

Get content items

March 28, 2023 • Jan Cerman • 4 min read • JavaScript

While your content creators draft articles and add finishing touches to the content in your project, you can deliver that content to web and mobile applications using Delivery API.

Delivery API is a read-only API available as both REST API and [GraphQL API](#). This tutorial covers the basics of using the [Delivery REST API](#).

The Delivery API can work in two modes: production and preview. In production mode, you get published content that is publicly available. In [preview](#) mode, you get the latest version of your content, be it published or unpublished.



Let's find out what you need to retrieve a list of published content items like articles from your project.

Get content items

To retrieve content items from a project, you first need to specify the project's environment using its unique ID.

1. Find your environment ID

You use the environment ID to tell Delivery API where to look for content. For example, an environment ID might look like this: `8d20758c-d74c-4f59-ae04-ee928c0816b`.

1. In Kontent.ai, use the two drop-downs at the top to select a project and its environment.
2. In  **Environment settings** > **General**, click  for the **Environment ID**.

With the environment ID, you can now make queries to Delivery API.

2. Make a request

Let's see how to get all content items from the specified environment in JSON format.

JavaScript

```
1 // Tip: Find more about JS/TS SDKs at https://kontent.ai/learn/javascript
2 const KontentDelivery = require('@kontent-ai/delivery-sdk');
3
4 const deliveryClient = KontentDelivery.createDeliveryClient({
5   environmentId: '8d20758c-d74c-4f59-ae04-ee928c0816b7'
6 });
7
8 const response = await deliveryClient.items()
9   .toPromise();
```

Recently published content items may appear in the API after a slight delay.

💡 Paging the results

If you don't need all content items at once, you can play with the paging by specifying the `limit` and `skip` query parameters.

For example, calling the `/items` endpoint with the `limit=3&skip=6` query parameters sets the page size to 3 and gives you the third page.

Filter content items

Now that you retrieved all content items, you need to apply filters to get only a few specific content items. In this example, you'll retrieve items based on the *Article* content type.

1. Find codenames

Before moving further, you need to find the **codename** of the content type.

📘 Quick facts about codenames

Codenames are alphanumeric identifiers of objects in Kontent.ai. Codenames are initially generated from the object's name, like content item name, when saved for the first time.

You can copy codenames by clicking `{#}` near the name of a content type, content element, or other objects in your project.

For example, to find the codename of a content type named *Article*, go to `Content model > Content types > Article > {#}`.



Example: Displaying the codename of the Article content type.

Once you have the codename (in this case, `article`) you can use it to filter the requested content items by their type.

2. Filter by codenames

The information about a content item's type is stored in the content item's *System* object, specifically, in its `type` property. The *System* object contains metadata about the content item, such as the last content modification date, language, collection, and more.

JSON

```
1 | "system": {
2 |   "id": "31f8470f-8a94-438a-8a47-f4cdb9c90ada",
```

```

3 |   "collection": "default",
4 |   "name": "Why structured writing needs structured content",
5 |   "codename": "structured_writing",
6 |   "language": "en-US",
7 |   "type": "article",
8 |   "sitemap_locations": [],
9 |   "last_modified": "2020-01-27T13:43:47.134249Z",
10 |   "workflow_step": "published"
11 | }

```

To filter the content items by type, you need to compare the value in the `type` property to `article` using the following notation: `system.type=article`. Any content items that are not based on the *Article* content type will be omitted from the response.

JavaScript

```

1 | // Tip: Find more about JS/TS SDKs at https://kontent.ai/learn/javascript
   | const KontentDelivery = require('@kontent-ai/delivery-sdk');
2 |
3 | const deliveryClient = KontentDelivery.createDeliveryClient({
4 |   environmentId: '8d20758c-d74c-4f59-ae04-ee928c0816b7'
5 | });
6 |
7 | const response = await deliveryClient.items()
8 |   .type('article')
9 |   .toPromise();

```

The value comparison is done using the equals operator.

 **Tip:** Check out more [examples of filtering content](#) with Delivery API.

Order content items

With content like articles, you usually want to retrieve and display them in a certain order and get, for example, only the three latest articles.

You can specify the order by using the `order` query parameter. The value of the `order` query parameter must be in the following format: `<PropertyToOrderBy>[<asc|desc>]`. Where the `PropertyToOrderBy` value specifies either a System property (such as `system.type`) or a content element within a content item (such as `elements.title`).

The Delivery API sorts content items alphabetically by their codenames by default. If you don't specify the order when retrieving content, it is the equivalent of adding `order=system.codename[asc]` to your query.

To get the three latest articles, you need to provide the following query parameters:

- `system.type=article` – specifies the content type of the content items.
- `limit=3` – sets the number of content items to return (sometimes also referred to as page size).

- `order=system.last_modified[desc]` – sorts the content items by last modification date in descending order.

JavaScript

```
1 // Tip: Find more about JS/TS SDKs at https://kontent.ai/learn/javascript
2 const KontentDelivery = require('@kontent-ai/delivery-sdk');
3
4 const deliveryClient = KontentDelivery.createDeliveryClient({
5   environmentId: '8d20758c-d74c-4f59-ae04-ee928c0816b7'
6 });
7
8 const response = await deliveryClient.items()
9   .type('article')
10  .limitParameter(3)
11  .orderParameter('system.last_modified', KontentDelivery.SortOrder.desc)
12  .toPromise();
```

What's next?

You've learned how to get specific content items from your Kontent.ai project with filtering and sorting. Besides fetching content items, you can also use the [Delivery REST API](#) to get content types, elements, and taxonomies.

- Future-proof your app with [best practices for getting content](#).
- [Set up content preview](#) so that editors can preview unpublished content.
- Map your project's content types to [strongly typed models](#) to streamline your development process.
- [Share content between projects](#) either directly in the UI or programmatically with [Delivery SDKs](#).

Want to create an SDK for your preferred technology? Check out our [guidelines for SDK developers](#).