Use strongly-typed models

December 29, 2021 • Jan Cerman • 3 min read • TypeScript

Strongly-typed models are programmatic representations of the content types in your Kontent by Kentico project. They help you map the content retrieved from the Delivery REST API (which comes in the form of JSON objects) directly to a model you can use in your code.

This works a little differently for each platform, so we recommend that you have a look at the documentation of the SDK you are using:

- Using strongly typed models with Delivery .NET SDK
- Using strongly typed models with Delivery JavaScript/TypeScript SDK
- Using strongly typed models with Delivery Java SDK
- Using strongly typed models with Delivery PHP SDK
- Using strongly typed models with Delivery Swift SDK

Life without strongly typed models

When getting content directly from the Delivery REST API, you receive content items as JSON objects. Your app then needs to parse the JSON to display your content. For example, to access the *Headline* element in a content item, you need to use notation like this .

Here is an example of retrieving, parsing, and displaying a content item.

See the code example on https://stackblitz.com/edit/typescript-kk-parse-item-json

In a more complicated scenario, this approach becomes cumbersome. It forces you to remember the JSON structure of your content items and the codenames of your content elements.

To make this easier, use a Delivery SDK to map the retrieved content items to their **strongly typed models**.

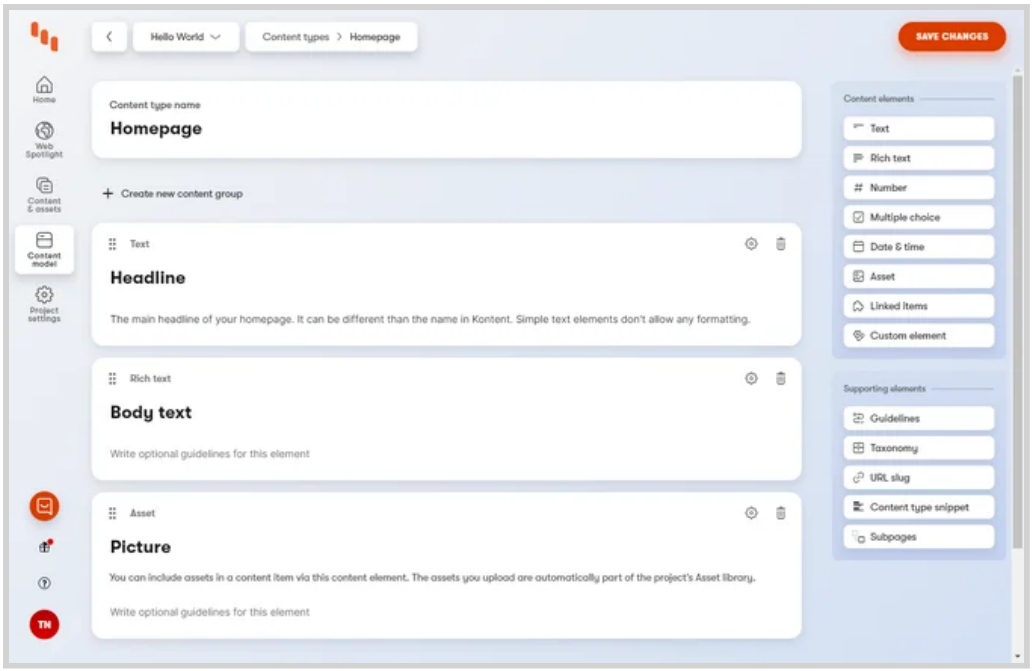
Use strongly typed models

This practice has several advantages:

- type safety during compile-time
- convenience (IntelliSense remembers content type properties for you)
- support of type-dependent functionalities (such as display templates in MVC)

The models are plain classes that don't have any attached behavior or dependency on an external framework. Each model corresponds to a content type in your Kontent project.

Here is the *Homepage* content type used in the previous TypeScript code example.



To create a content type's representation in code, you need a class with properties representing the individual content elements:

- Each property is mapped to its content element's codename either explicitly or using a naming convention. For example, codename becomes property .
- Properties are usually typed using classes provided by the SDK.
- Depending on the SDK, simple content elements (like text and number) are sometimes represented using native types (like string and integer).

The following example is a strongly typed model of the *Homepage* content type.

TypeScript

Generate models

We recommend that you generate the models using the TypeScript/JavaScript model generator.

Use the generator to create strongly typed models from your Kontent project by providing your Project ID.

shell

Get strongly typed content

With your models defined and included in your application, you can use them when retrieving items from Kontent.

TypeScript

A complete example

This code example is equivalent to the first one but maps the retrieved content item to a strongly typed model.

See the code example on https://stackblitz.com/edit/typescript-kk-strongly-typed-model

What's next?

You'll often have to define how your application resolves rich text elements containing assets, links, components, and other content items. For more advanced examples of using strongly typed models, see our sample applications.