

Rapport TP2

Louis Prigneaux 304

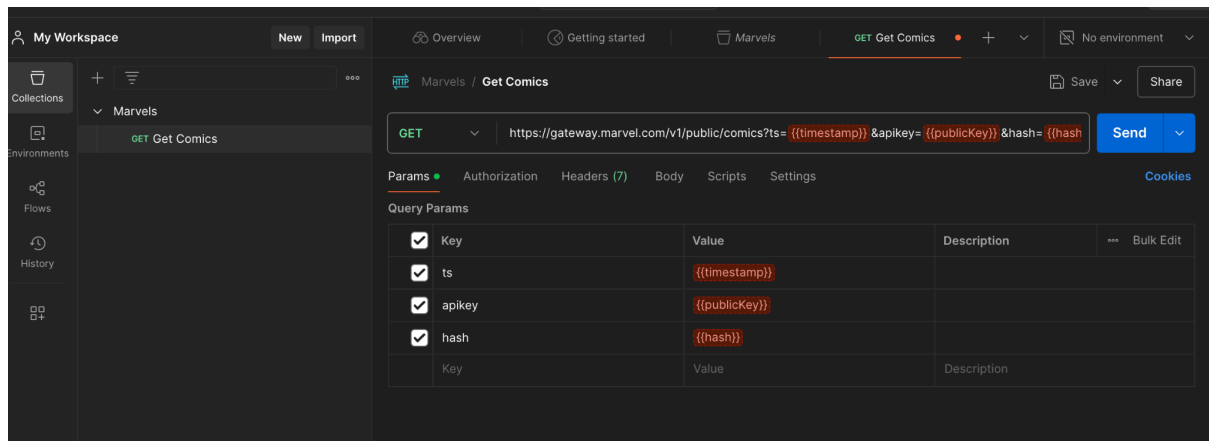
Question 1 :

J'ai créé un compte sur le site de marvel developers. On accède au site avec un timestamp à "1", l'apikey est notre clé publique et le hash est généré sur MD5 generator et représente le hash de ts+ privatekey + public key :

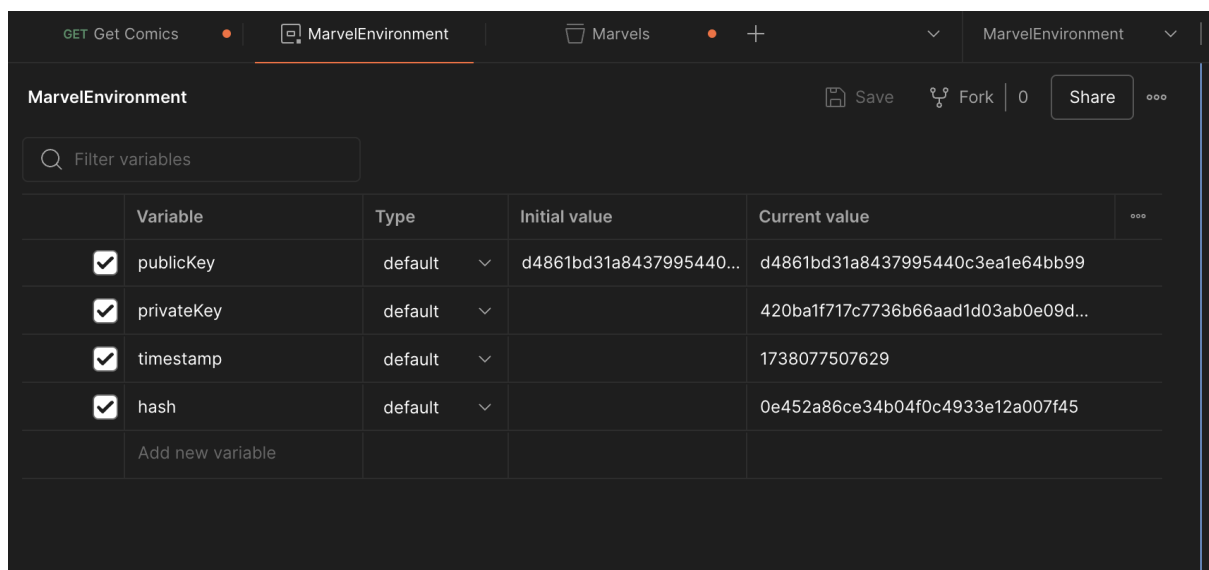
<https://gateway.marvel.com/v1/public/comics?ts=1&apikey=d4861bd31a8437995440c3ea1e64bb99&hash=df885ac4098063a9422c72a0c1103d23>

Question 2 :

Mise en place des params qui vont être utilisés dans la requête (ils appellent des variables d'environnement):



Création d'un environnement "Marvels" et le remplissage de ces champs :



Création du pre-script pour calculer le timestamp ainsi que le hash avec MD5 :

The screenshot shows the Postman interface with the 'Scripts' tab selected. The 'Pre-request' section is active, and the following JavaScript code is entered:

```
1  const timestamp = Date.now().toString();
2
3  const publicKey = pm.environment.get("publicKey");
4  const privateKey = pm.environment.get("privateKey");
5
6  const toHash = timestamp + privateKey + publicKey;
7
8  const hash = CryptoJS.MD5(toHash).toString();
9
10 pm.environment.set("timestamp", timestamp);
11 pm.environment.set("hash", hash);
12
13
14
```

On the right side, the 'Snippets' panel is visible, showing various actions like 'Get an environment variable', 'Set an environment variable', etc.

Résultat :

The screenshot shows the Postman interface with the 'Body' tab selected. The request was a GET to `https://gateway.marvel.com/v1/public/comics?ts={{timestamp}} &apikey={{apiKey}} &hash={{hash}}`. The response status is **200 OK**. The response body is JSON, showing the following structure:

```
1  {
2    "code": 200,
3    "status": "Ok",
4    "copyright": "© 2025 MARVEL",
5    "attributionText": "Data provided by Marvel. © 2025 MARVEL",
6    "attributionHTML": "<a href='\"http://marvel.com/\">Data provided by Marvel. © 2025 MARVEL</a>",
7    "etag": "58e1f7f98c3c4f5eb876f572b558f23db03e1388",
8    "data": {
9      "offset": 0,
10     "limit": 20,
11     "total": 63236,
12     "count": 20,
13     "results": [
14       {
15         "id": 82967,
16         "digitalId": 0,
17         "title": "Marvel Previews (2017)",
18         "issueNumber": 0,
19         "variantDescription": "",
20         "description": "",
21         "modified": "2019-11-07T08:46:15-0500",
22         "isbn": "",
23         "upc": "75960608839302811",
```

Question 3 :

- La fonction `getHash` permet de générer un hash MD5 nécessaire à l'authentification de l'API Marvel. Ce hash est calculé à partir de la concaténation du timestamp, de la clé privée et de la clé publique :

```
const dataToHash = `${timestamp}${privateKey}${publicKey}`;
```

```
const hash = crypto.createHash("md5").update(dataToHash).digest("hex");
```

- La fonction `getData` permet d'effectuer une requête GET asynchrone sur l'endpoint de l'API Marvel, en incluant les paramètres nécessaires (`ts`, `apikey`, `hash`) pour l'authentification.

```
const timestamp = Date.now().toString();

const hash = await getHash(publicKey, privateKey, timestamp);

const fullUrl = `${url}?ts=${timestamp}&apikey=${apiKey}&hash=${hash}`;
```

Utilisation de `fetch` :

```
const response = await fetch(fullUrl);
```

Enfin, on vérifie si "thumbnail" existe ainsi que l'image soit disponible.

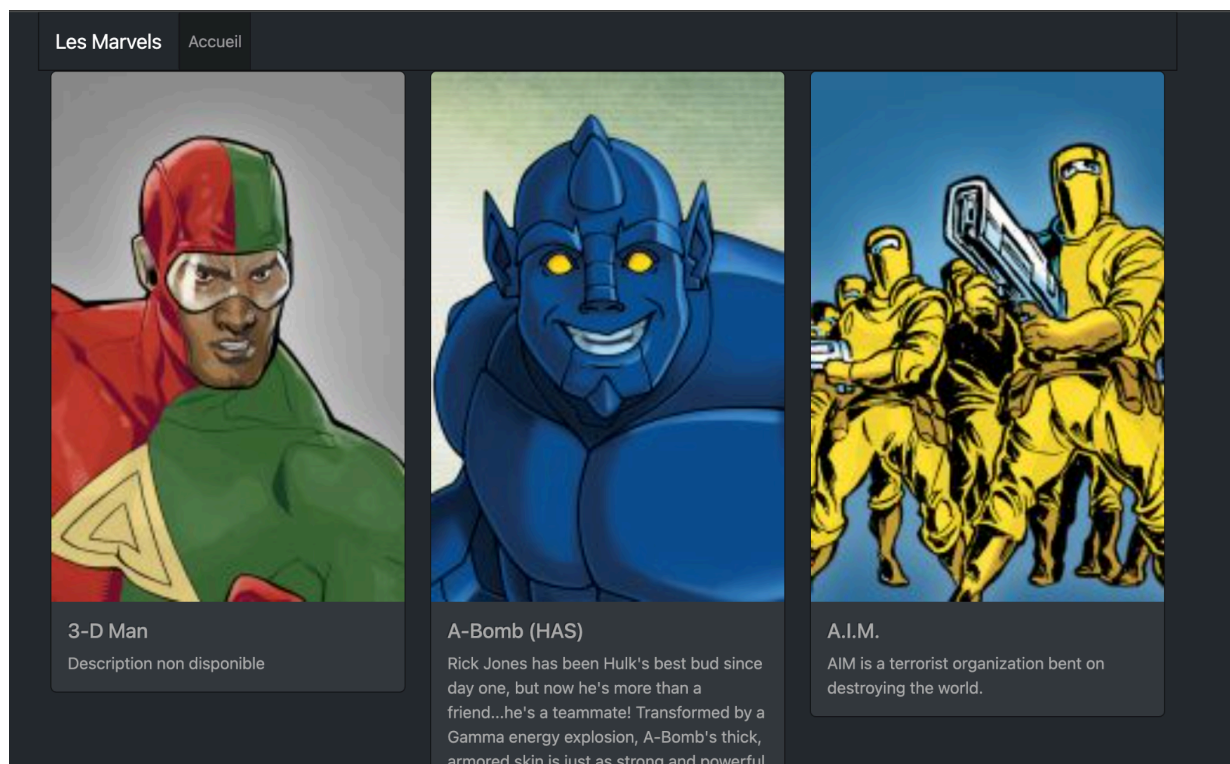
Résultats :

```
(base) louis@MacBook-Pro-de-Louis LesMarvels % node src/api.js
[
  {
    name: '3-D Man',
    description: 'Description non disponible',
    imageUrl: 'http://i.annihil.us/u/prod/marvel/i/mg/c/e0/535fecbbb9784/portrait_xlarge.jpg'
  },
  {
    name: 'A-Bomb (HAS)',
    description: "Rick Jones has been Hulk's best bud since day one, but now he's more than a friend...he's a teammate! Transformed by explosion, A-Bomb's thick, armored skin is just as strong and powerful as it is blue. And when he curls into action, he uses it like a ball of destruction!",
    imageUrl: 'http://i.annihil.us/u/prod/marvel/i/mg/3/20/5232158de5b16/portrait_xlarge.jpg'
  },
  {
    name: 'A.I.M.',
    description: 'AIM is a terrorist organization bent on destroying the world.',
    imageUrl: 'http://i.annihil.us/u/prod/marvel/i/mg/6/20/52602f21f29ec/portrait_xlarge.jpg'
  },
  {
    name: 'Abomination (Emil Blonsky)',
    description: 'Formerly known as Emil Blonsky, a spy of Soviet Yugoslavian origin working for the KGB, the Abomination gained his powers by receiving a dose of gamma radiation similar to that which transformed Bruce Banner into the incredible Hulk.',
    imageUrl: 'http://i.annihil.us/u/prod/marvel/i/mg/9/50/4ce18691cbf04/portrait_xlarge.jpg'
  },
  {
    name: 'Absorbing Man',
    description: 'Description non disponible',
    imageUrl: 'http://i.annihil.us/u/prod/marvel/i/mg/1/b0/5269678709fb7/portrait_xlarge.jpg'
  }
]
```

Question 4 :

- Installation et mise en place de fastify et des fichiers hds
- L'application utilisant Fastify et Handlebars ne s'affichait pas sur `http://localhost:3000`.
Aucun contenu HTML ne s'affichait, même avec un simple `<h1>Test: La page s'affiche</h1>`.
- Vérification du lancement du serveur avec `lsyf -i :3000`
Ajout d'un log dans `fastify.listen()` pour confirmer son démarrage
Modification de `host: '0.0.0.0'` pour éviter un problème de liaison locale

Résultats



Question 5 :

Je n'ai pas eu le temps de la finir