

The following section provides examples on how you can retrieve Kentico data in MVC applications. You can work with the following Kentico objects:

- Pages
- Page attachments
- Form records
- Custom table items
- Custom module classes
- Media library files

To retrieve content from page, form, custom table, and custom module class objects, we recommend [generating classes](#) for these objects and using the generated code in your application.

## Retrieving pages

To work with pages, you need to use [generated](#) strongly typed classes for page types.

The following example uses a generated code for the *DancingGoat.Article* page type. You can use the generated providers when retrieving individual pages.

```
// Gets the specified articles using the generated provider
IEnumerable<Article> articles = ArticleProvider.GetArticles()
    .OnSite("MySite")
    .Culture("en-US")
    .Path("/Articles/", PathTypeEnum.Children);

// Gets the published version of a single article using the generated provider
Article article = ArticleProvider.GetArticle(nodeGuid, "en-US", "MySite");
```

By default, the generated code returns published version of pages. If you need to retrieve the latest edited version of a page, use the following approach:

```
// Gets the latest version of a single article using the generated provider
Article article = ArticleProvider.GetArticle(nodeGuid, "en-US", "MySite")
    .LatestVersion()
    .Published(false);
```

To filter the retrieved pages, you can use any of the query parameters for a single page type listed in [Working with pages in the API](#).

## Working with retrieved page data

You can access the data of a retrieved page through the properties of the object. The following types of page data are available:

- Page form data – the data that is editable on a page's *Form* tab.
- General page data – properties available for all pages (via the *TreeNode* class). For example identifiers, values indicating the position in the content tree, etc.
- Metadata – data like the *page title* and *page keywords*.

### Accessing page form data

The fields available on a page's Form tab are specific to its Page type. You can see the fields that each page type uses:

- In *Pages types (application)* -> *edit page type* -> *Fields* tab
- In the corresponding `<namespace>_<page_type_code_name>` database table

We recommend accessing specific fields of a page type via the *Fields* property:

```
// Retrieves a specific page
News news = NewsProvider.GetNews(nodeGuid, "en-US", "MySite");

// Accesses the 'Title' field
string title = news.Fields.Title;

// Accesses the 'Text' field
string text = news.Fields.Text;
```



### Resolving HTML tags and relative URLs

Fields which are populated by the **Rich text editor** form control may contain HTML tags and relative links. To ensure that the content is displayed correctly, use one of the following methods:

- *Html.Raw* method disables the HTML encoding for the values.
- *Html.Kentico().ResolveUrls* extension method disables the HTML encoding for the values and resolves relative URLs to their absolute form.

```
@Html.Raw(Model.<Rich text field content>)
@Html.Kentico().ResolveUrls(Model.<Rich text field content>)
```

## Retrieving page attachments

For information on retrieving page attachments, see [Working with page attachments in MVC applications](#).

## Retrieving form data

The following example requires the classes [generated](#) for the *ContactUs* form included in the solution.

```
// Gets all form items
IEnumerable<ContactUsItem> items = BizFormItemProvider.GetItems<ContactUsItem>();

// Gets a single form item for a given form item ID
ContactUsItem item = BizFormItemProvider.GetItem<ContactUsItem>(1);
```

## Retrieving custom table data

The following example requires the classes [generated](#) for the *SampleTable* custom table included in the solution.

```
// Gets all custom table items
IEnumerable<SampleTableItem> items = CustomTableItemProvider.
GetItems<SampleTableItem>();

// Gets a single custom table item for a given custom table item ID
SampleTableItem item = CustomTableItemProvider.GetItem<SampleTableItem>(1);
```

## Retrieving custom module class data

The following example requires the classes generated for the *Office* custom module class included in the solution.

See the [Creating custom modules](#) page for information on creating custom module classes and generating code to work with them in your application.



```
// Gets all custom module class items
IEnumerable<OfficeInfo> items = OfficeInfoProvider.GetOffices();

// Gets a single custom module class item for a given class ID
OfficeInfo item = OfficeInfoProvider.GetOfficeInfo(1);
```

## Retrieving media library data

For information on retrieving media files from media libraries, see [Working with media libraries on MVC sites](#).