TP2 : développement avancé.

RABIARIVELO Ilo Andrianaly 303

Etape 1 :

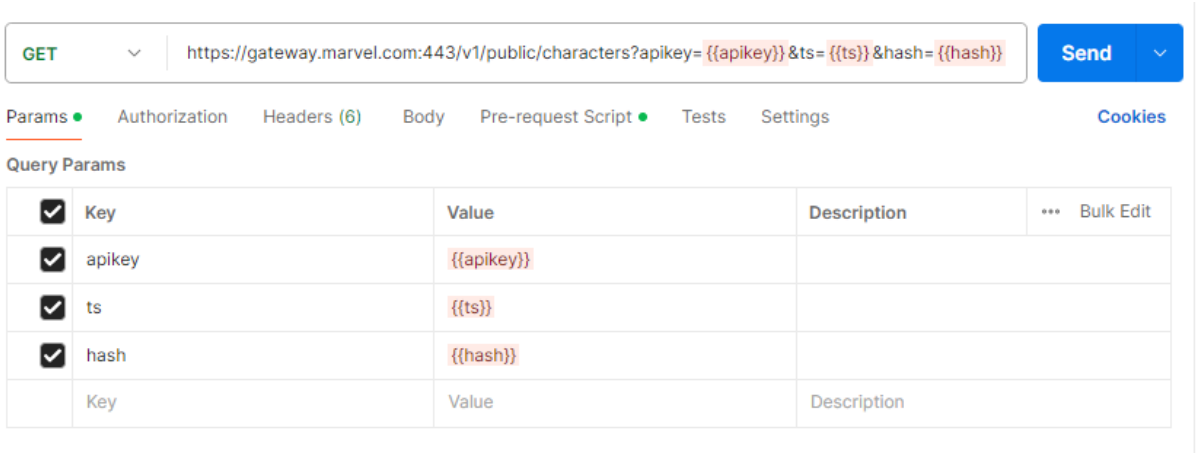
Après la création de mon compte sur le site de développeur marvel j'ai tapé la requête suivante /v1/public/characters et chercher le champ "spiderman" pour observer les changements dans l'URL. J'ai remarqué que pour l'instant que le champ "apikey" est vide.

Request URL

```
https://gateway.marvel.com:443/v1/public/characters?name=spiderman&apikey=
```

Etape 2 :

Suite à la création de la nouvelle collection "collection Marvel" j'ai créé les variables d'environnement qui pourraient être utiles pour l'utilisation de l'API.



Après avoir écrit le bon lien j'ai dû remplir le script donné dans l'énoncé pour pouvoir initialiser les variables d'environnement et ainsi avoir le bon timestamp, le bon hachage et les bonnes clefs.

```
const publicKey = "744082de314eee9f7763f53b14c8c14c";
const privateKey = "9e0f38824a227da0b28f447c332c6f2abea8a49c";

const ts =new Date().toISOString();
const hash=CryptoJS.MD5(ts+privateKey+publicKey).toString();

pm.environment.set("apikey", publicKey);
pm.environment.set("ts", ts);
pm.environment.set("hash", hash);
```

Etape 3 :

Suite à la création du projet github, j'ai complété les fonctions getHash et getData avec ce qui a été demandé et je les ai testé. J'ai un peu plus de difficulté à réaliser getData au vu de la création du json qui retourne les noms et les chemins vers les images correspondantes.

```
1 usage  ilo.rabiarivelo
export const getHash = async (publicKey, privateKey, timestamp) : Promise<ArrayBuffer>  => {
    const hash : Promise<ArrayBuffer> = crypto.createHash('md5').update(timestamp+privateKey+publicKey).digest( algorithm: "hex");
    return hash;
}
```

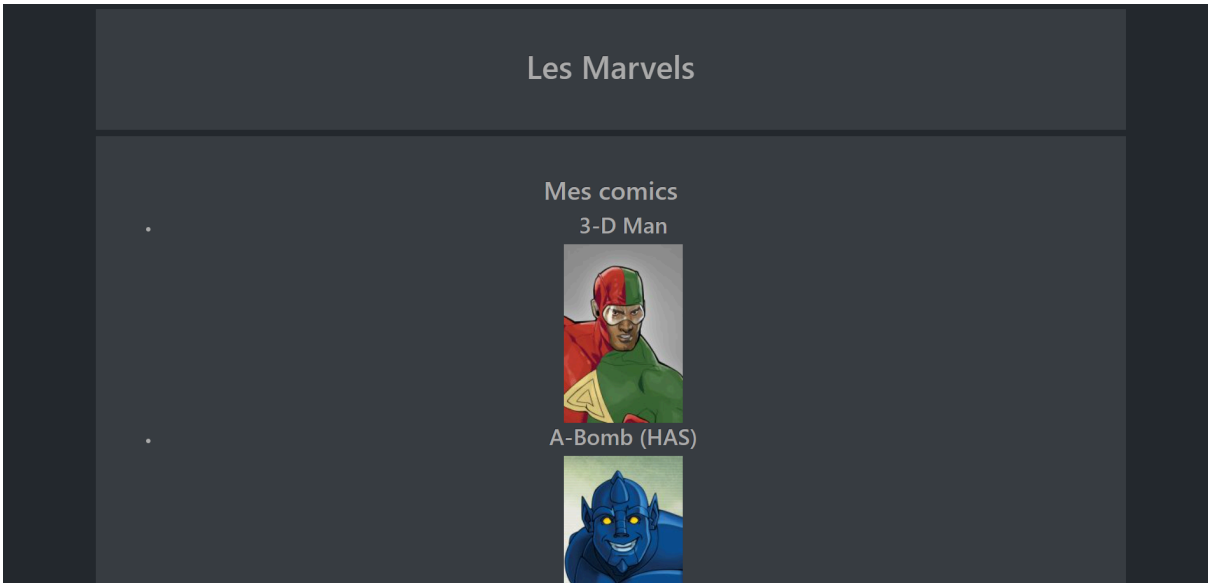
```
1 usage  ilo.rabiarivelo
export const getData = async (url) : Promise<json>  => {
    const publicKey : string  = "744082de314eee9f7763f53b14c8c14c";
    const privateKey : string  = "9e0f38824a227da0b28f447c332c6f2abea8a49c";

    const ts : string  =new Date().toISOString();
    const hash : ArrayBuffer  =await getHash(publicKey,privateKey,ts)

    url += "?apikey="+publicKey+"&ts="+ts+"&hash="+hash;
    const response = await (await fetch(url)).json()
    const trueresponse : ...  = [];
    response.data.results.forEach((element) : void => {
            if (!element.thumbnail.path.includes("image_not_available")){
                trueresponse.push({nom : element.name, path : element.thumbnail.path+"/portrait_xlarge."+element.thumbnail.extension})
            }
        })
    //console.log(trueresponse)
    return trueresponse;
}
```

Etape 4 :

Après avoir installé les modules Fastify, @Fastify/view et Handlebars avec la commande npm. J'ai recherché la documentation de ces différents modules et ainsi initialisé le moteur avec handlebars et assigné la bonne route et les partials header et footer.
Et après une modification de l'index pour boucler sur le fichier json "characters" que j'ai envoyé en retour dans le get j'ai obtenu ceci.



Etape 5 :

Après l'installation de docker (petit problème de machine virtuelle avec mon pc personnel mais j'ai réussi à le résoudre) et la création des fichiers Dockerfile et .dockerignore j'ai pu build l'image.

```
PS C:\Users\andri\PhpstormProjects\tp2\TP2DevA> docker build . -t nomdelimage
```

Et ensuite faire run l'image sur le bon port.

```
C:\Users\andri\PhpstormProjects\tp2\TP2DevA> docker run -p 8000:3000 nomdelimage
```

J'ai dû modifier certaines parties de mon code, comme par exemple le chemin du dossier template dans le code de mon serveur car docker ne le prend pas au même endroit.

Concernant l'utilisation du fichier .env et de dotenv pour le stockage des deux variables d'environnement j'ai juste modifier mon approche dans api.js après avoir bien installer le module avec npm.

```
const publicKey = process.env.PUBKEY;
const privateKey = process.env.PRIKEY;
```

Conclusion :

Pour conclure j'ai trouvé ce TP beaucoup plus long à réaliser que le premier. Mais il était aussi plus intéressant, et m'a permis de découvrir l'utilité de certains modules comme Fastify. J'avais déjà entendu parlé des Dockers mais ce TP m'a permis d'en apprendre beaucoup plus. Je suis plutôt fier de cette séance car j'ai pu terminer le sujet jusqu'à l'étape 5, et j'ai fait des commits "propre" à chaque partie du TP.