

TP2 - GraphQL

WIS 3 / Module Cloud et système web

I. Pré requis : Générer une API avec un SAAS : GraphCMS

Découvrir le service GraphCMS, un CMS headless en SAAS : <https://graphcms.com/>

Créer un compte avec l'offre gratuite, puis créer un nouveau projet : "From Template" / "Travel". Ceci créera un projet exemple contenant déjà des données utiles pour la suite du TP.

Observer le schéma et le contenu créé via l'interface de GraphCMS :

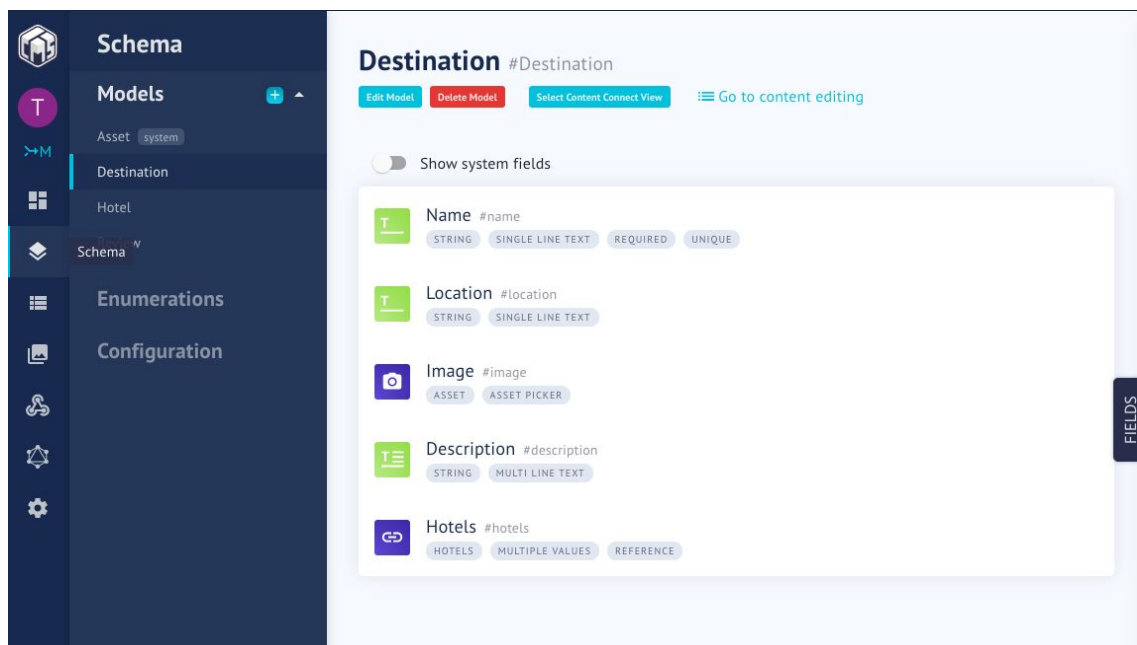
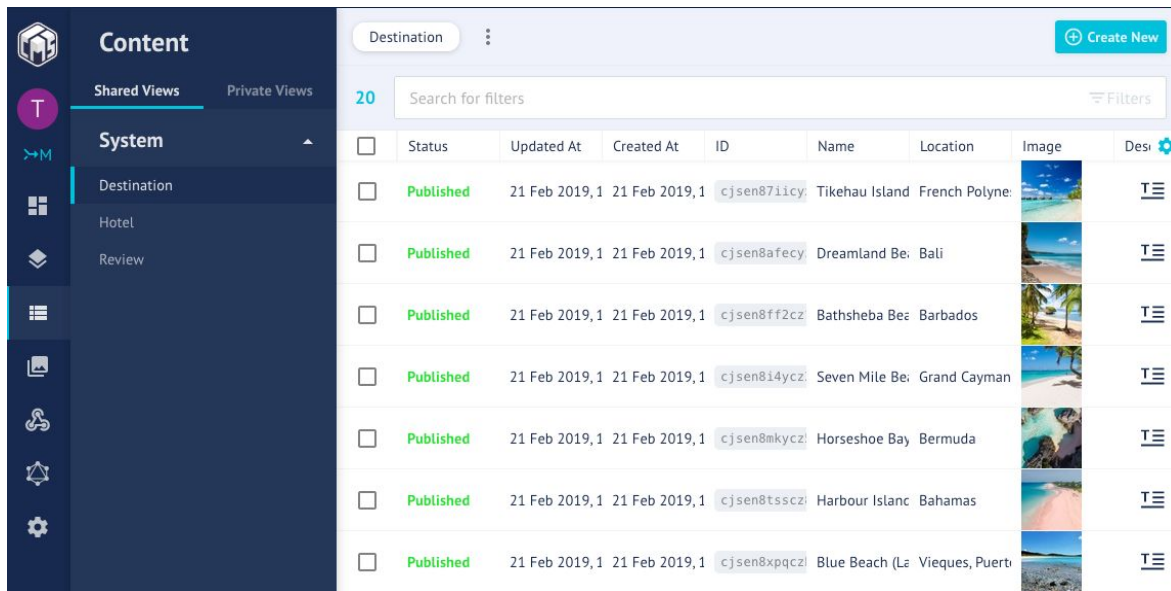









Schéma du projet




The screenshot shows a web application interface for managing content. On the left is a dark sidebar with a 'Content' header and a 'System' menu. The main area displays a table of destinations. The table has columns for Status, Updated At, Created At, ID, Name, Location, Image, and a 'Desi' column with a settings icon. All items in the table are marked as 'Published' and have a creation/update date of '21 Feb 2019, 1'. The destinations listed are Tikehau Island, Dreamland Be, Bathsheba Bez, Seven Mile Be, Horseshoe Bay, Harbour Islanc, and Blue Beach (La Vieques, Puerto Rico).

Status	Updated At	Created At	ID	Name	Location	Image	Desi
Published	21 Feb 2019, 1	21 Feb 2019, 1	cjsen87iicy	Tikehau Island	French Polynes		T
Published	21 Feb 2019, 1	21 Feb 2019, 1	cjsen8afecy	Dreamland Be	Bali		T
Published	21 Feb 2019, 1	21 Feb 2019, 1	cjsen8ff2cz	Bathsheba Bez	Barbados		T
Published	21 Feb 2019, 1	21 Feb 2019, 1	cjsen8i4ycz	Seven Mile Be	Grand Cayman		T
Published	21 Feb 2019, 1	21 Feb 2019, 1	cjsen8mkycz	Horseshoe Bay	Bermuda		T
Published	21 Feb 2019, 1	21 Feb 2019, 1	cjsen8tsscz	Harbour Islanc	Bahamas		T
Published	21 Feb 2019, 1	21 Feb 2019, 1	cjsen8xpcqz	Blue Beach (La	Vieques, Puerto Rico		T

Contenu du projet

II. Lire les données de l'API avec GraphQL

Se rendre sur la partie “API Explorer” pour écrire des requêtes GraphQL.



The screenshot shows the 'API Explorer' interface. It has a top bar with 'API Explorer', 'Prettify', 'Stage: master', and 'No Locale'. The main area is a text editor with a GraphQL query. The query is a simple query to fetch the 'name' of the 'destinations' field. The interface also includes a sidebar with various icons and a list of keyboard shortcuts.

```

1 # Welcome to GraphQL
2 #
3 # GraphQL is an in-browser tool for writing, validating, and
4 # testing GraphQL queries.
5 #
6 # Type queries into this side of the screen, and you will see intelligent
7 # typeheads aware of the current GraphQL type schema and live syntax and
8 # validation errors highlighted within the text.
9 #
10 # GraphQL queries typically start with a "{" character. Lines that starts
11 # with a # are ignored.
12 #
13 # An example GraphQL query might look like:
14 #
15 # {
16 #   field(arg: "value") {
17 #     subField
18 #   }
19 # }
20 #
21 # Keyboard shortcuts:
22 #
23 # Prettify Query: Shift-Ctrl-P (or press the prettify button above)
24 #
25 # Merge Query: Shift-Ctrl-M (or press the merge button above)
26 #
27 # Run Query: Ctrl-Enter (or press the play button above)
28 #
29 # Auto Complete: Ctrl-Space (or just start typing)
30 #
31
32 query {
33   destinations {
34     name
35   }
36 }

```

Api Explorer

Écrire les requêtes pour récupérer les informations suivantes :

- 1) La liste de toutes destinations (nom, adresse, description) et leurs hôtels correspondants (nom, description, site web)
- 2) La liste des hôtels classés par ordre alphabétique
- 3) Les reviews (descriptions) de l'hôtel "Caneel Bay"
- 4) La liste des destinations qui ont au moins un hôtel avec plus de 50 chambres

III. Mutations avec GraphQL

Écrire des mutations qui permettent les actions suivantes :

- 1) Rédiger une review sur l'hôtel "Caneel Bay"
- 2) Mettre à jour le nombre de chambre "Pink Sands" à 25
- 3) Créer une nouvelle destination
- 4) Supprimer une review

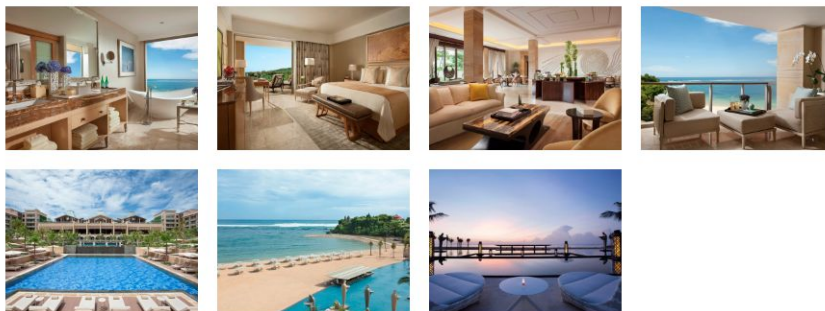
IV. Intégration dans un projet Angular

Utiliser l'api créée pour afficher la liste des hôtels disponibles sur une page web.

Tikehau Pearl Beach Resort



The Mulia & Mulia Villas



Fairmont Royal Pavilion



Nous nous baserons sur les fondamentaux appris dans le projet Angular [“weather” du TP1](#). La logique est la même. Les requêtes sont exécutées dans un service. Le component appelle le service et affiche les données dans le page HTML.

Vous pouvez trouver L’URL de l’API dans la partie “Settings” de votre projet graphCMS. Penser à changer les permissions de “Public API Permissions” pour ouvrir votre API sur l’extérieur.

Affichage des hôtels

En quelques étapes :

- Créer un nouveau projet angular ([tutoriel](#))
- Créer un service “**GraphqlService**”. Récupérer le code suivant : [graphql.service.ts](#)
- Créer un component “**list-hotels**” (idem weather)
 - Ajouter la propriété “**hotelsDatas**” (idem weatherDatas)
 - Injecter le service “**GraphqlService**” dans le constructor
 - Faire l’appel à la méthode “**getHotels**” du service graphql dans le constructor et stocker le résultat dans “**hotelsDatas**” (inspirez vous de loadWeather())
 - Réaliser l’affichage dans le fichier HTML. Commencez par afficher le nom des hôtels.
[Exemple de code HTML](#)
- Vous pouvez modifier la requête GraphQL dans le service pour ajouter des champs (photos, descriptions...) et améliorer votre rendu