

# Run the Vue sample app

March 21, 2023 • Juraj Uhlár • 6 min read • JavaScript

In this guide, you'll get a brief introduction to website development practices with Kontent.ai. You'll start by running a Vue sample application on your machine and updating an article in the Kontent.ai's Sample Project. You'll also explore how the content of the article is structured.

## New to modular content platforms?

If you are new to the world of modular content platforms or headless CMSs, you might want to start by [building a Hello world application](#).

After you grasp the core idea behind these platforms, everything in the sample application will make a lot more sense much faster.

## Looking for your favorite Javascript framework?

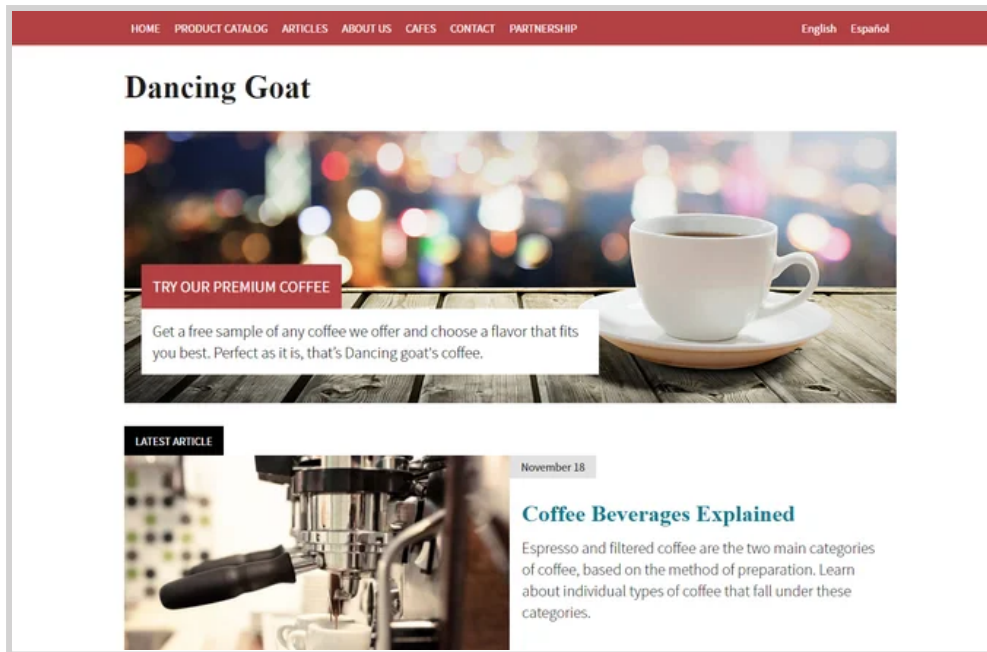
Depending on your preferences, you can choose from the following tutorials:

- [Run the Vue sample app](#)
- [Run the Express sample app](#)
- [Run the React sample app](#)

## Sample project for the app

The sample app displays data from the Sample Project that demonstrates our best practices and features of Kontent.ai. If you don't have it yet, [create your own sample project](#), it only takes a minute.

This full-featured project contains marketing content for Dancing Goat – an imaginary chain of coffee shops. Once you run the sample app, you'll see content from the Sample Project displayed in your browser.



## Running the Vue sample app

Before going any further, make sure you have the following:

- [Node v18](#) or newer installed
- [Kontent.ai](#) account with a Sample Project (you can also set one up during app configuration)
- Your favorite code editor, for example, [Visual Studio Code](#)

### 1. Download the app

The source code of the sample application lives in a [GitHub repository](#).

If you are used to working with [Git](#), you can clone the repository to your computer by typing the following into your console:

shell

```
1 | git clone https://github.com/kontent-ai/sample-app-vue.git
```

If you don't have Git installed, you can directly download the [sample application as a ZIP file](#), extract it, and continue from there.

### 2. Configure the app

Navigate to the root folder of the application in the command line and install the required NPM packages.

shell

```
1 | npm install
```

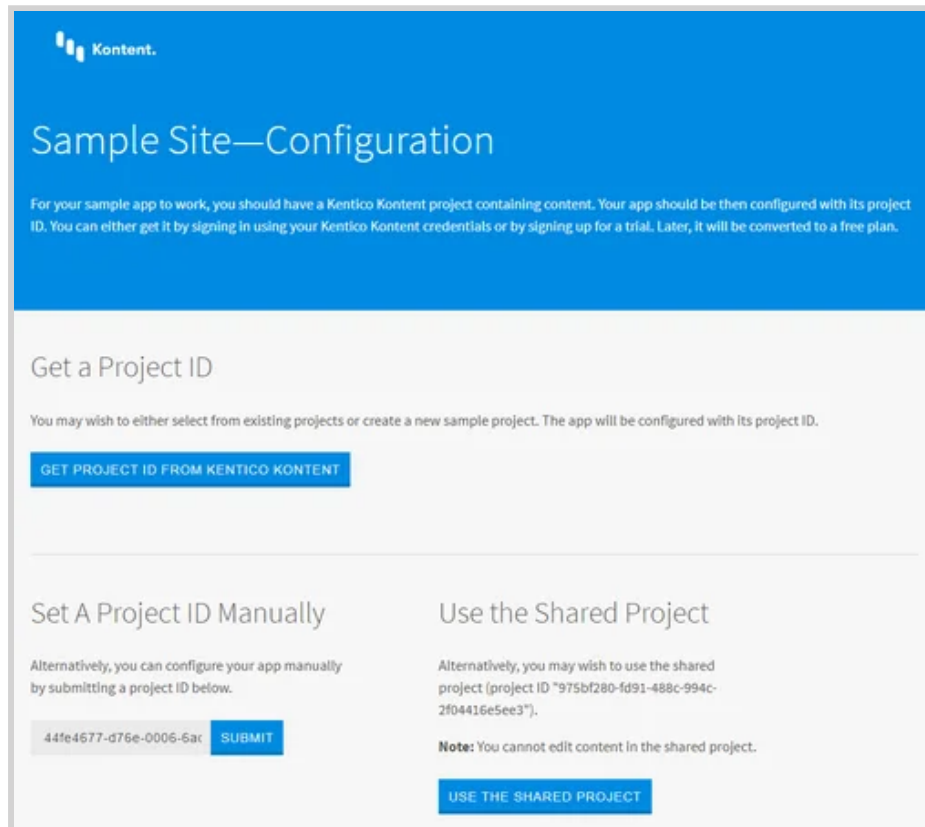
### 3. Run the app

Start a Node development server.

shell

```
1 | npm run dev
```

The application will open automatically in your browser at <http://localhost:8080>.



*First run of the sample app*

### 4. Connect your Kontent.ai project

When you run the application for the first time, you will see a Configuration page. Use it to connect the app to your Sample Project in Kontent.ai

1. Click **Get project ID from Kontent.ai**. A new browser window will open.
2. Sign in to your account or create a new one.
3. From the list of your projects, select the *Sample project*.
4. Click **Select project**.

The application now displays content from your Sample Project.


 You can also [set the project ID manually](#).

Now, let's explore the Sample Project and update its content so that you can see the changes reflected in your locally running application.

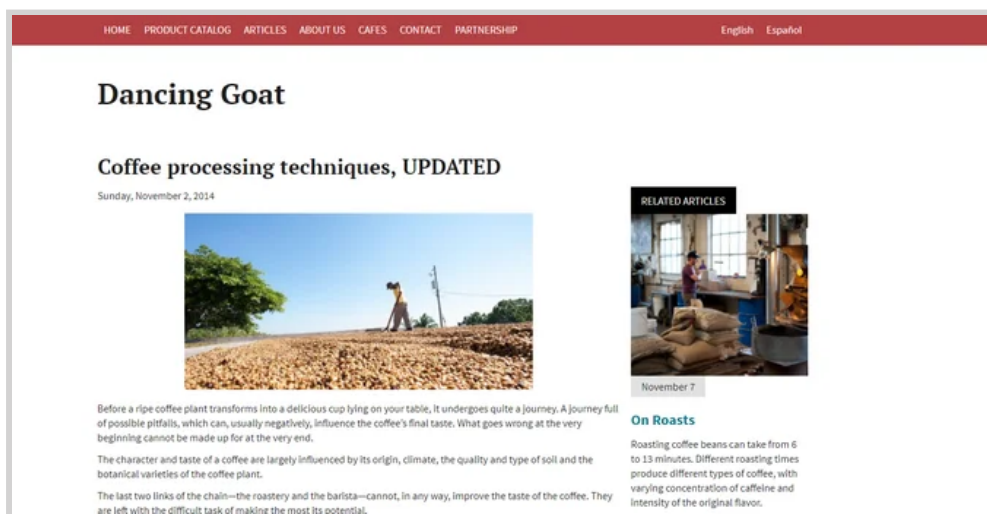
## Making changes to your project

After you sign in to [Kontent.ai](https://kontent.ai), you'll see your Sample Project to play around with. We recommend you [get familiar with content items](#) and how to [find your content](#) in your project. It's simple yet powerful.

Try to edit an article from the sample project:

1. From the app menu, choose  **Content & assets**.
2. Open a published article, for example, the *Coffee processing techniques* one.
  - Look for items with a green *Published* label and the word *Article* in the *Type* column.
3. [Create a new version](#) to edit the article.
4. Make a change in the article, for example, edit the *Title* text or some other element.
5. [Publish](#) the changes to see them in the sample app.


Voila! You have successfully updated content in your Sample Project. Refresh the browser window with your application to see the changes on the website.



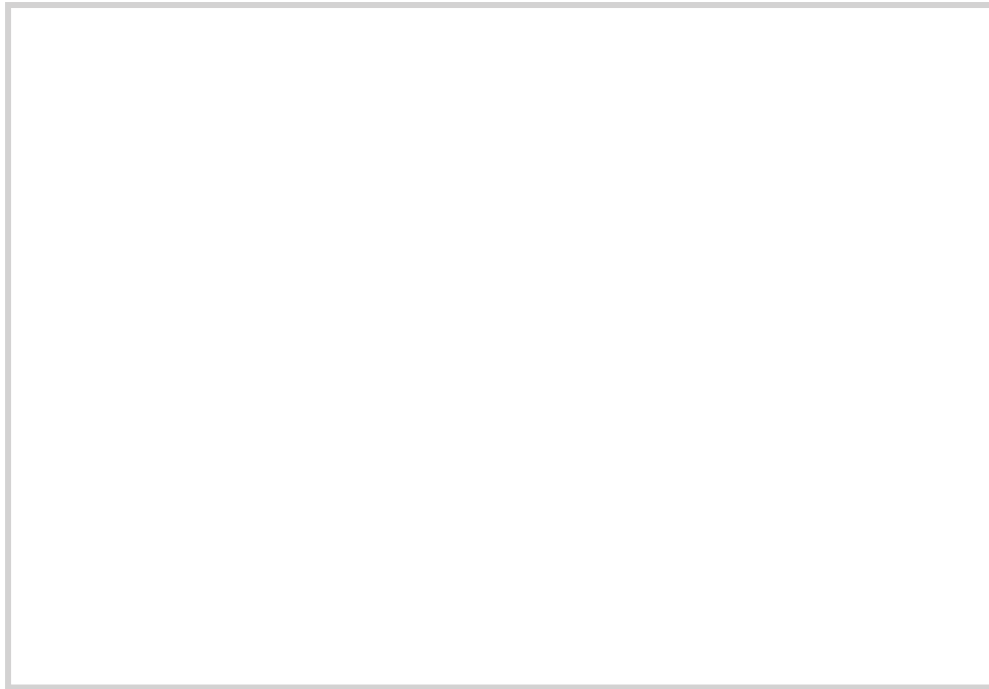
*Looks like someone changed the title of the article.*

Feel free to further explore how the content in your sample project is structured.

## Exploring content structure

To see details about the opened content item, go back to Kontent.ai and click  **Content details** in the top right. Under Content type, notice that the item is based on the *Article* content type.

To open the content type for editing, click *Article* in the content details.



*How the Article content type is modeled*

## How content is structured in Kontent.ai

A **content item** is any piece of content in your project. It is made up of **elements**, such as texts, images, files, etc.

Each content item is based on a single **content type**. A content type is a template that defines the *structure* of the content item – which elements it contains, their names, limitations, guidelines, and so on.



Mix and match the available elements to define your own content types.

## Deploying the Vue sample app

When using a headless CMS like Kontent.ai power a single-page application, you don't necessarily have to write any server-side code. Because the application only requires static resources, you have many interesting deployment options available to you.

Check out platform-specific [deployment guides in Vue.js docs](#).

## Summarized

Kontent.ai makes no assumptions about how your content is displayed. It simply delivers your content via a RESTful [Delivery API](#) to any application that asks for it – be it a website, [mobile app](#), [chatbot](#), electronic billboard, [virtual voice assistant](#), or an internet-enabled toaster.

We deliver the content, you do whatever you want with it – using the technology of your choice.

A traditional CMS would force you to put your content into neatly arranged boxes to fit inside a pre-defined template. The power of Kontent.ai comes from not getting in your way. Content structure and content presentation are truly separated and you have the flexibility to define both according to the needs of your project.

## What's next?

If you'd like to see more than [Vue](#), we also have these JavaScript-based sample apps:

- [Angular](#) – [Angular 9](#) sample app, [Angular Universal Starter](#)
- [Express.js](#) web app with examples of Algolia search integration and automated content translation via webhooks
- [React](#) single page app
- [Svelte](#) web app with Kontent.ai in a JAMStack scenario
- [Vanilla JavaScript](#) sample app

If you want to build your own app, consider using our [Nuxt.js module](#) for Vue.js to create a statically generated app. Lastly, we recommend you look at our [SDKs for JavaScript](#).