

Développement Avancé : TP2

Etape 1 : “Avengers, rassemblement !”

L'étape à été faite en classe.

Etape 2 : Toujours plus loin

L'étape à été faite en classe.

Etape 3 : Tu veux ma photo ?

```
export const getData = async (url) : Promise<json> => {  
  try {  
    const response = await fetch(url);  
    const data = await response.json();  
    return data;  
  } catch (error) {  
    console.error('Error fetching data:', error);  
    throw error;  
  }  
}
```

```

// Usage
export const getHash = async (publicKey, privateKey, timestamp) : Promise<ArrayBuffer> => {
  try {
    const hash = CryptoJS.MD5(`${timestamp}${privateKey}${publicKey}`).toString();
    return hash;
  } catch (error) {
    console.error('Error generating hash:', error);
    throw error;
  }
}

const publicKey : string = "4127f72f71e72c08b584350b149981f6";
const privateKey : string = "51420b3b2ced406ece06dd286c9b5d4bda5c9b3b";
const ts : number = Date.now();

const hash : ArrayBuffer = await getHash(publicKey, privateKey, ts);
console.log(ts);
console.log(hash);

const url : string = `https://gateway.marvel.com/v1/public/characters?ts=${ts}&apikey=${publicKey}&hash=${hash}&name=Hulk`;
const marvelData : json = await getData(url);

console.log(marvelData);

const hulkData = marvelData.data.results.find(character => character.name === 'Hulk');

if (hulkData) {
  console.log('Données de Hulk:', hulkData);

  if (hulkData.thumbnail && hulkData.thumbnail.path && hulkData.thumbnail.extension) {
    console.log('Thumbnail URL:', `${hulkData.thumbnail.path}.${hulkData.thumbnail.extension}`);
  } else {
    console.log('Il n\'a pas d\'image thumbnail valide pour Hulk.');
  }
} else {
  console.log('Hulk n\'a pas été trouvé dans les résultats.');
```

```

const charactersArray = marvelData.data.results.map(character => {
  const imageUrl : string | null = character.thumbnail && character.thumbnail.path && character.thumbnail.extension
    ? `${character.thumbnail.path}/portrait_xlarge.${character.thumbnail.extension}`
    : null;

  return {
    name: character.name,
    imageUrl: imageUrl,
  };
});

console.log('Tableau de personnages :', charactersArray);
```

Donc dans cette étape, nous avons utilisé l'API Marvel pour obtenir des données de personnages de bande dessinée. Après avoir récupéré le dépôt de démarrage et installé les dépendances nécessaires, nous avons implémenté des fonctions pour récupérer les données et calculer une valeur de hachage MD5 conformément aux spécifications de l'API. En utilisant les identifiants fournis, nous avons effectué une requête pour obtenir les données du personnage Hulk, filtrant ensuite les résultats pour ne conserver que ceux avec une image thumbnail valide. Enfin, nous avons construit un tableau de personnages en extrayant les noms et en construisant des URL d'images spécifiques pour chaque

personnage, démontrant ainsi l'utilisation pratique des services Web externes et des bibliothèques JavaScript pour manipuler et présenter des données.