

Run the Express sample app

March 28, 2023 • Jan Cerman • 7 min read

In this guide, you'll learn how to run a JavaScript sample app built on Express.js and see your project's content in your browser. The sample app will show content from a sample project, and you'll also get to change something in the project.

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?

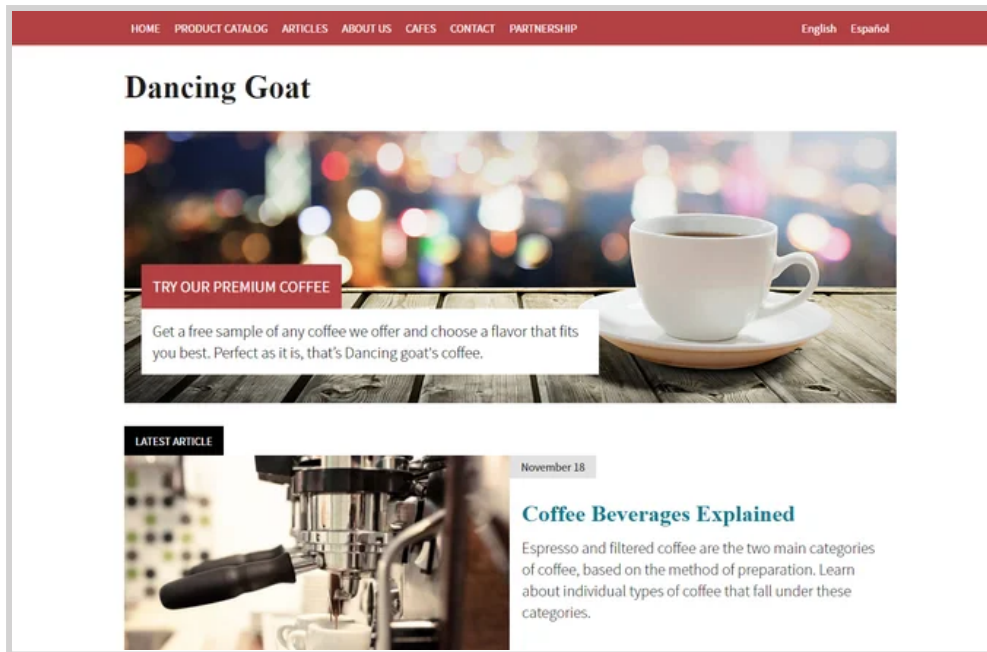
Depending on your preferences, you can choose from the following tutorials:

- [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 JavaScript sample app

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

- [Node and npm](#) installed
- [Kontent.ai](#) account with a [sample project](#)
- 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-express-js.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. Connect to your Kontent.ai project

Before you run the app, you need to tell it which Kontent.ai project it should use.

1. In Kontent.ai >  **Environment settings** > **General** > **Environment ID**, copy the Environment ID for Production.
2. In the sample app's root folder, create a new file named `.env`.
3. Edit the `.env` file.
4. On a new line, insert `projectId=<YOUR_ENVIRONMENT_ID>` where you replace `<YOUR_ENVIRONMENT_ID>` with the ID from step 1.
5. Save the changes.

3. 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
```

4. Run the app

Start a Node development server.

shell

```
1 | npm start
```


The application will open automatically in your browser at <http://localhost:3000> .

Now, let's try updating the project's content so that you can see the changes reflected in your locally running application.

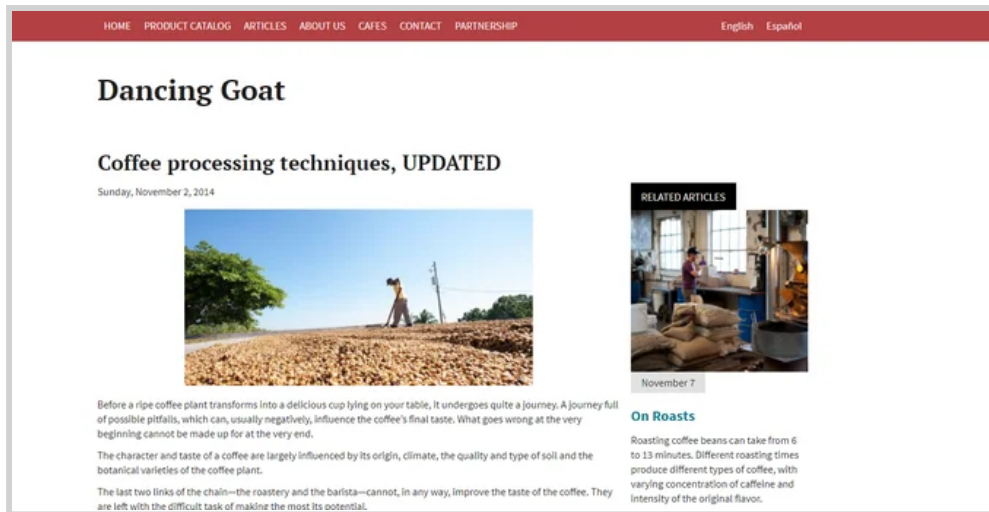
Making changes to your project

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.



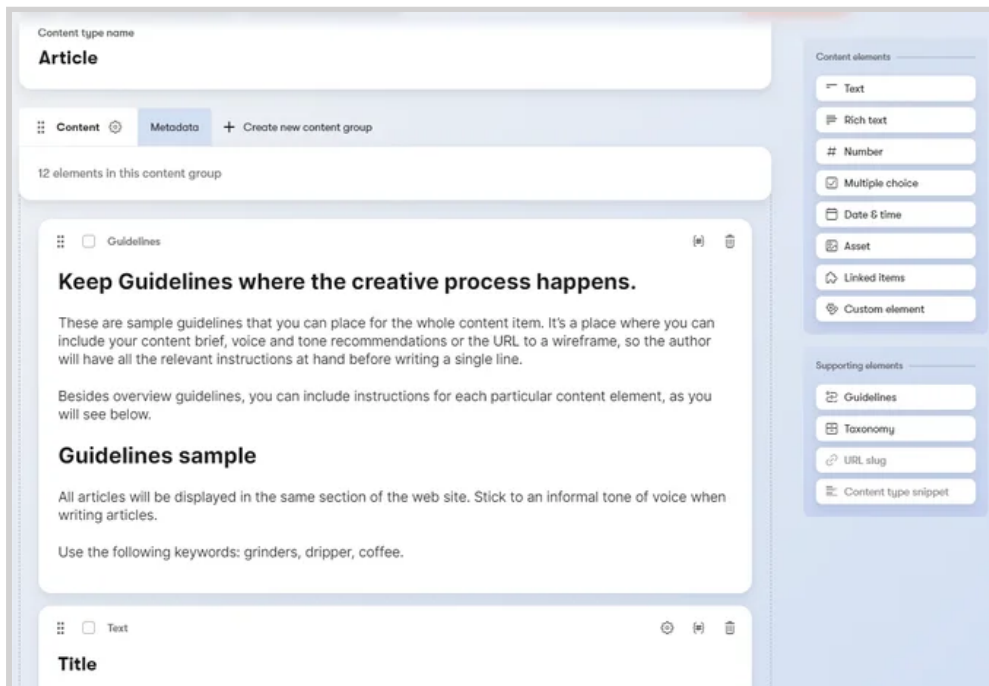
Looks like someone changed the title of the article.

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 **i** **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 JavaScript sample app

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

Here, you're going to use [Surge](#) to deploy your Express.js sample app:

1. Navigate to the root folder of the application in the command line.
2. Type `npm install --global surge` to install Surge on your computer.
3. Type `npm run build` to build the application and make it production-ready.
4. Type `cd build` to navigate to the build folder.
5. Type `surge` to deploy your application. The first time you do this, Surge will prompt you to set up your account. Every other time it will confirm the build directory and generate a funny subdomain for your app.

You can change the subdomain if you like and can even use your own domain, all for free. After that, you're done. The site is deployed and ready to be accessed on the domain specified during deployment.

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 [Express.js](#), we also have these JavaScript-based sample apps:

- [Angular](#) – [Angular 9](#) sample app, [Angular Universal Starter](#)
- [React](#) single page app
- [Svelte](#) web app with Kontent in a JAMStack scenario
- [Vue.js](#) single page app
- [Vanilla JavaScript](#) sample app

Check out our tutorial on [how to build vanilla JavaScript app](#). We also recommend you take a look at our [SDKs for JavaScript](#).