

To manage objects in Kentico using the REST service, send requests using the POST, PUT or DELETE [HTTP method](#) to the appropriate URL — append the **resource paths** described below to the **base URL** of your REST service.

The base URL of the REST service is **<site domain name>/rest**. For example, if your site is running at *http://localhost/Kentico*, use *http://localhost/Kentico/rest* as the base URL of the service.

Object resource paths start with an **object type** value. To find the value for specific object types, open the **System** application in the Kentico administration interface and select the **Object types** tab.



Important: Do NOT use object requests to create, update or delete the pages of Kentico websites. See [Managing pages using REST](#) for information about working with pages.

Creating objects

HTTP method: **POST**

Resource format:

- **/<object type>** - creates a new object of the specified type.
- **/<object type>/currentsite** - creates a new object of the specified type and assigns it to the website running on the domain in the base URL.
- **/<object type>/site/<site name>** - creates a new object of the specified type and assigns it to the specified website.
- **/customtableitem.<custom table code name>** - creates a new data record inside the specified [custom table](#).
- **/bizformitem.bizform.<form code name>** - creates a new data record inside the specified [form](#).

Set the name and other fields of the new object in the data of the POST request. Both [XML](#) and [JSON](#) formats are supported for the data.



Important:

- The service automatically sets the system fields of the new object (such as the ID and timestamp fields).
- Creating multiple objects of the same type in a single request is not supported.
- You can create child or binding objects along with the primary object. When using the XML format for the data of such requests, enclose the data into the **<data>** element to ensure valid syntax with a single root element.

Examples

Creates a new user with the Editor privilege level and an empty password. Assigns the user to the site running on the domain in the base URL.

XML

JSON



URL: ~/rest/cms.user/currentsite	URL: ~/rest/cms.user/currentsite?format=json
Data:	Data:
<pre><CMS_User> <UserName>Editor< /UserName> <FullName>Content editor</FullName> <Email>editor@localh ost.local</Email> <UserEnabled>true< /UserEnabled> <UserPrivilegeLevel> 1< /UserPrivilegeLevel> </CMS_User></pre>	<pre>{ "UserName": "Editor", "FullName": "Content editor", "Email": "editor@localhost. local", "UserEnabled": true, "UserPrivilegeLevel": 1 }</pre>

Creates a new *country* object with a child state

XML	JSON
URL: ~/rest/cms.country	URL: ~/rest/cms.country?format=json
Data:	Data:
<pre><data> <CMS_Country> <CountryDisplay Name>New country< /CountryDisplay Name> <CountryName>N ewCountry< /CountryName> < /CMS_Country> <CMS_State> <StateDisplayN ame>New state< /StateDisplayN ame> <StateName>New State< /StateName> <StateCode>NS< /StateCode> < /CMS_State> < /data></pre>	<pre>{ "CountryDisplayName": " New country", "CountryName": " NewCountry", "CMS. State": [{ "StateDisplayName": " New state", "StateName": " NewState", "StateCode": "NS" }] }</pre>

Creates a *page alias* for the *Home* page of the sample Corporate site

XML	JSON
-----	------

URL: <code>~/rest/cms.documentalias/site/corporatesite</code> Data: <pre><CMS_DocumentAlias> <AliasNodeID>4</AliasNodeID> <AliasURLPath>/HomeAlias< /AliasURLPath> < /CMS_DocumentAlias></pre>	URL: <code>~/rest/cms.documentalias/site/corporatesite?format=json</code> Data: <pre>{ "AliasNodeID":4, "AliasURLPath": "/HomeAlias" }</pre>
--	--

Adds a new data record into the *Sample table* custom table

XML	JSON
URL: <code>~/rest/customtableitem.customtable.sampletable</code> Data: <pre><customtable_SampleTable> <ItemText>Record added via REST< /ItemText> < /customtable_SampleTable></pre>	URL: <code>~/rest/customtableitem.customtable.sampletable?format=json</code> Data: <pre>{ "ItemText": "Record added via REST" }</pre>

Updating existing objects

HTTP method: **PUT**

Resource format:

- `/<object type>/<id >` - updates the object with the specified identifier (primary key ID).
- `/<object type>/<code name>` - updates the object with the specified code name. For object types with site bindings, always updates the object assigned to the site running on the domain in the base URL.
- `/<object type>/site/<site name>/<code name or guid>` - updates the object with the specified code name or GUID value on the specified website.
- `/<object type>/global/<code name or guid>` - updates the global object with the specified code name or GUID value.
- `/customtableitem.<custom table code name>/<item id or guid>` - updates the data record with the specified identifier (ID or GUID value) inside the given [custom table](#).
- `/bizformitem.bizform.<form code name>/<item id or guid>` - updates the data record with the specified identifier (ID or GUID value) inside the given [form](#).

Update the values of the object's fields using the data of the PUT request. Both [XML](#) and [JSON](#) formats are supported for the data. Updating multiple objects in a single request is not supported.

Examples

Updates the email address of the *administrator* user

XML	JSON
-----	------

URL: <code>~/rest/cms.user/administrator</code> Data: <pre><CMS_User> <Email>newAddress@localhost.local</Email> </CMS_User></pre>	URL: <code>~/rest/cms.user/administrator?format=json</code> Data: <pre>{ "Email": "newAddress@localhost.local" }</pre>
--	--

Updates a data record of the sample *Contact us* form

XML	JSON
URL: <code>~/rest/bizformitem.bizform.contactus/1</code> Data: <pre><Form_ContactUs> <Email>newMail@localhost.local</Email> <Message>Updated message</Message> </Form_ContactUs></pre>	URL: <code>~/rest/bizformitem.bizform.contactus/1?format=json</code> Data: <pre>{ "Email": "newMail@localhost.local", "Message": "Updated message" }</pre>

Deleting objects

HTTP method: **DELETE**

Resource format:

- **/<object type>/<id>** - deletes the object with the specified identifier (primary key ID).
- **/<object type>/<code name>** - deletes the object with the specified code name. For object types with site bindings, always deletes the object assigned to the site running on the domain in the base URL.
- **/<object type>/site/<site name>/<code name or guid>** - deletes the object with the specified code name or GUID value from the specified website.
- **/<object type>/global/<code name or guid>** - deletes the global object with the specified code name or GUID value.
- **/customtableitem.<custom table code name>/<item id or guid>** - deletes the data record with the specified identifier (ID or GUID value) from the given [custom table](#).
- **/bizformitem.bizform.<form code name>/<item id or guid>** - deletes the data record the specified identifier (ID or GUID value) from the given [form](#).

URL examples:

- `~/rest/cms.user/53`
- `~/rest/cms.country/usa`
- `~/rest/cms.emailtemplate/site/corporatesite/Blog.NotificationToModerators`
- `~/rest/customtableitem.customtable.SampleTable/5`

Deleting multiple objects in a single request is not supported. You need to send a separate DELETE request for each object.