Afpa

Secteur Tertiaire Informatique
Filière « Etude et développement »

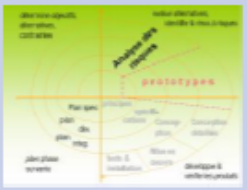
Séquence « Développer des pages Web »

Découverte du langage JavaScript

Apprentissage

Mise en pratique

Evaluation



1 Présentation

Nous allons travailler sur les données issues du site Swapi, permettant de récupérer les informations sur les films de la saga **Star Wars**.

Pour comprendre cette API, vous pouvez faire des essais sur le site (vous pourrez vous appuyer sur la documentation https://swapi.dev/documentation).

Le premier terme après l'URL https://swapi.dev/api/ est le type d'informations voulues (people pour les personnages, planets, starships, vehicles, films, species).

Le second (optionnel) est pour spécifier une entité particulière (1 pour le premier objet, et ainsi de suite).
Par exemple, la requête dev https://swapi.dev/api/films/1/ permet de récupérer le premier film dans la liste, et la requête https://swapi.dev/api/films/ permet de tous les récupérer. L'objet retourné est bien un JSON.

Pour info
Films : 6 entrées
Planètes : 60 entrées
Personnages : 82 entrées

Pour faire une recherche vous pouvez utiliser le paramètre « search » qui filtra le résultat retourné.
Cela permet des requêtes du type :
https://swapi.dev/api/people/?search=kenobi

Vous obtiendrez ainsi les informations sur Obi-Wan Kenobi.

2 Demande

a) Maquetter les formulaires

b) Formulaire index.html :

Titre d'onglet : "Star wars "
Encodage en UTF-8
Titre de la page : "Star wars : Recherche"
Style des pages :

fond gris foncé, écriture en blanc
titre en jaune, centré, taille de 2em, sur un fond noir
police d'écriture Roboto pour toute la page
largeur de page d'environ 80% de la fenêtre

Le curseur de la souris sera le gif fourni.

Le reste est assez libre.

Prévoir les éléments suivants
Un menu avec trois choix (se référer à l'initiation BootStrap de OpenClassrooms):

- Films
- Personnages
- Planètes

Afficher plusieurs photos de votre choix sur l'univers Star Wars… Rendez votre accueil attractif.

Le choix d'un élément dans le menu renverra vers une page spécifique :

People.html

Planet.html

Movie.html

c) Formulaires People, Planet, Movie

Pour garder une certaine cohérence, le css utilisé pour ces pages sera le même que pour index.html

Par défaut, chacune de ses pages renverra les dix premiers éléments de chaque endpoint :

Exemple pour les films qui n'a que 6 éléments :
https://swapi.dev/api/films/6/

Sur la partie gauche, nous aurons une liste des éléments dans un flex. Chaque élément sera affiché dans une card.

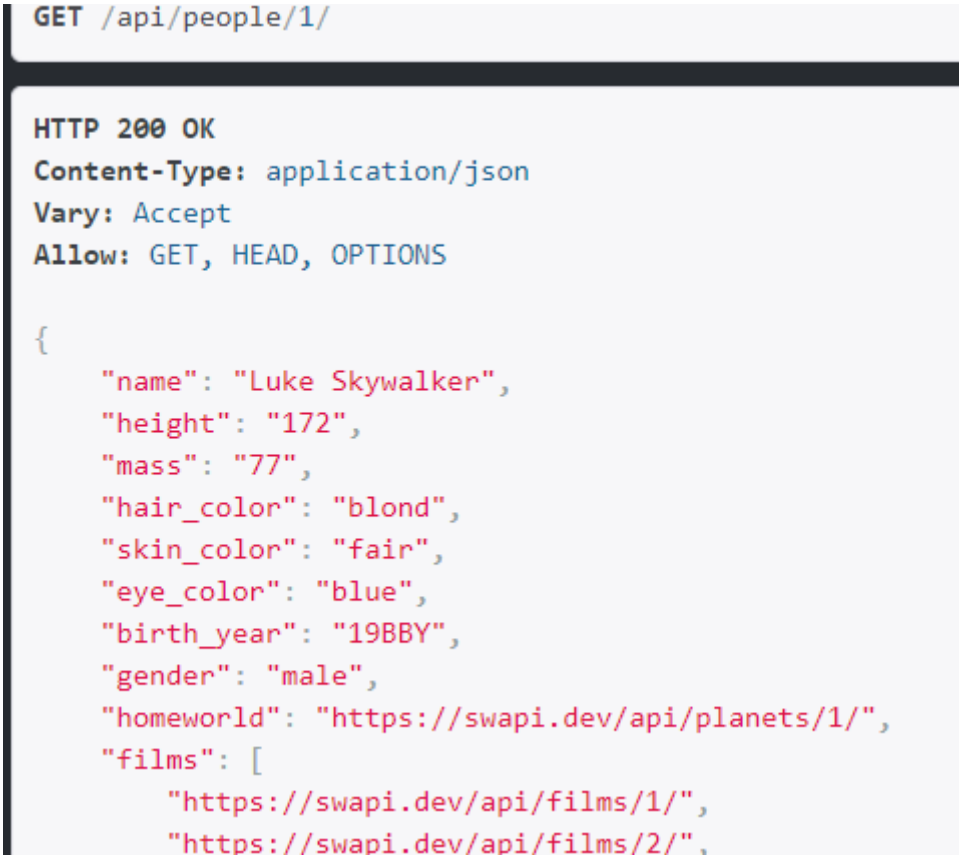
Sur la partie droite, le détail de la card sélectionnée.

Pour chaque page nous aurons une fonction Search qui fera un appel spécifique, et permettra comme son nom l'indique une recherche.

Pour chaque objet renvoyé dans le JSON nous avons parfois des éléments de ce type (renvoyant à une api)


```
{
    "title": "Revenge of the Sith",
    "episode_id": 3,
    "opening_crawl": "War! The Republic is
    "director": "George Lucas",
    "producer": "Rick McCallum",
    "release_date": "2005-05-19",
    "characters": [
        "https://swapi.dev/api/people/1/",
        "https://swapi.dev/api/people/2/",
        "https://swapi.dev/api/people/3/",
```


Prenons l'exemple des films comme spécifié via la copie d'écran.
Nous voudrons faire apparaitre uniquement dans la description (partie droite), le nom du personnage.
Donc si nous allons sur l'API du personnage 1, nous utiliserons l'attribut name à savoir Luke Skywalker.



Quand nous voudrons voir les films sur la page de personnage, nous afficherons l'attribut « title » , et pour les planètes le « name » etc.…

L'affichage de ces éléments apparaitra comme un lien, qui renverra sur une page spécifique.
Par exemple, si nous sommes sur la page People.html avec le détail de Luke Skywalker. Sur « Homeworld », nous aurons Tatooine. Le click sur Tatooine, nous renverra sur la page Planet chargée sur Tatooine…

➔ Se référer au passage de paramètres d'une page à une autre

NB : Dans ces cas là (affichage du formulaire suite à un clic sur un lien), le bouton rechercher sera invisible

Bien entendu, cela ne concerne que les informations People, Planètes et Films. Les autres informations telles que véhicules ne disposeront pas de cette fonctionnalité de « lien cliquable »

Créer **une classe API** (gère les appels aux différentes API) et **une classe Affichage** (manipulation du DOM etc.…).

Pour chacune de ces classes, un constructeur (un seul attribut à savoir le type pour people, planet ou film), des méthodes ...

Ce travail est individuel mais vous pouvez quand même vous concerter et demander de l'aide à vos collègues du moment que ça ne les empêche pas d'avancer (n'est pas jedi qui veut…).

Difficulté supplémentaire :
Le formateur n'interviendra qu'un quart d'heure par personne, par jour. Ceci afin de mettre en situation vis-à-vis du monde professionnel (votre maître de stage ou votre responsable, ne sera pas forcément toujours présent).
Préparer en amont et de façon claire vos demandes, afin de gagner du temps.

Rappels :

- Maquetter chaque page
- Créer une fonction qui aura comme paramètre une API et qui renvoie les informations (via JQuery ou JS)
- Créer les classes Affichage et API avec les constructeurs et les méthodes correspondantes (le nombre de méthodes n'est pas défini, du moment que toutes les fonctionnalités sont opérationnelles). La fonction précédemment créée sera une fonction de la classe API.
- Créer une page Accueil avec un menu avec trois choix (Personnages, Films, Planètes)
- Créer trois pages pour Personnages, Films, Planètes qui renverront les dix premiers enregistrements par défaut au chargement (sauf Films qui n'en renvoie que 6)
- Si l'attribut affiché dans votre page est un lien vers une API, afficher le premier attribut de l'objet renvoyé par le JSON à la suite de l'appel de cette API.
  Ces attributs seront des liens qui permettront d'afficher d'autres pages avec les informations correspondantes.
- Ajouter à chaque page une fonction Recherche

- *Attentions aux identations aux noms de variable (éviter d'utiliser plusieurs fois le même non dans des fonctions différentes par ex….)*

Optionnel :

- Sur chaque page, au lieu de ne renvoyer que les dix premiers enregistrements par défaut, faire une pagination (10 enregistrements par page).
- Inclure la lecture de musique via l'API Spotify (nécessite une authentification à la première lecture ; idéalement votre compte google) (https://developer.spotify.com/documentation/web-api/) dans toutes vos pages avec idéalement une musique starwars joué par défaut. Vous pouvez vous inspirer des exemples fournis.
  Ou de cette vidéo : https://www.youtube.com/watch?v=d0FFlTeyAY8