



**FREE eBook**

# LEARNING kentico

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#kentico**

# Table of Contents

<b>About</b> .....	<b>1</b>
<b>Chapter 1: Getting started with kentico</b> .....	<b>2</b>
Remarks.....	2
Examples.....	2
Installation or Setup.....	2
<b>Chapter 2: Culture-specific URLs</b> .....	<b>3</b>
Examples.....	3
Configuring culture-specific URLs.....	3
<b>Credits</b> .....	<b>7</b>

---

# About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [kenticio](#)

It is an unofficial and free kenticio ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official kenticio.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Chapter 1: Getting started with kentico

## Remarks

This section provides an overview of what kentico is, and why a developer might want to use it.

It should also mention any large subjects within kentico, and link out to the related topics. Since the Documentation for kentico is new, you may need to create initial versions of those related topics.

## Examples

### Installation or Setup

Detailed instructions on getting kentico set up or installed.

Read **Getting started with kentico** online: <https://riptutorial.com/kentico/topic/1013/getting-started-with-kentico>

# Chapter 2: Culture-specific URLs

## Examples

### Configuring culture-specific URLs

Having culture-specific URLs can be beneficial in terms of SEO.

E.g. English version of the following page:

```
http://www.mydomain.com/insurance
```

Would translate into:

```
http://www.mydomain.nl/verzekering
```

Instead of:

```
http://www.mydomain.nl/nl-nl/insurance
```

There are more approaches of achieving this:

1. Usually, you'll want the URLs to be derived from documents' names. To do that, make sure you set **Use name path for URL path** to true. By default, it's false.

### Page URLs

Default URL path prefix:  ?

**Use name path for URL path:**  ?

Use permanent URLs:  ?

Remember original URLs when moving pages:  ?

Automatically update page alias:  ?

2. If the default URL creation pattern doesn't work for you, you can set the URLs manually as described in the [official documentation](#). However, this option is viable only if you need to adjust small amounts of URLs.
3. If you want to automate creation of URLs based on a custom pattern you can implement a [custom module](#).

```

using System;
using System.Text;

using CMS;
using CMS.DataEngine;
using CMS.DocumentEngine;
using CMS.Helpers;

[assembly: RegisterModule(typeof(CultureSpecificUrlsModule))]
public class CultureSpecificUrlsModule : Module
{
    public CultureSpecificUrlsModule() : base("CultureSpecificUrlsModule")
    {
    }

    protected override void OnInit()
    {
        base.OnInit();
        /**
         * Before the node gets saved, we'll update it's DocumentUrlPath.
         * The system will ensure it'll be saved in a valid URL format.
         */
        DocumentEvents.Update.Before += Update_Before;
    }

    private void Update_Before(object sender, DocumentEventArgs e)
    {
        /**
         * Here you can apply conditions before you actually update the DocumentUrlPath.
         * E.g. you can check for the document's culture.
         */
        UpdateUrlPath(e.Node);
    }

    public static void UpdateUrlPath(TreeNode node)
    {
        /**
         * You can set the DocumentUrlPath to whatever you want.
         * In this example we're using a method extracted from
CMS.DocumentEngine.TreePathUtils.
         * The same method is used to generate a URL for the default culture.
         */
        node.DocumentUrlPath = GetUrlPathFromNamePathInternal(node.DocumentNamePath);
    }

    internal static string GetUrlPathFromNamePathInternal(string namePath, int level = -1)
    {
        // Check valid path
        if (String.IsNullOrEmpty(namePath) || (namePath == "/"))
        {
            return null;
        }

        // For top level the path is always /
        if (level == 0)
        {
            return "/";
        }

        // Ensure maximal level if not set
        if (level < 0)

```

```

    {
        level = Int32.MaxValue;
    }

    // Get the path parts
    string[] pathParts = namePath.Split(new[] { '/' },
StringSplitOptions.RemoveEmptyEntries);
    int currentLevel = 1;
    var path = new StringBuilder();

    foreach (string part in pathParts)
    {
        string shortPart = part;

        // Shorten the part to the allowed maximum
        if (shortPart.Length > TreePathUtils.MaxAliasLength)
        {
            shortPart = shortPart.Substring(0, TreePathUtils.MaxAliasLength);
        }

        path.Append("/", shortPart);

        if (++currentLevel > level)
        {
            break;
        }
    }

    return path.ToString();
}
}

```

4. If you need to update existing pages (e.g. if you've forgotten to check **Use name path for URL path** before you started developing your project) you can use a simple console application that will update the URLs for you:

```

using System;

using CMS.DataEngine;
using CMS.DocumentEngine;

namespace CultureUrlsUtil
{
    class Program
    {
        static void Main(string[] args)
        {
            CMSApplication.Init();

            /*** Here you can narrow down the scope of documents that should be updated using
DocumentQuery ***/
            var pages = DocumentHelper.GetDocuments().Culture("es-es");

            foreach (var page in pages)
            {
                /*** Here we are calling code from the example above. ***/
                CultureSpecificUrlsModule.UpdateUrlPath(page);
                page.Update();
            }
        }
    }
}

```

```
        Console.WriteLine("URLs created!");  
        Console.ReadLine();  
    }  
}  
}
```

Read Culture-specific URLs online: <https://riptutorial.com/kentico/topic/7736/culture-specific-urls>



---

# Credits

S. No	Chapters	Contributors
1	Getting started with kentico	<a href="#">Community</a> , <a href="#">Kristian Bortnik</a>
2	Culture-specific URLs	<a href="#">rocky</a>