

Projet React - Pokédex interactif

Contexte du projet.....	1
Outils à votre disposition.....	2
Avant de commencer.....	2
Fonctionnalités attendues.....	2
Listing des Pokémon.....	3
Consultation des données d'un Pokémon.....	3
Gestion de l'équipe.....	3
Consultation de l'équipe.....	4
Rendu côté serveur (SSR).....	4
API légère.....	5
Utilisation.....	5
/pokemon.....	6
limit.....	6
offset.....	6
with.....	6
search.....	7
/pokemon/:slug.....	7
/type.....	7
Barèmes d'évaluation.....	7
Barème "Listing des Pokémon".....	8
Barème "Consultation des données d'un Pokémon".....	8
Barème "Gestion de l'équipe".....	8
Barème "Consultation de l'équipe".....	9
Barème "Rendu côté serveur (SSR)".....	9
Barème "Compétences transverses".....	9
Conseils, infos et astuces.....	9

Contexte du projet

Vous travaillez pour **Center Of Data Analysis** (CODA, quelle originalité 😊). Ils attendent de vous le développement d'un Pokédex interactif.

CODA vous donne un délai de **cinq jours** pour réaliser le projet. Bien entendu, ils n'attendent pas de vous que le Pokédex soit l'application la plus parfaite : ils veulent cependant qu'elle fonctionne.

L'apparence n'a pas d'importance, tant que les bonnes pratiques sont respectées.

Outils à votre disposition

- [PokéAPI](#) (pour ceux qui veulent utiliser une API complète, mais plus complexe)
- [API légère](#) (recommandée pour cet exercice)
- [Memento Dev](#)
- [Valideur W3C](#)
- [Color Contrast Checker](#)

Avant de commencer

Le projet doit être versionné sur Github. Le repository peut être laissé privé, à l'unique condition de m'en donner les accès de lecture.

Tous les commits, branches et merges seront analysés pour valider la bonne utilisation de Git (*mais ça reste un bonus 😊*).

Pseudo Github : [GauthierWebDev](#)

Pour le livrable, vous avez jusqu'à dimanche 23h59 pour me faire parvenir le lien vers votre repository en message privé sur Discord (*et autoriser l'accès en lecture si besoin*). Passé ce délai, le livrable sera considéré comme étant réalisé hors-délai.

Fonctionnalités attendues

- [Listing des Pokémon](#)
- [Consultation des données d'un Pokémon](#)
- [Gestion de l'équipe](#)
- [Consultation de l'équipe](#)
- [Rendu côté serveur \(SSR\)](#)

Listing des Pokémon

La page d'accueil doit afficher une liste de Pokémon.

Il est possible d'effectuer des recherches par nom (*français*), ainsi que de filtrer les résultats par type (*feu, acier, glace etc*).

Au clic sur un Pokémon, rediriger l'utilisateur vers la page de consultation des données du Pokémon cliqué.

Consultation des données d'un Pokémon

Il est nécessaire de pouvoir afficher sur une page dédiée toutes les informations d'un Pokémon, à savoir :

- Son apparence de face (*défaut, femelle si existante, shiny et shiny femelle si existante*)
- Son nom français
- Ses types (*un minimum, deux maximum*)
- Ses statistiques de base :
 - HP
 - Attaque
 - Défense
 - Attaque spéciale
 - Défense spéciale
 - Vitesse

Gestion de l'équipe

Un utilisateur devra pouvoir ajouter des Pokémon dans son équipe.

L'équipe ne sera pas persistante, c'est-à-dire qu'au rechargement de la page : l'équipe sera réinitialisée.

Il pourra ajouter jusqu'à six Pokémon à cette équipe.

L'utilisateur peut préciser si le Pokémon est shiny ou non.

Consultation de l'équipe

Il doit pouvoir accéder à toutes les informations de l'équipe.

Une moyenne des statistiques de l'équipe (*pour chaque statistique*) doit être affichée, avec la liste des différents types qui composent l'équipe (*par ordre de Pokémon concerné puis par ordre alphabétique*).

Exemple pour les types :

- Eau (x3), Feu (x3), Acier (x2), Glace (x1)

Rendu côté serveur (SSR)

Le Pokédex doit être rendu côté serveur, puis rendre le reste des pages dynamique.

L'outil utilisé (*Next, Vike etc*) n'a pas d'importance, tant que la première requête navigateur permet de récupérer une réponse HTML valide et complète selon la page consultée.

API légère

Étant donné que PokéAPI, bien qu'elle soit extrêmement complète, demande de mixer plusieurs requêtes entre elles et ne permette pas de faire de la recherche de Pokémon facilement (*par son nom dans une langue précise par exemple*) il peut être compliqué de mettre en place toutes les fonctionnalités attendues dans l'application.

C'est pourquoi vous avez à votre disposition une API qui se veut légère et qui permettra de réaliser toutes vos requêtes plus facilement !

L'API restera ouverte le temps de l'exercice. Par la suite, l'API sera rendue indisponible.

Aucun exemple de retour n'est communiqué, c'est à vous de déterminer (et typer 😊) le format de retour.

Utilisation

Toutes les requêtes devront avoir un jeton d'authentification (*Bearer Token*), avec cette valeur :

`advanced-pokedex-api-key-9sd1u98cvg4t98yi`

Attention à ne pas utiliser ce jeton d'authentification côté client 😊

URL de l'API : `https://pokedex.coda.memento-dev.fr`

Pour effectuer les requêtes, tout devra être fait avec la méthode GET.

/pokemon

Ce endpoint permet la récupération d'une liste de Pokémon.

Exemple : `https://pokedex.coda.memento-dev.fr/pokemon`

Vous pouvez aussi spécifier des arguments dans la querystring :

- limit
- offset
- with
- search

Tous les arguments sont facultatifs, il n'est pas nécessaire de les spécifier pour chaque requête !

limit

- Description : Nombre de résultats à récupérer
- Type : Entier
- Contraintes : 1 à 90
- Valeur par défaut : 30

offset

- Description : Décalage des résultats à récupérer
- Type : Entier
- Contraintes : >0
- Valeur par défaut : 0

with

- Description : Relations à récupérer pour chaque Pokémon
- Type : Chaîne de caractères (*valeurs séparées par virgule*)
- Associations disponibles :
 - types : Types du Pokémon
 - stats : Statistiques du Pokémon
- Exemples :
 - {endpoint}?with=types
 - {endpoint}?with=stats
 - {endpoint}?with=types,stats

search

- Description : Nom de Pokémon, partiel ou complet, à rechercher
- Type : Chaîne de caractères

/pokemon/:slug

Ce endpoint permet la récupération d'un Pokémon individuel, ainsi que le Pokémon se trouvant avant et après selon l'ordre du Pokédex. Le slug est le nom anglais du pokémon, en kebab-case.

Exemple : <https://pokedex.coda.memento-dev.fr/pokemon/pikachu>

Ce end point ne prend aucun argument autre que le slug.

/type

Ce endpoint permet la récupération de la liste de tous les types de Pokémon.

Exemple : <https://pokedex.coda.memento-dev.fr/types>

Ce end point ne prend aucun argument.

Barèmes d'évaluation

La note finale est calculée sur un maximum de **50 points**. Elle sera ensuite transposée sur une note **sur 20**, arrondie à la première décimale selon le palier le plus proche (0,5).

Exemples :

- 10,5 -> 10,5
- 10.6 -> 11
- 10,4 -> 10,5

10 points additionnels (*en dehors des 50 points*) sont présents en tant que bonus (*compétences transverses*).

Barème “Listing des Pokémon”

Total : 10 points

Objet	Points	Critères d'évaluation
Affichage des Pokémon	4	La liste des Pokémon (<i>nom et image</i>) s'affiche correctement sur la page.
Recherche	3	La recherche par nom (<i>français</i>) affiche les résultats attendus.
Filtres	3	Les Pokémon sont filtrés correctement selon le type sélectionné (<i>feu, glace, etc</i>).

Barème “Consultation des données d'un Pokémon”

Total : 10 points

Objet	Points	Critères d'évaluation
Affichage de l'apparence	4	L'apparence du Pokémon est correctement affichée et l'utilisateur peut sélectionner l'apparence parmi celles disponibles.
Affichage du nom et des types	3	Le nom et les types sont affichés.
Statistiques de base	3	Les statistiques sont affichées de manière distincte et lisible.

Barème “Gestion de l'équipe”

Total : 10 points

Objet	Points	Critères d'évaluation
Création d'une équipe	5	L'utilisateur peut ajouter jusqu'à six Pokémon dans son équipe.
Personnalisation d'une équipe	5	L'utilisateur peut supprimer un Pokémon de son équipe.

Barème "Consultation de l'équipe"

Total : **10 points**

Objet	Points	Critères d'évaluation
Affichage de l'équipe	5	La liste des Pokémon de l'équipe s'affiche correctement.
Statistiques de l'équipe	5	L'équipe affiche correctement la moyenne des statistiques et l'ensemble des types présents dans l'équipe.

Barème "Rendu côté serveur (SSR)"

Total : **10 points**

Objet	Points	Critères d'évaluation
Rendu côté serveur	5	Le serveur retourne un HTML complet, valide et fonctionnel.
Récupération des données initiales	5	Les données récupérées via l'API sont initialement effectuées avant l'affichage de la page.

Barème "Compétences transverses"

Il s'agit ici de bonus non obligatoires !

Total : **5 points**

Objet	Points	Critères d'évaluation
Utilisation de Git	5	L'utilisation de Git est comprise et correctement appliquée, avec le respect d'un gitflow conventionnel et d'un bon nommage des branches et commits.
Base de données	5	La base de données a été modélisée et permet la persistance des données pour l'équipe Pokémon (<i>commune à tous les utilisateurs, pas d'authentification pour différencier l'utilisateur qui gère l'équipe</i>).

Conseils, infos et astuces

- Les données des Pokémon sont présentes sur l'API, ne cherchez pas ailleurs !
- Certains Pokémon ont la même apparence, qu'il s'agisse d'un Pokémon femelle ou non.
- Commencez par la conception, au lieu de démarrer directement avec le code. Ça vous évitera de devoir déconstruire ce que vous avez réalisé 😊