By sending requests to the Kentico REST service, you can get data from the system or manipulate pages and objects.
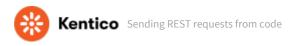
The following example demonstrates how to send REST requests using server-side code. The sample code is incomplete — you need to perform the following according to your requirements:

- integrate the request sending into your custom logic or application
- supply the required values (HTTP method, request URL and data)
- process the response data
- ensure exception handling

C# example

```csharp
using System;
using System.Web;
using System.Net;
using System.IO;
using System.Text;

...

string httpMethod; // "GET", "POST", "PUT" or "DELETE"
string requestUrl; // The URL of the REST request (base URL + resource path)
string requestData; // Data for POST or PUT requests

string responseDescription; // Stores the description of the response status
string responseData; // Stores data retrieved by the request

// Creates the REST request
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(requestUrl);

// Sets the HTTP method of the request
request.Method = httpMethod;

// Authorizes the request using Basic authentication
request.Headers.Add("Authorization: Basic YWRtaW5pc3RyYXRvcjo=");

// Submits data for POST or PUT requests
if (request.Method == "POST" || request.Method == "PUT")
{
        request.ContentType = "text/xml";

        Byte[] bytes = Encoding.GetEncoding("utf-8").GetBytes(requestData);
        request.ContentLength = bytes.Length;

        using (Stream writeStream = request.GetRequestStream())
        {
                writeStream.Write(bytes, 0, bytes.Length);
        }
}

// Gets the REST response
HttpWebResponse response = (HttpWebResponse)request.GetResponse();

// Stores the description of the response status
responseDescription = (Int32)response.StatusCode + " - " + response.StatusDescription;

// Gets the response data
using (Stream responseStream = response.GetResponseStream())
{
        if (responseStream != null)
                using (StreamReader reader = new StreamReader(responseStream))
                {
                        responseData = HttpUtility.HtmlEncode(reader.ReadToEnd());
                }
}

...
```