEN/Google Drive/Enseignement/2019-2020/ComWeb_CIR2'
```

Récupération du texte :

Requête AJAX et affichage dans la console :

1. Télécharger le dossier [tp3ex1Input](#)
2. Démarrer le serveur apache
3. Ecrire une fonction en JS : `function ajaxRequest(type, url)` qui crée une requête AJAX de type "type" sur l'url "url". Une fois la réponse du serveur est obtenue. La réponse de la requête doit être affichée dans la console du navigateur
4. Tester la fonction en envoyant une requête de type GET au fichier "timestamp.php" et vérifier si l'heure s'affiche dans la console du navigateur. L'affichage dans la console doit être comme suit :



Requête AJAX et affichage statique dans la page HTML :

5. Ajouter un troisième argument à la fonction ajaxRequest : `function ajaxRequest(type, url, callback)` qui crée une requête AJAX de type "type" sur l'url "url". Une fois la réponse du serveur est obtenue, la réponse de la requête doit appeler une fonction `callback` ([Rappel sur les callback](#)). Cette fonction aura pour rôle de modifier le contenu de la section HTML d'id timestamp qui se trouve dans le fichier « index.html »
6. Tester la fonction en envoyant une requête de type GET au fichier timestamp.php et vérifier si l'heure s'affiche dans la page index.html. L'affichage dans la page index.html doit être comme suit :

AJAX

Date et heure

🕒 13/05/2020 09:19:06

Requête AJAX et affichage dynamique dans la page HTML :

7. Appeler la fonction `ajaxRequest(type, url, callback)` chaque seconde et vérifier si l'heure se modifie dynamiquement dans la page `index.html`

Traitement des erreurs:

8. Lorsqu'une erreur survient lors de la requête AJAX, l'utilisateur doit en être informé. Pour ce faire, créer une fonction `httpErrors(errorCode)` qui va être appelée en cas d'erreur. Cette fonction va afficher le message d'erreur dans l'élément `section` du fichier `index.html` qui l'ID `errors`.

Exemples de messages d'erreur :

400: 'Requête incorrecte', 401: 'Authentifiez vous', 403: 'Accès refusé', 404: 'Page non trouvée', 500: 'Erreur interne du serveur', 503: 'Service indisponible'

9. Tester la fonction en envoyant une requête de type GET au fichier `timestampp.php` et vérifier si le message d'erreur va s'afficher dans la page `index.html`. L'affichage dans la page `index.html` doit être comme suit :

AJAX

🚫 Page non trouvée

Date et heure

Récupération d'un fichier JSON :

Encodage de l'information à envoyer au code Javascript en utilisant PHP:

10. Installer l'extension JSON view sur Chrome ou Firefox

11. Implémenter un fichier PHP (time.php) qui :

- a. extrait l'heure courante à l'aide de la fonction PHP `getdate` et la stocke dans un tableau comme suit :

```
["Il est : 20:50:14",{"hours":20,"minutes":50,"seconds":14}]
```

- b. encoder le tableau en JSON
- c. afficher le résultat du tableau encodé en JSON dans le navigateur.

Exemple d'affichage :

```
[
  "Il est : 20:50:14",
  - {
    hours: 20,
    minutes: 50,
    seconds: 14
  }
]
```

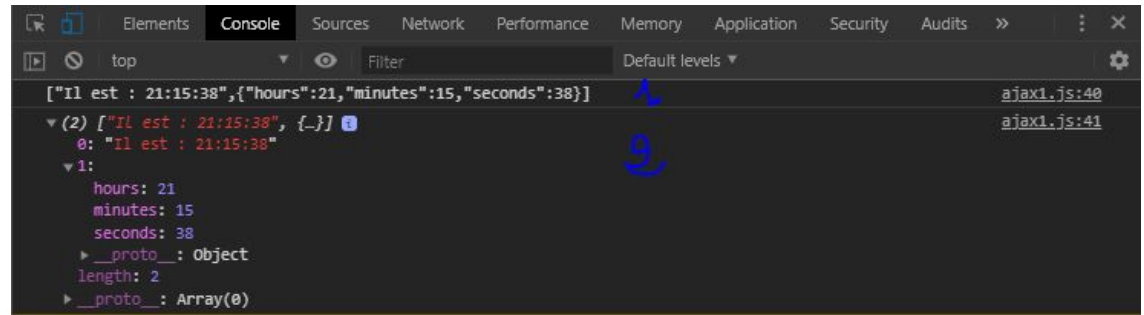
Décodage de l'information JSON en utilisant JavaScript:

12. Télécharger le dossier [tp3ex2Input](#) et déplacer time.php dans le dossier php

13. Créer un fichier "ajax.js" dans le dossier js dans lequel il est demandé de :

- a. Ecrire une fonction en JS : `function ajaxRequest(type, url)` qui crée une requête AJAX de type "type" sur l'url "url". Une fois la réponse du serveur est obtenue. La réponse de la requête doit être affichée dans la console du navigateur de deux manières : texte brute et JSON décodé = tableau
- b. Tester la fonction en envoyant une requête de type GET au fichier "time.php" et vérifier si l'heure s'affiche dans la console du navigateur (texte brute (1) et JSON décodé = tableau (2)). L'affichage

dans la console doit être comme suit :



- c. Ajouter un troisième argument à la fonction `ajaxRequest` : `function ajaxRequest(type, url, callback)` qui crée une requête AJAX de type "type" sur l'url "url". Une fois la réponse du serveur est obtenue, la réponse de la requête doit appeler une fonction `callback` ([Rappel sur les callback](#)). Cette fonction aura pour rôle de modifier le contenu du fichier "index.html", plus précisément de l'élément h4 d'id "title" et de l'élément p d'id "detail"
- d. Tester la fonction en envoyant une requête de type GET au fichier "time.php" et vérifier si l'heure s'affiche dans la page index.html. L'affichage dans la page index.html doit être comme suit :

AJAX-JSON

```
Il est : 22:43:47
*** Détail ***
hours : 22
minutes : 43
seconds : 47
```

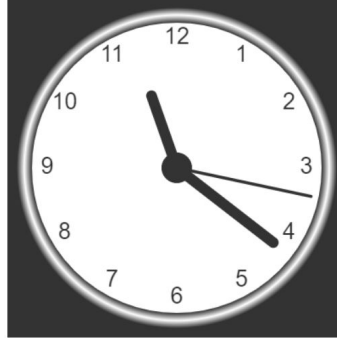
- e. Appeler la fonction `ajaxRequest(type, url, callback)` chaque seconde et vérifier si l'heure se modifie dynamiquement dans la page index.html
 - f. Traiter les erreurs en cas d'échec de la requête GET
14. Dans le fichier "clock.js" déjà existant dans le dossier tp4ex2Input:
- a. Modifier les lignes 104, 105 et 106 pour récupérer l'heure, les minutes et les secondes
 - b. appeler la fonction `drawHands` avec les arguments nécessaires dans la ligne 28
 - c. tester la fonction `displayclock` en envoyant une requête de type GET au fichier "time.php" et vérifier si l'horloge s'affiche dans la page

index.html. L'affichage dans la page index.html doit être comme suit :
AJAX-JSON

Il est : 23:23:16

*** Détail ***

hours : 23
minutes : 23
seconds : 16



- d. appeler la fonction `ajaxRequest(type, url, callback)` chaque seconde et vérifier si l'horloge se modifie dynamiquement dans la page index.html