

Develop .NET apps

November 30, 2021 • Jan Cerman • 4 min read • .NET

Learn how to develop your own .NET applications that use Kontent APIs.

SDKs and API references

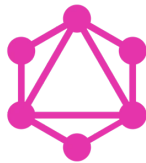
We highly recommend using SDKs to make your app development easier.

Delivery API

Get content from your Kontent project with Delivery API. Check out our [best practices for getting content](#) and examples on [filtering content](#).



Delivery REST API
reference



Delivery GraphQL API
reference



Delivery REST .NET
SDK
(GitHub)

Management API

Manage and [import](#) content into your Kontent projects. With Management API, you can create, edit, delete, publish your content, and more.



Management API
reference

Postman collection

Get to know the Kontent REST APIs with the popular [Postman](#) client. [Install our Postman collection](#) and try any of the API endpoints.

▶ Run in Postman



Learn about changes in Kontent APIs and their availability in the [API reference](#).

Get started tutorials

If you are new to the .NET world, go through these tutorials. They should get you up to speed before you [dig into the sample apps](#).

Sample apps

Do you already know what framework you want to explore? Check one of these sample apps. If you have any questions, ideas for improvement, or you get stuck in code, feel free to submit a GitHub issue.



ASP.NET Core MVC



ASP.NET Core MVC
Boilerplate



.NET Core Razor
Pages



Blazor

Dive deeper

Take your Kontent-powered apps further with the following APIs and features.

- ▶ [Smart link SDK](#) automatically creates [edit links in your web app](#) based on HTML data attributes provided in your app. The SDK also lets you connect your website with [Web Spotlight](#).
- ▶ [Webhooks](#) provide an event-based notification mechanism. They let you know about specific changes in your project as they happen, which is useful for [caching](#).
- ▶ [Image transformation API](#) is a real-time image manipulation and optimization API. Check our [image optimization guide](#) to speed up image delivery in your app.
- ▶ [Custom Elements API](#) lets you implement your own custom elements and use methods from this JavaScript API.

Strongly-typed models

Both the [Delivery](#) and [Management SDK](#) support strongly-typed ([POCO](#)) models based on the content types in your Kontent project.

You can [generate strongly-typed models](#) automatically by using the .NET Generator.

By using strongly-typed models, you get:

- Type safety during compile-time
- Convenient usage in code – `@Article.ArticleTitle` vs. `@Article.GetString("article_title")`
- Support for type-dependent features such as [display templates](#) in MVC



ASP.NET Core extensions

If you're developing an ASP.NET Core web app, the ASP.NET Core extensions package will give you helper classes for easier development. For example, you can use `img-asset` tag helper for more convenient work with responsive images via [Image transformation API](#), and a webhook validator to verify your [webhook's](#) payload integrity.



Jamstack

[Jamstack](#) is an architecture designed to make the web faster, more secure, and easier to scale out of the box.

The first step is to choose a static site generator (SSG). A static site generator combines data from Kontent with specific visual templates, and prerenders a static site, which you can host on a CDN.



Choose your .NET static
site generator

What's next?

- Follow [best practices for getting content](#) to ensure your content and code is scalable and maintainable.
- Explore Kontent's [integration capabilities](#) to build coherent and robust experience for developers and content editors.
- Once you have your Kontent-powered web app running, focus on [SEO optimization](#) and explore the benefits of [Edge computing](#).
- Not sure what's what in Kontent? Check our [glossary of terms](#).