RAPPORT DÉVELOPPEMENT AVANCÉ TP2

2024

Maryam BINT IBRAHIM

SOMMAIRE

N° 01 - Étape 1 – « Avengers, rassemblement ! »

L'étape 1 de ce projet consiste à accéder à l'API de marvels pour ce faire différent processus sont mis en place à savoir la vérification du bon fonctionnement de l'API puis l'observation et la compréhension du des données récupérées.

Nous remarquons que le format de la réponse est une structure JSON.Les informations tels que "noms des personnages, leurs descriptions, l'image" sont stockées dans la partie "results" sous la clé "data".
Voici comment on peut accéder à ces informations:

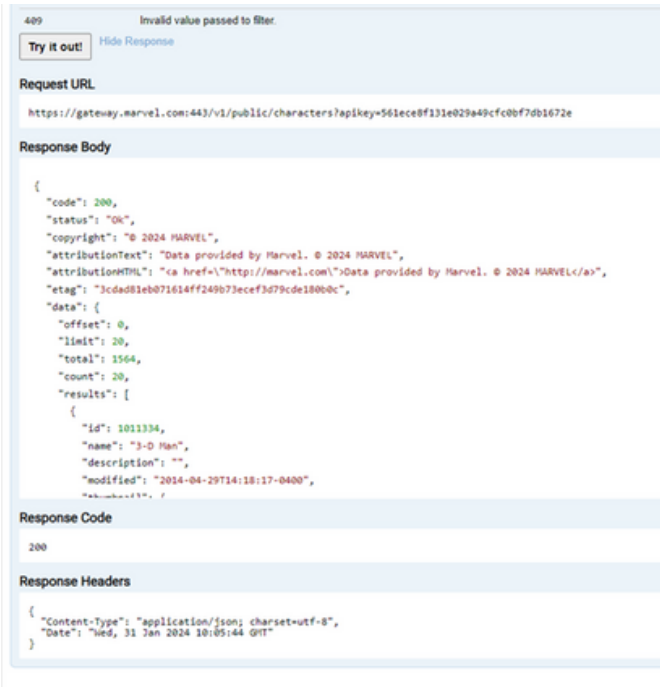
- Noms des personnages:
  - Chemin: `data.results[].name`
  - Exemple: `response.data.results[0].name`
-
- - Descriptions des personnages:
  - Chemin: `data.results[].description`
  - Exemple: `response.data.results[0].description`

- - Image miniature correspondante:
  - - Chemin: `data.results[].thumbnail.path` et `data.results[].thumbnail.extension`
  - - Exemple: `response.data.results[0].thumbnail.path + "." + response.data.results[0].thumbnail.extension`

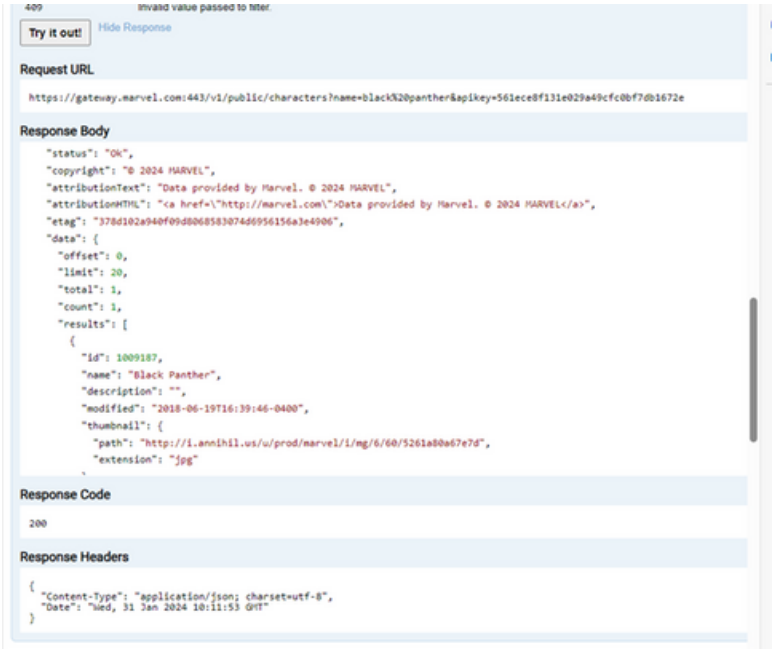
Apres le lancement de la requête j'obtient cette requête http:

- https://gateway.marvel.com:443/v1/public/characters?apikey=561ece8f131e029a49cfc0bf7db1672e

Cette requête renvoie une réponse qui contient le body, header...



- Après l’ajout black panther pour le paramètre personnage “characters” j'obtiens cette la réponse json contenant toute les infos de black panther
    - https://gateway.marvel.com:443/v1/public/characters?name=black%20panther&apikey=561ece8f131e029a49cfc0bf7db1672e

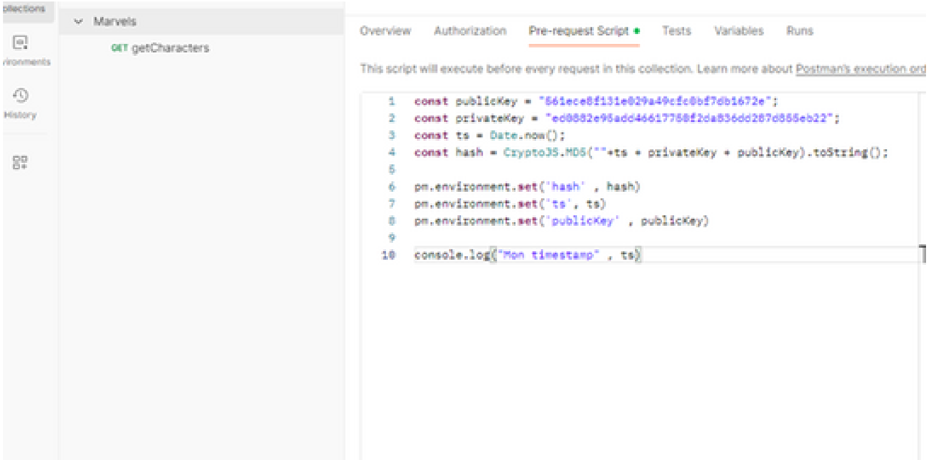


N° 02 - Étape 2 – Toujours plus loin

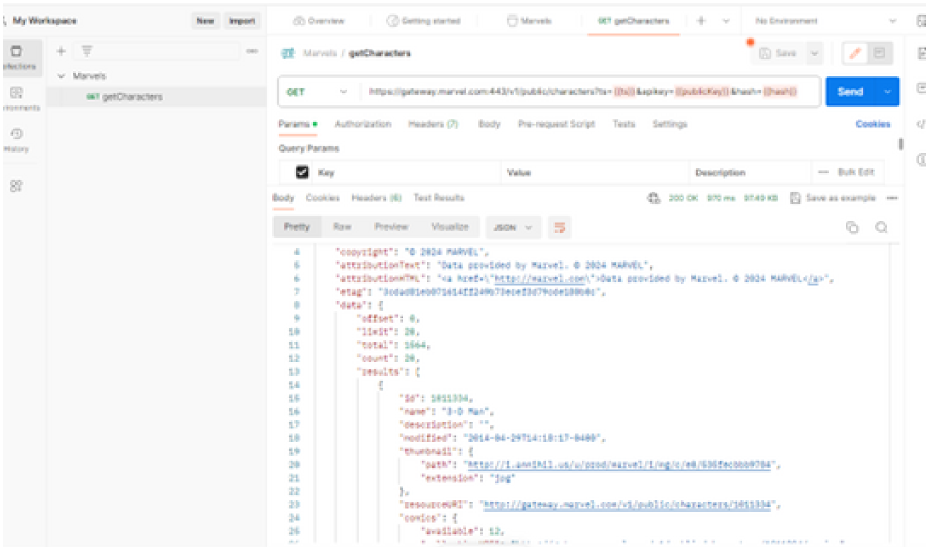
Dans cet exercice, l'idée était de récupérer des résultats depuis une page web de manière à distance, en utilisant une authentification basée sur un chiffrement asymétrique. On a obtenu des clés publiques et privées dans l'onglet "My Developer Account". Pour s'authentifier, on devait effectuer un calcul de hachage sur notre ordinateur local avec la clé publique, la clé privée et un timestamp.

J'ai utilisé Postman pour effectuer les requêtes et un script JavaScript dans l'onglet "pre-request" pour automatiser le calcul de hachage avant chaque demande. J'ai créé une Collection nommée "Marvels" pour regrouper ces requêtes. Les résultats du calcul sont stockés dans des variables d'environnement pour faciliter les échanges.

Le code JavaScript utilise la bibliothèque CryptoJS pour le hachage, et les variables d'environnement de Postman sont utilisées pour stocker les résultats de manière efficace. En bref, le script simplifie le processus d'authentification nécessaire pour accéder aux résultats distants.



Marvels
GET getCharacters
Overview
Authorization
Pre-request Script
Tests
Variables
Runs
This script will execute before every request in this collection. Learn more about Postman's execution ord
const ts = Date.now();
const hash = CryptoJS.MD5(""+ts + privateKey + publicKey).toString();
pm.environment.set('hash' , hash)
pm.environment.set('ts', ts)
pm.environment.set('publicKey' , publicKey)
console.log("Mon timestamp" , ts)



My Workspace
New
Import
Marvels
GET getCharacters
Marvels / getCharacters
GET
Send
Params
Authorization
Headers (7)
Body
Pre-request Script
Tests
Settings
Cookies
Query Params
Key
Value
Description
Bulk Edit
Pretty
Raw
Preview
Visualize
JSON

N° 03/04 - Étape 3 – Tu veux ma photo ? & Étape 4 – On s'accroche au guidon

- Récupération du Projet Initial :
  - J'ai commencé par utiliser la commande `git clone` pour obtenir les fichiers de départ depuis un dépôt Git. Cela m'a donné la base de départ nécessaire pour construire mon application.
- Installation du Module Node-Fetch :
  - Je me suis assuré de faciliter les requêtes asynchrones vers l'API Marvels en intégrant le module `node-fetch` avec la commande `npm`. Cela a rendu la gestion des interactions avec l'API plus fluide dans mon code.

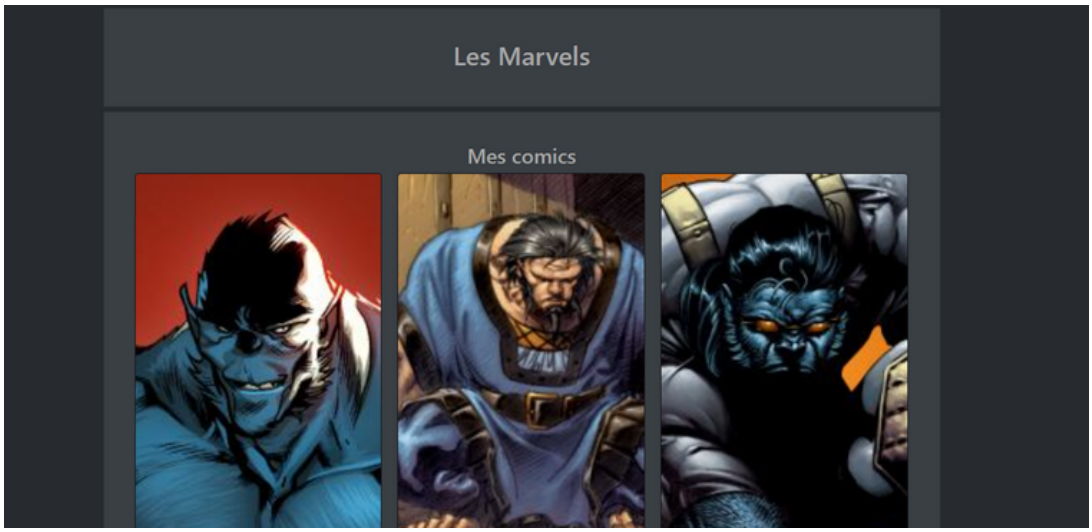
- Écriture des Fonctions pour l'Authentification et la Récupération des Données :
  - Dans le fichier `api.js`, j'ai créé deux fonctions essentielles. Tout d'abord, `getHash` génère un hachage en suivant les spécifications d'authentification, utilisant les clés d'API, le timestamp, et la méthode de hachage MD5. Ensuite, `getData` utilise ce hachage pour récupérer les données depuis l'API Marvels. J'ai également inclus une étape de filtrage pour éliminer les personnages sans image, construisant ainsi un tableau structuré de personnages.

- Configuration du Serveur avec Fastify :
  - J'ai configuré un serveur en utilisant Fastify, une bibliothèque légère pour créer des applications web en NodeJS. J'ai enregistré le moteur de rendu Handlebars pour organiser la présentation des données dans une vue, et j'ai défini une route principale pour gérer les requêtes.

- Affichage des Résultats dans une Vue Handlebars :
  - En utilisant le moteur de rendu Handlebars, j'ai élaboré une vue (fichier `index.hbs`) pour présenter de manière structurée les informations récupérées depuis l'API Marvels. Cela a ajouté une dimension visuelle à mon application, facilitant la compréhension des données du côté des utilisateurs.

- Lancement du Serveur :
  - Enfin, j'ai initié le serveur, le rendant accessible sur le port 3003. Cela a mis en marche mon application, permettant aux utilisateurs de visualiser facilement le nom des personnages et leurs images associées dans un navigateur.

En résumé, j'ai suivi des étapes claires et organisées du tp pour construire mon application NodeJS. J'ai interagi avec l'API Marvels, filtré les données, et présenté ces informations de manière conviviale grâce à Fastify et Handlebars.



Amélioration : Il est courant de voir de nombreuses pages web utiliser la fonction "offset" pour faciliter la navigation entre les différentes pages. Pour simplifier cette expérience pour l'utilisateur, j'ai implémenté un système où il peut naviguer en cliquant sur un bouton, augmentant l'offset de 1 à chaque clic.

N° 05 - Étape 5 – C’est dans la boite !

Pour ce dernier exercice, j'ai suivi toutes les étapes que vous avez énoncées dans le sujet. Vous trouverez dans mon dépôt Git les gits que j'ai eu à faire. Malheureusement, j'ai rencontré quelques problèmes lors du lancement de la machine virtuelle (VM) et avec le fichier .ova disponible dans la clé USB que vous nous avez fournie. C'est pour ces raisons que je n'ai pas pu terminer ce projet.