Retrieve linked items and subpages

January 24, 2023 • Aaron Collier • 6 min read • Ruby

Sometimes you want to reuse bits of your content in various places. Other times, you want to show how different pieces of content are related, such as who wrote an article. In situations like these, you're going to want to structure your content as separate items and then use linked items to show their connections.

If you're using Web Spotlight to manage your website, your content is organized in a page tree for easy navigation. You can work with the pages the same way as you would with linked items. See how to retrieve subpages of a single page.

How linked items help you

One of the benefits of using an API-first CMS like Kontent.ai is that your content is all structured and ready to be reused. You can define many elements for a given content item and use different ones in different contexts. For example, you might have a full article with catchy copy and captivating images, but only want to use its title and a brief description in a list of articles. When you put your content into various structured items and then link those together to build relationships, you can define what gets used where.

Some benefits from having content in separate items:

- You can have separate workflows for separate pieces of content. For example, you can have a homepage with many testimonials and have some published and some in other steps.
- You can reuse the items, so you only have to edit content once for the changes to appear in many places. For example, you can have an author's bio appear on all the articles they've written.
- You don't need to plan exactly what content will be included in each section – you only need to give the opportunity to add linked items. For example, you can have a product listing where you later decide which products will appear and in which order.

With Kontent.ai, you can link content together in two ways:

1. Using a Linked items element – here, you can limit the items, including what types can be included and how many. This way is covered here.
2. In a Rich text element – here, your items are included within the text, which requires additional work from your app to render them. Read how to deal with structure in your rich text.

This article will look at how to use linked items to add an author with a name and bio to your articles. You might do this if you have an author who is writing multiple articles and wants the chance to edit their bio separately. If you have not yet done so, create at least one author and link it to an article.

Add your relationships to your model

If you're using strongly typed models, you should remember to add your Author model as well as add a linked items element to your Simple Article model.

Retrieve authors with articles

Now, you can retrieve any articles along with their associated authors. When retrieving articles, make sure the `depth` query parameter is higher than 0 (the default value is 1) so that the authors will be returned. Note that any other filtering or order you do to the article will not affect what items are returned within the Linked items element.

The following examples show how to retrieve a single article (named *The Origin of Coffee*) with its author (a linked item).

> If you want to include more than one level of linked items in the response, set the `depth` query parameter to 2 or more.

Ruby

```ruby
# Tip: Find more about Ruby SDKs at https://kontent.ai/learn/ruby
require 'delivery-sdk-ruby'

delivery_client = Kontent::Ai::Delivery::DeliveryClient.new project_id: '8d20758c-d74c-4f59-ae04-ee928c0816b7'
delivery_client.item('the_origin_of_coffee')
               .depth(1)
               .execute do |response|
                 item = response.item
                 puts response.json
               end
```

In the default JSON response, the codename for your author (`jane` in our example) will appear in the `value` of your `author` element. If you had multiple authors, they would appear in the same order in the `author` element as they do in the UI.

Jane's name and bio will appear in the `modular_content` collection under her codename. So to use Jane within your article, all you have to do is match the codename from the `author` element with the values you want to use from the `jane` collection.

JSON

```json
{
  "item": {
    "system": {
      "id": "4735c956-4be4-482c-b874-bba201a22f44",
      "name": "The Origin of Coffee",
      "codename": "the_origin_of_coffee",
      "language": "en-US",
```

```
      "type": "simple_article",
      "sitemap_locations": [],
      "last_modified": "2020-02-03T12:36:02.7767984Z"
    },
    "elements": {
      "title": {
        "type": "text",
        "name": "Title",
        "value": "The Origin of Coffee"
      },
      "author": {
        "type": "modular_content",
        "name": "Author",
        "value": [
          "jane_doe"
        ]
      },
      "body": {
        "type": "rich_text",
        "name": "Body",
        "images": {},
        "links": {},
        "modular_content": [],
        "value": "<p>The history of coffee is patchy and full of myth and hearsay. Although it's far more likely that uses for coffee were developed over time and were discovered by people tasting various parts of the cherry, the old fables do add a bit of romance and are very cute.</p>"
      }
    }
  },
  "modular_content": {
    "jane_doe": {
      "system": {
        "id": "3bf1c09b-4552-4e2c-8de5-ffce79f773e0",
        "name": "Jane Doe",
        "codename": "jane_doe",
        "language": "en-US",
        "type": "author",
        "sitemap_locations": [],
        "last_modified": "2020-02-03T12:30:14.641708Z"
      },
      "elements": {
        "name": {
          "type": "text",
          "name": "Name",
          "value": "Jane Doe"
        },
        "bio": {
          "type": "rich_text",
          "name": "Bio",
```

```
            "images": {},
            "links": {},
            "modular_content": [],
            "value": "<p>Jane is a fun-loving writer who enjoys her coffee black as hell,
strong as death, and sweet as love.</p>"
          }
        }
      }
    }
  }
}
```

Retrieve subpages

When using Web Spotlight to manage your website content, you can retrieve subpages the same way as your linked items.

The following example shows how to retrieve a single page (Insurance listing) with its subpages (travel, car, and health insurance). Both the *Insurance listing* page and its subpages are of the *Page* type.

> If you're using strongly typed models, you should remember to add a subpages element to your Page model.

When retrieving pages, make sure the `depth` query parameter is higher than 0 (the default value is 1) so that the subpages will be returned.

Ruby

```ruby
# Tip: Find more about Ruby SDKs at https://kontent.ai/learn/ruby
require 'delivery-sdk-ruby'

delivery_client = Kontent::Ai::Delivery::DeliveryClient.new project_id: '8d20758c-d74c-
4f59-ae04-ee928c0816b7'
delivery_client.item('insurance_listing')
               .depth(1)
               .execute do |response|
                 item = response.item
                 puts response.json
               end
```

In the default JSON response, the codenames for the types of insurance (`travel_insurance`, for example) will appear in the `value` of the `navigation_links` element. They appear in the same order in the `navigation_links` element as they do in the UI.

You can find the details about each type of insurance page in the `modular_content` collection.

JSON

```
{
  "item": {
```

```
    "system": {
      "id": "47e1a69d-9205-4be4-9ce9-78042dd4560b",
      "name": "Insurance listing",
      "codename": "insurance_listing",
      "language": "default",
      "type": "page",
      "sitemap_locations": [],
      "last_modified": "2020-09-02T12:24:44.4297022Z"
    },
    "elements": {
      "title": {
        "type": "text",
        "name": "Title",
        "value": "Insurance listing"
      },
      "url": {
        "type": "url_slug",
        "name": "URL",
        "value": "insurance-listing"
      },
      "show_in_navigation": {
        "type": "multiple_choice",
        "name": "Show in navigation",
        "value": [
          {
            "name": "Yes",
            "codename": "yes"
          }
        ]
      },
      "navigation_links": {
        "type": "modular_content",
        "name": "Subpages",
        "value": [
          "travel_insurance",
          "car_insurance",
          "health_insurance"
        ]
      },
      "target_content": {
        "type": "modular_content",
        "name": "Content",
        "value": [
          "insurance_landing_page"
        ]
      }
    }
  },
  "modular_content": {
    "car_insurance": {
```

```
      "system": {
        "id": "242d3030-e21f-488b-965f-0b170dcb144c",
        "name": "Car insurance",
        "codename": "car_insurance",
        "language": "default",
        "type": "page",
        "sitemap_locations": [],
        "last_modified": "2020-09-02T12:26:18.1255074Z"
      },
      "elements": {
        "title": {
          "type": "text",
          "name": "Title",
          "value": "Car insurance"
        },
        "url": {
          "type": "url_slug",
          "name": "URL",
          "value": "car-insurance"
        },
        "show_in_navigation": {
          "type": "multiple_choice",
          "name": "Show in navigation",
          "value": [
            {
              "name": "Yes",
              "codename": "yes"
            }
          ]
        },
        "navigation_links": {
          "type": "modular_content",
          "name": "Subpages",
          "value": []
        },
        "target_content": {
          "type": "modular_content",
          "name": "Content",
          "value": []
        }
      }
    },
    "health_insurance": {
      "system": {
        "id": "522624cb-afe3-4f60-b1b0-72a8b1c9ea03",
        "name": "Health insurance",
        "codename": "health_insurance",
        "language": "default",
        "type": "page",
        "sitemap_locations": [],
```

```
        "last_modified": "2020-09-02T12:26:37.1424023Z"
      },
      "elements": {
        "title": {
          "type": "text",
          "name": "Title",
          "value": "Health insurance"
        },
        "url": {
          "type": "url_slug",
          "name": "URL",
          "value": "health-insurance"
        },
        "show_in_navigation": {
          "type": "multiple_choice",
          "name": "Show in navigation",
          "value": [
            {
              "name": "Yes",
              "codename": "yes"
            }
          ]
        },
        "navigation_links": {
          "type": "modular_content",
          "name": "Subpages",
          "value": []
        },
        "target_content": {
          "type": "modular_content",
          "name": "Content",
          "value": []
        }
      }
    },
    "insurance_landing_page": {
      "system": {
        "id": "22f67998-5042-4456-947a-81ba7eed0467",
        "name": "Insurance landing page",
        "codename": "insurance_landing_page",
        "language": "default",
        "type": "article",
        "sitemap_locations": [],
        "last_modified": "2020-09-02T12:28:44.4123144Z"
      },
      "elements": {
        "title": {
          "type": "text",
          "name": "Title",
          "value": "Pick your insurance today!"
```

```
        },
        "body": {
          "type": "rich_text",
          "name": "Body",
          "images": {},
          "links": {},
          "modular_content": [],
          "value": "<p>If you’re thinking about taking out insurance, you’ve probably asked yourself, ‘Why is it so complicated?!” It can be trying to pick a policy that suits your lifestyle, but the search will be worth it if you go about things the right way.</p>"
        }
      }
    },
    "travel_insurance": {
      "system": {
        "id": "4b43567d-c0de-463e-94cc-62b51fe4023f",
        "name": "Travel insurance",
        "codename": "travel_insurance",
        "language": "default",
        "type": "page",
        "sitemap_locations": [],
        "last_modified": "2020-09-02T12:26:01.7874635Z"
      },
      "elements": {
        "title": {
          "type": "text",
          "name": "Title",
          "value": "Travel insurance"
        },
        "url": {
          "type": "url_slug",
          "name": "URL",
          "value": "travel-insurance"
        },
        "show_in_navigation": {
          "type": "multiple_choice",
          "name": "Show in navigation",
          "value": [
            {
              "name": "Yes",
              "codename": "yes"
            }
          ]
        },
        "navigation_links": {
          "type": "modular_content",
          "name": "Subpages",
          "value": []
        },
```

```
        "target_content": {
          "type": "modular_content",
          "name": "Content",
          "value": []
        }
      }
    }
  }
}
```

What's next?

You've seen how to link content together through the example of adding an author to your articles. First, you model and create content in Kontent.ai. Then, you adjust your models in your app and retrieve your articles with the new relationship. You can take a similar approach in linking other content, such as adding related articles.

- Use your linked content items to create hierarchical URLs.
- See how to add structure to your rich text.
- Read more about linked items in our API reference.
- Read about getting content and see how to filter your articles.
- See how to manage your website by using Web Spotlight.
- Create a sitemap for your website by linking content items.