

Documentation Technique du Projet FreeMeal

1. Introduction

Le projet FreeMeal est une application web qui utilise l'API FreeMeal (TheMealDB) pour afficher, rechercher et filtrer des recettes de cuisine. Cette documentation technique décrit l'architecture, les fonctionnalités et les composants du projet.

2. Technologies Utilisées

- Langage: JavaScript (ES6+)
- API: TheMealDB (<https://www.themealdb.com/api.php>)
- HTML5 pour la structure
- CSS3 pour le style
- Fetch API pour les requêtes HTTP

3. Architecture du Projet

```
freemeal-project/
├── index.html      # Page principale
├── styles/
│   └── style.css   # Feuille de style principale
├── scripts/
│   └── app.js      # Script JavaScript principal
└── assets/         # Dossier pour les images/icons
```

4. Fonctionnalités Principales

4.1. Affichage des Recettes

- Récupération des recettes aléatoires au chargement
- Affichage sous forme de cartes avec:
 - Image de la recette
 - Nom du plat
 - Catégorie
 - Zone géographique

4.2. Recherche de Recettes

- Barre de recherche pour trouver des plats par nom
- Recherche en temps réel avec debounce pour optimiser les performances

4.3. Filtrage par Catégorie

- Menu déroulant pour filtrer les recettes par catégorie
- Mise à jour dynamique de l'interface

4.4. Détails des Recettes

- Affichage des détails complets lorsqu'une recette est sélectionnée:
 - Instructions de préparation
 - Liste des ingrédients avec mesures
 - Vidéo de préparation (si disponible)

5. Structure du Code JavaScript

5.1. Variables Globales

```
``javascript
const mealsEl = document.getElementById('meals');
const favoriteContainer = document.getElementById('fav-meals');
const mealPopup = document.getElementById('meal-popup');
const mealInfoEl = document.getElementById('meal-info');
const popupCloseBtn = document.getElementById('close-popup');
const searchTerm = document.getElementById('search-term');
const searchBtn = document.getElementById('search');
``
```

5.2. Fonctions Principales

```
`getRandomMeal()`
``javascript
async function getRandomMeal() {
  const resp = await
fetch('https://www.themealdb.com/api/json/v1/1/random.php');
  const respData = await resp.json();
  const randomMeal = respData.meals[0];

  addMeal(randomMeal, true);
}
```

```
}  
...
```

Récupère une recette aléatoire depuis l'API et l'affiche.

```
`getMealById(id)`  
``javascript  
async function getMealById(id) {  
  const resp = await  
  fetch('https://www.themealdb.com/api/json/v1/1/lookup.php?i=' + id);  
  
  const respData = await resp.json();  
  const meal = respData.meals[0];  
  
  return meal;  
}  
...
```

Récupérer les détails complets d'une recette par son ID.

```
#### `getMealsBySearch(term)`  
``javascript  
async function getMealsBySearch(term) {  
  const resp = await  
  fetch('https://www.themealdb.com/api/json/v1/1/search.php?s=' + term);  
  
  const respData = await resp.json();  
  const meals = respData.meals;  
  
  return meals;  
}  
...
```

Effectue une recherche de recettes par terme.

```
`addMeal(mealData, random = false)`  
``javascript  
function addMeal(mealData, random = false) {  
  const meal = document.createElement('div');  
  meal.classList.add('meal');
```

```

meal.innerHTML = `
  <div class="meal-header">
    ${random ? `<span class="random"> Recette Aléatoire </span>`
: ""}
    
  </div>
  <div class="meal-body">
    <h4>${mealData.strMeal}</h4>
    <button class="fav-btn"><i class="fas fa-heart"></i></button>
  </div>
`;

mealsEl.appendChild(meal);
}
...

```

Ajoute une recette à l'interface utilisateur.

6. Gestion des Événements

- Écouteurs pour:
 - Clic sur le bouton de recherche
 - Clic sur les boutons favoris
 - Clic sur les recettes pour afficher les détails
 - Fermeture de la popup de détails

7. Optimisations

- Debouncing pour la recherche en temps réel
- Cache local pour les recettes favorites
- Chargement asynchrone des données
- Gestion des erreurs pour les requêtes API

8. Points d'Amélioration

1. Ajout de tests unitaires
2. Pagination pour les résultats de recherche
3. Animation de chargement
4. Meilleure gestion des erreurs réseau

5. Version mobile responsive améliorée

9. Exemple d'Appel API

Requête pour une recette aléatoire:

...

GET <https://www.themealdb.com/api/json/v1/1/random.php>

...

Réponse exemple:

```json

{

"meals": [

{

"idMeal": "52772",

"strMeal": "Teriyaki Chicken Casserole",

"strDrinkAlternate": null,

"strCategory": "Chicken",

"strArea": "Japanese",

// ... autres propriétés

}

]

}

```

10. Installation et Exécution

1. Cloner le dépôt

2. Ouvrir `index.html` dans un navigateur web

3. Aucune dépendance à installer (fonctionne avec les APIs natives du navigateur)

Cette documentation couvre les aspects techniques essentiels du projet FreeMeal. Pour toute question complémentaire, n'hésitez pas à consulter le code source ou la documentation de l'API TheMealDB.