Haute École Bruxelles-Brabant
École Supérieure d'Informatique
Bachelor en Informatique

2020 - 2021
WEBR4-DIN

Développement Internet

Format de données JSON

JSON est un format de données dons les objectifs sont similaires à ceux de XML. Il se distingue par son format très léger.

1 Objets JavaScript

Les objets JavaScript [1] sont une structure de données permettant de stocker plusieurs informations dans une seule variable. Ces informations sont de deux types :

- Des **propriétés**, qui contiennent des valeurs
- Des **méthodes**, qui contiennent des fonctions

Contrairement à un langage typé tel que Java, il est possible de créer un objet sans définir au préalable une classe [2] qui régit la structure de cet objet.

Un objet est défini via des accolades `{}`, contenant des déclarations sous la forme `champ:valeur`, séparées par des virgules. Par exemple :

```
var person = {
    firstName: "John",
    lastName: "Doe",
    age: 50,
    eyeColor: "blue",
    fullName : function() {
        return this.firstName + " " + this.lastName;
    }
};
```

Cet objet possède quatre propriétés `firstName`, `lastName`, `age`, et `eyeColor`, et une méthode `fullName`. Notez que les propriétés et méthodes se déclarent exactement de la même façon : c'est la valeur assignée au champ qui détermine s'il s'agit d'une propriété ou méthode.

1. https://www.w3schools.com/js/js_objects.asp
2. Des classes *peuvent* être définies en JavaScript, mais leur rôle sert plutôt de générer des objets qui possèdent tous la même structure, puisque JavaScript n'est pas un langage typé.

Version du document : 2 mars 2021

L'accès aux propriétés peut se faire de deux façons `nomDeVariable.propriété` ou `nomDeVariable["propriété"]`. Ces notations sont équivalentes.

```
person.firstName; // "John"
person["firstName"]; // "John"
person.age; // 50"
```

Quant aux méthodes, il est possible d'appeler la méthode de façon similaire à un appel de fonction classique `nomDeVariable.méthode()`. Il est toutefois également possible de passer la déclaration de la fonction dans une variable en omettant les parenthèses `nomDeVariable.méthode`.

```
person.fullName // Déclaratin de la méthode
person.fullName() // Appel de la méthode : renvoie "John Doe"
```

On peut donc voir un objet JavaScript comme un dictionnaire mettant en relation des chaînes de caractère à des valeurs ou des fonctions.

2 JSON

JSON (JavaScript Object Notation) [3] est un format texte qui peut être facilement interprété comme un objet JavaScript, ou un tableau d'objets JavaScript, du moment que ces objets ne possèdent pas de méthodes. Etant un format très léger, il a tendance à supplanter XML dans les services Web et l'utilisation d'AJAX.

Les éléments constitutifs sont :

- Des objets, représentés comme dans du code JavaScript, entre accolades `{}` contenant des propriétés :
  - Un nom de propriété sous la forme d'une chaîne de caractère. Contrairement à du code JavaScript, le nom de chaque propriété **doit être noté entre guillemets** `"`.
  - Une valeur, soit :
    - Une chaîne de caractère entre guillemets `""`, avec certains caractères d'échappements possibles.
    - Un nombre
    - Un booléen `true` ou `false`
    - La valeur vide `null`
    - Un objet
    - Un tableau

  Les propriétés sont séparées par des virgules `,`
- Des tableaux, représentés par des crochets `[]` contenant des objets séparés par des virgules.

Les espaces blancs (retours à la ligne, tabulations, etc) ne sont généralement pas pris en compte.

3. https://www.json.org/

Par exemple, les données représentées dans le fichier XML suivant :

```xml
<?XML version="1.0" encoding="UTF-8"?>
<classe>
        <cours>Développement Internet</cours>
        <etudiants>
                <etudiant>
                        <no>23456</no>
                        <nom>Dupont</nom><prenom>Marc</prenom>
                </etudiant>
                <etudiant>
                        <no>24235</no>
                        <nom>Durand</nom><prenom>Hélène</prenom>
                </etudiant>
        </etudiants>
</classe>
```

Pourraient être représenté via le fichier JSON :

```json
{ "classe": {
    "cours": "Développement Internet",
    "etudiants": [
        { "no": 23456, "nom": "Dupont", "prenom": "Marc" },
        { "no": 24235, "nom": "Durand", "prenom": "Hélène" }
    ]
}
}
```

3 Utilisation de JSON en JavaScript

La plupart des langages actuels offrent un support à JSON ; nous allons aborder ici l'utilisation de JSON avec jQuery, comme il pourrait être utilisé avec AJAX.

Nous allons ici construire une page HTML dotées de trois boutons, comme ci-dessous :

```
<!DOCTYPE html>
<html>
    <head>
        <title>Exemple JSON</title>
        <meta charset="utf-8" />
        <script type="text/JavaScript"
            src="http://code.jquery.com/jquery-latest.js"></script>
        <script type="text/JavaScript" src="buttons.js"> </script>
    </head>
    <body>
        <div id="personne">
            <input type="button" id="b1" value="getPersonne">
        </div>
        <div id="personnes">
            <input type="button" id="b2" value="getPersonnes">
        </div>
        <div id="personnesAjax">
            <input type="button" id="b3" value="getPersonnesAjax">
        </div>
    </body>
</html>
```

Avec le fichier `personnes.json` :

```
[
    { "nom": "TonNom", "prenom" : "TonPrenom"},
    { "nom": "SonNom", "prenom" : "SonPrenom"}
]
```

Et le script `buttons.js` :

```
var json1 = '{ "nom": "MonNom", "prenom" : "MonPrenom"}';
var json2 = '[{ "nom": "MonNom", "prenom" : "MonPrenom"}, \
    { "nom": "SonNom", "prenom" : "SonPrenom"}]';
$(document).ready(function () {
    $("#b1").click(function () {
        var unePersonne = $.parseJSON(json1);
        $("#personne").html("<p class='personne'>"
            + unePersonne.nom + ", " + unePersonne.prenom + "</p>");
    });
    $("#b2").click(function () {
        var desPersonnes = $.parseJSON(json2);
        var liste = "<ul>";
        $.each(desPersonnes, function (idx, personne) {
            liste += '<li class="personne">' + personne.nom
                + ", " + personne.prenom
                + "</li>";
        });
        liste += "</ul>";
        $("#personnes").html(liste);
    });
    $("#b3").click(function () {
        $.ajaxSetup({ mimeType: "text/plain" });
        $.getJSON("personnes.json", function (desPersonnes) {
            var liste = "<ul>";
            $.each(desPersonnes, function (key, personne) {
                liste += "<li class='personne'>"
                    + personne.nom + ", "
                    + personne.prenom + "</li>";
            });
            liste += "</ul>";
            $("#personnesAjax").html(liste);
        });
    });
});
```

Pour les deux premiers boutons, un gestionnaire d'événement `onclick` est assigné comme nous l'avons déjà fait. La fonction `$.parseJSON()` de jQuery permet de transformer une chaîne de caractère déjà stockée dans une variable, et de transformer celle-ci en un objet JavaScript, si le format de la chaîne est un format JSON valide.

Pour le troisème bouton, la fonction de jQuery `$.getJSON()` est appelée à la place. Cette fonction, déjà vue au chapitre précédent, envoie une requête `GET` à l'adresse renseignée, reçoit une réponse qui est supposée au format JSON, et convertit celle-ci automatiquement en un objet JavaScript (en appelant implicitement la méthode `parseJSON()`).

4 Exercices

Rappel : les navigateurs modernes bloquent les requêtes AJAX pour le protocole *file*. Vérifiez la console de votre navigateur en cas de problèmes.

4.1 JSON basique

Reprenons le fichier JSON présenté plus haut :

```
{ "classe": {
    "cours": "Développement Internet",
    "etudiants": [
        { "no": 23456, "nom": "Dupont", "prenom": "Marc" },
        { "no": 24235, "nom": "Durand", "prenom": "Hélène" }
    ]
}
}
```

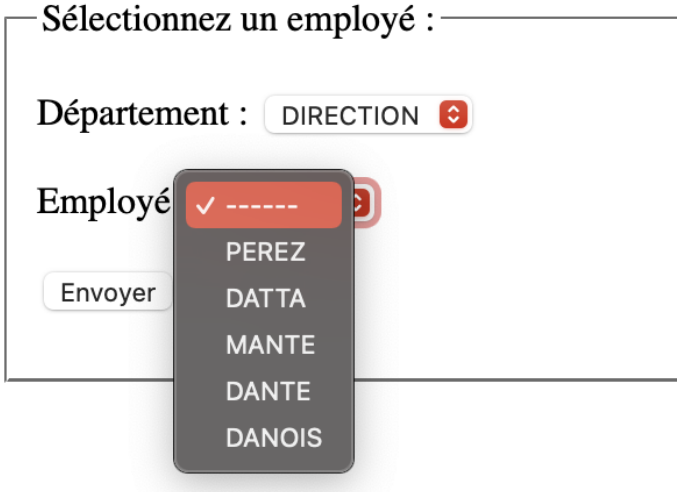
À partir de ce fichier, ajoutez un bouton à l'exemple ci-dessus qui fait apparaître lors d'un clic :

Liste des étudiants du cours Développement Internet

- g23456 Dupont, Marc
- g24235 Durand, Hélène

4.2 Retour à AJAX

Maintenant que vous savez manipuler le format JSON, écrivez un document HTML qui présente le formulaire présenté lors de l'introduction du chapitre précédent :



En utilisant les fichiers JSON fournis dans le dossier `departements` :

— Remplissez la liste des départements

— Remplissez dynamiquement la liste des employés lorsqu'un département est sélectionné dans la liste.

4.3 Recherche via AJAX et JSON

Flickr, site de partage de photographies, offre des Web-services d'accès aux photographies.

Le script `recherche_flickr.js` permet de placer dans l'élément identifié par images les 5 premières images portant les tags rue royale et Bruxelles que Flickr référencera.

Créez une page HTML et modifiez le script afin d'afficher la galere d'images reçues. Modifiez ensuite votre page et ajoutez-y un formulaire permettant d'entrer des tags de recherche manuellement.

Voici un exemple possible de présentation :