**Laboratoire #2**

Requête http et asynchronisme

**Étape 1 - Ouvrir un projet Angular créé par un programmeur maladroit 📦❓**

* Réinstallez les dépendances du projet **Angular✨** fourni à l’aide de Powershell.
* Ouvrez le projet avec Visual Studio Code et exécutez l’application Web.

**Étape 2 - Monsieur, vous êtes sûr qu’on a vu cette matière ? 🤔**

* L’objectif de ce petit laboratoire sera simplement de créer une **requête HTTP** pour afficher le résultat dans le HTML de notre application Web.
  + En gros, grâce à l’API de **last fm**, nous allons afficher une liste d’artistes similaires à celui recherché par l’utilisateur.
  + Si vous souhaitez tenter ce défi 🧩 sans trop d’aide, vous pouvez vous contenter de ces explications. Sinon, il y a les indications pas-à-pas dans l’**étape** **3**.
  + Clé d’API : e34ebf8561ba7c653a21d1d99a1a0070





Une image contenant texte, outil, conception, algèbre

Description générée automatiquement**Étape 3 - Main dans la main** 🤝

* Commencez par vous rendre sur <https://www.last.fm/api> et trouvez une méthode d’API (Dans le menu de gauche) qui permettrait de « trouver des artistes similaires à celui recherché par l’utilisateur ». Indice : c’est artist.getSimilar 🙄
  + Gardez la page de la méthode ouverte de côté pour le moment.



* Préparation du projet en vue de réaliser des requêtes
  + Dans le template HTML du composant app, on a déjà un mini formulaire pour spécifier le nom d’un artiste. Dans la classe du composant, il y a une variable nommée **artist**. Liez la variable **artist** avec l’input textuel servant à écrire le nom d’un artiste. (Two-way binding)
  + **4**  Nous aurons besoin d’une certaine modification dans **app.config.ts** pouvoir faire des **requêtes HTTP**.
  + **5**  *Injectez la dépendance* nécessaire dans le constructeur du composant app.
  + Créez une autre variable de classe dans le composant app :
    - Nommez-la **similarArtists**.
    - Son type est un **tableau de string**, et on initialise la variable avec un **tableau vide**.
    - Réponse car vous le valez bien 🤗 -> **similarArtists : string[] = [];**
    - Cette variable servira à stocker touuuuuuuus les noms des artistes similaires à celui recherché.
* Requête HTTP
  + **6-8**  Créez la **requête HTTP** de type Get à l’aide de la variable de type **HttpClient** injectée précédemment.
    - Attention ! Dans ce cas-ci, la requête ne sera pas lancée au lancement de la page, mais seulement lorsque l’utilisateur appuiera sur le bouton du formulaire. **Pas besoin d’utiliser ngOnInit()** !
    - L’URL de la requête est disponible dans la page **last fm** que vous avez trouvé tout à l’heure. Normalement, il y a 2 paramètres obligatoires à glisser dans l’URL...
    - **Premier paramètre** : Le **nom de l’artiste**. Il va falloir se servir de la valeur de la variable **artist** (qui est two-way bind à un input). On va simplement la *concaténer* à l’endroit désiré dans le string de l’**URL**. (Comme ça, on peut faire varier ce qu’on demande de chercher à l’API !)
    - **Deuxième paramètre** : La **clé d’API**. Simplement la « hard-coder » à l’endroit désiré dans le string de l’**URL**. e34ebf8561ba7c653a21d1d99a1a0070
  + Maintenant que vous avez une variable qui contient le résultat de la requête (un objet JSON) … :
    - **13**  Assurez-vous que l’objet reçu par la requête n’est pas forcé d’être de type « Object ».
    - **14**  Commencez par simplement mettre l’instruction **console.log(x)** pour jeter un coup d’œil à **l’objet JSON** qui est reçu et imprimé dans la **console du navigateur**. Testez une recherche avec l’artiste « **mariah carey** » dans votre application Web. Essayez d’identifier les **chemins vers les NOMS des artistes similaires** dans l’objet JSON.
    - Notre but sera donc de remplir la variable **similarArtists**. (Le tableau de string) On souhaite ajouter TOUS les artistes similaires. Pour rappel, pour ajouter un élément à la fin d’un tableau, on utilisera la méthode monTableau.**push**(ma\_donnée). Pour vider un tableau, il suffit de faire monTableau = [];
    - **15-18**  On a donc une **boucle foreach** avec une instruction **push** à l’intérieur à rédiger pour faire ces ajouts. Voici une piste à compléter :

**for(let artist of /\* ??? \*/){**

**this.similarArtists.push(/\* ??? \*/);**

**}**

* Si vous êtes un peu plus rustique il y a également :

**for(let i = 0; i < /\* ??? \*/.length; i++){**

**this.similarArtists.push(/\* ??? \*/);**

**}**

* Affichez les données trouvées dans le **template HTML**.
  + Pas trop compliqué, on aura besoin d’un **<h3>**, d’un **<ul>** et d’un **<li>** avec un **\*ngFor**.

