 This page is a part of a tutorial, which you should follow sequentially, from the beginning to the end. [Go to the first page: Getting started with Kentico.](#)

You will learn about:

- [Content only page types in Kentico MVC](#)
- [Creating the page type for the Home page](#)
- [Creating the page type for the Medical center page](#)
- [Creating the page types for the website navigation](#)

In [step 1](#) and [step 2](#) of this tutorial, you have set up your environment for MVC development. Let's start building your tutorial website.

The website will have two pages and a simple navigation menu. The *Home* page will have a *header*, a *text heading*, and the *text* itself. The *Medical center* page will have a *header* text field and a content field that can be formatted using HTML. To allow content editors full control over the menu, you will create a menu item page type with a page selector and another page type representing a content tree section that will separate the menu items from the rest of the pages. The menu item page type will allow editors to hand-pick pages that will display in the website's navigation.

In this tutorial step, you will create page types for the dynamic menu and the pages that will store the content of the MVC site.

Content only page types in Kentico MVC

Content only page types are the most suitable page type for the development of an MVC site. They use fields to define a reusable structure of the page, and content editors use these fields to edit content on their website. Each page type has a separate database table storing the values of the defined fields, and also uses two general tables to store additional page information.

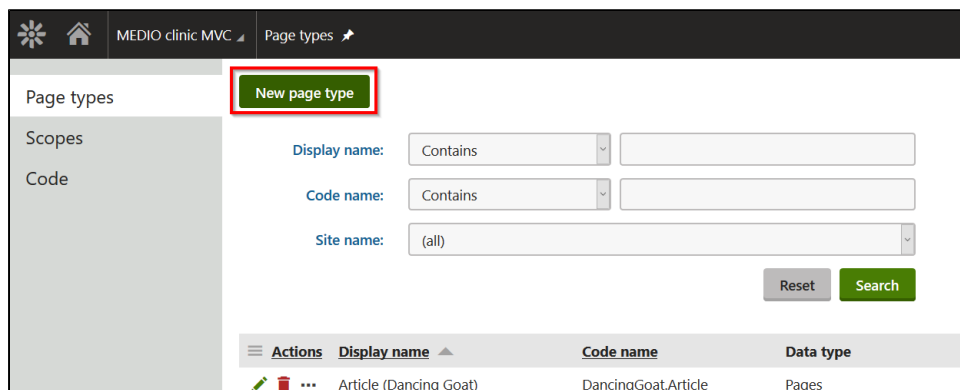
Content only pages are not based on traditional page templates. Their content, e.g., text, HTML hyperlinks, images, etc., is stored and edited via the Kentico administration interface, but the presentation of the content is handled by corresponding MVC views. To help your website editors navigate around their content, you can create a hierarchical structure on the Kentico site. This structure will be just a visual representation, because content only pages do not define actual pages on the live site.



How is the data stored?

For example, your *Home* page type for the tutorial website will have three fields. The first two *Text* fields will hold the page header and the heading, and the third field will be a *Text area*. Any page based on this new page type will have its content stored in the *MEDIO_O_Home* database table. Additional data, such as general page properties, is stored in the *CMS_Document* table, and data related to the page's position in the content tree hierarchy is stored in the *CMS_Tree* table.

Creating the page type for the Home page

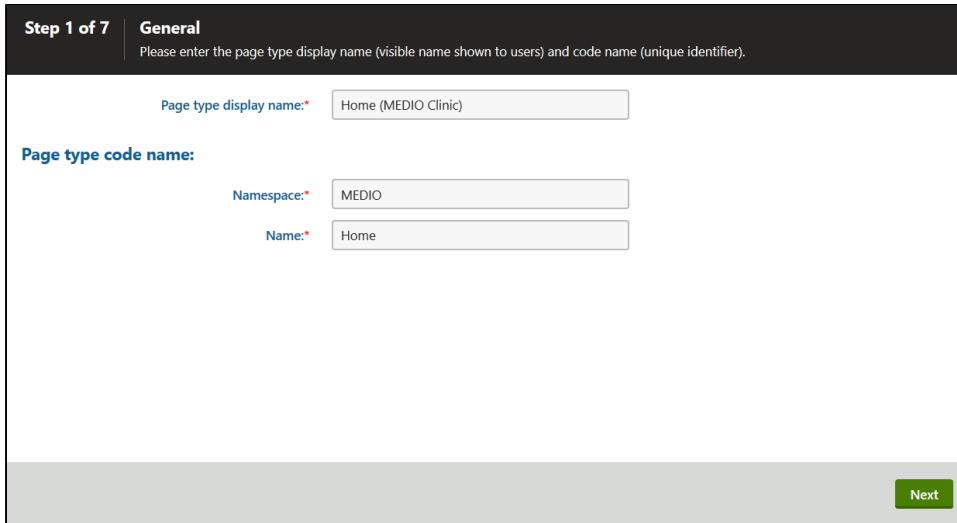
1. In the **Page types** application, click the **New page type** button to start the wizard.



Actions	Display name	Code name	Data type
  ...	Article (Dancing Goat)	DancingGoat.Article	Pages

2. In **Step 1** of the wizard, fill in the following values:

- **Page type display name:** Home (MEDIO Clinic)
- **Namespace:** MEDIO
- **Name:** Home



Step 1 of 7 | General
Please enter the page type display name (visible name shown to users) and code name (unique identifier).

Page type display name:* Home (MEDIO Clinic)

Page type code name:

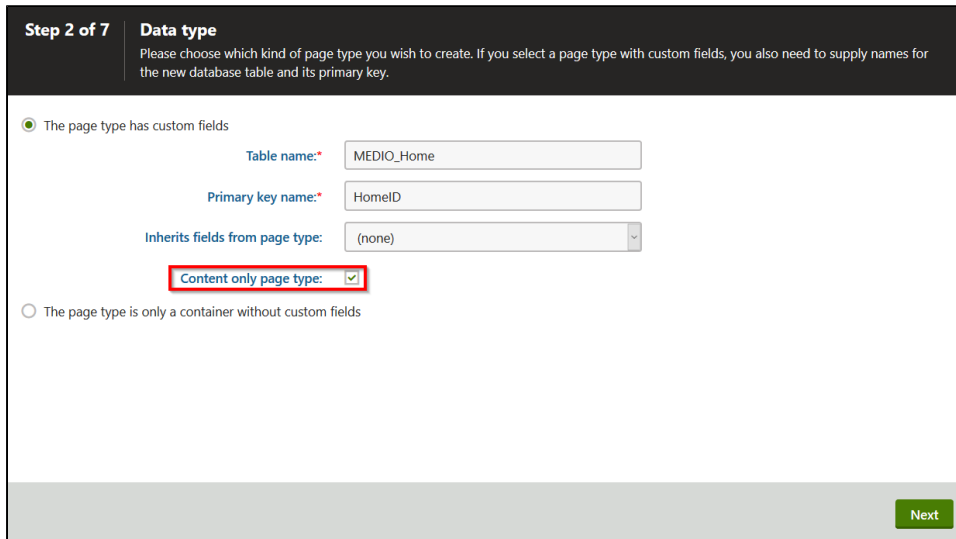
Namespace:* MEDIO

Name:* Home

Next

3. Click **Next**.

4. In **Step 2**, make sure the **Content only page type** option is selected, and leave the other values as they are.



Step 2 of 7 | Data type
Please choose which kind of page type you wish to create. If you select a page type with custom fields, you also need to supply names for the new database table and its primary key.

☒ The page type has custom fields

Table name:* MEDIO_Home

Primary key name:* HomeID

Inherits fields from page type: (none)

Content only page type: ☒

☐ The page type is only a container without custom fields

Next

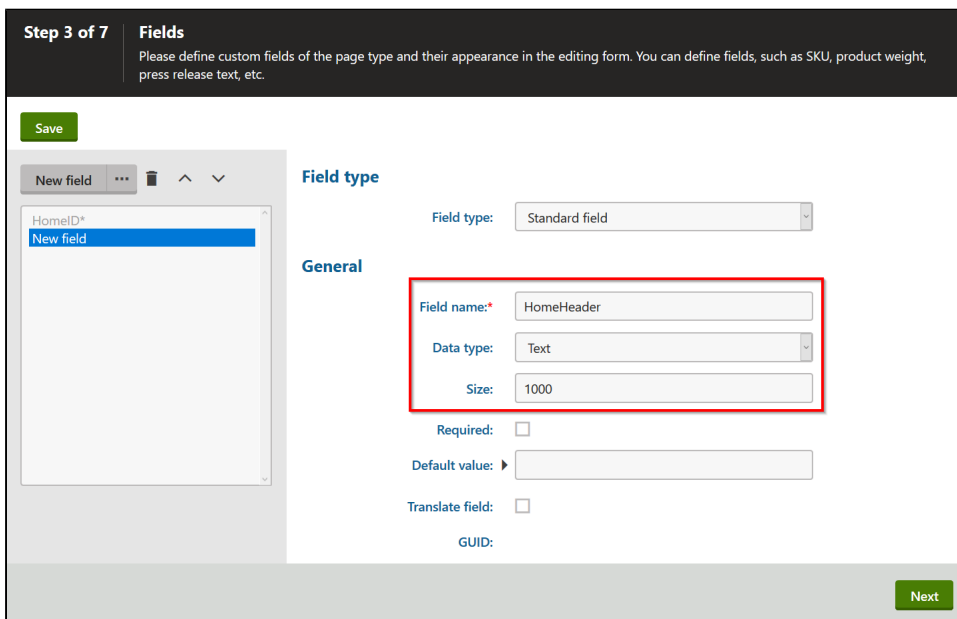
5. Click **Next**.

6. In **Step 3**, click the **New field** button to create a field for the *page header*.

7. Fill in the following values:




- **Field name:** HomeHeader
- **Data type:** Text
- **Size:** 1000
- **Field caption:** Header

- **Form control:** Text area



Step 3 of 7 **Fields**
Please define custom fields of the page type and their appearance in the editing form. You can define fields, such as SKU, product weight, press release text, etc.

Save

New field ...   

HomeID*
New field

Field type

Field type: Standard field


General

Field name*: HomeHeader

Data type: Text

Size: 1000

Required: ☐

Default value: 

Translate field: ☐

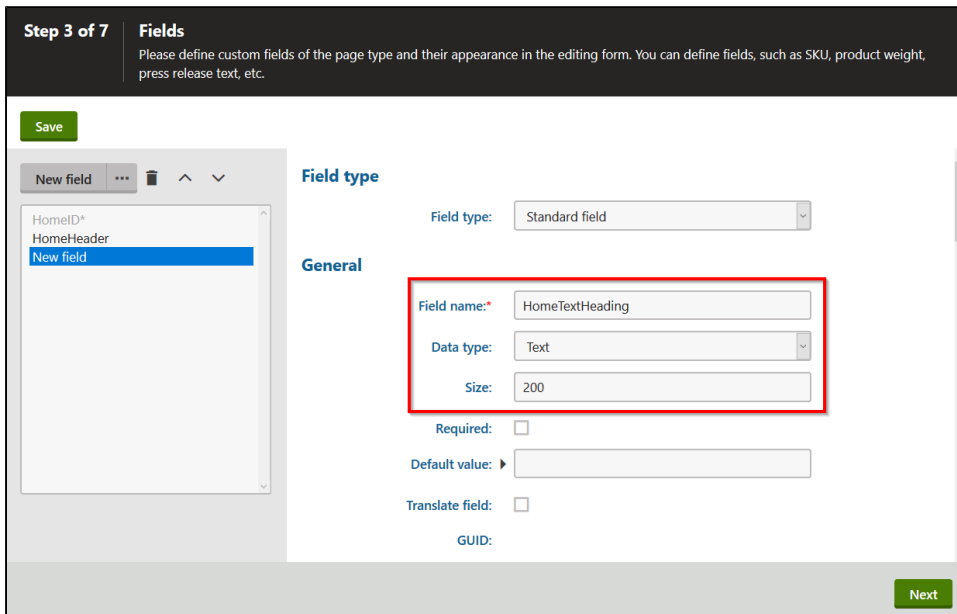
GUID:

Next

8. Click **Save**.




9. Click the **New field** button to create the second field for the *page heading* with the following values:

- **Field name:** HomeTextHeading
- **Data type:** Text
- **Size:** 200
- **Field caption:** Text heading
- **Form control:** Text box



Step 3 of 7 **Fields**
Please define custom fields of the page type and their appearance in the editing form. You can define fields, such as SKU, product weight, press release text, etc.

Save

New field ...   

HomeID*
HomeHeader
New field

Field type

Field type: Standard field


General

Field name*: HomeTextHeading

Data type: Text

Size: 200

Required: ☐

Default value: 

Translate field: ☐

GUID:

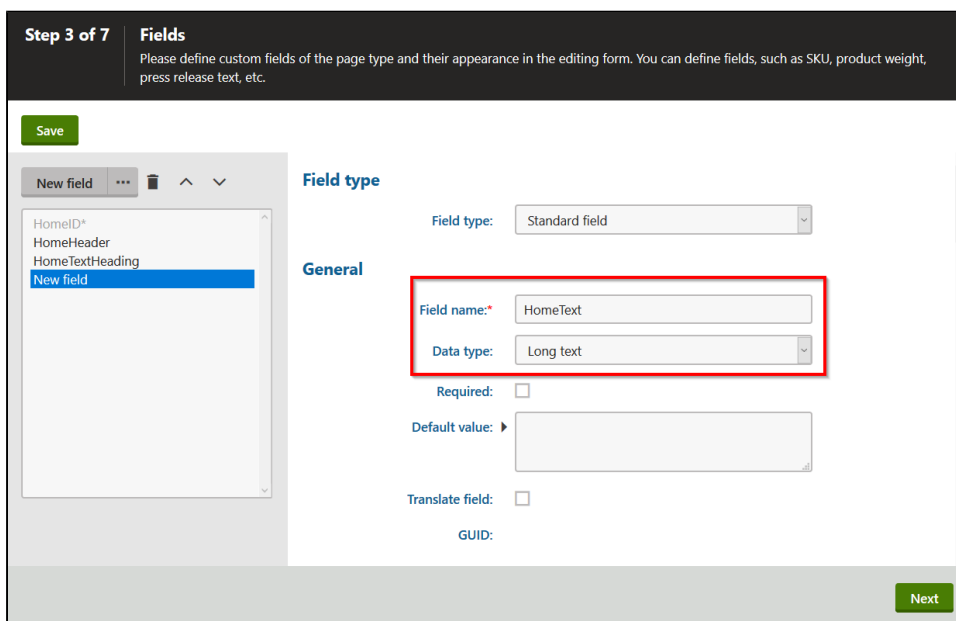
Next

10. Click **Save**.

11. Create the third field for the page text with the following values:

- **Field name:** HomeText
- **Data type:** Long text
- **Field caption:** Text

- **Form control:** Text area



Step 3 of 7 **Fields**
Please define custom fields of the page type and their appearance in the editing form. You can define fields, such as SKU, product weight, press release text, etc.

Save

New field ...

HomeID*
HomeHeader
HomeTextHeading
New field

Field type
Field type: Standard field

General

Field name*: HomeText

Data type: Long text

Required: ☐

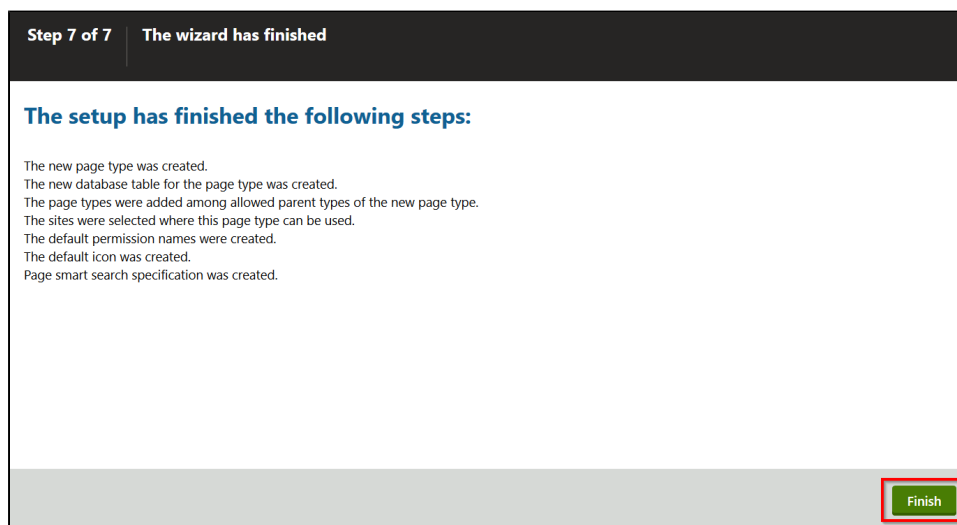
Default value:

Translate field: ☐

GUID:

Next

12. Click **Save**.
13. Click **Next**.
14. In **Step 4**, do not change anything, and click **Next**.
15. In **Step 5**, click **Next**.
16. In **Step 6**, click **Next**.
17. In **Step 7**, click **Finish** to complete the new page type wizard.



Step 7 of 7 **The wizard has finished**

The setup has finished the following steps:

The new page type was created.
The new database table for the page type was created.
The page types were added among allowed parent types of the new page type.
The sites were selected where this page type can be used.
The default permission names were created.
The default icon was created.
Page smart search specification was created.

Finish

Setting the page type's URL pattern

Content only pages just store data, and the presentation of actual web pages is handled by the MVC application. As a result, the Kentico administration application does not automatically know the live URLs under which individual pages will be available.

You can add this information into the system by setting a **URL pattern** for your content only page types.



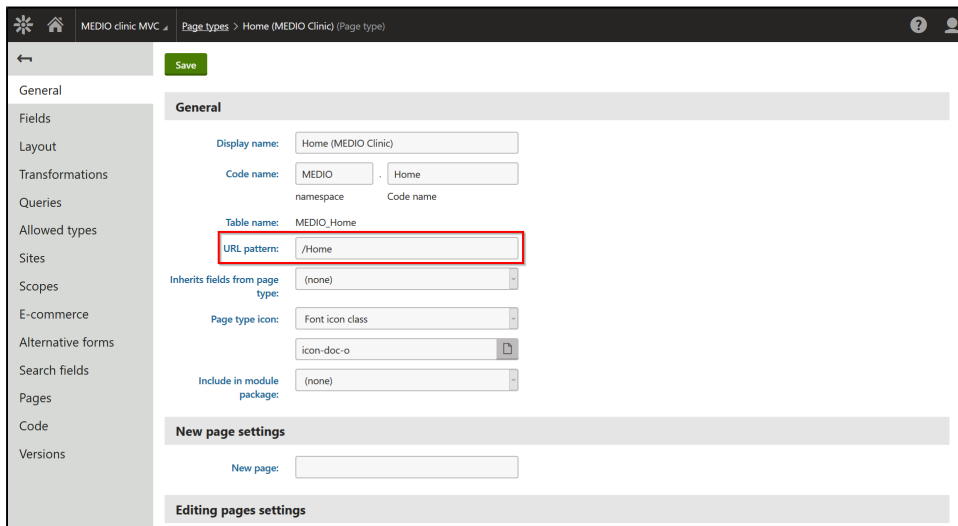
Features that use the URL pattern

By setting the correct URL pattern for your page types, you allow content editors to do the following:

- Preview content in the *Pages* application, and use the *Live site* button in the application list to open the appropriate page on the MVC site
- Add page links into the content of rich text fields using the system's WYSIWYG editor

Additionally, the sample implementation of the navigation menu in this tutorial will use the URL pattern values to generate page links.

1. In the **Page types** application, make sure you are on the **General** tab of your *Home (MEDIO clinic)* page type.
2. Type the following value into the **URL pattern** field: */Home*



The screenshot shows the 'General' tab of the 'Home (MEDIO Clinic)' page type configuration. The 'URL pattern' field is highlighted with a red box and contains the value '/Home'. Other fields include 'Display name' (Home (MEDIO Clinic)), 'Code name' (MEDIO), 'Table name' (MEDIO_Home), 'Inherits fields from page type' (none), 'Page type icon' (icon-doc-o), and 'Include in module package' (none).

3. Click **Save**.

There will only be a single Home page of the *Home (MEDIO clinic)* type, so a static */Home* URL pattern is sufficient. On more complex sites, you may need to add macro variables into the pattern to generate URLs dynamically (e.g. for identifiers when using the same page type for multiple pages, or for different language versions of the same page). The system combines the URL pattern with the site's presentation URL, so the final live URL of the Home page will be: *http://localhost/MedioMVC/Home*

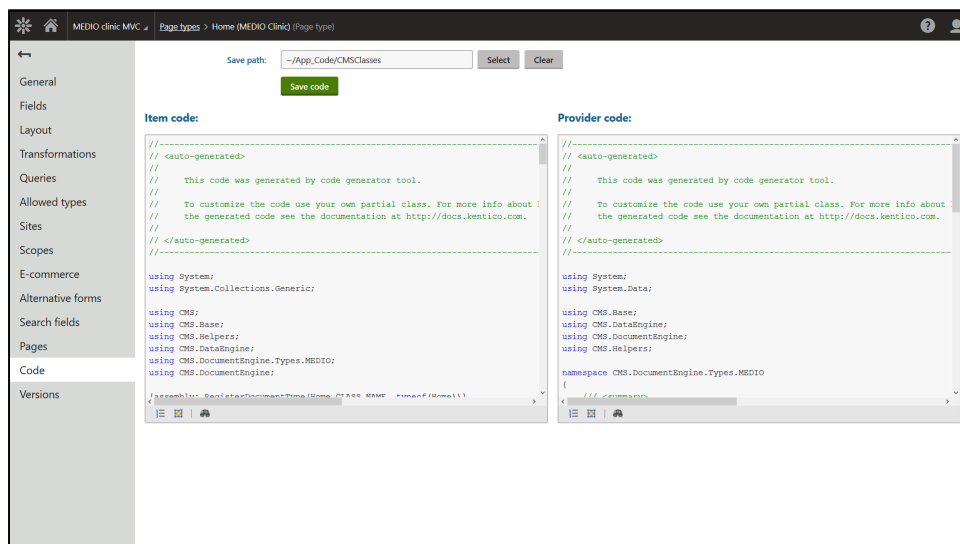
Preparing the page type's wrapper classes

Before you start adding new pages and their content, let's make sure you'll be able to work with this page type in your MVC application. Kentico allows you to generate code for your page type directly from the user interface.

1. To generate code for the *Home (MEDIO Clinic)* page type, switch to the **Code** tab.
2. Verify the **Save path** (by default, the path is set to *~/App_Code/CMSClasses*), and click **Save code**.



If you have installed Kentico as a web application, your default path to save classes will be set to *~/Old_App_Code/CMSClasses*.



The code generator creates two wrapper classes, *Home* and *HomeProvider*. The *Home* class contains properties matching the page type fields you have defined, and the provider allows developers to work with pages of the given type in the MVC site's code.

Later in the tutorial, you will use these two classes to retrieve and display content on the website. But first, let's create all page types you will need to build your tutorial website.

Creating the page type for the Medical center page

1. In the **Page types** application, click the **New page type** button to start the wizard.
2. In **Step 1** of the wizard, fill in the following values:
 - a. **Page type display name:** Medical center (MEDIO Clinic)
 - b. **Namespace:** MEDIO
 - c. **Name:** MedicalCenter
3. In **Step 2**, make sure the **Content only page type** option is selected.
4. In **Step 3**, create two fields with following values:
 - a. The field for the *Medical center* header
 - **Field name:** MedicalCenterHeader
 - **Data type:** Text
 - **Size:** 200
 - **Field caption:** Header
 - b. The field for the *Medical center* text
 - **Field name:** MedicalCenterText
 - **Data type:** Long text
 - **Field caption:** Text
 - **Form control:** Rich text editor
5. Finish the page type wizard.
6. On the **General** tab of the *Medical center (MEDIO Clinic)* page type, enter */MedicalCenter* into the **URL pattern** field.
7. Click **Save**.
8. Switch to the **Code** tab.
9. Verify that the **Save path** is set to *~/App_Code/CMSClasses* and save the code of the page type.

You have created the page types for your pages. Now, let's create the page types you will use to build the navigation menu.

Creating the page types for the website navigation

There are several ways to implement navigation on MVC sites. This tutorial demonstrates how to create a dynamic navigation menu that allows content editors to control the content. Your website's editors will create new menu items as separate "pages" in the content tree, and select which page from the content tree will be the link target for each menu item. You will then code the menu's appearance and linking functionality in the MVC application. This dynamic approach gives editors full control over the navigation menu items, without the need to adjust the MVC site's code.

Creating the page type for the menu section

To keep the content tree organized, you will first create a page type that will serve as a parent for the menu items and separate them from the remaining pages.

1. In the **Page types** application, click the **New page type** button.
2. In **Step 1** of the wizard, fill in the following values:
 - **Page type display name:** Menu items (MEDIO Clinic)
 - **Namespace:** MEDIO
 - **Name:** MenuItem
3. In **Step 2**, make sure the **Content only page type** option is selected.
4. In **Step 3**, create a new required field with the following values:
 - **Field name:** MenuItem
 - **Data type:** Text
 - **Size:** 200
 - **Required:** Yes (selected)
 - **Field caption:** Name
5. In **Step 4**, leave the *MenuItem* field as the **Page name source** and click **Next**.
6. Finish the wizard.
7. On the **General** tab, type / (*forward slash*) into the **URL pattern** field. This ensures that the preview and live URL for the menu items page in the *Pages* application always displays the site's root page.
8. Click **Save**.

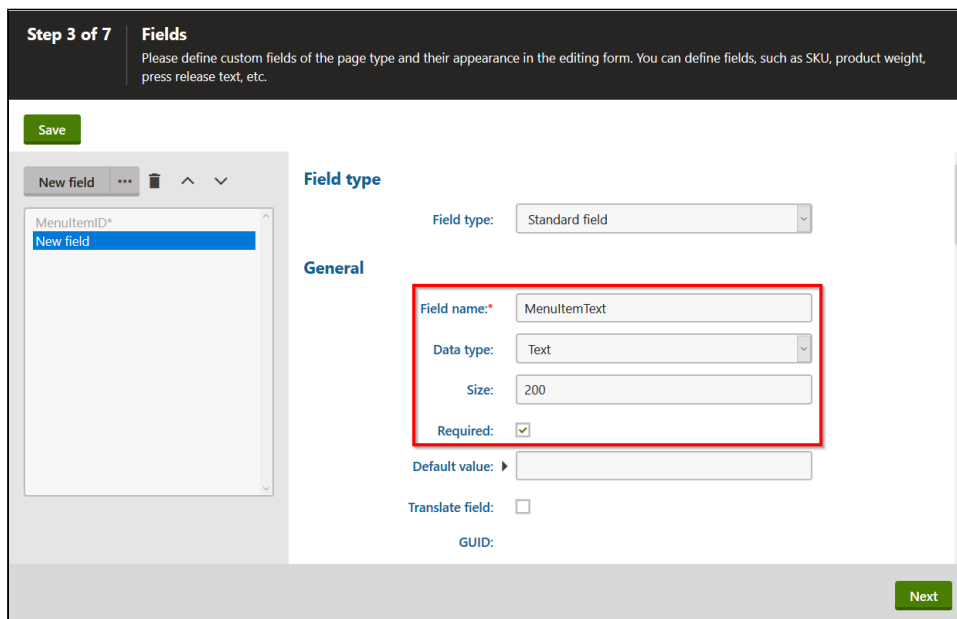
This page type will be used to create a content tree section for menu items, so they don't visually blend in with the rest of the pages. The resulting page will only serve as a visual separator, and you will not work with it in the MVC application.

Creating the page type for menu items

For the individual items in the menu, you need a *text* caption which will be displayed on the live site, and a way to choose where the hyperlink will send visitors. For the latter, you will use a *page selector* which allows editors to hand-pick the items in the content tree. While building this page type, you will also limit the scope so content editors can only add menu items as child pages under the *Menu items* section.

1. In the **Page types** application, click the **New page type** button.
2. In **Step 1** of the wizard, fill in the following values:
 - **Page type display name:** Menu item (MEDIO Clinic)
 - **Namespace:** MEDIO
 - **Name:** MenuItem
3. In **Step 2**, make sure the **Content-only page type** option is selected.
4. In **Step 3**, create fields with the following values:
 - a. The field for the menu item text:
 - **Field name:** MenuItemText
 - **Data type:** Text
 - **Size:** 200
 - **Required:** Yes

- **Field caption:** Menu text



Step 3 of 7 Fields
Please define custom fields of the page type and their appearance in the editing form. You can define fields, such as SKU, product weight, press release text, etc.

Field type
Field type: Standard field

General

Field name: MenuItemText

Data type: Text

Size: 200

Required: ☒

Default value:

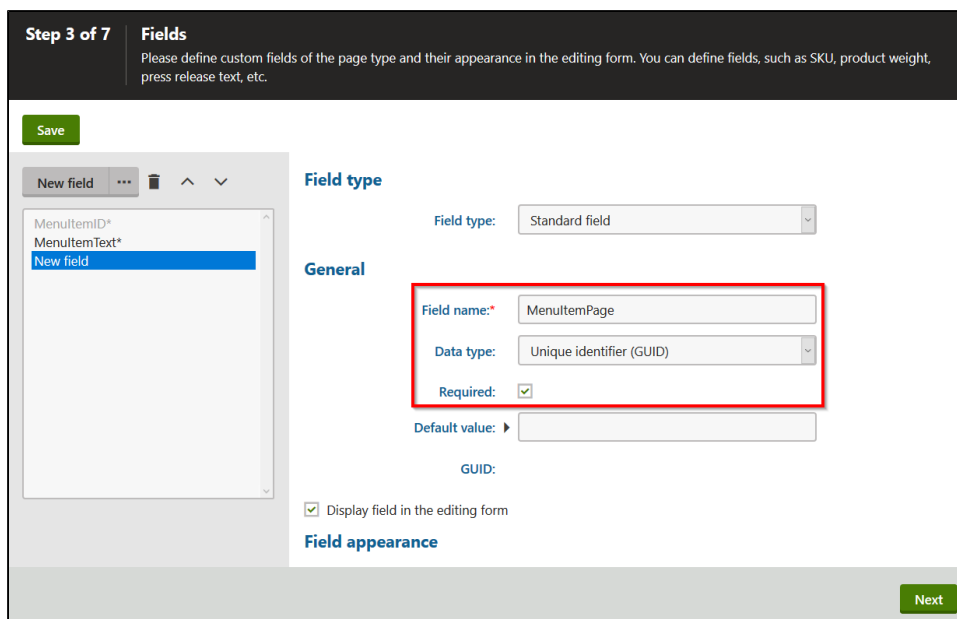
Translate field: ☐

GUID:

Next

b. The field for the page selector:

- **Field name:** MenuItemPage
- **Data type:** Unique identifier (GUID)
- **Required:** Yes
- **Field caption:** Page
- **Form control:** Page selector



Step 3 of 7 Fields
Please define custom fields of the page type and their appearance in the editing form. You can define fields, such as SKU, product weight, press release text, etc.

Field type
Field type: Standard field

General

Field name: MenuItemPage

Data type: Unique identifier (GUID)

Required: ☒

Default value:

GUID:

☒ Display field in the editing form

Field appearance

Next

5. In **Step 5**, update the **Allowed parent page types**:

- Remove **Root (CMS.Root)** as an allowed parent page type – select *Root (CMS.Root)* in the list, click the **Remove selected** button, then click **OK**.



- Click the **Add page types** button and select the **Menu items (MEDIO.MenuItems)** page type as the allowed parent.

Step 5 of 7 | Parent types

Please select page types under which this page template can be placed.

☐ Page type name

☐ Menu items (MEDIO Clinic) (MEDIO.MenuItems)

Items per page 10

Remove selected ... Add page types

Next

- Finish the wizard.
- On the **General** tab, type `/` (forward slash) into the **URL pattern** field. This ensures that the preview and live URL for menu items in the *Pages* application always displays the site's root page.
- Click **Save**.
- Switch to the **Code** tab and save the code for your *Menu item (MEDIO Clinic)* page type.

You have now prepared all the page types you need for your tutorial website. Let's add some content to your *MEDIO Clinic MVC* site.

Previous page: [Creating a new MVC application](#) — **Next page:** [Adding content only pages to the MVC site](#)

Completed pages: 4 of 10

0