université PARIS-SACLAY

Langages Dynamiques
Master 1 ISD

TP nº 4

Requête Wikimedia

Le but de ce TP est de fournir une page HTML permettant de rechercher des images sur Wikipedia (anglais) à partir d'un mot clé. On utilise pour cela l'API Wikimedia qui permet d'interroger la base Wikimedia (contenant entre autres les données wikipedia) de manière automatisée.

Le principe général est le suivant. On effectue une requête AJAX POST à l'URL :

```
https://www.mediawiki.org/w/api.php?format=json&formatversion=2&p1=v1&p2=v2&...&pn=vn
```

où les `pi=vi` sont des paramètres bien choisis. On obtient en réponse (que la requête soit correcte ou resulte en erreur) une chaîne de caractère représentant un objet JSON que l'on peut décoder et utiliser depuis Javascript.

Analyse *Cette partie est à réaliser au brouillon, pour se donner une idée de la complexité de la tâche.*

On suppose données deux URLs :

- `searchImage?query=str` renvoie tous les noms de fichiers d'image en rapport avec la chaîne de caractère passée en argument.
- `getURL?file=filename` renvoie l'URL du fichier donner en argument.

Par exemple, si on effectue une requête HTTP sur :

```
http://wikimedia.org/searchImage?query=str
```

on obtient :

```
File:Bread rolls.jpg
File:French Bread.jpg
File:Rye Bread.png
...
```

et si on effectue maintenant une requête HTTP sur :

```
http://wikimedia.org/getURL?file=File:Bread rolls.jpg
```

on obtient :

```
https://upload.wikimedia.org/wikipedia/commons/6/61/Bread_rolls.jpg
```

On rappelle que la seule façon de lire une URL depuis du code Javascript est de faire une requête AJAX asynchrone (i.e. qui appelle un callback lorsque la réponse du serveur est disponible).

On suppose que l'utilisateur a fourni une chaîne de caractère à chercher (par exemple « Bread ») et on souhaîte afficher toutes les images correspondantes.

Donner la suite d'action à effectuer, d'abord grossièrement, puis en détaillant de plus en plus chaque action, pour arriver à un pseudo-code exhibant les problèmes d'asynchronisme.

Implémentation par promesses

1. Ouvrir le fichier `mediawiki.js`, lire le code de la fonction `MediaWiki.ajax` et le commenter judicieusement.
2. Compléter le code de la fonction `MediaWiki.query` aux endroits indiqués. Cette fonction doit utiliser `MediaWiki.ajax` pour effectuer la requête et renvoyer une promesse qui resoudra vers un objet Javascript qui sera la résultat de la requête MediaWiki décodée.
3. Compléter la fonction `MediaWiki.getImageURL(file)`. Cette dernière prend en argument un nom de fichier (`file`). La fonction `getImageURL` effectue une requête auprès de l'API MediaWiki (en utilisant `MediaWiki.query`) avec les paramètres suivants :

   `titles` : le nom de fichier (`file`)

   `action` : la chaîne fixe `"query"`

   `prop` : la chaîne fixe `"imageinfo"`

   `iiprop` : la chaîne fixe `"url"`

   La fonction doit renvoyer une promesse Javascript se résolvant sur un objet décrivant l'URL de l'image. Tester la fonction dans la console Javascript de la manière suivante :

   ```
   MediaWiki.getImageURL("File:Voiture 1905.jpg").then((r) => {
                  console.log('OK');
                  console.log(r)
              });
   ```

   Vérifier que la fonction affiche le résultat attendu et explorer l'objet affiché dans la console pour voir comment accéder à l'URL d'une image dans un tel résultat. Donner une expression Javascript permettant d'aller chercher l'URL dans un tel objet `o`.

> **Réponse:** Voir le fichier joint. En explorant l'objet résultat, on se rend compte que pour accéder à l'URL, il faut aller chercher l'URL grace à l'expression :
>
> `o.query.pages[0].imageinfo[0].url`

4. Compléter la fonction `MediaWiki.searchImages(str)`. Cette dernière prend en argument un nom de fichier (`file`). La fonction `searchImages` effectue une requête auprès de l'API MediaWiki (en utilisant `MediaWiki.query`) avec les paramètres suivants :

   `srsearch` : la chaîne (`str`)

   `action` : la chaîne fixe `"query"`

   `srnamespace` : la chaîne fixe `"6"`

   `list` : la chaîne fixe `"search"`

   `utf8` : la chaîne fixe `"1"`

   `srlimit` : la chaîne fixe `"20"`

   Dans un premier temps, afficher simplement dans la console le résultat de la requête AJAX et constater que c'est un objet javascript dont la propriété `query` contient un tableau dont chaque élément est un objet dont la propriété `title` contient le nom de fichier recherché.

   Modifier ensuite `searchImages` pour convertir la propriété `query` mentionnée ci-dessus en un tableau d'URL menant aux images correspondantes (en utilisant `MediaWiki.getImageURL`). On ne conservera que les URLs se terminant par `svg`, `jpg`, `jpeg` ou `png` (majuscule ou minuscule).

   Votre fonction `searchImages` doit renvoyer une promesse se résolvant sur le tableau d'URLs.

5. Tester la fonction dans la console Javascript de la manière suivante :

   ```
   MediaWiki.searchImages( "Bread").then((e) => {
                                          console.log('OK');
                                          console.log(e);
                          });
   ```

Vérifier que l'on obtient bien un tableau de 20 URLs (ayant rapport avec du pain).

6. Lire le fichier `index.html` et compléter enfin le fichier `init.js`. On souhaite que lorsque le bouton `Rechercher` est cliqué (évènement `click`), alors la chaîne contenue dans la boite de texte (propriété `value` de l'objet Javascript représentant la boite de saisie) soit utilisée pour une recherche d'image et que les images trouvées soient affichées dans le `div` d'id `searchResult` (on créera pour chaque URL *u* un élément `<img src=u />`.

   La requête pouvant prendre du temps, que se passe-t'il si on clique plusieur fois rapidement sur le bouton ? proposer une solution.

> **Réponse:** Si on appuie rapidement sur le bouton, une requête peut se lancer alors que la précédente n'est pas finie. Les résultats peuvent arriver de manière asynchrone et se mélanger. On choisit donc de désactiver le bouton le temps que la requête finisse.