Tutoriel API Anilist - GraphQL 👍

L'API d'Anilist permet d'avoir de nombreuses informations sur des milliers d'animés et de mangas de tout genre et de toute époque.

Pour utiliser l'API d'Anilist, il faudra faire des requêtes en GraphQL. L'avantage de GraphQL est qu'on peut récupérer seulement les champs voulus. 🙂

GraphQL

Même si un tutoriel complet sur GraphQl ne sera pas fait ici, nous verrons un minimum les bases pour comprendre ce qu'on fait quand on souhaite faire une requête à l'API d'AniList. Pour ceux qui souhaitent mieux comprendre, je vous redirige vers la doc officielle : https://graphql.org/learn/

Tout d'abord, il faut comprendre que GraphQL, comme la majorité des autres langages de requête pour les API, se base sur le principe du JSON. Cela signifie que les données sont structurées de manière similaire, ce qui les rend faciles à comprendre et à manipuler.

Voici un exemple de données au format JSON :

```
{
  "anime": {
    "id": "16498",
    "title": {
      "english": "Titre en anglais},
      "romaji": "Titre en romaji"
    }
  }
}
```

Ici, l'objet JSON "anime" est composé de deux attributs distincts :

- "id" : Un simple attribut int qui contient l'id de l'animé.
- "title" : Un objet JSON représentant les titres associés à l'animé. Cet objet comprend un attribut “english” et “romaji” qui correspond au titre de l'animé dans la langue indiqué.

Cette structure souligne comment les données peuvent être organisées de manière hiérarchique dans un objet JSON, avec des attributs simples et des structures plus complexes imbriquées à différents niveaux.

👍 Maintenant que l'on sait comment est structuré le corps d'un objet JSON, et donc en GraphQL aussi, il reste un point à aborder, celui des variables.

Prenons une demande GraphQl avec une requête qui a un "corps" comme celui-ci :

```
{
  anime(numero: 5, id: $idParam){
    id,
    numero,
    title {
      english,
      romaji"
    }
  }
}
```

Ici, on cherche à obtenir l'id, le numéro, et le titre en anglais et romaji des animés qui ont 5 en numéro et un id qui a la même valeur que la variable idParam représenté par un "$".

Il faut donc bien définir la valeur de cette variable quelque part et cela ce fait dans la partie "variable" de la requête.

Une fois qu'on a le corps de la requête ainsi que les options de celle-ci (les variables dans notre cas) on peut désormais formuler nos propres requêtes vers une API GraphQL.

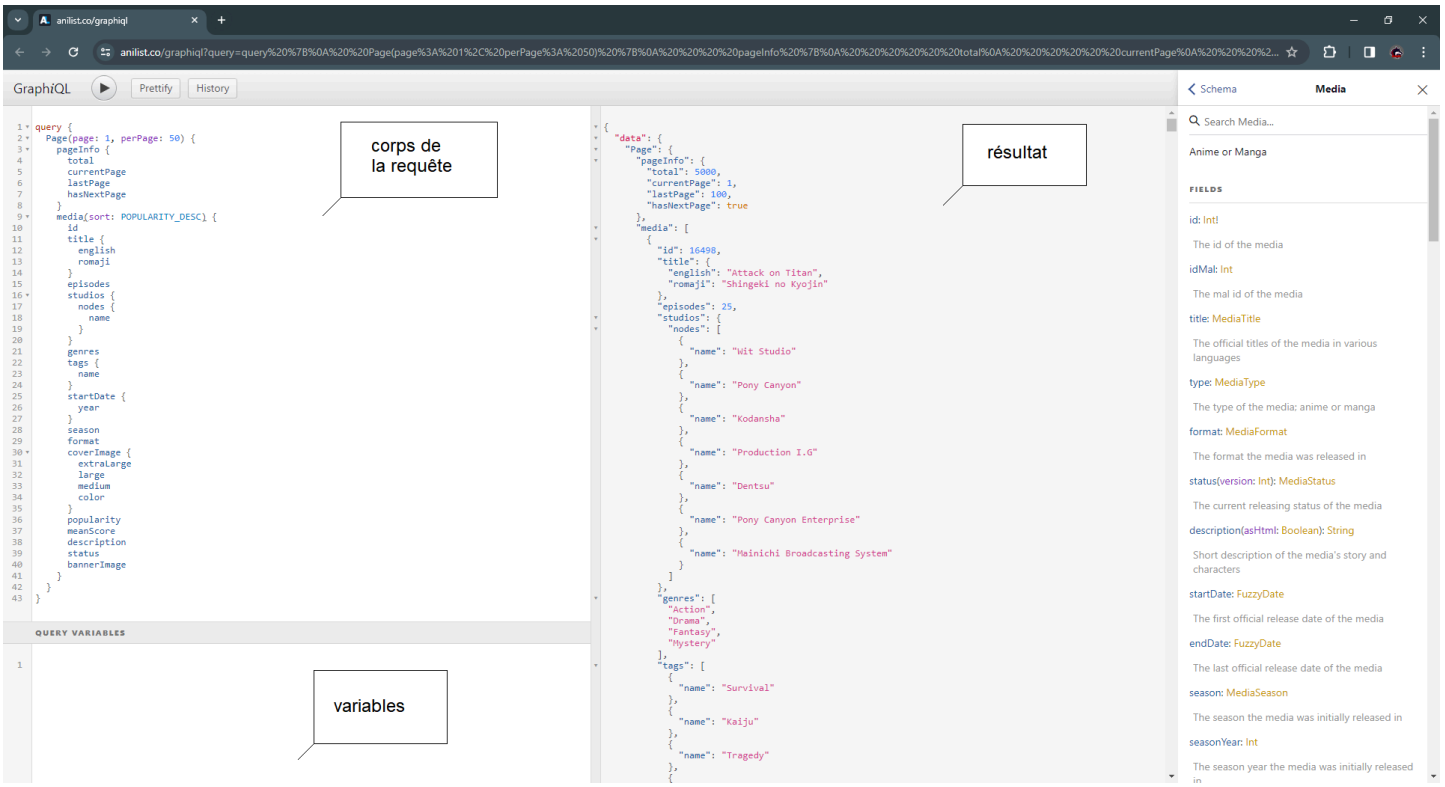
API Anilist

👋 Maintenant que vous en savez un peu plus sur l'utilisation de GraphQL, il est temps de rentrer dans le vif du sujet.

Pour se familiariser avec l'API et voir toutes les informations et champs disponibles, il faut aller à l'adresse suivante : https://anilist.co/graphiql

Chaque œuvre est représentée par l'attribut "media" dans lequel il y a toutes les informations relatives à l'animé ou manga. Vous pouvez chercher les attributs disponibles dans la barre de recherche à droite.

Voici un exemple de requêtes via l'outil d'AniList :



Cette requête renvoie les 50 œuvres (Anime/Manga) les plus populaires avec de nombreuses caractéristiques comme l'id, le titre en romaji et en anglais, la date de sortie, etc..

❗ ❗ ❗ ❗ ❗ Le résultat d'une requête peut contenir au maximum les informations de 50 animés à la fois. Même si on augmente la valeur du perPage au-delà de 50. Pour avoir les suivants, il faut donc faire plusieurs requêtes en changeant le paramètre page . ❗ ❗ ❗ ❗ ❗

L'attribut "page info" donne des informations sur le nombre d'animés correspondant à votre requête. Ici, il y a 5000 animés correspondants (sûrement le max affichable), vous êtes sur la page 1, il y a donc 100 pages en tout et il existe une prochaine page.

Requête Javascript

Maintenant que l'on sait comment fonctionne GraphQl ainsi que l'API d'AniList, il est temps de passer à l'exploitation via une requête JavaScript.

Vu qu'un exemple serait trop verbeux, je vous invite à aller visiter le site officiel d'AniList sur leur API où vous trouverez un modèle complet du code JS pour faire ce genre de requêtes.
https://anilist.gitbook.io/anilist-apiv2-docs/overview/graphql/getting-started

Ici, je vous expliquerai seulement les différents points à prendre en compte.

En général, pour faire une requête JS vers l'API d'anilist, on a 5 éléments importants.

Le premier étant l'objet "query" qui correspond au corps de la requête comme nous l'avons vu précédemment.Le second est l'élément variable dont l'utilité a aussi été vue précédemment. Le troisième correspond à l'url qui vers l'API d'anilist : "https://graphql.anilist.co"

Le quatrième élément n'est autre que les options passées lors de la requête. C'est dans ces options que l'on spécifie la méthode en POST et dans le "body" on retrouve la "query" et les "variables".

Enfin, la dernière étape est d’utiliser la fonction fetch qui prend en paramètre l’url et l’option de la requête pour effectuer celle-ci.

🎉 Maintenant il ne reste plus qu'à mettre en pratique 🎉

Le Mans Université, L3 Informatique
Yamis MANFALOTI
Nathan RACCOUARD
Cody THEARD