

CONTENT TYPES	1
DATA FIELDS - UNDERSØG ALLE FIELD TYPER	1
RELATION FIELDS - UNDERSØG MULIGHEDERNE	2
COMPONENTS	3
CONTENT MANAGER	3
GLOBAL CONFIGURATION PARAMETERS	3
SEO	3
AUTHENTICATION	3
JSON	3
API	3

## Content types

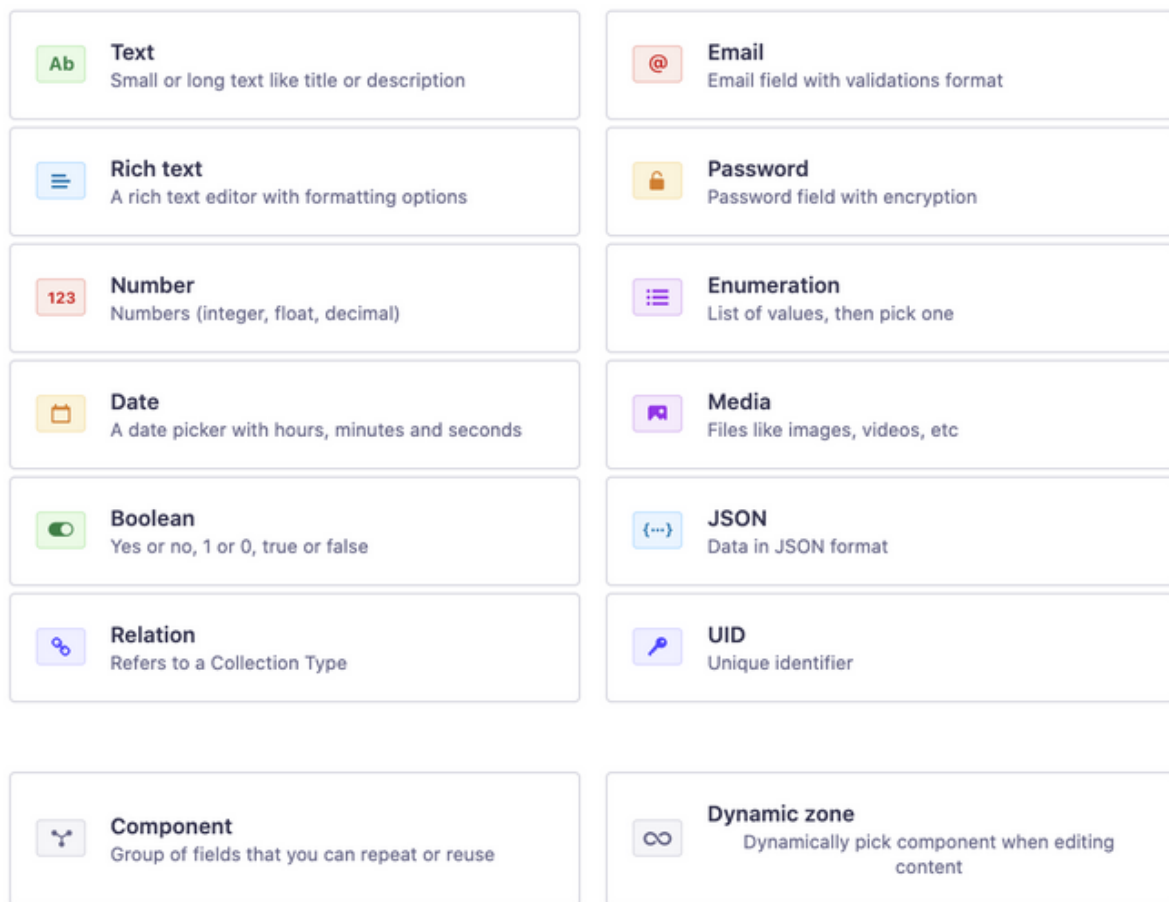
Content types er inddelt i to hovedkategorier. Collections og single. Collections kan indeholde mange forskellige entries, hvor single er til en enkelt.

Collections er derfor nok den mest åbenlyse til de fleste formål som en tabel over entries.

Single type er oplagt at bruge til f.eks. en hjemmesides undersider. "Om"-side eller "Kontakt"-siden, som der i sagens natur kun skal være én af.

## Data fields - Undersøg alle field typer

Content-types består af de her data fields. De er:



Mange af felterne giver sig selv. Boolean er et binært felt, rich text giver formatteringsmuligheder mv.







**Components** er en kombination af felter, så man hurtigt kan tilføje et sæt af felter til content-types. De kan nestes.

**Dynamic zones** er en måde at kombinere components på, som giver en fleksible struktur til måden, man opbygger sit indhold.

## Relation fields - Undersøg mulighederne

Relations er en måde at kæde content-types sammen på. En åbenlys måde er at tilføje kategorier eller anmeldelser til andre collection content-types.

There are 6 different types of relations:

-  One way: Content-type A *has one* Content-type B
-  One-to-one: Content-type A *has and belong to one* Content-type B
-  One-to-many: Content-type A *belongs to many* Content-type B
-  Many-to-one: Content-type B *has many* Content-type A
-  Many-to-many: Content-type A *has and belongs to many* Content-type B
-  Many way: Content-type A *has many* Content-type B

Som man kan se i ovenstående, er der 6 forskellige typer af relations.

**Many-to-many:** F.eks. kan en film have mange forskellige genrer, ligesom en genre kan indeholde mange forskellige film.

**Many-to-one:** En restaurant kan også have mange forskellige reviews, hvor et review er bundet op på en enkelt restaurant.

**May-to-one:** Men en restaurant kan kun have én rating, som til gengæld kan være forbundet med mange restauranter (eks. 4-stjernede restauranter)

## Components

Jf. ovenstående er components en kombination af felter, så man hurtigt kan tilføje samme data fields til en content-type.

## Content Manager

This is where the magic happens. Det er her, man kan tilføje content til APIet.

Content-Type builderen er det mere strukturelle plugin, hvor man opbygger collections og relations.

## Global Configuration Parameters

Jeg er ikke helt sikker på, hvad der menes med det her. Er det global settings?

## SEO

SEO-felter kan laves som components, så man hurtigt kan tilføje dem til det relevante content.

Man kan f.eks. lave et slug-felt, der har name-felt som attached field, og når man så opretter nyt content i den collection, vil den autogenerere slug-feltet ud fra name-feltet.

<https://strapi.io/blog/headless-cms-strapi-seo-best-practices>

## Authentication

Det er muligt at bruge tokens til authentication. Der er tre roles med i den gratis plan. De betalte planer har ubegrænsede custom roller

<https://docs.strapi.io/developer-docs/latest/plugins/users-permissions.html>

## JSON

Det er at tilføje json som et data field til en content-type.

## API

REST API gør det muligt at tilgå content-types gennem API endpoints. Endpoints genereres automatisk, når en content-type bliver tilføjet.

Rest API'et tilføjer dog ikke relations, components og media fields. Her skal man bruge populate, fields og select parametrene for at returnere specifikke felter.

På min localhost bruger jeg f.eks. [http://localhost:1337/api/movies?populate=\\*](http://localhost:1337/api/movies?populate=*)

<https://docs.strapi.io/developer-docs/latest/developer-resources/database-apis-reference/rest/populating-fields.html>