

# Utiliser un CMS Headless pour stocker les données

---

- CMS Headless ???
- Un exemple
- Une api toute faite

# CMS Headless

---

- Un CMS Headless : un CMS sans frontend
- Un backoffice pour structurer et saisir du contenu stocké dans une base de données
- Une API pour accéder à ce contenu à l'aide de requêtes REST ou GraphQL
- En général disponible en version cloud ou self-hosted

# directus

---

- Cms headless écrit en node.js
- Disponible en cloud ou en version auto-hébergée
- Image docker disponible
- Un backoffice pour définir la structure du contenu et insérer des données
- Une api REST + GraphQL

# architecture

- Une application web
  - Création de la structure
  - Gestion des droits
  - Création du contenu
- Une base de données SQL
  - postgres, mysql, mariaDB, Oracle, SQLite, S3 ...
- Capable d'utiliser une base existante et d'exploiter son contenu

# Installation - documentation

---

- Avec docker compose

<https://docs.directus.io/self-hosted/docker-guide.html>

- Documentation

<https://docs.directus.io/>

# Le modèle des données

- Une collection = 1 table avec un identifiant (PK)

The image shows the Directus Data Model interface. On the left, a sidebar contains navigation links: Directus, Data Model, Access Control, Webhooks, Flows, Extensions, Settings, Appearance, Bookmarks, Translations, Report Bug, Request Feature, and Directus 10.8.2. An orange arrow points to the 'Data Model' link. The main area displays 'Settings Data Model' with a search bar and a 'No Collections' message. A 'Create Collection' button is visible. An orange arrow points from this button to a 'Creating New Collection' dialog. The dialog has a 'Name' field with 'blog\_post' and a 'Primary Key Field' field with 'id'. The 'Singleton' section has a 'Treat as single object' checkbox and a list of field types: Auto-incremented integer, Auto-incremented big integer, Generated UUID, and Manually entered string. The 'Auto-incremented integer' option is selected and highlighted with a blue border.

Directus

Data Model

Access Control

Webhooks

Flows

Extensions

Settings

Appearance

Bookmarks

Translations

Report Bug

Request Feature

Directus 10.8.2

Settings Data Model

Search Collection...

No Collections

You don't have any Collections. Click the button below to get started.

Create Collection

Collection Setup

Optional Fields

Creating New Collection

Name the collection and setup its unique "key" field...

Name \*

blog\_post

Collection names are case sensitive

Primary Key Field

id

Singleton

☐ Treat as single object

Auto-incremented integer

Auto-incremented big integer

Generated UUID

Manually entered string

Auto-incremented integer

# ■ Des champs de différents types

**Data Model**  
**Blog Post**

Fields & Layout **Saves Automatically**

id

status

date\_created

user\_created

**Create Field**

Create Field in Advanced Mode

**New Field (blog\_post)** Search Field...

**Text & Numbers**

Input

Autocomplete Input (...)

Block Editor

Code

Textarea

WYSIWYG

Markdown

Tags

**Selection**

Toggle

Datetime

Repeater

Map

Color

Dropdown

Icon

Checkboxes

Checkboxes (Tree)

Dropdown (Multiple)

Radio Buttons

**Relational**

# ■ Une interface de saisie


Data Model  
← **Blog Post**

Fields & Layout **Saves Automatically**

id	
status	user_created
date_created	
titre*	
contenu	

Create Field

Create Field in Advanced Mode












Blog Post  
← **Creating Item in Blog Post**

Status  
Draft

Titre\*  
titre

Contenu

**B I U** H1 H2 H3         



# ■ Et une api pour accéder au contenu

GET api-1 x +

GET http://localhost:8055/items/blog\_post

Query Body Headers Auth Vars Script

Assert Tests Docs

Name	Value
------	-------

+ Add Param

Response Headers Timeline Tests

200 OK 19ms 742B

```
1 {
2   "data": [
3     {
4       "id": 3,
5       "status": "draft",
6       "user_created": "b57e2d58-0567-4de7-b75d-2f43f1e19eaa",
7       "date_created": "2023-12-20T13:59:59.108Z",
8       "titre": "Mon premier Post",
9       "contenu": "<p>Salut les loulous,</p>\n<p>Ceci est mon premier post sur ce blog consacré à la vie des grenouilles sur les bords du lac de Chantebleu. C'est un thème qui me passionne :</p>\n<ul>\n<li>j'adore les grenouilles,</li>\n<li>je suis fan de l'écologie,</li>\n<li>j'habite moi-même près du lac.</li>\n</ul>\n<p>Bref, je vous raconte tout ça dans la suite ....</p>\n<p><br></p>",
10      "description": "le premier post qui plante le décor"
11    }
12  ]
13 }
```

# ■ Création d'associations entre collections

- Type

- Collection associée et FK

**Posts (categorie)** Search Field...

Checkboxes (Tree) Dropdown (Multiple) Radio Buttons

**Relational**

File Image Files Builder (M2A)

Many to Many **One to Many** Tree View Many to One

**Key \*** posts

**Type** Alias

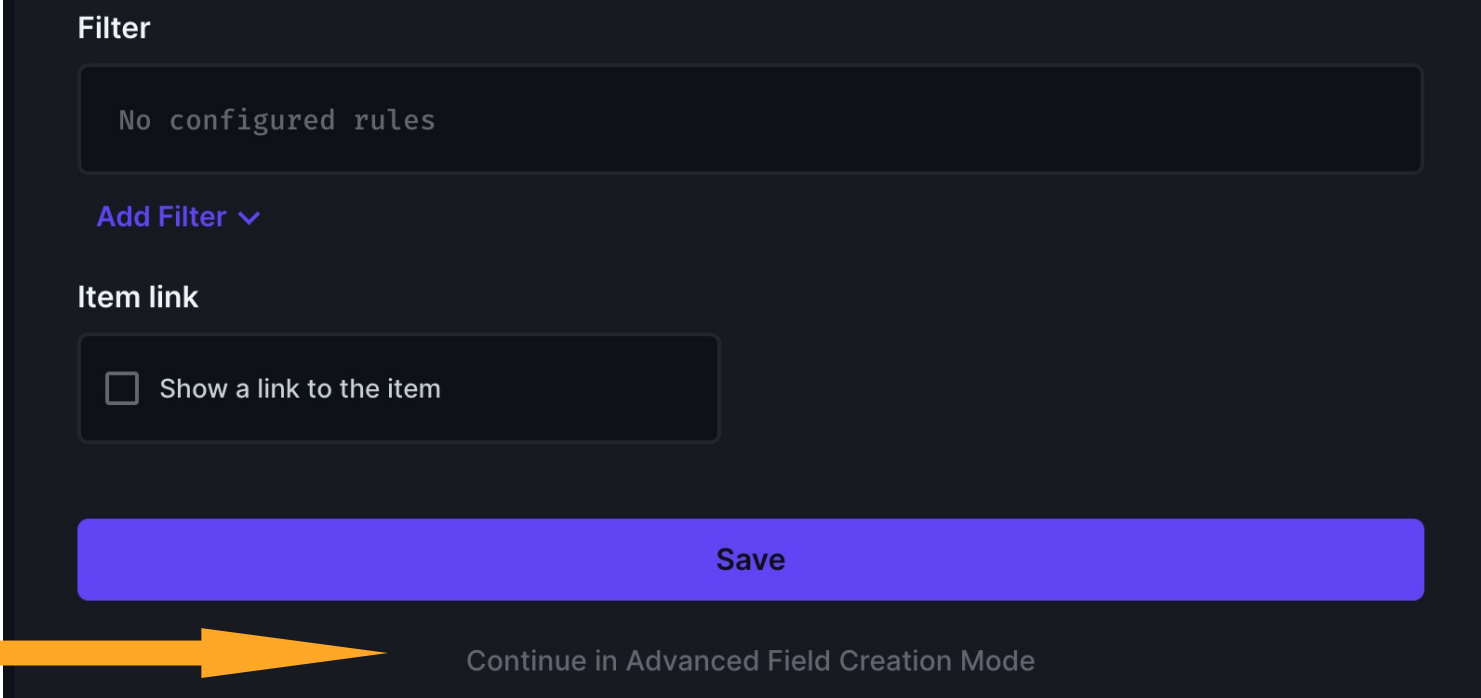
**Default Value** NULL

**Required** ☐ Require value to be set on creation

**Related Collection \*** blog\_post

**Foreign Key \*** cat\_id

- Attention : pour créer l'association inverse, notamment dans la cas Many-To-Many, il faut passer en *Advanced Field Creation Mode* avant de sauvegarder



The screenshot shows a dark-themed configuration panel. At the top is a 'Filter' section with a text box containing 'No configured rules' and a blue 'Add Filter' link with a dropdown arrow. Below this is an 'Item link' section with a checkbox labeled 'Show a link to the item'. A large blue 'Save' button is positioned at the bottom of the panel. An orange arrow points from the left towards the 'Continue in Advanced Field Creation Mode' text, which is located just below the 'Save' button.

Filter

No configured rules

[Add Filter](#) ▾


Item link



☐ Show a link to the item

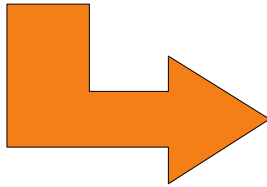
Save

[Continue in Advanced Field Creation Mode](#)

# ■ Des droits d'accès basés sur des rôles











Settings **Access Control** 

	Name	Users	Description
	Public	...	Controls what API data is available without aut
	Administrator	1	Initial administrative role with unrestricted App



Access Control **Public**

Permissions **Saves Automatically**

Collection					
blog_post					
categorie					

System Collections ▾

# Autorisation et authentification dans Directus

- Autorisation et droits d'accès
  - Des rôles
  - Des droits attribués aux rôles sur chaque collection
  - Des utilisateurs

# rôles

Directus

Data Model

Access Control

Webhooks

Flows

Extensions

Settings

Appearance

Settings

Access Control

Name	Users	Description
Public	--	Controls what API data is available without authenticating.
Administrator	1	Initial administrative role with unrestricted App/API access.
api_users	--	accès au travers de l'api uniquement

Role Name

api\_users

Role Icon

Supervised User Circle

Description

accès au travers de l'api uniquement

App Access

☐ Enabled

Admin Access

☐ Enabled

IP Access

Add allowed IP addresses, leave empty to allow all...


Require 2FA


☐ Enabled


# droits


Collection	+	👁	✎	🗑	🔗
categorie	🚫	✓	🚫	🚫	🚫
ingredients	🚫	✓	🚫	🚫	🚫
ingredients_recette	🚫	✓	🚫	🚫	🚫
lieu	✓	✓	🚫	🚫	🚫
point_vente	🚫	✓	🚫	🚫	🚫
point_vente_produit	🚫	✓	🚫	🚫	🚫
point_vente_recette	🚫	✓	🚫	🚫	🚫
produit	🚫	✓	🚫	🚫	🚫
recette	🚫	✓	🚫	🚫	🚫
seance	✓	✓	🚫	🚫	🚫
spectacle	✓	✓	🚫	🚫	🚫
taille	🚫	✓	🚫	🚫	🚫
tarif	🚫	✓	🚫	🚫	🚫


# utilisateurs


 Directus


 All Users


 Administrator

 api\_users














User Directory

← Adding User   


First Name

michel 


Last Name

michel 


Email

miche@gmail.com 

Password

..... 

Avatar ▾



 No File Selected

## Admin Options

Status

Active ▾

Role

api\_users  

Token

Click "Generate Token" to create a new static access token



Provider

Default ▾

External Identifier





# authentication

---

- Basée sur deux types de tokens
- Access tokens statiques
  - Définis à la création de l'utilisateur, sans expiration
  - Utiles pour des communication serveur-serveur
- Access token temporaire JWT
  - Obtenus auprès de l'api login
  - Accompagnés d'un refresh token

# Transport des access tokens

- Dans les requêtes REST ou GraphQL

## Query Parameter

```
?access_token=<token>
```

## Authorization Header

```
Authorization: Bearer <token>
```

# login

## Request

REST GraphQL SDK

POST /auth/login

POST /auth/login/:provider

```
{  
  "email": user_email,  
  "password": user_password  
}
```

## Request

REST GraphQL SDK

POST /graphql/system

```
mutation {  
  auth_login(email: "user_email", password: "user_password") {  
    access_token  
    refresh_token  
  }  
}
```

# refresh

## Request

REST GraphQL SDK

POST /auth/refresh

```
{  
  "refresh_token": refresh_token_string,  
  "mode": refresh_mode  
}
```

## Request

REST GraphQL SDK

POST /graphql/system

```
mutation {  
  auth_refresh(refresh_token: "abc...def", mode: json) {  
    access_token  
    refresh_token  
  }  
}
```

# logout

## Request

REST

GraphQL

SDK

POST /graphql/system

```
mutation {  
  auth_logout(refresh_token: "refresh_token")  
}
```

# API REST

- Directus permet d'accéder au contenu au travers d'une API REST
- Un point d'entrée par modèle de données défini dans le backoffice
- Des paramètres pour sélectionner les champs, filtrer, trier ...

`http://host/items/spectacles`

`?fields=titre,duree`

`?filter={"titre":{"_eq": "stones"}}`

`?sort=duree`

# récapitulatif

---

- Backend pour création d'une api d'accès à des données très rapide à mettre en place
- Types de données riches
- Très facile d'emploi pour des actions simples de type CRUD
- Possibilité d'ajouter du métier au travers de hooks et de flows (enchaînement d'actions sur événements)