Making a Blog

So far we only considered contributing pages visually, directly in Page Builder. This is great for individual, unique pages but it's not the best way to manage a blog. In a blog, you want all your articles to have the same structure and style, and you want to be able to manage them all in one place.

Blog Post Type

Let's start by creating a node type called `blogPost`. We'll keep it concise but feel free to add more fields if you need them. Create a new file named `src/components/BlogPost/definition.cnd`:

```
[hydrogen:blogPost] > jnt:content, mix:title, jmix:mainResource,
hydrogenmix:component
 - subtitle (string) i18n mandatory
 - authors (string) multiple
 - cover (weakreference, picker[type='image']) mandatory < jmix:image
 - body (string, richtext) i18n mandatory
```

You might notice the `multiple` keyword next to authors: its allows setting a list of strings instead of a single one. Because it is not marked as mandatory, it may be undefined: we will treat this case as an empty list. `mix:title` is a mixin that adds a `jcr:title` field to the node type.

Create a `types.ts` file in the same folder:

```
import type { JCRNodeWrapper } from "org.jahia.services.content";

export type Props = {
  "jcr:title": string;
  "subtitle": string;
  "authors"?: string[];
  "cover": JCRNodeWrapper;
  "body": string;
};
```

Let's also create a simple view to render the blog post as a card:

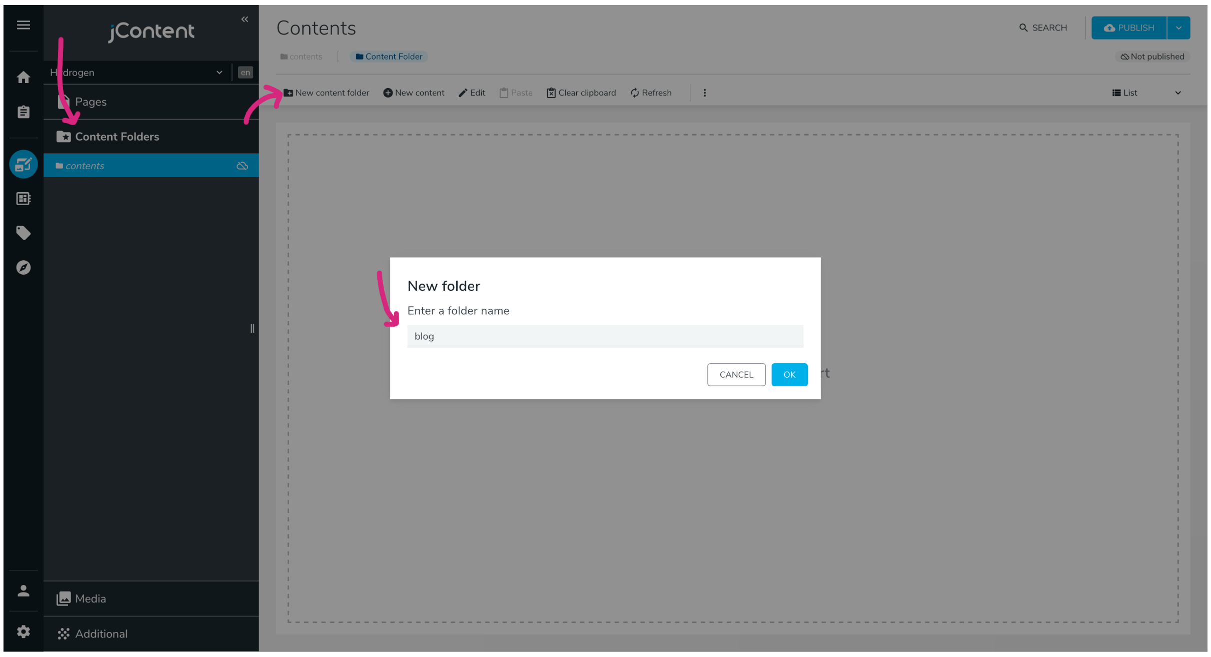
▶ `src/components/BlogPost/default.server.tsx`

▶ `src/components/BlogPost/component.module.css`

Content Folders

Instead of creating blog posts directly in Page Builder, we'll store our posts in **Content Folders**. This way, we can manage them all in one place and easily list them.

Create a new folder named `blog` in the `contents` folder:

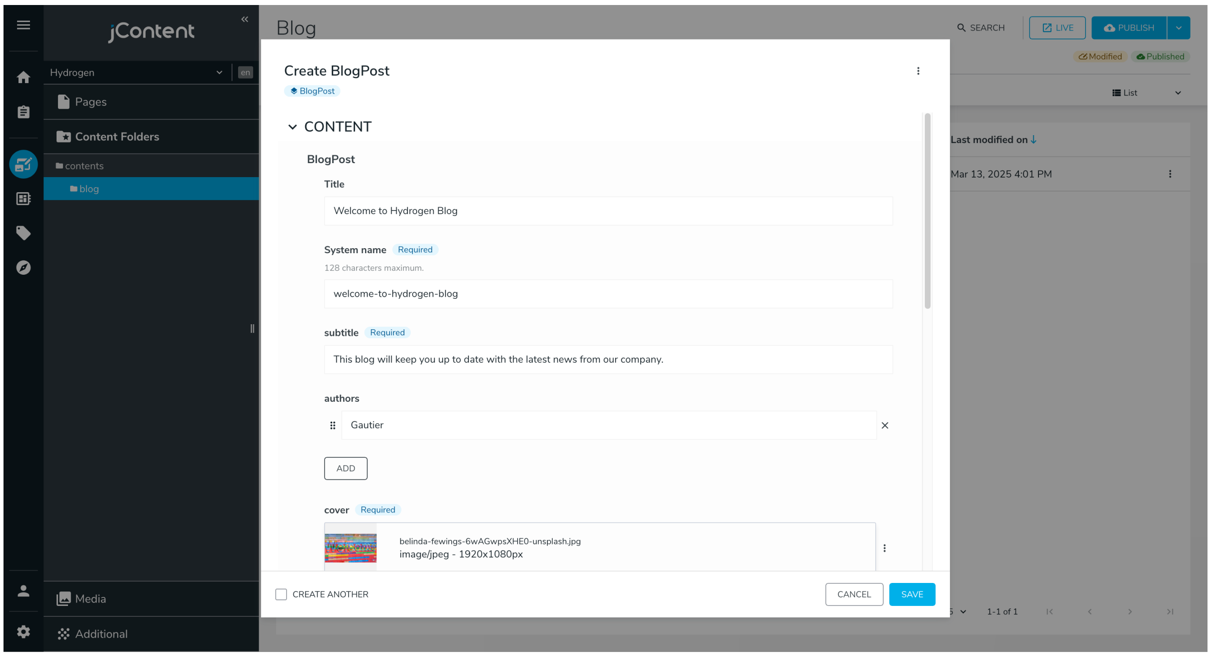


New folder

Enter a folder name

blog

Create a few blog posts in this folder:



Create BlogPost

BlogPost

CONTENT

BlogPost

Title: Welcome to Hydrogen Blog

System name (Required; 128 characters maximum.): welcome-to-hydrogen-blog

subtitle (Required): This blog will keep you up to date with the latest news from our company.

authors: Gautier

cover (Required): belinda-fewings-6wAGwpsXHE0-unsplash.jpg, image/jpeg - 1920x1080px

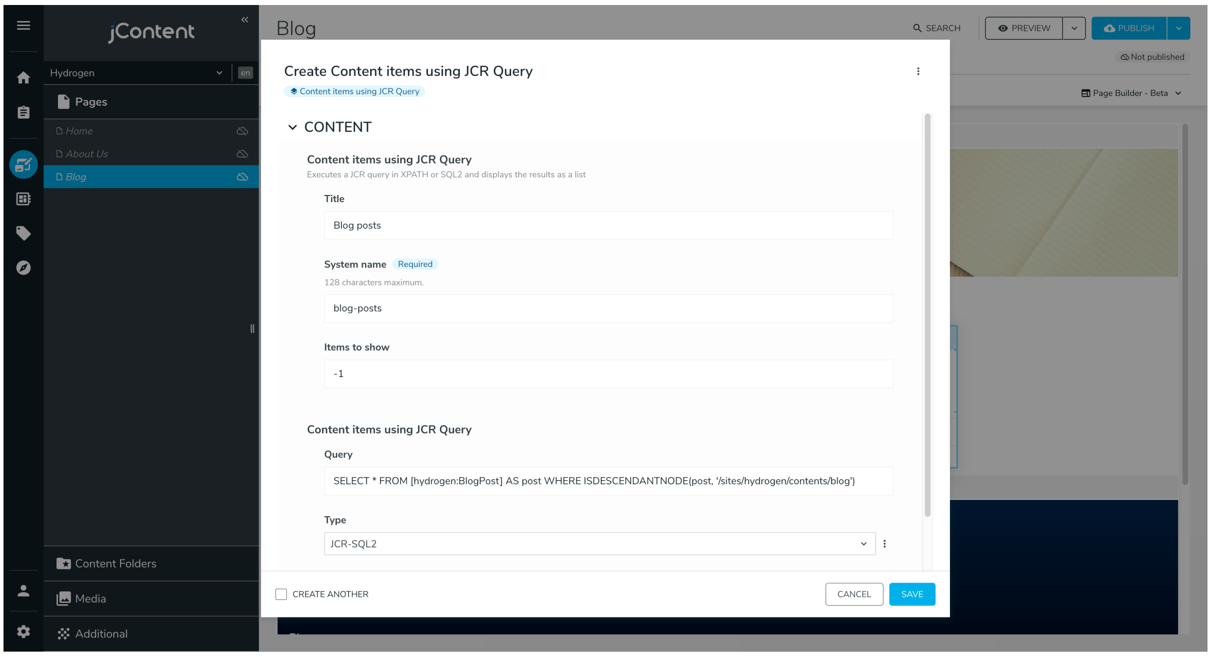
Listing Articles

Let's get back to the **Pages** tab of the sidebar (above Content Folders). Create a new page named "Blog" with a **Single column** layout. Create a Hero section on top of the page to introduce the list of articles.

In the main area, add a new **Jahia - Queries > Content items using JCR Query** component. The JCR is the tree-like database where all your content is stored, pages and content folders alike. We will use a JCR query to list all the blog posts. In the query field, enter:

```
SELECT *
FROM [hydrogen:blogPost] AS post
WHERE ISDESCENDANTNODE(post, '/sites/hydrogen/contents/blog')
```

JCR-SQL2 is a SQL-like language to query the JCR. This query selects all nodes of type `hydrogen:blogPost` that are descendants of the `/sites/hydrogen/contents/blog` node. In this query, `hydrogen` is the site key, you might have set it to something else when creating your site.



Create Content items using JCR Query

Content items using JCR Query

CONTENT

Content items using JCR Query

Executes a JCR query in XPATH or SQL2 and displays the results as a list

Title: Blog posts

System name (Required) — 128 characters maximum.: blog-posts

Items to show: -1

Content items using JCR Query

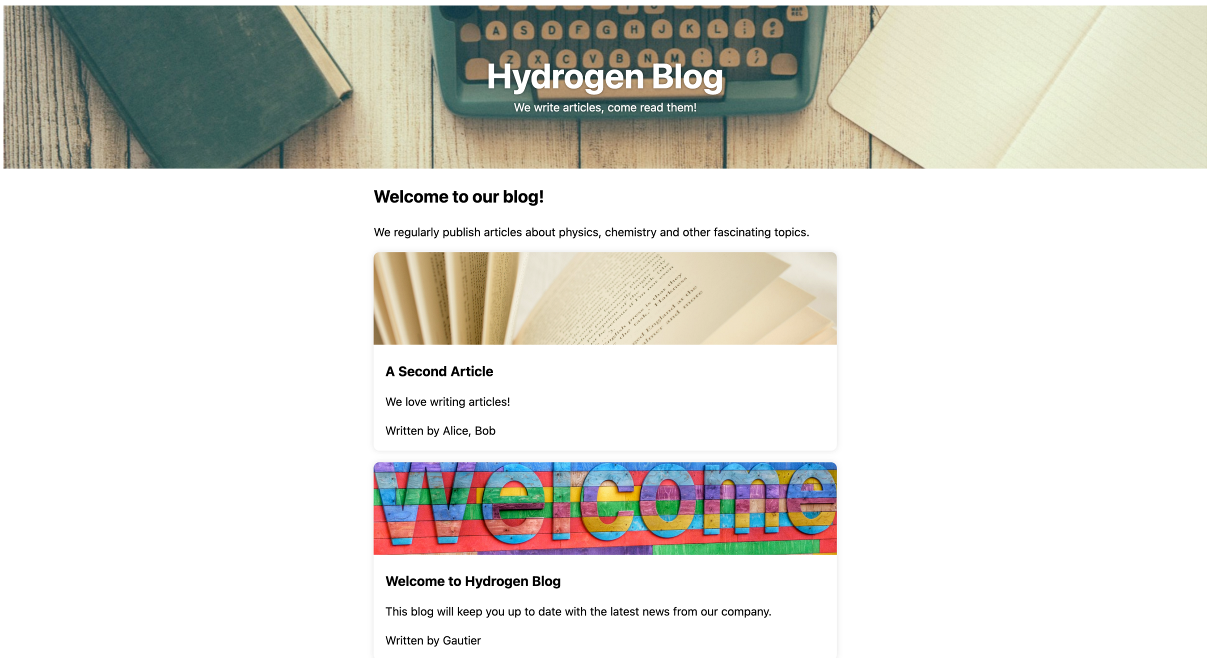
Query: SELECT * FROM [hydrogen:BlogPost] AS post WHERE ISDESCENDANTNODE(post, '/sites/hydrogen/contents/blog')

Type: JCR-SQL2

CREATE ANOTHER   CANCEL   SAVE

You can set *Items to show* to a positive number to limit the number of articles displayed, or -1 to show all of them.

You can also add a rich text above and below the list of articles to introduce and conclude the blog page, and update the footer to reference your Blog page. When done, you should have a page that looks like this:



Hydrogen Blog

We write articles, come read them!

Welcome to our blog!

We regularly publish articles about physics, chemistry and other fascinating topics.

A Second Article

We love writing articles!

Written by Alice, Bob

Welcome to Hydrogen Blog

This blog will keep you up to date with the latest news from our company.

Written by Gautier

Cover image by RetroSupply on Unsplash, article images by Belinda Fewings on Unsplash and Olga Tutunaru on Unsplash

If you click on a blog post, you'll end up on a 404 error page. We'll take care of that in the next section. But for now...

Publication Date

Our JCR query is pretty simple: it lists all blog posts. However, our posts are not ordered and we don't have a way to publish them at a specific date. Let's add a `publicationDate` field to our `BlogPost` type:

```
 - publicationDate (date)
```

It's not marked as `mandatory` because we can't add a mandatory field to an existing type, it would be a breaking change. We will consider it a feature: if the date is not set, the post is a draft.

You also need to update the `types.ts` file:

```
import type { JCRNodeWrapper } from "org.jahia.services.content";

export type Props = {
  "jcr:title": string;
  "subtitle": string;
  "authors"?: string[];
  "cover": JCRNodeWrapper;
  "body": string;
  "publicationDate": string;
};
```

`publicationDate` is a string because it's a date in the ISO 8601 format. As the native `Date` object does not properly handle time zones, it's easier to handle dates as strings if your application requires time zone support.

We can also update our JCR query to order the posts by publication date, and only show published posts:

```
SELECT *
FROM [hydrogen:blogPost] AS post
WHERE ISDESCENDANTNODE(post, '/sites/hydrogen/contents/blog')
  AND post.[publicationDate] IS NOT NULL
ORDER BY post.[publicationDate] DESC
```

As you can see, this query looks like a regular SQL query. The `DESC` keyword orders the posts by publication date in descending order, so the most recent posts are displayed first. Don't forget to update your posts with a publication date!

Finally, it would be nice to display the publication date on the blog post card. Update the `src/components/BlogPost/default.server.tsx` file:

```
// Add `currentResource` to the second argument of the function
<p>
  Written {authors && authors.length > 0 && <>by {authors.join(", ")} </
  {publicationDate && (
    <>
      on{" "}
      {new Date(publicationDate).toLocaleDateString(currentResource.getL
        dateStyle: "long",
      })}
    </>
  )}
</p>
```

We used `currentResource.getLocale().toString()` to get the locale of the current resource. This is useful when you want to format dates or numbers according to the current locale. We forward this locale to `Date.toLocaleDateString` to format the publication date in a human-readable way.

There are a lot of things to discover about JCR queries, as it allows you to query, filter, order and paginate contents in a tree-like structure. You can find more information in the Jahia documentation.

Next: View Content in Full Page