ARTRAGIS

Geste de savoir

Strapi un CMS headless open source

31 août 2022

Table des matières

Introduction

Bonjour,

Dans ce tutoriel on va parler d'une classe d'outils de plus en plus utilisés: les CMS headless.

i Prérequis

Pour suivre ce tutoriel il faut simplement que vous soyez familier avec l'utilisation (en tant que lecteur ou que rédacteur) des outils qui permettent de publier du contenu sur internet. Pour la fin du tutoriel, surtout les TP, un peu de connaissance en javascript vous sera nécessaire.
Strapi utilise souvent la ligne de commande. Il est donc conseillé de se familiariser un peu avec cet outil.

1. CMS ? Headless ? De quoi tu parles ?

1.1. Content Management System

Abrégé SGC (Système de Gestion de Contenu) en français, les CMS sont une gamme de logiciels assez large.

Un CMS distingue deux types d'utilisateurs: les propriétaires et les visiteurs. Le concept est le suivant: Les propriétaires de l'application créent un «contenu», lui donnent des «métadonnées» et les visiteurs vont pouvoir accéder à la fiche ainsi rédigée de ces contenus.

Vous vous dites sûrement que tout ça est assez flou et donc qu'il y a un loup. Pourtant, si un vocabulaire aussi flou est utilisé c'est parce que la principale qualité des CMS modernes, c'est leur souplesse.

Ainsi vous pouvez proposer un blog, une vitrine pour télécharger des logiciels, une boutique dynamique… L'immense majorité des CMS sont capables de faire ça sans aucun souci. Certains ont une spécialisation qui leur permet d'avoir des fonctionnalités avancées, mais globalement l'idée restera toujours la même.

Pour rentrer dans le vif du sujet, on peut citer quelques CMS qui sont encore très connus en 2022:

— Wordpress : CMS open-source le plus connu de tous[1], il est généraliste bien qu'encore très connoté «blog». Il est codé en PHP. Vous pouvez choisir de l'héberger vous-même ou utiliser un hébergeur spécialisé.
— Shopify: CMS propriétaire et disponible «dans le cloud»[2] il est spécialisé dans la création rapide de boutiques en ligne.
— Wix: solution commerciale de création «NoCode» de site web vitrine et de CMS.
— Drupal , Prestashop : CMS Open source spécialistes de la boutique en ligne, eux aussi codés en PHP.

1.2. Headless

? Ok pour CMS, mais pourquoi *headless*?

Pas de panique! Un CMS *headless* n'est pas un CMS codé par Louis Croix V Bâton. Il s'agit d'un CMS qui n'a pas de… présentation. Autrement dit, sans manipulation supplémentaire, vous avez vos contenus qui sont publiés, mais pas d'interface graphique définie pour les voir.
Mieux, vous avez même le droit de définir vous-même de quoi sont constitués vos contenus!

? Mais ça sert à quoi si je peux pas les voir?

1. D'après l'enquête du site w3techs , il a à lui seul plus de la moitié du marché des CMS.
2. Plus précisément c'est un logiciel en SaaS, c'est à dire que l'application n'est pas hébergée par l'utilisateur.

En vérité, vous pouvez les voir, mais il faudra que vous définissiez vous-même la manière de les afficher et d'interagir avec eux[3]. Contrairement à Wordpress qui définit une structure rigide autour de laquelle des **thèmes** permettent de personnaliser la vue, les CMS *headless* vous proposent une **API REST** ou compatible **GraphQL** et à vous de définir la structure de la vue.

Pour vous y aider vous pourrez vous-même définir ce qu'est un contenu pour vous!

Par exemple, pour un vendeur de bijou fantaisie un produit c'est quoi sinon

— un nom,
— une (ou plusieurs) photos,
— un prix,
— un lien vers la boutique?

À l'opposé, pour un vendeur de location estivale, un produit c'est:

— un nom,
— une géolocalisation,
— une ou plusieurs marques différenciantes (ex: camping 4 étoiles),
— un prix **par personne**,
— un calendrier des disponibilités.

Avec un CMS habituel, vous devriez utiliser une extension spécialisée dans la réservation pour proposer le second cas, avec un CMS Headless, rien à ajouter pour l'instant.

Autre aspect, la possibilité de *choisir* votre GUI vous permet de partager la même source de données pour votre site web, une application mobile et même pour toute autre source de vente.

?

Du coup ce sont des CMS spécialisés dans la vente?

Eh bien non, et je vous propose que pour la suite du tutoriel, on se mette dans la peau de l'équipe de communication de votre site fruité préféré.

?

Mais du coup il faut savoir programmer?

Dans l'idéal, ce n'est pas nécessaire. En effet, certains logiciels incluent déjà le code capable de se brancher à Strapi pour afficher vos contenus, vous n'avez qu'à définir quel champ afficher à quel endroit et le logiciel se charge de faire le reste. C'est le cas par exemple de gatsby ou encore jekyll .

Cependant, dans les faits, pour vraiment tout customiser, il va falloir coder à un moment ou à un autre. C'est ce que nous allons faire dans le TP puisque nous allons utiliser javascript pour modifier ZDS.

3. Vous verrez parfois sur internet des gens parler de **GUI** pour *Graphical User Interface* pour parler de l'affichage. Et pour cause, ça veut dire «interface graphique d'utilisation», dans notre cas, une page web ou une application téléphone le plus souvent.

2. Préparons le TP

Afin de vous donner un exemple réel d'une utilisation de strapi, le CMS *headless* open source que je vais vous présenter, nous allons donc permettre à l'équipe de communication du site de fournir deux fonctionnalités importantes:

— promouvoir un contenu,
— envoyer des «news» en anglais et en français.

Afin que vous ne soyez pas perdu, je vous propose d'imaginer que ces deux éléments seront affichés sur ZDS de plusieurs manières différentes:

— pour les contenus «promus», s'ils sont marqués comme «star», ils doivent être affichés dans le menu principal (bibliothèque) à la place de la liste des tags les plus utilisés,
— pour les contenus «promus», s'ils sont marqués comme «live», ils doivent être affichés à la une, à la place du système actuel de unes.
— Pour les news on aura deux types d'éléments:
  — La news «majeure» affichée en entête du site, là où se trouvent d'habitude les alertes telle que «nous allons faire une maintenance», c'est la version française qui est toujours affichée.
  — Les news «mineures»:
    — Si elles sont marquées «privées», elles seront affichées sous la petite cloche, dans la langue de l'utilisateur.
    — Si elles sont marquées «publiques» elles seront affichées selon votre choix, mais un conseil, ne faites pas un de ces overlay qui rendent une partie de la page inutilisable.

A la fin nous aurons un code javascript qui fonctionne sur une instance locale de zds.

3. Découvrons Strapi

Passons au vif du sujet, et découvrons un CMS headless open source qui nous aidera à résoudre notre exercice: Strapi.

Strapi est, comme vous l'aurez deviné, un CMS headless généraliste opensource. Il est édité par une entreprise dont le principal modèle économique est de vendre du support ainsi que des fonctionnalités spécifiques aux entreprises telles que l'authentification par LDAP ou le *Single Sign-On*.

Si vous avez un projet personnel ou associatif, la version «community» qui est donc totalement open source, vous suffira amplement.

Notons que strapi est développé avec NodeJS, il vous faudra donc installer node et npm avant de démarrer l'outil. Pour les utilisateurs de Windows, vous trouverez les installateurs sur le site officiel .

3.1. Installation

Une fois les outils JS installés, l'entreprise éditrice de strapi vous fournit un moyen simple de démarrer votre instance de test:

```
npx create-strapi-app@latest zds-news --quickstart
```

?

Démarrer notre instance de test alors qu'on n'a toujours pas installé strapi?

En fait cette commande est deux en un: elle télécharge strapi le déploie dans le dossier qui ici s'appellera «zds-news» et démarre l'instance de test.

Dans notre cas «zds-news» est le nom de votre projet, cela va créer un sous-dossier que vous pourrez par exemple sauvegarder dans un dépôt git.

Une fois strapi installé, le système lance l'application et ouvre un onglet dans votre navigateur. Vous serez invité à créer votre compte administrateur.

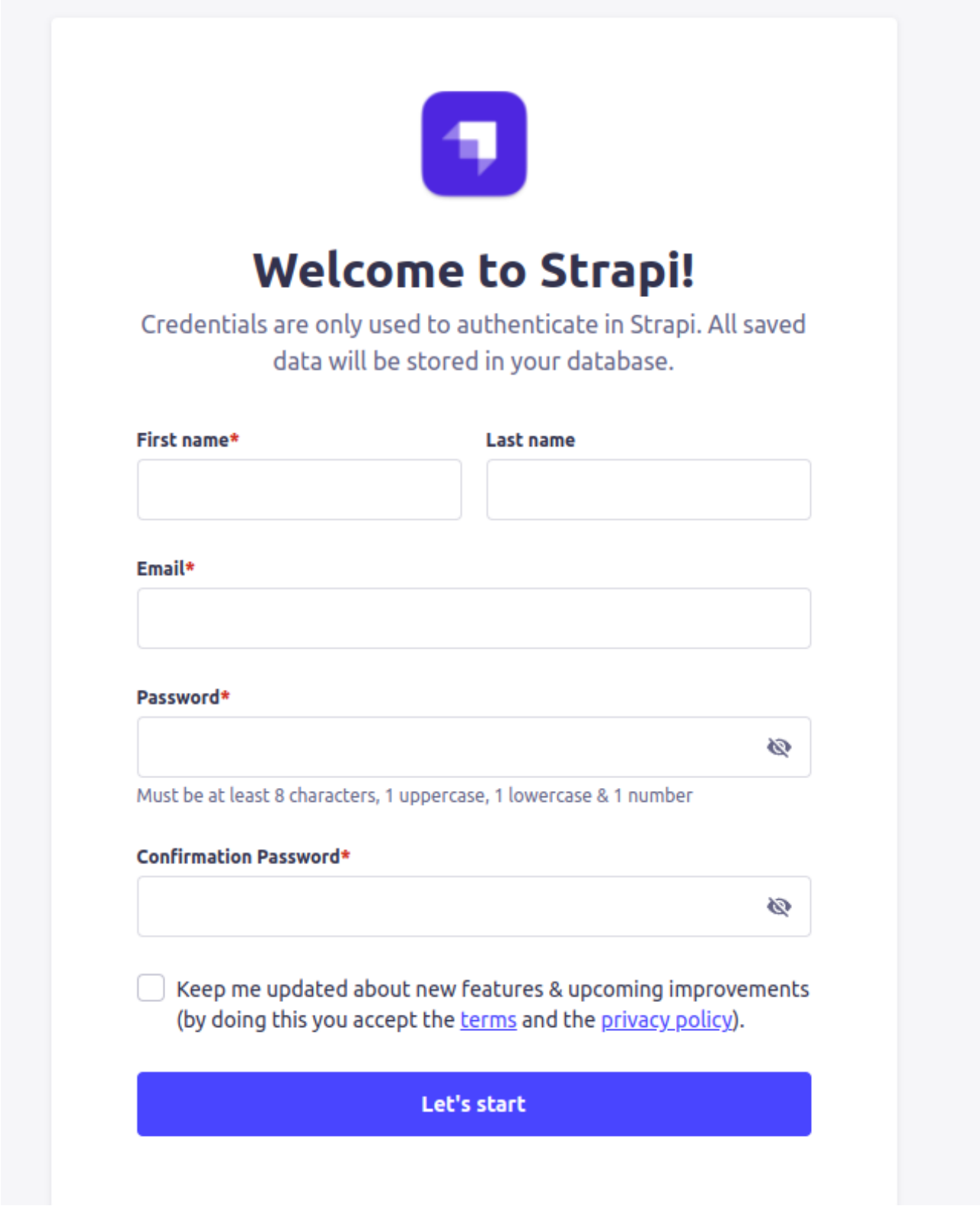


FIGURE 3.1. – Formulaire de création du compte administrateur

Une fois le compte administrateur créé, une page d'accès rapide se présente à vous et vous invite à créer votre premier type de contenu.

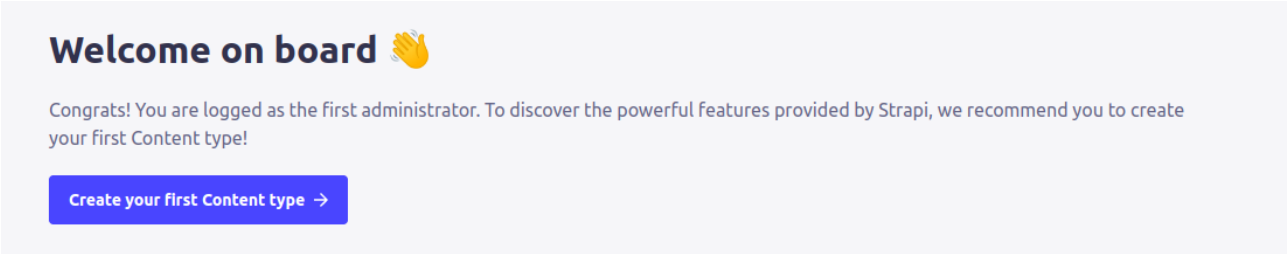


FIGURE 3.2. – Cliquez sur le bouton "Create your first content type"

3.2. Les types de données

Une fois que vous avez cliqué dessus, vous voyez une page vous invitant à créer des types de données. Le système les sépare en trois catégories: Single Type, Collection Type et Component.

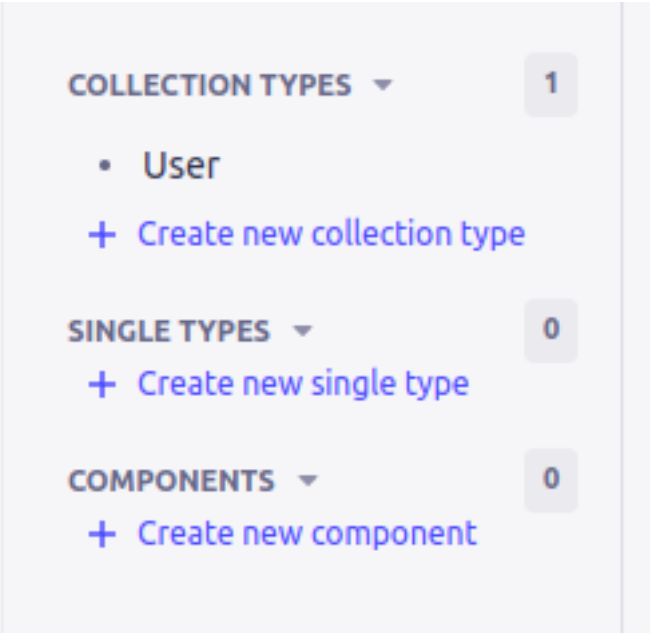


FIGURE 3.3. – Trois types de données: Single Type, Collection Type, Component

Vous êtes actuellement sur la page vous proposant de créer des «Collection type» alors commençons par ce modèle-là.

3.2.1. Collection type

Un type «collection» est un type de données dont vous vous attendez à avoir plusieurs exemplaires dans le fonctionnement de vos applications et dont vous allez fréquemment manipuler les listes.

Par exemple vous allez avoir plusieurs «unes» dans le TP et ZDS affichera les 5 dernières.

D'ailleurs, je vous propose de commencer par le type «Unes» afin de nous familiariser avec l'interface et de comprendre les concepts sous-jacents.

3.2.2. Single Type et Component

Les deux derniers types de données seront revus plus tard dans le cours, mais maintenant que vous avez pris en main les types collections, vous avez peut-être deviné ce qu'ils signifient:

— Un **Single Type**: est un type de données qui ne sera présent qu'une seule fois dans l'application. Dans notre cas, le TP a été créé pour que le message de news affiché en page d'accueil soit de ce type.
— Un **Component** est un ensemble de champs qui sera réutilisable par plusieurs types de données. Cela permet de garder des noms communs mais aussi de ne pas se répéter. Ça rend le développement plus rapide.

i Node de service

De part leur nature, les composants ne seront pas visibles de l'extérieur, ils sont à usage interne seulement.

3.2.3. Commençons à utiliser strapi

3.2.3.1. Créons les types On peut donc commencer par cliquer sur Create new Collection Type et remplir le formulaire qui arrive avec le nom de notre type: «une» (pour se simplifier la vie on va parler français 🍊).

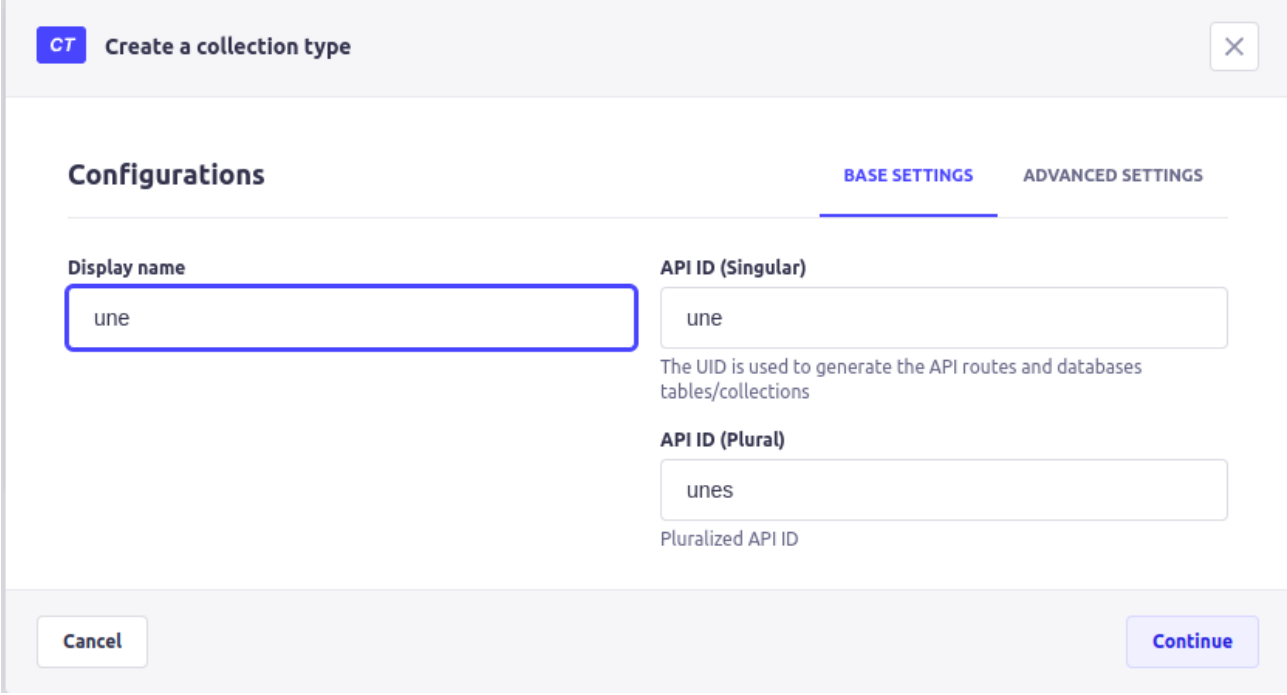


FIGURE 3.4. – Créer le type "une", on nous dit que l'endpoint d'API sera /unes (pluriel) ou /une (singulier) selon le besoin.

Une fois ce formulaire rempli, nous allons pouvoir le valider. Cela nous mène à une page remplie de petits widgets décrivant des types de champs.

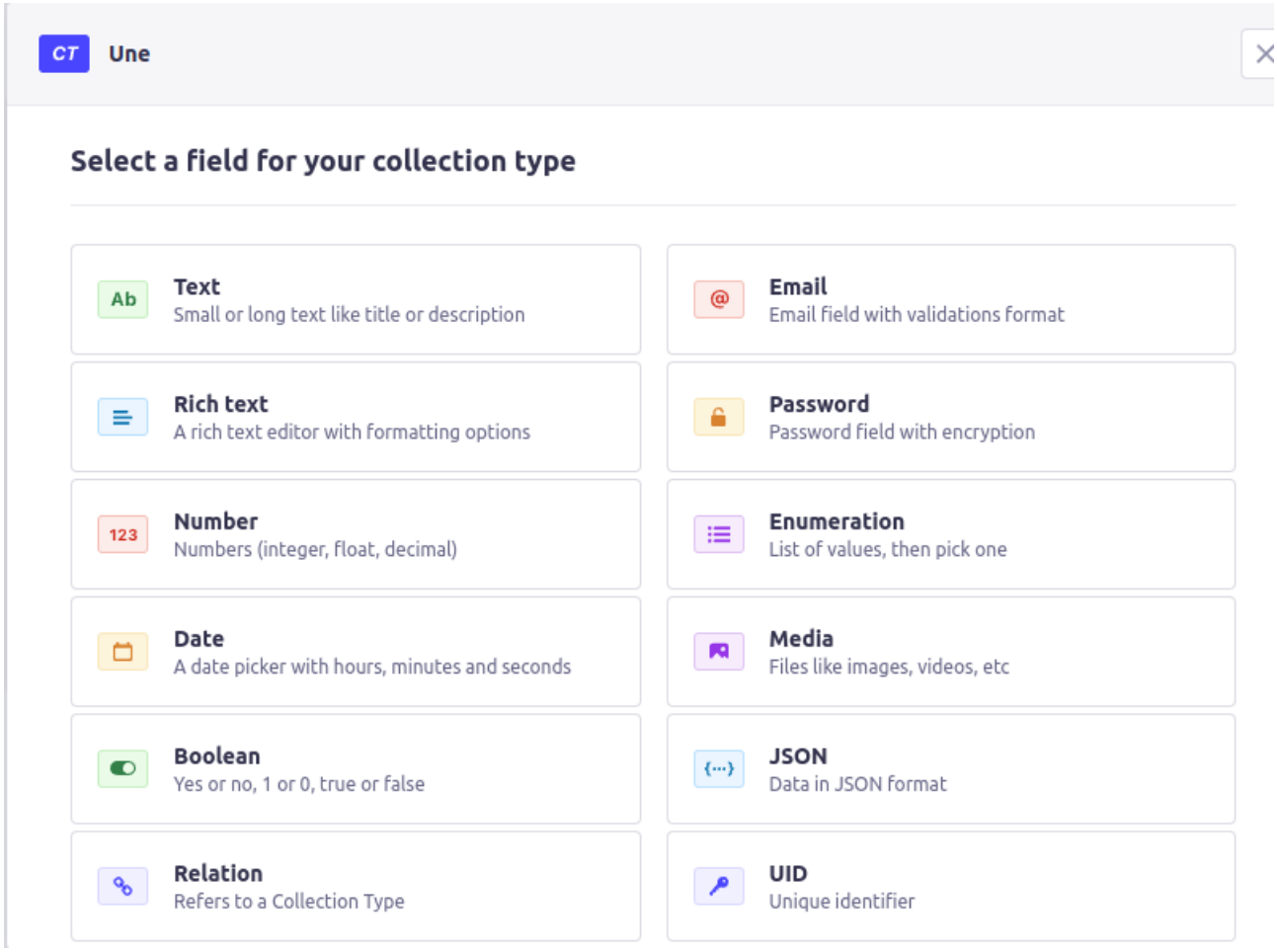


FIGURE 3.5. – Les types d'objets à ajouter, par exemple texte, image, lien…

Une *Une* est composée d'un titre (un texte), un type (tutoriel/article/forum), un lien et une image. Nous allons donc commencer par sélectionner «texte» et nommons-le «titre». Ensuite cliquons sur Advanced Settings.

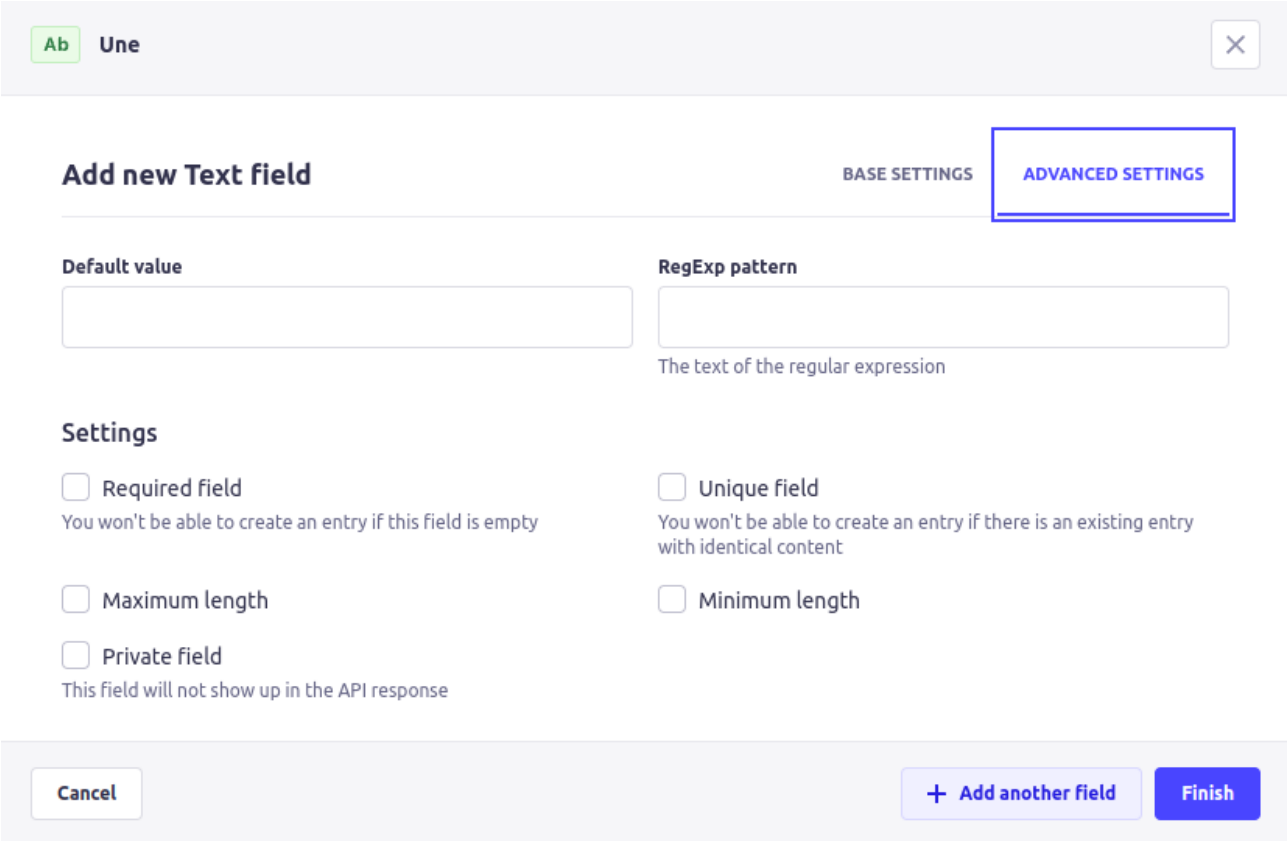


FIGURE 3.6. – Advanced Settings - on peut y décider du format du titre mais aussi du fait qu'il est obligatoire.

Comme le titre est un élément obligatoire, nous allons cocher la case Required Field.

Maintenant que le premier champ est rempli, nous pouvons passer au second en cliquant sur Add Another Field. Nous allons nous concentrer sur le type de une. Pour cela nous allons choisir le champ «énumération». Il nous permettra de définir les types acceptés. On l'appelle donc «type» et on entre les types:

— Tutoriel,
— Article,
— Billet,
— Forum.

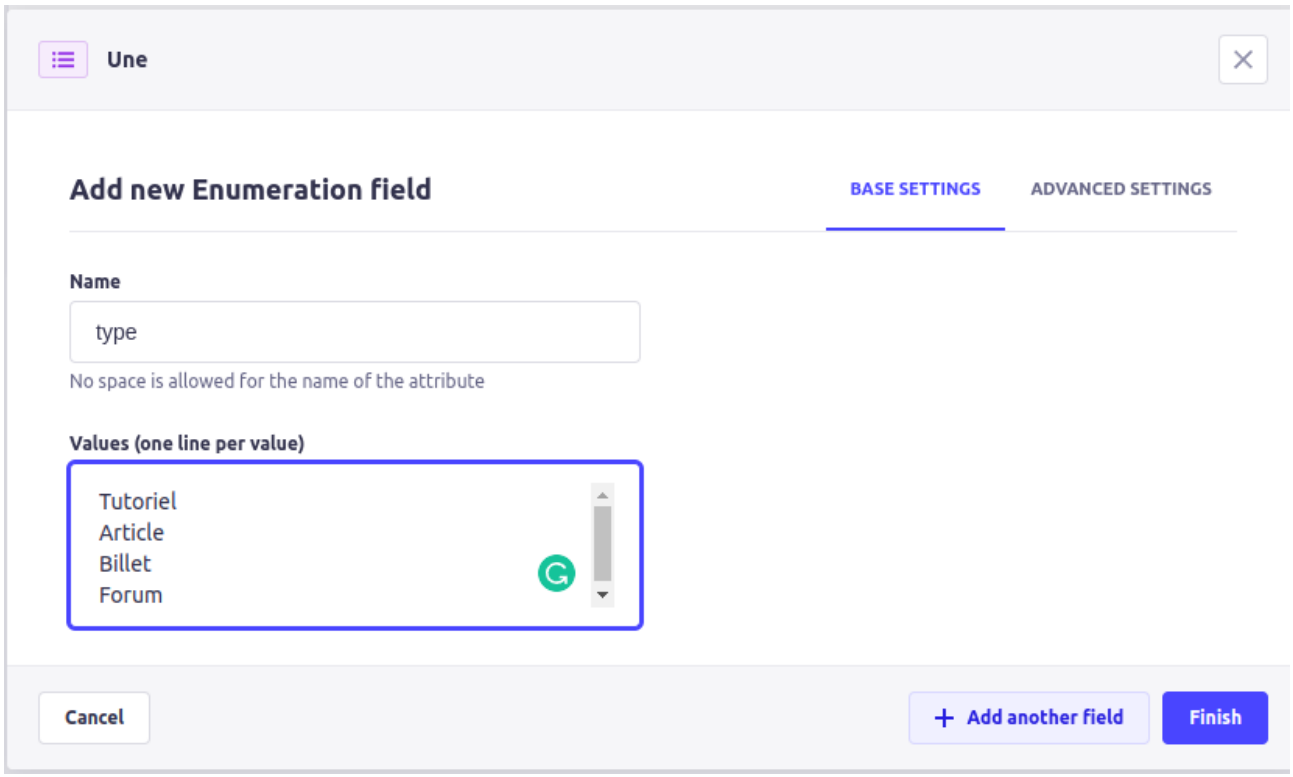


FIGURE 3.7. – La configuration du champ «type».

On peut alors recommencer la manipulation pour l'URL qui est un simple texte qui commence par http:// ou https:// (la regexp sera alors `^https?://`).

Enfin, nous allons dire qu'il nous faut une image, prenons donc le type «Media».

Comme toujours, il faut donner un nom (image) mais il faudra aussi définir le fait que nous n'allons ajouter qu'une seule image et non plusieurs.

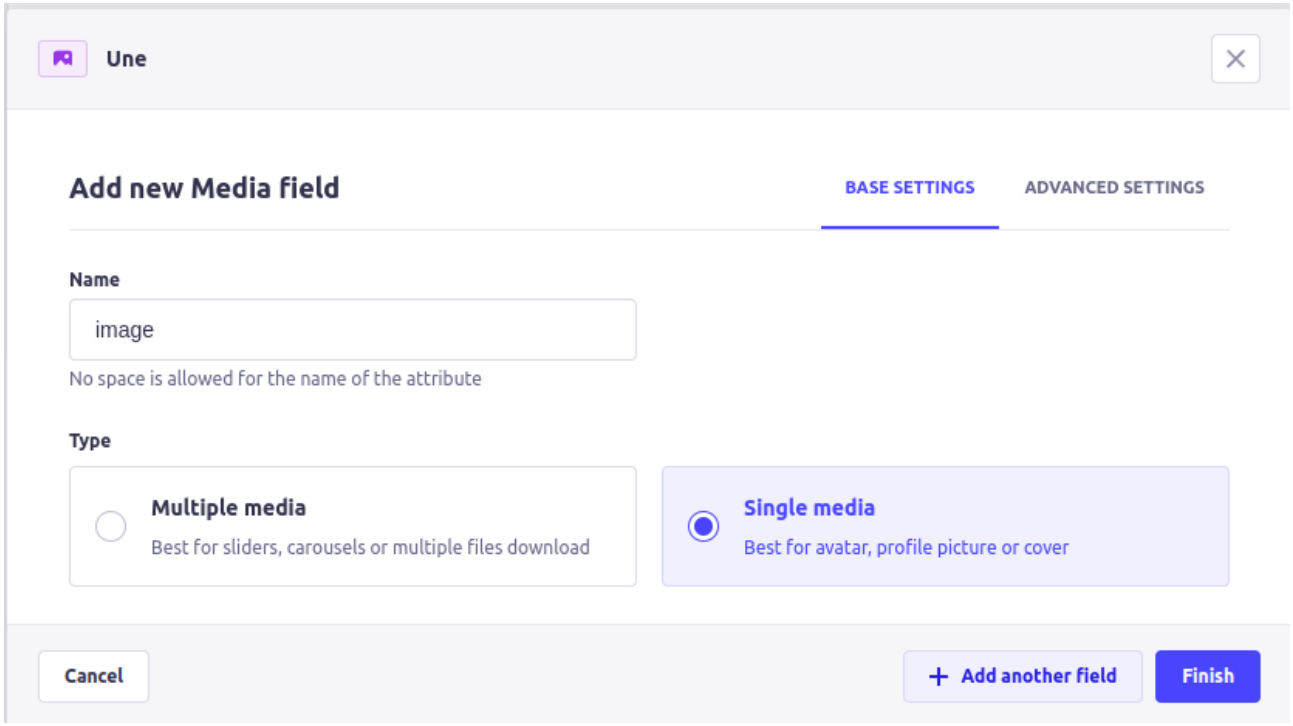


FIGURE 3.8. – Configurer le champ «image» pour qu'il n'intègre qu'un seul fichier.

De plus, il faut explorer les paramètres avancés pour restreindre l'ajout aux images.

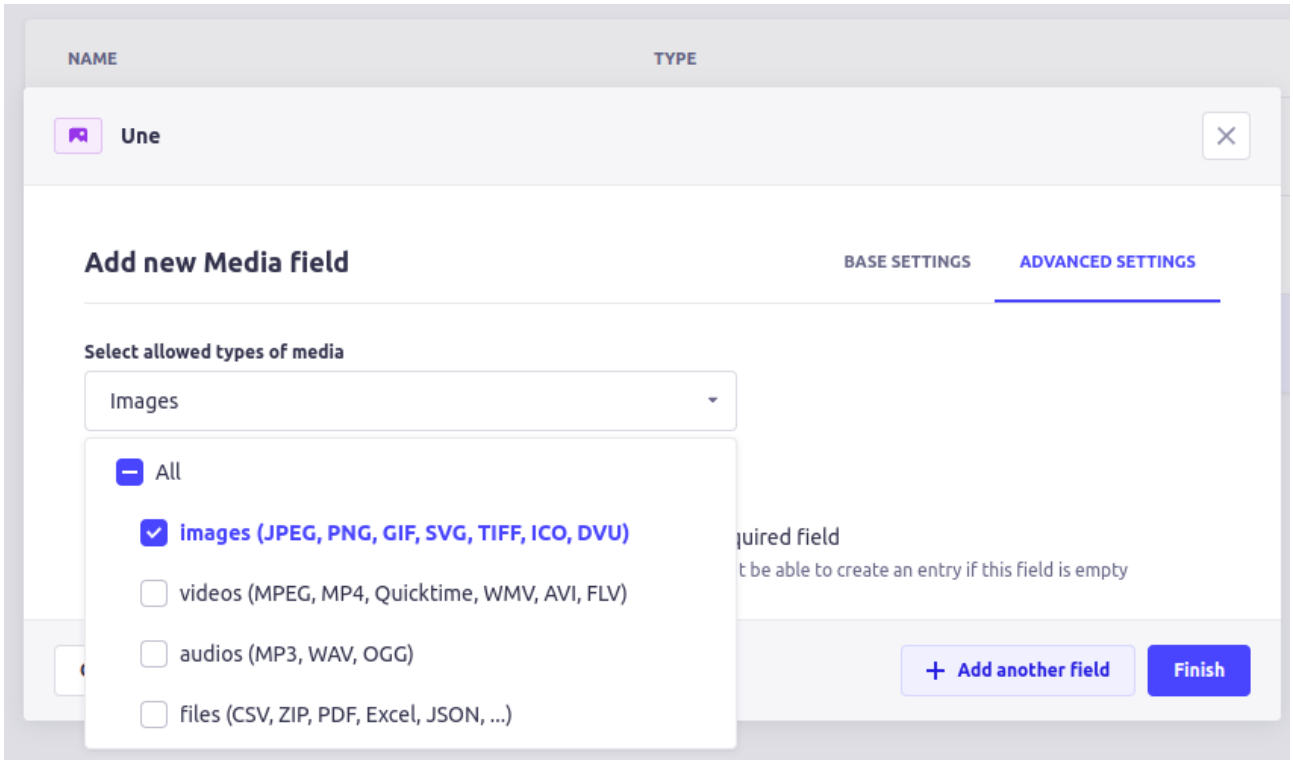
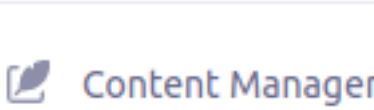


FIGURE 3.9. – Pour restreindre les médias aux images, il faut désélectionner les trois autres types de fichiers (vidéo, son, fichier).

Enfin nous pouvons terminer notre conception. **Il est important de cliquer sur le bouton «save» en haut, à droite**.

3.2.3.2. Créer notre première une Maintenant que nous avons défini ce qu'est une *une*, il va fal-



loir en écrire. Pour cela, nous allons nous rendre dans le **Content Manager**.

Comme c'est votre première visite, une pop-in vous invite à rédiger votre premier contenu, sinon, le bouton Create New Entry se trouve en haut à droite de votre page.

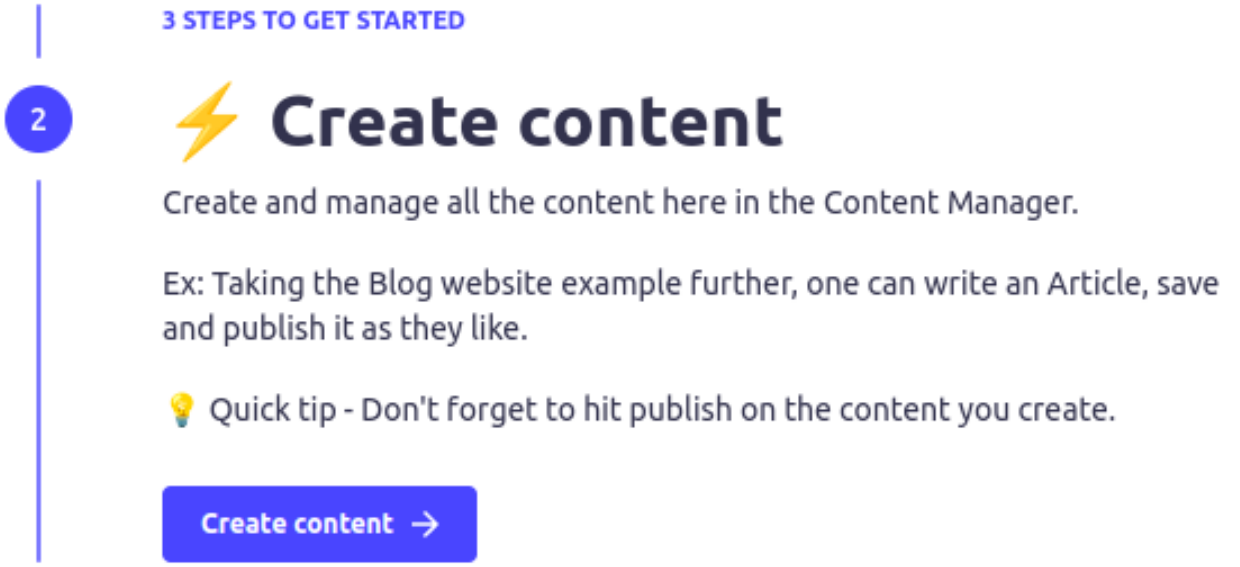


FIGURE 3.10. – Wizard vous guidant vers la première création de contenu.

Point vocabulaire

On notera que dans le langage de strapi, créer un contenu est appelé **entry**.

Nous voici donc sur un formulaire qui liste les champs à entrer.

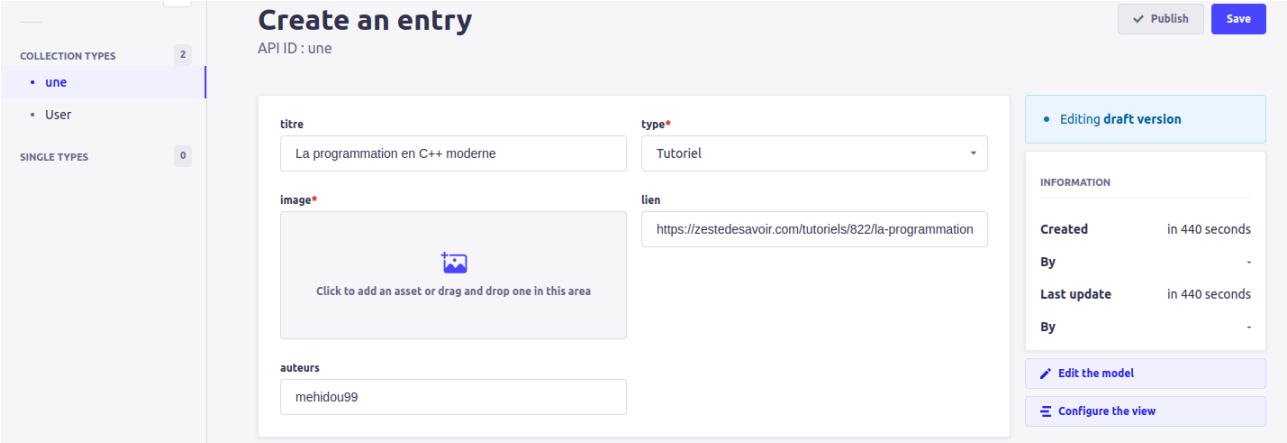


FIGURE 3.11. – Le formulaire marque d'une étoile rouge les champs obligatoires. J'ai fait en sorte de volontairement en mettre optionnels pour que vous le voyiez. Ici j'ai prérempli la une telle qu'on peut la voir sur zds le 03 mai 2022.

Il nous reste à envoyer une image. Afin de limiter l'empreinte disque, strapi repose sur un système de galerie. Il va donc falloir ajouter une image à ladite galerie en la téléchargeant. Une fois cela fait il suffit de sélectionner l'image pour que cela fonctionne.

Cliquer sur le bouton Save pour enregistrer votre une.

? Eh, tu aurais pas fait une erreur sur le pseudo de mehdidou99?

Si, et j'espère qu'il me le pardonnera, mais c'était le meilleur moyen pour moi de faire une transition vers la suite.

Logiquement, le «quick tour» vous propose d'aller voir l'API, pour l'instant nous allons fermer cet encart et nous allons cliquer sur le type «une» pour qu'il nous montre la liste des unes.

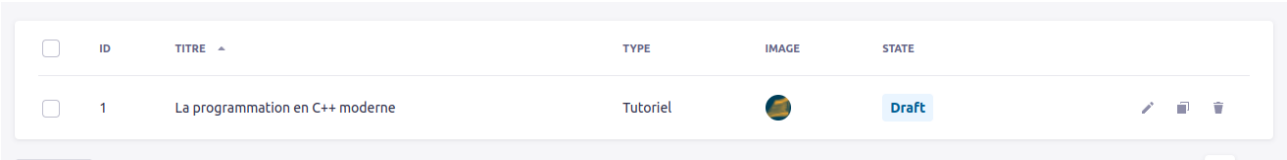


FIGURE 3.12. – La liste des unes avec notre une (La programmation en C++ moderne) affichée comme un draft.

Comme vous pouvez le constater notre une est encore un brouillon (Draft signifie brouillon en anglais). Cela veut dire que personne ne peut la voir autre que nous. Et nous allons pouvoir cliquer sur l'icône crayon pour éditer notre une et corriger cette erreur au pseudo de l'auteur du livre La programmation en C++ Moderne.

Une fois que vous avez sauvegardé, vous pouvez voir que «Publish» est activé. Cliquons dessus afin de rendre notre une publique.

3.3. Mode développement/mode production

Avant de passer à la suite, une petite note s'impose à propos du démarrage de strapi.

Tout ce que nous venons de faire est arrivé car nous avons utilisé le «mode développement» qui permet de définir la structure des contenus. En «mode production» seuls l'édition de l'affichage et la création de contenus sont possibles.

Pour démarrer strapi en mode production, il suffit de faire `npm start`[1] ou `npx strapi start`, pour le démarrer en mode développement, il faut faire `npm run develop` ou `npx strapi develop`.

4. Dialoguer avec l'API

4.1. Récupérer les unes

4.1.1. L'url

L'API strapi est assez complète et permet aussi bien de lire les entités que de les modifier. Pour ne pas rendre ce tutoriel trop long, nous allons nous contenter d'utiliser l'url pour lire le contenu du type que nous venons de créer.

Comme il s'agit d'un type collection, l'url de base est construite ainsi: `http://localhost:1337/api/{nom au pluriel}`. Dans notre cas il faudra donc accéder à `http://localhost:1337/api/unes`.

? Pourquoi j'ai une erreur ?

Eh oui, nous sommes en train de découvrir le logiciel alors notre première touche aura forcément eu pour réponse:

```
{
    "data":null,
    "error":{
        "status":403,
        "name":"ForbiddenError",
        "message":"Forbidden",
        "details":{}
    }
}
```

Listing 1 – Une erreur 403 - c'est à dire que vous n'avez pas le droit d'accéder à la ressource.

1. si vous êtes venus sur ce tutoriel en ayant déjà l'habitude d'utiliser npm et l'outil de ligne de commande, vous avez peut-être par automatisme ajouté un "-g" au départ. Si tel est le cas, vous pouvez simplement utiliser les commandes `strapi start` et `strapi develop`.

Je vous recommande de lire ce tutoriel pour mieux comprendre pourquoi le code associé à cette erreur est 403, mais en attendant, il va falloir que nous réglions ce problème.

4.1.2. Régler les permissions

Strapi vient avec un système de sécurité assez complet basé sur l'octroi de privilège en fonction de votre statut. On pourrait par exemple imaginer, dans le cas de ZDS d'avoir ce genre de statuts:

— «Président»: peut tout faire,
— «Développeurs»: peuvent changer le modèle,
— «Communication»: peuvent poster des unes,
— «Les anonymes»: peuvent juste afficher les unes mais rien de plus.

Le modèle de permission sépare les utilisateurs en deux types les «administrateurs» et les «end-users». Nous y reviendrons dans une prochaine partie, mais sachez que les utilisateurs non authentifiés (ce que vous étiez quand vous avez simplement copié l'url d'API dans votre navigateur) sont des end-users.
Pour donner le droit de voir les unes à tout le monde on doit aller dans settings > Users and Permissions Plugin > Roles.
Une fois à cet endroit, on peut déplier le volet «Public» puis configurer les droits sur le endpoint «unes».
Nous allons autoriser «find» car cela permet de lister les unes.
Après avoir sauvegardé, vous obtiendrez:


```
{
    "data":[
        {
            "id":1,
            "attributes":{
                "titre":"La programmation en C++ moderne",
                "type":"Tutoriel",
                "lien":
                    "https://zestedesavoir.com/tutoriels/822/la-programmation-e
                    ,
                "auteurs":"mehdidou99",
                "createdAt":"2022-05-03T11:00:42.277Z",
                "updatedAt":"2022-05-03T11:07:12.519Z",
                "publishedAt":"2022-05-03T11:07:12.517Z"
            }
        }
    ],
    "meta":{
        "pagination":{
            "page":1,
            "pageSize":25,
            "pageCount":1,
            "total":1
        }
    }
}
```

Listing 2 – Notre une est là avec toutes les données sauf l'image.

Le résultat est déjà plus prometteur! 
On peut noter que l'image n'est pas présente dans le résultat et pour cause: c'est un média qui possède plusieurs versions: complète, thumbnail… Afin de ne pas surcharger la réponse par défaut, le système se retient de la proposer, mais il est possible d'obtenir les informations nécessaires en ajoutant un paramètre à l'URL: `?populate=image`.


```
{
    "data": [
        {
            "id": 1,
            "attributes": {
                "titre": "La programmation en C++ moderne",
                "type": "Tutoriel",
                "lien": "https://zestedesavoir.com/tutoriels/822/la-programmation-e",
                "auteurs": "mehdidou99",
                "createdAt": "2022-05-03T11:00:42.277Z",
                "updatedAt": "2022-05-03T11:07:12.519Z",
                "publishedAt": "2022-05-03T11:07:12.517Z",
                "image": {
                    "data": {
                        "id": 1,
                        "attributes": {
                            "name": "article-illu.83268abf72a5.png",
                            "alternativeText": "article-illu.83268abf72a5.png",
                            "caption": "article-illu.83268abf72a5.png",
                            "width": 177,
                            "height": 177,
                            "formats": {
                                "thumbnail": {
                                    "name": "thumbnail_article-illu.83268abf72a5.pn",
                                    "hash": "thumbnail_article_illu_83268abf72a5_95",
                                    "ext": ".png",
                                    "mime": "image/png",
                                    "path": null,
                                    "width": 156,
                                    "height": 156,
                                    "size": 22.09,
                                    "url": "/uploads/thumbnail_article_illu_83268a
```

```
                                }
                            },
                            "hash": "article_illu_83268abf72a5_957806f6e1",
                            "ext": ".png",
                            "mime": "image/png",
                            "size": 3.18,
                            "url": "/uploads/article_illu_83268abf72a5_957806f6e1",
                            "previewUrl": null,
                            "provider": "local",
                            "provider_metadata": null,
                            "createdAt": "2022-05-03T10:59:59.579Z",
                            "updatedAt": "2022-05-03T10:59:59.579Z"
                        }
                    }
                }
            }
        ],
        "meta": {
            "pagination": {
                "page": 1,
                "pageSize": 25,
                "pageCount": 1,
                "total": 1
            }
        }
}
```


Listing 3 – Une une complète!

4.1.3. Publier

Essayons maintenant de créer un nouveau contenu mais sans le publier. Si on appelle de nouveau l'URL de l'API, on obtient uniquement la une déjà publiée.
Si pour quelque raison que ce soit nous voulions aussi afficher les unes en draft, il suffirait d'ajouter le paramètre suivant à l'URL: `?publicationState=preview`.

i

Cette notation étrange à base de `?`, `=` ou encore `&` est un moyen de définir des paramètres dans une URL quel que soit le site. Vous en avez par exemple l'habitude avec Youtube qui, pour voir la 10ème seconde de cette vidéo de Rick Ashley vous propose: `https://www.youtube.com/watch?v=dQw4w9WgXcQ&t=10s`: le `v` est le nom du paramètre pour identifier la vidéo, le `t` le nom du paramètre pour définir le temps. Le `?` sépare les paramètres du reste de l'URL et `&` sépare les paramètres pour en mettre plusieurs.

? Et si je voulais *planifier* la publication, comment je pourrais faire?

Strapi ne propose pas une fonctionnalité officiellement appelée ”planifier une planification”. Bien que certains de ses concurrents le fassent, on peut noter qu'il existe une astuce que tous les CMS *headless* proposent pour faire vous-même votre système de publication: ajouter un champ ”publication date” à votre entité.
On pourrait donc éditer le modèle des unes, ajouter un champ date de publication puis explorer la documentation de l'API pour savoir comment on ne sélectionne que les entrées dont la date de publication est antérieure à maintenant.
On obtiendrait alors une url qui vaut `http://localhost:1337/api/unes?populate=image&filters[publication_date][$le][2022-05-16T13:00:00Z]` et ça marcherait.

? Il faut formater la date comme ça? C'est vraiment bizarre.

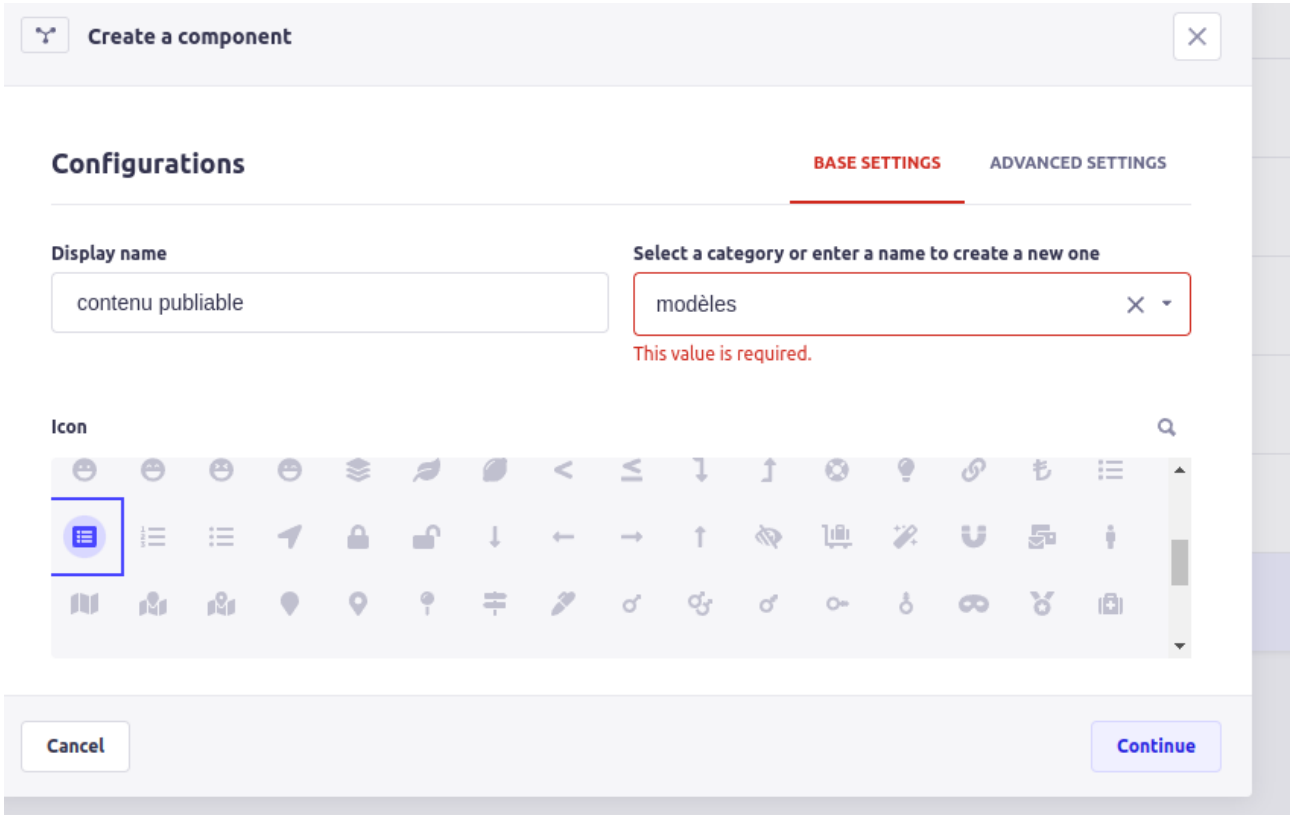
Il s'agit d'un format normalisé, tous les langages de programmation savent le produire. Car il faut se souvenir que ***consommer une API*** c'est avant tout le travail d'un programme.
On parlera plus tard de la manière d'intégrer facilement Strapi dans un programme déjà existant si vous n'êtes pas développeur. Pour l'instant, je vous propose de découvrir les *composants*.

5. Les composants réutilisables

On l'avait déjà vu avant, il existe un type de contenu appelé les «composants». Ces derniers sont utilisés pour définir un ensemble de champs identiques à plusieurs *types*.
Par exemple, nous voulons absolument gérer le fait que les unes peuvent être publiées dans le futur, mais nous voulons aussi, plus tard, afficher le message d'accueil seulement durant un temps donné.
Pour cela, on va dire que nos objets auront tous deux champs:

— une date de publication (obligatoire),
— une date de fin de publication (optionnelle).

Allons dans le *Content-Builder* et créons un nouveau **Composant**:



Legende : Entrer le nom du composant. Je vous laisse tester avec «modèle» pour voir ce que dit Strapi. Comme beaucoup de logiciels, il n'aime pas les accents.
Une fois le composant créé, on va ajouter deux champs de type `datetime`: `publication_date` et `end_of_publication`:

FIGURE 5.13. – Création du champ publication_date de type datetime.

Il faut penser à sauvegarder une fois le formulaire validé.

Maintenant, nous allons éditer le modèle `une` et cliquer sur le bouton Add a new field.
Tout en bas du panneau des types, sélectionnez «Component». Nous allons pouvoir ajouter un *composant déjà existant*:

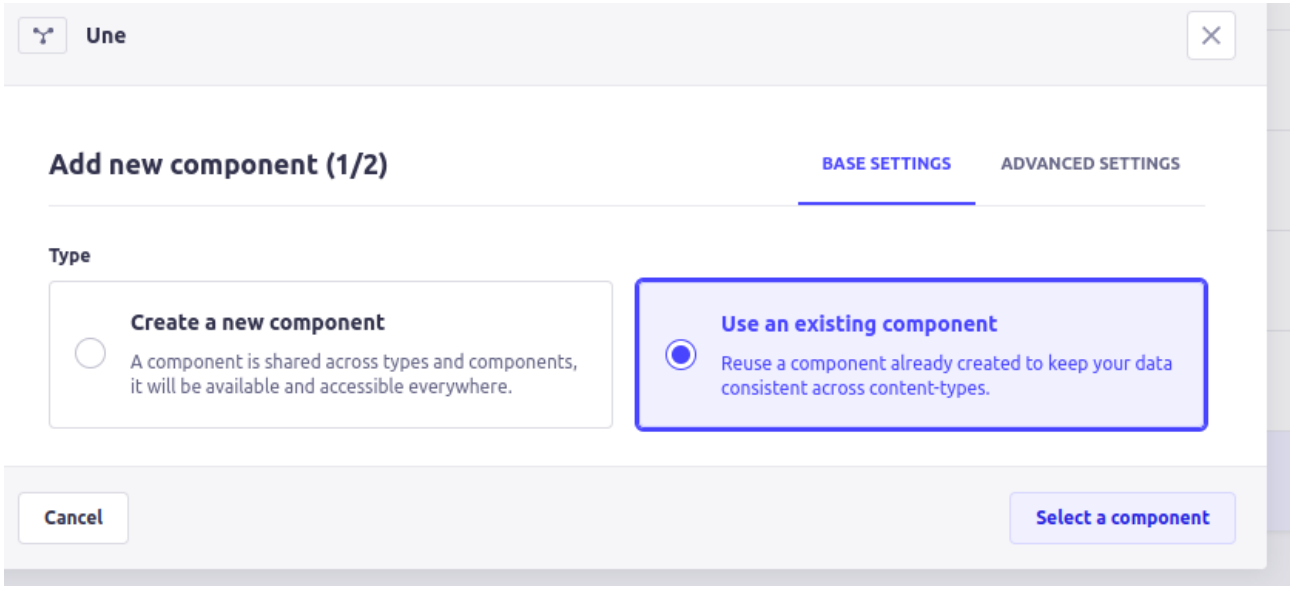


FIGURE 5.14. – Le formulaire d'ajout de composant, avec Select Existing Component activé.

La suite du formulaire nous invite à faire le choix entre **Repeatable Component** et **Single Component**.
Le premier est là pour nous permettre de gérer des listes d'objets, alors que l'autre permet de réutiliser une structure prédéfinie sans avoir à la recréer à la main. C'est donc cette deuxième option que nous allons ici choisir:

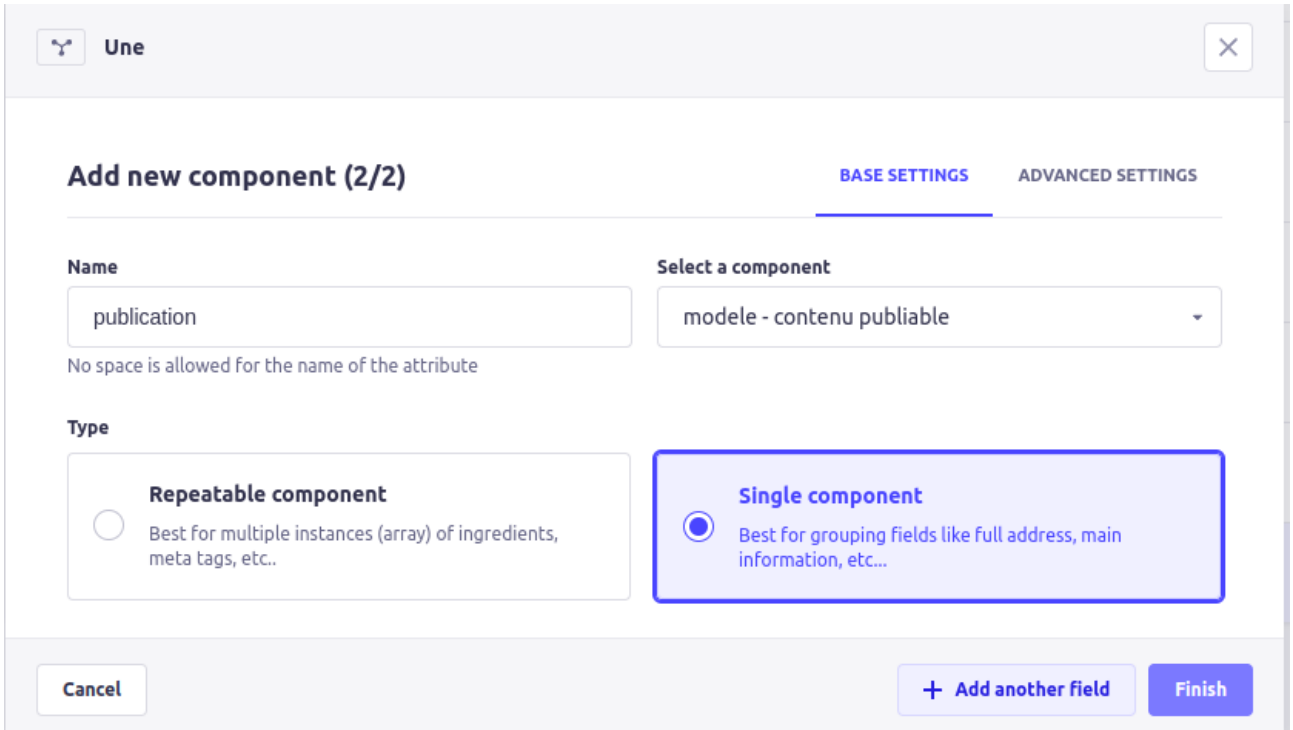


FIGURE 5.15. – Ajout du composant: on sélectionne le type contenu publiable et on le prend en tant que single component.

? Que se passe-t-il pour les objets déjà publiés?

Ils n'ont pas de composant sélectionné. Pour notre cas c'est parfait, cela permet de dire que si le composant n'est pas ajouté alors l'objet est publié tout de suite et pour l'éternité. À l'opposé, si le composant existe, nous pourrons utiliser les dates sélectionnées.

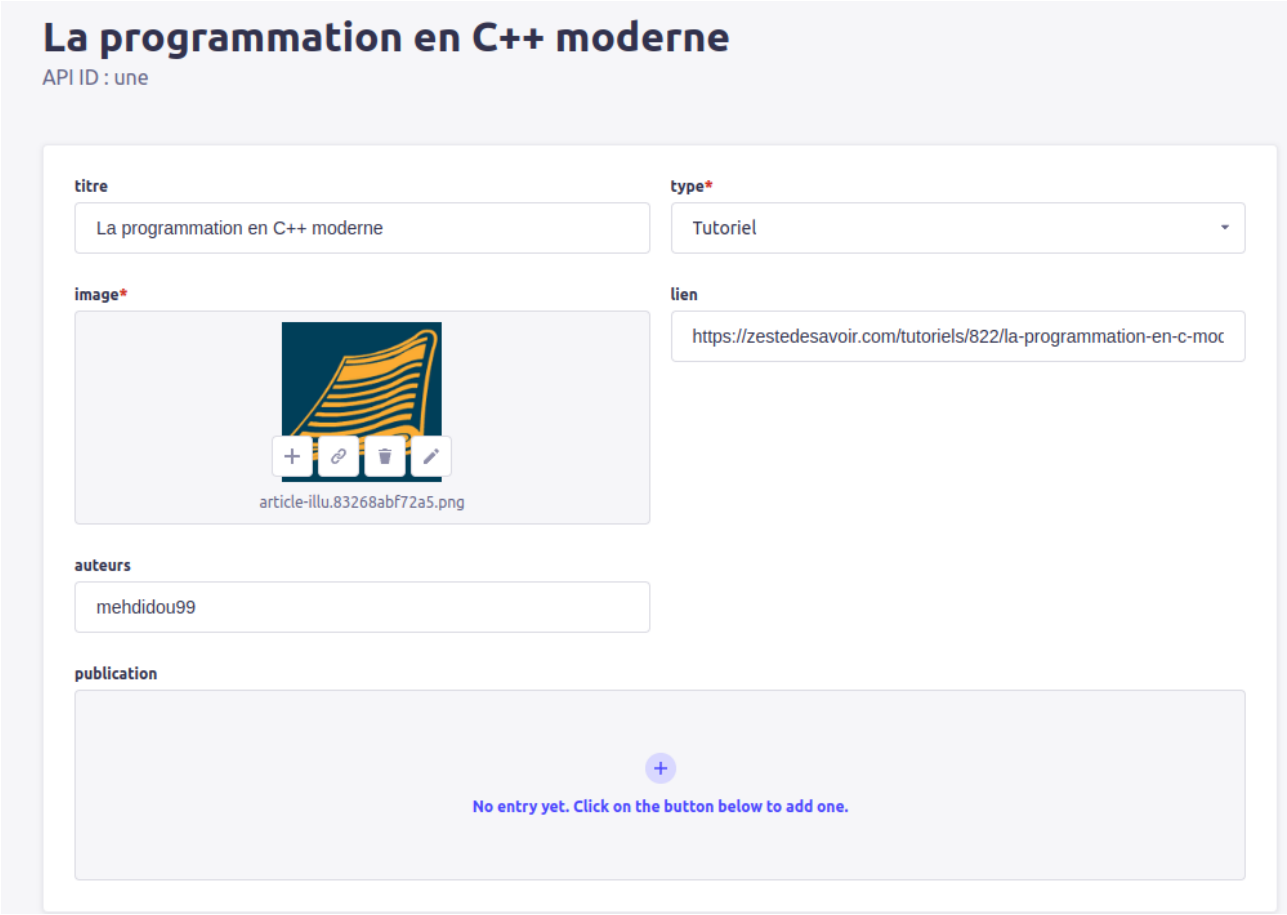


FIGURE 5.16. – La une sur le C++ sans le composant de publication.

6. TP time : on va changer les unes de ZDS

6.1. Rappel des faits

Notre but est de découvrir les avantages d'utiliser un CMS headless. Si vous n'avez pas l'habitude de coder, cela sera difficile pour vous de faire le code complexe qui va gérer l'utilisation des dates de planification.
Dans un premier temps on va donc faire le TP en mode simple: pas de planification. Le but sera simplement d'afficher les unes dans ZDS. Une fois qu'on aura affiché ces unes, on va explorer un peu plus les avantages d'avoir un CMS pour faire tout ça.
Enfin, en exercice personnel, vous pourrez ajouter un peu de complexité avec la planification des unes.

6.1.1. Version simple : on ne tient pas compte de la planification

6.1.1.1. Quelques données ZDS affiche ses unes de manière assez simple.
Les unes sont mises dans des balises `<article>` qui ont la classe `featured-resource-item`.

Dans ces articles on a une balise `a` pour le lien, une balise `img` pour l'image, et une balise `p` pour l'auteur ou l'autrice.
Pour obtenir les 5 unes les plus récentes, il faudra trier par date de publication en ordre décroissant puis ne prendre que 5 éléments.
Vous pourrez vous aider de la documentation du tri et de la pagination .
En créant 3 unes, j'ai eu le résultat suivant:

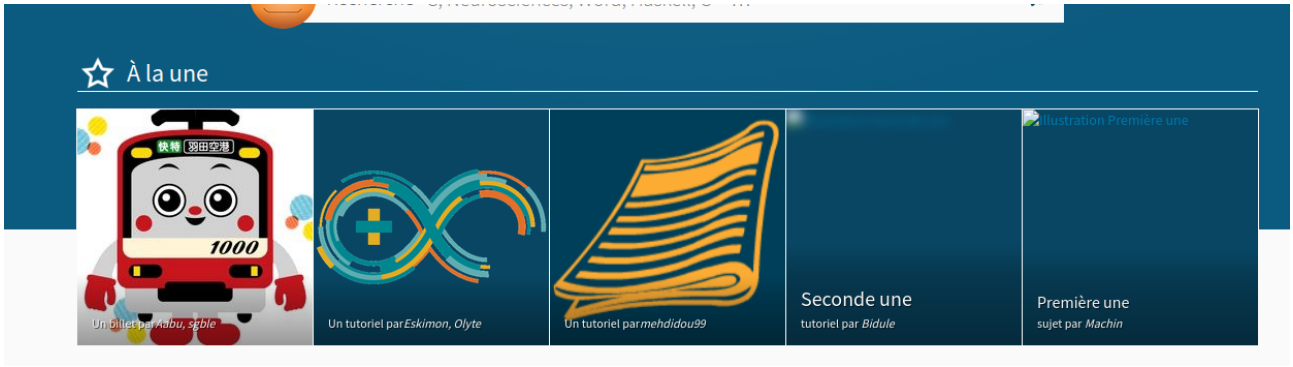


FIGURE 6.17. – Le résultat: trois unes les plus récentes

6.1.1.2. Correction

code JS

6.1.2. Un CMS Headless reste un CMS

Les CMS comme Wordpress et concurrents sont devenus très célèbres car ils permettent de gérer facilement tout ce qui est relatif à la publication et la mise en avant d'un contenu.
Il en est de même avec les CMS headless. Pour vous expliquer ce que je veux dire par là, imaginons qu'en plus de votre site web, vous ayez un Discord pour discuter avec votre communauté ou un Mattermost pour discuter avec votre équipe. Vous aimeriez qu'ils soient prévenus quand un contenu est publié?
Si vous avez lu le tutoriel d'Eskimon vous savez qu'il est possible de faire un **WebHook** pour communiquer avec Discord mais aussi avec tout un tas de plateformes.
Strapi vous permet de faire de même. La seule limite c'est que les webhook produits par strapi ont leur propre structure qui n'est pas celle qu'attend discord par exemple.
Heureusement il existe d'autres ressources sur le Web pour vous permettre de transformer cet envoi en quelque chose de compatible avec toutes les applications. On peut par exemple penser à ifttt .

6.1.3. Version avec la planification et les autres consignes

Avec la planification, il faudra faire un filtre plus complexe: soit publication.publication_date doit être `null` soit il doit être arrivé avant aujourd'hui et publication.end_date doit être null ou arriver plus tard.
Comme le filtre est plus sélectif, il faudra sûrement gérer le cas où vous manquez de unes de ce fait.
Autre aspect, souvenez-vous, nous voulions deux types de contenu: star et live. Il va donc falloir que vous trouviez un moyen pour avoir ces deux types de contenus et que chaque entrée puisse

être l'un, l'autre, ou les deux. Une fois les entités changées il faudra faire d'autres filtres et modifier d'autres éléments du site pour les afficher dans le menu plutôt que dans la partie des unes.
Pour l'internationalisation, on en parle juste après. 🍊
Je vous laisse faire l'exercice pour vous-même, vous pourrez venir sur les forums pour discuter de vos codes.

7. Internationalisation

Une problématique courante dans le monde des CMS consiste à présenter deux versions d'un même contenu adapté à deux langues différentes ou plus.
Spacefox vous a déjà parlé des pièges de l'internationalisation, et ici nous allons nous concentrer sur ne proposer qu'une traduction aux différentes personnes.
Pourquoi ne se concentrer que sur ça? Rappelez-vous, vous êtes sur un CMS headless, cela permet par exemple d'avoir deux interfaces différentes en fonction des langues de chacun et ainsi s'adapter plus facilement au sens de lecture, à la longueur des mots, à leur potentielle hyphenation…
Nous allons nous concentrer sur le contenu. Mais avant toute chose, il va falloir dire à strapi que nous comptons utiliser plusieurs langues.
Aller dans Settings puis Internationalization. Là, vous aurez une liste de *Locales* qui normalement se réduit à l'anglais pour l'instant.
Nous allons choisir français sans préférence régionale. C'est peut-être une erreur à septante millions[1] mais pour l'instant on va faire simple.
Dans les *Advanced Settings*, nous allons la placer en tant que locale par défaut de nos contenus. Cela veut dire que si quelqu'un demande un contenu sans préciser la locale, il l'aura en Français.
Pour continuer notre exploration de strapi, nous allons créer un nouveau type de contenu. Cette fois-ci nous allons prendre un single type content. Il nous permettra d'afficher notre message d'accueil.
Après lui avoir donné un nom (j'ai choisi `home message`), il faudra aller dans Advanced Settings pour activer l'internationalisation pour ce contenu:

1. Vous l'avez?

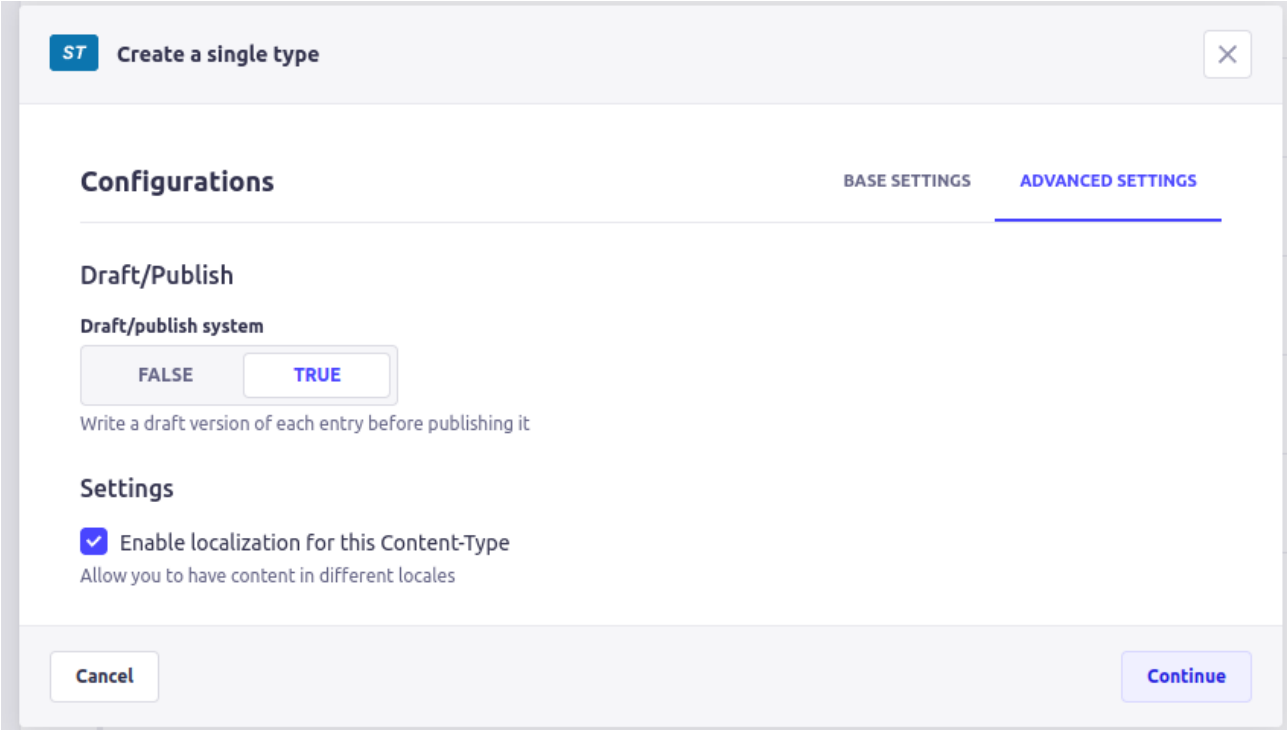


FIGURE 7.18. – Les paramètres avancés avec l'internationalisation activée.

Pour l'instant le contenu ne sera constitué que d'un simple texte. Après avoir validé, on peut donc créer un message, avec «Bonjour tout le monde par exemple».

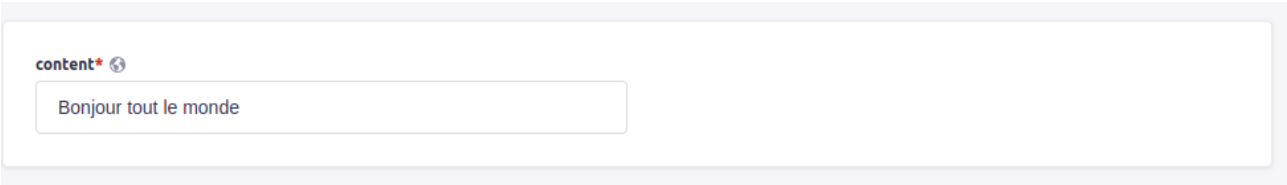


FIGURE 7.19. – Mettre le message.

> Pour que le message soit disponible à l'URL `http://127.0.0.1:1337/api/home-message` il faudra le publier et donner au public le droit de le voir.

Maintenant, si on demande le contenu en anglais en ajoutant `locale=en` à l'URL, on obtient:

```
{
    "data":null,
    "error":{
        "status":404,
        "name":"NotFoundError",
        "message":"Not Found",
        "details":{}
    }
}
```

Le système ne trouve pas de contenu en anglais. Et pour cause: nous n'en avons pas créé.

Il faut retourner dans l'édition du home message et regarder le menu à droite, là on va sélectionner la locale anglaise:

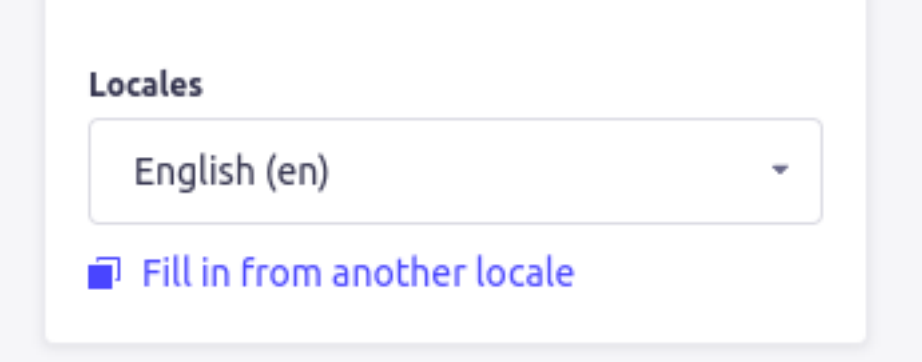


FIGURE 7.20. – Sélection de la locale anglaise.

On va ainsi pouvoir créer un contenu spécifique à l'anglais, qu'il faudra aussi publier.

```
{
    "data":{
        "id":2,
        "attributes":{
            "content":"Hello world!",
            "createdAt":"2022-05-23T13:10:15.092Z",
            "updatedAt":"2022-05-23T13:18:37.672Z",
            "publishedAt":"2022-05-23T13:12:21.145Z","locale":"en"
        }
    },
    "meta":{}
}
```

Comme le type est complètement internationalisé, vous pourrez aussi ajouter des champs numériques qui convertissent unité par unité.
Et comme vous êtes sur un CMS, vous profitez des plugins, de l'automatisation, etc.

8. D'autres CMS headless ?

Le monde des CMS headless est un écosystème commercial très riche. Au moment d'écrire ce tutoriel, le plus important d'entre eux est Contentful qui est proposé en SaaS.
Strapi est un logiciel qui doit être déployé sur un serveur par vos propres soins. Une offre SaaS est à l'étude cependant.
Chez les logiciels open source, on a sanity.io qui propose un système équivalent à strapi mais a déjà une offre SaaS en plus de son offre On-Premise.
J'ai aussi eu l'occasion de tester le CMS français prismic qui est en SaaS, non open source mais qui vous proposera une version gratuite.
Quel que soit le modèle économique de ces logiciels, l'augmentation du prix se fait toujours sur trois critères globaux:

— le nombre d'utilisateurs administrateurs (qui ont accès à tout)/non administrateurs (qui peuvent créer le contenu ou juste accéder à du contenu privé) disponibles,

— l'espace disque disponible pour les médias ainsi que la présence, ou non, de CDN,
— la présence ou non d'un support.

Vous pourrez trouver une liste exhaustive sur jamstack.org .

Conclusion

Voilà qui termine ce tutoriel qui, je l'espère, vous aura fait découvrir cet outil.
Je remercie @Ekron pour la validation ainsi que mon collègue julien bras qui m'a fait découvrir cette technologie.

Contenu masqué

Contenu masqué n°1 : code JS

```
function flushText(featuredElement, une) {
  const authorNode = document.createElement("i")
  authorNode.textContent= une.attributes.auteurs
  let textNode;
  switch (une.attributes.type) {
    case "Tutoriel":
      textNode = document.createTextNode("Un tutoriel par")
      break
    case "Article":
      textNode = document.createTextNode("Un article par")
      break
    case "Billet":
      textNode = document.createTextNode("Un billet par")
      break
    default:
      textNode =
          document.createTextNode("Une discussion lancée par")
break;
  }
  featuredElement.querySelector(".featured-resource-description").
      innerHTML
      =""
  featuredElement.querySelector(".featured-resource-description").
      append(textNode)
  featuredElement.querySelector(".featured-resource-description").
      append(authorNode)
}
function getFeatured() {
  const featured =
      document.querySelectorAll('.featured-resource-item')
  if (!featured) {
    return
```

Contenu masqué

```
27    }
28    let featureNb = 0;
29    const nbMax = featured.length;
30    fetch(
          "http://localhost:1337/api/unes?| sort[0]=publishedAt:desc&populate=image
          )
31      .then(async r => {
32        const json = await r.json()
33        json.data.forEach(une => {
34          if (featureNb >= nbMax) {
35            // sinon on va ajouter plus de unes qu'on a de carrés
36            return
37          }
38          featured[featureNb].querySelector("img").setAttribute(
                "src",
39            `http://localhost:1337${une.attributes.image.data.attrib
                utes.url}`)
40          featured[featureNb].querySelector("img").parentNode.setAtt
                ribute("href",
                une.attributes.lien)
41          featured[featureNb].querySelector("h2").textContent =
                une.titre
42          flushText(featured[featureNb], une)
43          featureNb++;
44        })
45      })
46  }
47  getFeatured()
```

Retourner au texte.

Liste des abréviations

CDN Content Delivery Network. 27