

REST

Consommer les services WEB REST via AJAX

Application : consommer l'Api...

- ◇ Consommer les services WEB REST via l'objet XMLHttpRequest
 - ◇ `requeteAjax = new XMLHttpRequest();`
 - ◇ `requeteAjax.readyState`
 - ◇ `requeteAjax.status`
 - ◇ `requeteAjax.open()`
 - ◇ `requeteAjax.send()`
 - ◇ `JSON.parse()`
- ◇ Consommer les services WEB REST via l'API Fetch



◇ API fetch

- ◇ L'API Fetch est une interface disponible nativement avec JavaScript permettant de réaliser des requêtes HTTP.
- ◇ Cette interface fonctionne de manière asynchrone et repose sur l'utilisation des promesses (Promise) de JavaScript.
- ◇ La constitution d'un appel asynchrone HTTP via Fetch se fait par la fonction `fetch(url, options)` :
 - ◇ Le paramètre `url` est une chaîne de caractères représentant la route d'accès à la ressource ciblée.
 - ◇ Le paramètre optionnel `options` est un objet JavaScript contenant les différents paramètres de la requête.
- ◇ Par défaut (sans options), l'appel à la fonction `fetch()` se fait via une requête GET.

 https://developer.mozilla.org/fr/docs/Web/API/Fetch_API

 https://developer.mozilla.org/fr/docs/Web/API/Fetch_API/Using_Fetch

API Fetch

```
function maFonction() {  
  let url = Code  
  let data = {  
    Code  
  }  
  // Entête de la requête  
  let headers = {  
    'Content-type': 'application/json; charset=UTF-8'  
  }  
  // Options de la requête  
  let options = {  
    method: Code  
    body: JSON.stringify(data),  
    header: headers  
  };  
  // Requête  
  fetch(url, options)  
  .then(function(response) {  
    return response.json();  
  })  
  .then(function(json) {  
    // Utiliser l'objet JSON  
    Code  
  })  
}
```



Options

Structure type d'une fonction
Requête AVEC options

```
function maFonction() {  
  let url = Code  
  // Requête  
  fetch(url)  
  .then(function(response) {  
    return response.json();  
  })  
  .then(function(json) {  
    // Utiliser l'objet JSON  
    Code  
  })  
}
```

Structure type d'une fonction
Requête SANS options

Par défaut (sans options),
l'appel à la fonction fetch() se
fait via une requête GET.

Méthode HTTP : GET
Récupérer les produits

```
function getProduits() {  
    let url = 'http://localhost/LabREST_03/api/produit/list/';  
    fetch(url)  
    .then(function(response) {  
        // Afficher le code status de la réponse  
        document.getElementById('result').innerHTML = 'Status : '+response.status;  
        // Récupérer le corps de la requête en JSON  
        return response.json();  
    })  
    .then(function(json) {  
        // Appel de la méthode displayProduits() en lui passant l'objet JSON  
        displayProduits(json);  
    })  
}
```

API Fetch

◇ API fetch

◇ Remarque concernant les méthodes GET et HEAD

◇ GET et HEAD ne prennent pas en charge le corps dans la requête (body).

! Uncaught (in promise) TypeError: Window.fetch: HEAD or GET Request cannot have a body.

◇ Vous pouvez utiliser `URLSearchParams` pour convertir vos paramètres en paramètres de requête à envoyer avec l'URL.

```
let nom = document.getElementById('nameproduit').value;
const params = new URLSearchParams({
  'nom': nom
});
let url = `http://localhost/LabREST_03/api/produit/listone/${params}`;
```

EVALUATION

A VOUS DE JOUER...



A vous de jouer...



PHP - REST - TP1.pdf

Objectif 5