Use strongly-typed models

November 23, 2022 • Jan Cerman • 3 min read • JavaScript

Strongly-typed models are programmatic representations of the content types in your Kontent.ai project. They help you map the JSON objects you get from Delivery REST API directly to a model you can use in your code.

The mapping works a little differently for each platform. We recommend that you check the docs of the SDK you're 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

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'd use a notation like this `response.data.item.elements.headline.value`.

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

See the code example on https://stackblitz.com/edit/kontent-ai-learn-display-content-without-models

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

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.ai project.

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 `body_text` becomes property `bodyText`.
- 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.

No code sample for this language yet. Try selecting another language at the top.

Generate models

We recommend that you generate the models using the [TypeScript/JavaScript model generator](https://kontent.ai/learn/tutorials/develop-apps/build-strong-foundation/strongly-typed-models/).

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

shell

```shell
npm i @kontent-ai/model-generator -g

kontent-generate --projectId=8d20758c-d74c-4f59-ae04-ee928c0816b7 --addTimestamp=false --nameResolver=camelCase --sdkType=delivery --apiKey=<YOUR_MANAGEMENT_API_KEY>
```

Get strongly typed content

With your models defined and added to your application, you can use them when retrieving items from Kontent.ai.

No code sample for this language yet. Try selecting another language at the top.

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/kontent-ai-learn-display-content-with-models

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.