# **Content Delivery API**

The Optimizely Content Delivery API is a flexible programming interface that you can use for building this Headless CMS solution, using Optimizely as a hub for content.

Content Delivery API is used to extract the content out of the presentation layer of Optimizely CMS. It converts the IContent to ContentApiModel and wraps it and serializes to JSON format and provides it as a response.

## **Steps to integrate Content Delivery API with Optimizely CMS 12**

1. Install the package “EPiserver.ContentDeliveryAPI.CMS” in the project.
2. Configure it in the Startup.cs file.

services.AddContentDeliveryApi(options =>

{

options.SiteDefinitionApiEnabled = true;

options.RequiredRole = string.Empty;

});

* SiteDefinitionApiEnabled option is to define whether site details can be access through headless CMS. If it is set to true, it can be accessed using the endpoint

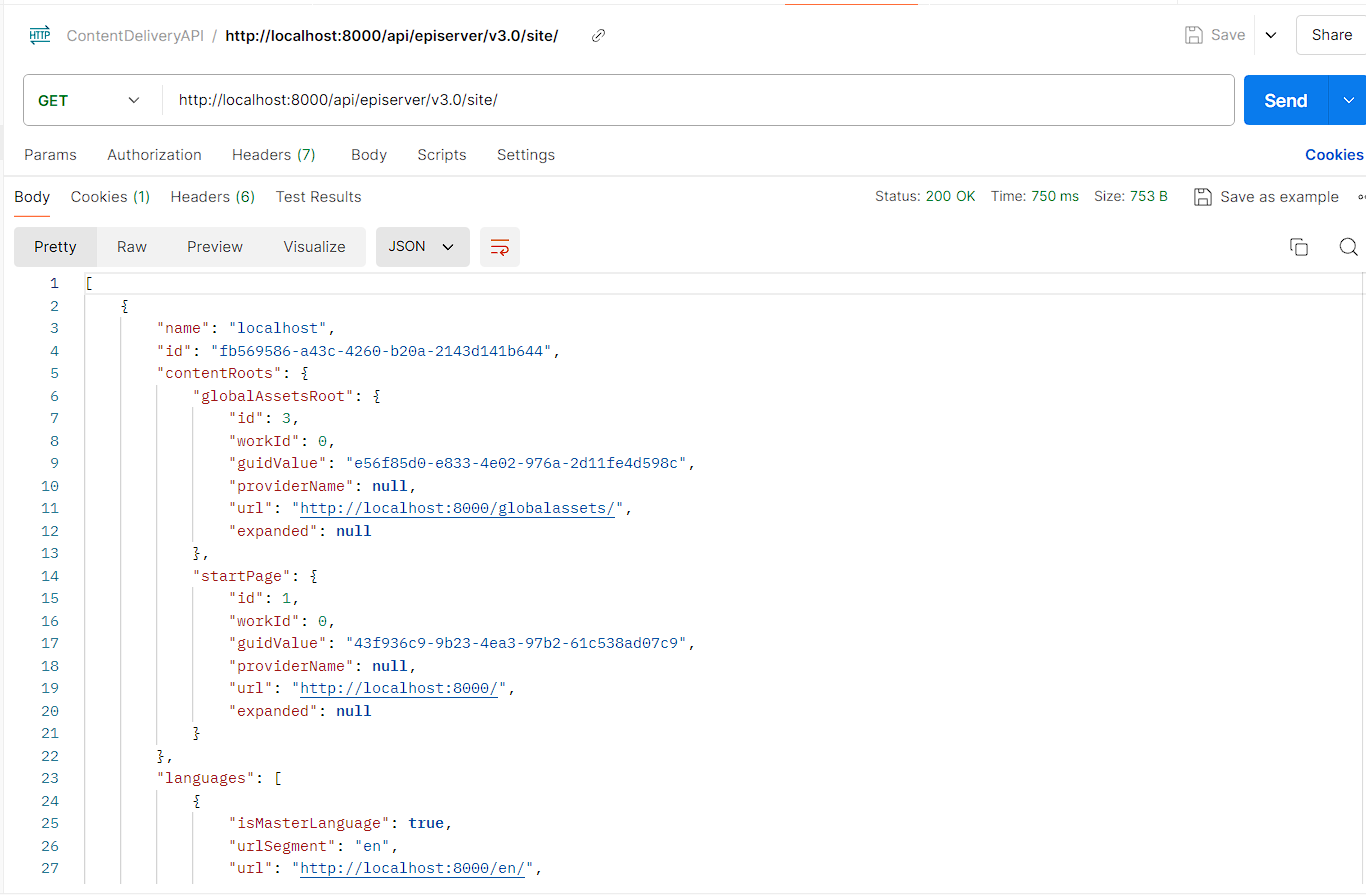
“**api/episerver/v3.0/site**/”

* RequiredRole option is to set who can access content using Headless CMS.

1. You can also add the other options to set cache duration and use “withfriendlyurl()” to access the content using its URL.
2. After configuring run the application.
3. Open the postman to test the Content Delivery API.
4. Postman collection(ContentDeliveryAPI.postman\_collection.json) is added in the same folder where this document resides. You can import it in the postman to test it.
5. To test the site definition API that we enabled, select the Http Method as GET and paste the below URL and click on Send.

<http://localhost:8000/api/episerver/v3.0/site/>

1. The Response will give the details of all the sites created in this CMS 12 application.



1. To access the content using the Content Delivery API, use the below URL to get the content by its ID,  
   [http://localhost:8000/api/episerver/v3.0/content/9/?expand=\*](http://localhost:8000/api/episerver/v3.0/content/9/?expand=*)

The expand=\* query string is used to get all the properties within contentarea, content reference etc.

1. The response will give the content data in JSON format.

