

TP : GretaEat

ECF Blanc

Vous arrivez dans votre nouvelle boîte, et pour les besoins de celle-ci, on vous demande de maquetter puis de coder un site permettant de pouvoir proposer des idées de repas à une famille sur une semaine. En somme, un "Suggestionneur de Repas", en utilisant l'API suivante : [TheMealDB] (<https://www.themealdb.com>).

TheMealDB est un site web exposant une API de recettes de cuisine du monde entier. Plusieurs webhooks permettent de consulter cette API afin de l'intégrer à nos applications.

En utilisant **TheMealDB** API (<https://www.themealdb.com/api.php>), vous créerez les pages de notre application **GretaEat** en réalisant les exercices suivants.

Les fonctionnalités du Site

- un header avec au minimum le logo de votre app
- un footer avec au minimum votre nom et prénom
- des vignettes de repas classifiés par Catégorie / Zone géographiques / Ingrédients
- les fiches plats doivent afficher les informations du plat, les ingrédients avec les dosages, la photo, la zone géographique et la catégorie

Les contraintes techniques

- L'intégration doit se faire en HTML5 & CSS3 & JS, avec ou sans framework
- Le code devra utiliser les balises sémantiques et ne doit contenir aucune erreur, au validateur W3C HTML et CSS. (<https://validator.w3.org>)
- Le HTML, le CSS et le JS doivent être séparés et **le dossier doit être organisé**.
- Le code doit être versionné avec git et doit avoir un repo distant sur Github ou Gitlab
- Si besoin d'icônes, utilisez [**FontAwesome**] (<https://fontawesome.com/icons>) ou [**GoogleFont**] (<https://fonts.google.com/icons>)
- Votre site ne doit pas faire style "nineties"
- **Bonus** : Créez la PWA.

L'évaluation

L'évaluation portera sur :

- l'utilisation d'un HTML sémantique et adapté au référencement
- bonnes pratiques CSS et JS
- design responsive
- la lisibilité de votre code

Critères de validation :

- Chaque exercice doit être réalisé sur une page indépendante.

- La présentation doit être propre et fonctionnelle. Design simple, efficace et responsive. Vous prendrez en compte l'accessibilité utilisateur.
- Votre code devra être optimisé en termes de performances (asynchronisme).
- Votre code devra gérer les erreurs.
- Votre code devra être sécurisé : validation des inputs.
- Vous écrirez les tests nécessaires à la validation du code.

Les exercices 1 à 8 sont obligatoires. Les exercices 9 à 12 sont en bonus.

Exercice 1

Créer la page **random.html** permettant d'afficher les détails d'un plat au hasard. Utiliser le bon webhook.

Exercice 2

Créer la page **categories.html** qui liste l'ensemble des catégories.

Exercice 3

Créer la page **categorie.html** qui affiche, sous forme de vignettes, les plats appartenant à une catégorie passée en paramètre dans l'URL.

Exercice 4

Créer la page **meal.html** qui affiche les détails d'un plat, dont l'id est passé en paramètre dans l'URL. Par exemple, les détails de ce plat :

www.themealdb.com/api/json/v1/1/lookup.php?i=52772

Astuce : Comment récupérer un paramètre d'url? Utiliser le DOM.

Exercice 5

Créer la page **first-letter.html** qui filtre les plats par première lettre. L'utilisateur cliquera sur une des 26 lettres présentées pour afficher dynamiquement les plats en dessous de l'alphabet.

Exercice 6

Créer une page **search.html** contenant un champ de recherche permettant de récupérer les plats par nom. Les résultats seront présentés sous forme de vignette, sous le champ de saisie. Un message devra alerter l'utilisateur en cas de résultats vides.

Exercice 7

Faire en sorte que chaque fiche recette générée dans les pages **categorie.html**, **first-letter.html**, et **search.html** contienne un lien fonctionnel vers la page **meal.html** correspondante (exercice 4).

Exercice 8

Créer une page d'accueil **index.html** permettant de rediriger l'utilisateur vers les différentes fonctionnalités de notre application. Chaque page disposera d'un menu permettant de naviguer vers les autres pages.

Exercice 9 Bonus

Sur le modèle de la page **Catégories** (exercice 2), créer la page **areas.html** permettant de lister les zones géographiques.

Exercice 10 Bonus

Sur le modèle de la page **Catégorie** (exercice 3), créer la page **area.html** permettant de filtrer les plats par zone géographique.

Exercice 11 Bonus

Créer la page **ingredient.html** qui affiche, sous forme de vignettes, les plats appartenant à un ingrédient passé en paramètre dans l'URL.

Exercice 12 Bonus

Modifier la page **meal.html** (exercice 2) afin que la catégorie, la zone géographique et les ingrédients pointent vers les pages correspondantes des exercices 3, 10 et 11.