
	Institut Supérieur des Technologies de l'Information et de la Communication Département Informatique			
Classe : LGLSI2		Module : Indexation & Web	Elément : Tech. Prog. Web	Code : ECUEF422
Enseignants : Mohamed Seifeddine AZZABI, Raoudha SOUABNI, Mouna BADIS, Akrem KHMIRI, Ferdaous JENHANI			Année universitaire : 2022-23	
TP N° 3: Fetch API GET, POST, PUT & Delete: to the Next Level				
Objectifs : <ul style="list-style-type: none">Utilisation avancée du DOMUtilisation du Fetch API pour faire des CRUD avec des API REST		Outils : VSCode, Chrome, NPM, http-server	Documentation : <ul style="list-style-type: none">https://developer.mozilla.org/fr/docs/Web/CSS/Referencewww.json.org	

Activité 1 : Fetch API & GET to the Next Level

Objectif de l'activité

Récupérer des informations à partir de différentes API REST en utilisant Fetch API JavaScript. Les informations récupérées seront ensuite affichées dans un tableau HTML créé avec Bootstrap en utilisant DOM. Les données à récupérer seront le nom, la nationalité, la date de naissance, l'âge et un fait sur l'âge à partir d'un formulaire saisi par l'utilisateur.

Documentation des API

nationalize.io

URL : `https://api.nationalize.io?name={nom}`

Method: GET

Response:

```
{
  "country": [
    {
      "country_id": "TN",
      "probability": 0.408
    },
    {
      "country_id": "EG",
      "probability": 0.178
    },
    {
      "country_id": "DZ",
      "probability": 0.126
    },
    {
      "country_id": "TZ",
      "probability": 0.089
    },
    {
      "country_id": "AE",
      "probability": 0.018
    }
  ],
  "name": "seif"
}
```

numbersapi.com

URL : `http://numbersapi.com/{mois}/{jour}/date?json`

Method: GET

	<pre>{ "text": "September 21st is the day in 1965 that Gambia, Maldives and Singapore are admitted as members of the United Nations.", "year": 1965, "number": 265, "found": true, "type": "date" }</pre>
	Reponse
agify.io	URL : <i>https://api.agify.io?name={nom}</i> Method: GET Response <pre>{ "age": 51, "count": 298106, "name": "Ali" }</pre>
numbersapi.com	URL : <i>http://numbersapi.com/{age}?json</i> Method: GET Reponse <pre>{ "text": "10 is the number of letters used in the traditional Snellen chart.", "number": 10, "found": true, "type": "trivia" }</pre>

Travail demandé

- Créer un projet web « AnalyzeMe »
- Créer un formulaire HTML (Bootstrap) avec les champs suivants :
 - Nom
 - Date de naissance
 - Bouton Analyze Me
- Dans la même page, créer un tableau HTML Bootstrap avec les colonnes suivantes :
 - Nom
 - Nationalité
 - Date de naissance
 - Comme ce jour
 - Âge
 - Fait sur l'âge
- Créer un script JavaScript qui utilise l'évènement **submit** pour déclencher une fonction **fAnalyzeMe()**

5. La fonction ***fAnalyzeMe()*** doit:
 - a. Récupérer le nom et la date de naissance
 - b. Utiliser l'API ***https://api.nationalize.io?name={nom}*** pour récupérer la nationalité la plus probable et l'afficher dans HTML
 - c. Utiliser l'API ***http://numbersapi.com/{mois}/{jour}/date?json*** pour récupérer un évènement dans cette date et l'afficher dans la cellule **Comme ce jour**
 - d. Utiliser l'API ***https://api.agify.io?name={nom}*** pour prédire l'âge et l'afficher dans HTML.
 - e. Utiliser l'API ***http://numbersapi.com/{age}?json*** pour avoir un fait sur l'âge et l'afficher dans la cellule **Fait sur l'âge**
6. Dans le script JavaScript, utilisez le DOM pour remplir le tableau avec les informations récupérées à partir des API.

Activité 2 : Fetch API & POST (CRUD with REST & DOM)

Objectif de l'exercice

Créer une application web CRUD en utilisant l'API Fetch de JavaScript et le DOM. Cette application doit comporter la création, la modification, la recherche et la suppression de produits. L'application sera créée en utilisant Bootstrap et sera divisée en quatre pages accessibles via une barre de navigation.

Travail demandé

- Créer un projet ProductApp

Partie HTML

1. Une page d'accueil (index.html), contenant:
 - a. Un tableau pour afficher les produits avec leur nom, leur description et leur prix. Les produits doivent être récupérés à partir de l'API
 - b. Une barre de recherche pour filtrer les produits par nom.
 2. Une page pour créer un nouveau produit (add.html), contenant :
 - a. un formulaire: nom, description et prix.
 - b. Les données doivent être soumis à l'aide de la méthode POST de l'API pour créer un nouveau produit.
 3. Une page pour modifier un produit existant, avec un formulaire Bootstrap prérempli avec les informations du produit à modifier. Le formulaire doit être soumis à l'aide de la méthode PUT de l'API pour mettre à jour le produit. (Pour y arriver on va utiliser les informations suivantes)
 4. Une page pour supprimer un produit existant, avec un bouton de confirmation qui envoie une requête DELETE à l'API pour supprimer le produit.
- ⇒ Dans tous les pages, une barre de navigation Bootstrap doit être utilisée pour permettre à l'utilisateur de naviguer entre les différentes pages de l'application.

Partie JavaScript

1. Lorsque l'utilisateur remplit le formulaire de création de produits et soumet le formulaire, le script doit effectuer une requête POST à l'API ***https://restful-api.dev/api/products*** avec les données saisies par l'utilisateur. Si la requête réussit, le script doit afficher un message de succès et vider le formulaire. Si la requête échoue, le script doit afficher un message d'erreur.
2. Lorsque l'utilisateur sélectionne un produit à partir de la liste de produits affichée et soumet le formulaire de modification de produits, le script doit effectuer une requête PUT à l'API ***https://restful-***

api.dev/api/products/{id} en remplaçant {id} par l'identifiant du produit sélectionné. Les données saisies par l'utilisateur doivent être incluses dans la requête. Si la requête réussit, le script doit afficher un message de succès et vider le formulaire. Si la requête échoue, le script doit afficher un message d'erreur.

3. Lorsque l'utilisateur saisit un terme de recherche et soumet le formulaire de recherche de produits, le script doit effectuer une requête GET à l'API https://restful-api.dev/api/products?q={search_term} en remplaçant {search_term} par le terme de recherche saisi par l'utilisateur. Les produits correspondant à la recherche doivent être affichés dans une liste. Si la requête échoue, le script doit afficher un message d'erreur.
4. Lorsque l'utilisateur sélectionne un produit à partir de la liste de produits affichée et clique sur le bouton de suppression, le script doit effectuer une requête DELETE à l'API <https://restful-api.dev/api/products/{id}> en remplaçant {id} par l'identifiant du produit sélectionné. Si la requête réussit, le script doit afficher un message de succès et rafraîchir la liste de produits

Documentation des API :

Méthode GET/Récupérer liste des produits	
Request	Response
https://api.restful-api.dev/objects	200
	<pre>[{ "id": "1", "name": "Google Pixel 6 Pro", "data": { "color": "Cloudy White", "capacity": "128 GB" } }, { "id": "2", "name": "Apple iPhone 12 Mini, 256G", "data": null },]</pre>
Méthode POST/Créer un produit	
Request	Response
https://api.restful-api.dev/objects	200
<pre>{ "name": "Apple MacBook Pro 16", "data": { "year": 2019, "price": 1849.99, "CPU model": "Intel Core i9", "Hard disk size": "1 TB" } }</pre>	<pre>{ "id": "7", "name": "Apple MacBook Pro 16", "data": { "year": 2019, "price": 1849.99, "CPU model": "Intel Core i9", "Hard disk size": "1 TB" }, "createdAt": "2022-11-21T20:06:23.986Z" }</pre>
Méthode DELETE/Supprimer produit	
Request	Response
https://api.restful-api.dev/objects/6	200
	<pre>{ "message": "Object with id = 6, has been deleted" }</pre>

Méthode PUT/Modifier produit

Request

<https://api.restful-api.dev/objects/7>

```
{
  "name": "Apple MacBook Pro 16",
  "data": {
    "year": 2019,
    "price": 2049.99,
    "CPU model": "Intel Core i9",
    "Hard disk size": "1 TB",
    "color": "silver"
  }
}
```

Response

200

```
{
  "id": "7",
  "name": "Apple MacBook Pro 16",
  "data": {
    "year": 2019,
    "price": 2049.99,
    "CPU model": "Intel Core i9",
    "Hard disk size": "1 TB",
    "color": "silver"
  },
  "updatedAt": "2022-12-25T21:08:41.986Z"
}
```

Code pour bloquer le submit par le navigateur

```
const FORM = document.querySelector('#create-product-form');
```

```
form.addEventListener('submit', (event) => {
  event.preventDefault();
  //le code doit être ajouter ici
})
```