

Parser un xml en javascript et jquery

Au fil du temps j'ajoute des articles selon les difficultés rencontrées tout au long de mes développements web. Aujourd'hui ce sera comment parser un xml en Javascript aidé de jquery.

Pourquoi le faire en javascript alors qu'en php on y arrive plus facilement ? Dans le but d'économiser les ressources de votre serveur. Dans le cas ou vous parsez un xml toutes les x secondes cela peut vite encombrer votre serveur.

Pour rappel ce script fonctionne avec la librairie [jquery](#).

Dans votre fichier html prévoyez une zone d'accueil du résultat :

```
<div id="resultxml"></div>
```

Dans votre fichier js commençons par l'appel du fichier xml en jquery via la fonction ajax() :

```
$.ajax({
    type: "GET",
    url: "VotreFichier.xml",
    dataType: "xml",
    success: function(data){parseXml(data)}
});
```

Si vous aviez dans l'idée d'appeler cette fonction pour plusieurs xml en même temps nous vous conseillons d'appliquer la méthode queue afin d'exécuter les traitements les uns après les autres :

```
function getfic(MonFichierXml){
    $.ajax({
        mode: 'queue',
        type: "GET",
        url: MonFichierXml,
        dataType: "xml",
        success: function(data){parseXml(data)}
    });
    $.ajax({mode: 'dequeue'});
}
```

En cas de succès d'obtention du fichier xml, la fonction ajax va appeler la fonction appeler parseXml() :

```
function parseXml(xml){
    dataxml = null;
    $(xml).find("item").each(function(){
        title = $(this).find("title").text();
        description = $(this).find("description").text();
```

```
    dataxml += title + " " + description + "  
";  
});  
$('#resultxml').html($data);  
}
```

La fonction va parser les tags item contenus dans votre xml et récupérer le contenu des tags de votre choix, à vous de décider quoi en faire. On affichera en fin de traitement le résultat dans le div "resultxml" que l'on a créé ci-dessus.

Voilà un début de piste pour parser un xml via jquery, à vous de jouer et n'hésitez pas à me poser vos questions.