

Kentico uses a **rewriting engine** to process URLs. The engine allows the system to use friendly URLs in format:

`http://www.example.com/products/kentico-cms`

instead of something like:

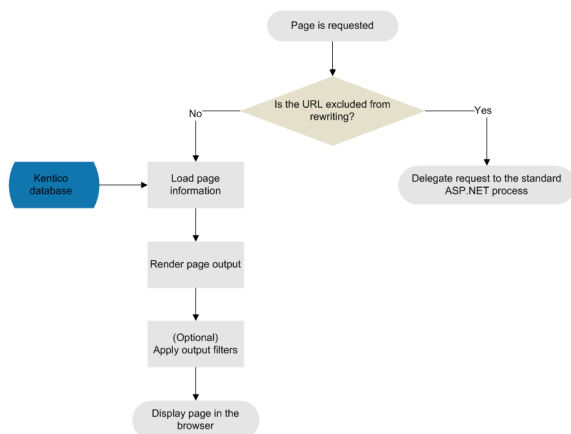
`http://www.example.com/products.aspx?id=527`

The friendly URLs provide several benefits:

- They are easy to remember and easy to write into the browser address bar.
- They are friendly towards search engines (for more information, see the [Search engine optimization](#) chapter).
- They represent a path that shows users where they are located on the website.
- Users can easily bookmark or send the URLs of pages

Every page has its own unique URL. If the website's content is available in [multiple languages](#), individual language versions of pages are recognized based on a combination of their **alias path** and a URL with a [culturespecific format](#), or through completely custom **URL paths** (if specified).

Figure: Basic page processing steps



After pages are processed by the rewriting engine, the system optionally applies [output filters](#) that perform additional changes in the rendered HTML code.

URL processing

The steps below explain how the system handles a request for the following URL: **`http://www.example.com/products/kentico-cms`**

1. Look up the website based on the domain name

The system checks the domain name in the URL and finds a matching site, either via the main site domain name or [domain aliases](#). If none of the currently running websites match the domain name, the `~/cmsmessages/invalidwebsite.aspx` page is displayed. If the requested domain name contains a port number that is not found, the domain is checked without the port number.

2. Process the URL path

The request in this example specifies a standard URL path, so the request is handled by the Kentico URL rewriting engine. The most prioritized way to identify standard pages is the **Page URL path**, so the engine first attempts to find a page with a URL path set to `/products/kentico-cms`.

If there is no page with a matching URL path, the rewriting engine tries to look up a page with an **Alias path** equal to `/products/kentico-cms` (the alias path is generated according to the page's position in the content tree).

3. Check for Page aliases

If a page cannot be found under the given URL or alias path, the system tries to find a page with a **Page alias** set to `/products/kentico-cms`.

See also: [Setting page aliases](#)

4. Handle errors

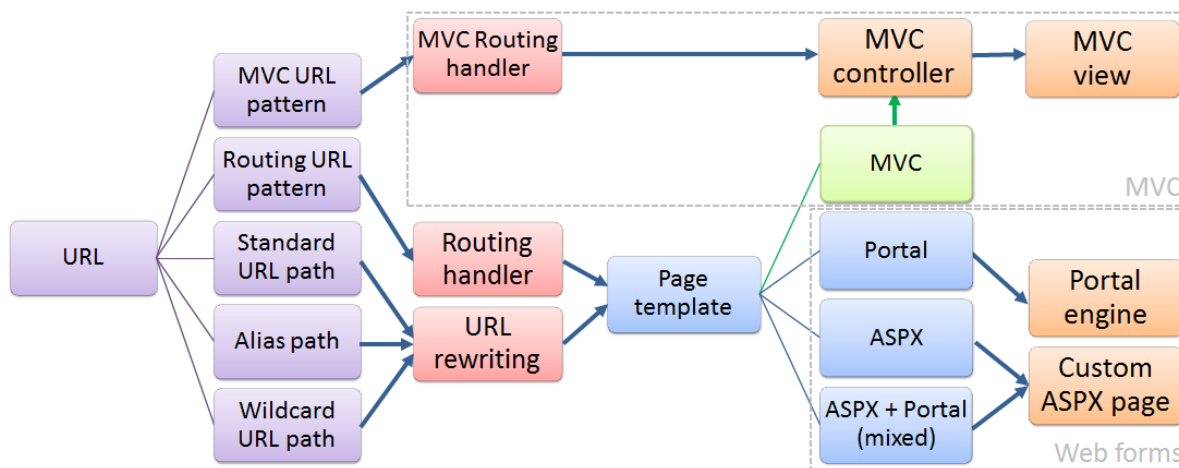
If the steps described above fail to find a matching page for the requested URL, the system does not process the request. The web server returns a 404 HTTP status code and displays the [Page not found error page](#) configured for the website.

5. Display the page

When the requested page is identified, the system displays the content according to the type of the page's template:

- If the page is managed by the [Portal engine](#), the `~/cmspages/portaltemplate.aspx` system page is called, which renders the final output according to the web parts placed on the given template and the content of the specific page.
- For [ASPX page templates](#), the appropriate template (web form) is loaded using an internal URL, for example: `/products.aspx?aliaspath=/products/kentico-cms`
- In the case of MVC templates, the system passes on the request to the MVC controller and action specified for the given template, which then renders the page using an appropriate MVC view.

Figure: Processing for all types of URL requests





Processing of MVC and Routing URL patterns

If the URL of an incoming request matches the MVC or Routing pattern defined for a page, the pattern always has priority over other types of URL processing (even if it is set through a page alias).

The rewriting engine evaluates MVC and Routing patterns in the same step of the process, so precedence between the two is not defined. You should always avoid collisions between MVC and Routing patterns.

Excluding URLs from the rewriting engine

If your website uses custom physical pages (*.aspx*, *.html* or any other file types) stored inside the web project directory, you can avoid unnecessary processing by excluding their URLs from the Kentico rewriting engine. When a visitor requests an excluded URL, the system skips all URL rewriting actions and attempts to access the specified page directly. This leads to better performance and also allows you to prepare your own custom URL rewriting logic.

To disable rewriting for specific pages, open the **Settings** application, select the **URLs and SEO** category, and type the matching URL paths into the **Excluded URLs** setting:

- Use URL paths without the website's domain name or virtual directory.
- The paths must always start with a forward slash (/), without the virtual path designator (~).
- Entering a value excludes all URLs that start with the given path, including sub-directories and all possible extensions.
- You can enter multiple URLs separated by semicolons.

Sample values:

- **/Custom.aspx** - excludes the *~/Custom.aspx* page stored directly under the website's root.
- **/Custom** - excludes all pages whose URL path starts with */Custom*, for example: *~/Custom*, *~/Custom2.aspx*, *~/Custom/Page.htm*
- **/Custom;/Static** - excludes all pages whose URL path starts with */Custom* or */Static*.



Warning: Be careful not to exclude the URLs used by the regular pages in the website's content tree. With URL rewriting disabled for a URL, the system always tries to load a matching physical page, which leads to a page not found error in most cases.