

TP 2 : Développement d'une plateforme de recettes culinaires interactive

"GourmeTech"

Introduction et contexte

Le client est ravi de la première version de *GourmeTech* : la plateforme est responsive, accessible, agréable à utiliser, et répond parfaitement aux besoins des utilisateurs. Cependant, après avoir testé l'application avec un panel d'utilisateurs, il a réalisé qu'une fonctionnalité essentielle manquait à son cahier des charges initial : la **possibilité de découvrir de nouvelles recettes aléatoires**.

Afin d'enrichir l'expérience utilisateur, il souhaite que vous ajoutiez une nouvelle page permettant de **générer dynamiquement des recettes à partir d'une API externe**. L'objectif est d'intégrer l'API publique TheMealDB (<https://www.themealdb.com/api.php>) pour permettre aux visiteurs de découvrir des plats variés en un clic.

Ce TP vous permettra d'explorer l'utilisation des APIs REST, de pratiquer la manipulation de données au format JSON, et de renforcer vos compétences JavaScript autour de la gestion des requêtes HTTP avec fetch.

Objectifs pédagogiques

Au travers de ce TP, vous serez amené à démontrer votre capacité à :

- Intégrer une API tierce dans une application web
- Envoyer et traiter des requêtes HTTP en JavaScript
- Afficher dynamiquement des données issues d'une API
- Mettre à jour l'interface utilisateur de manière fluide
- Gérer les erreurs liées aux appels réseau
- Maintenir une structure de code claire et modulaire

Consignes générales

Vous allez créer une nouvelle page dans le projet GourmeTech :

```
gourmeTech/
|
|— ...
|— generer.html          # Nouvelle page pour générer des
recettes aléatoires
|
|— js/
|   |— generer.js        # Script dédié à la gestion de cette
page
```

Cette page doit permettre à l'utilisateur de :

1. Générer **une recette aléatoire** à partir de l'API TheMealDB
2. Afficher dynamiquement les informations de cette recette, par exemple :
 - Image
 - Nom du plat
 - Catégorie
 - Origine (area)
 - Liste des ingrédients avec leurs mesures
 - Instructions de préparation
3. (Optionnel) Ajouter cette recette à ses favoris via le système déjà en place

Détails techniques de l'API TheMealDB

- URL de la recette aléatoire : <https://www.themealdb.com/api/json/v1/1/random.php>
- La réponse est un objet JSON contenant un tableau `meals` avec une seule recette.
- Il faudra extraire dynamiquement les ingrédients (les clés vont de `strIngredient1` à `strIngredient20`) et leurs mesures (`strMeasure1`, etc.).

Fonctionnalités à implémenter

Interface (HTML + CSS)

- Bouton « **Générer une recette** » (avec une animation ou effet visuel si possible)
- Conteneur vide qui sera alimenté dynamiquement avec la recette
- Style cohérent avec le reste du site (thèmes, polices, responsive)
- Respect de l'accessibilité (structure sémantique, aria-labels, etc.)

JavaScript (generer.js)

- Requête fetch vers l'API de TheMealDB
- Traitement de la réponse JSON
- Construction dynamique du DOM pour afficher la recette
- Gestion des erreurs réseau ou de parsing
- (Bonus) Possibilité d'ajouter la recette aux favoris (en utilisant le localStorage comme pour les autres recettes)

Exigences spécifiques

- Utilisation exclusive de **JavaScript vanilla**
- **Aucune bibliothèque externe** autorisée
- Code structuré et **commenté**
- Gestion propre des erreurs
- Le tout doit rester **responsive et accessible**

Bonus (facultatif)

- Ajouter un **bouton pour générer une nouvelle recette sans recharger la page**
- Permettre à l'utilisateur de **sauvegarder plusieurs recettes aléatoires**
- Ajouter un **système de partage** (copier un lien ou exporter en PDF)
- Afficher un **loader animé** pendant la récupération des données

Livrables attendus

- La page generer.html fonctionnelle
- Le fichier generer.js relié à la page et bien commenté
- Le style intégré au fichier styles.css existant (ou via une classe spécifique)
- Mise à jour du dépôt GitHub avec :
 - Le nouveau code
 - Une entrée dans le README.md expliquant l'ajout de la page et comment elle fonctionne

Conseils pour réussir

- Testez les appels à l'API dans un fichier JS à part avant de l'intégrer dans la page
- Préparez une fonction utilitaire pour extraire dynamiquement les ingrédients
- Ajoutez des vérifications dans le DOM avant de générer du contenu (ex : if (data.meals))