

# Implementation examples MediaStore

The MediaStore API have an online documentation in terms of Swagger documentation:  
<https://mediastore.tradesolution.no/api/swagger>

## Most common request:

GET for fetching metadata about a product:

<https://mediastore.tradesolution.no/api/products/{gtin}>

GET for fetching metadata for the main image:

<https://mediastore.tradesolution.no/api/products/{gtin}/main>

GET for fetching the actual main image:

<https://mediastore.tradesolution.no/api/products/{gtin}/main/image>

GET for fetching all modified GTINs for a given day:

<https://mediastore.tradesolution.no/api/products/getmodifiedgtins?date=2020-02-11>

GET for fetching alle modified images today:

<https://mediastore.tradesolution.no/api/products/modified>

GET for fetching alle modified images between two dates:

<https://mediastore.tradesolution.no/api/products/modified?from=2020-02-11&to=2020-02-11>

## API operations

The standard API Request URL structure to get the binary image data of a product looks like this:

<https://mediastore.tradesolution.no/api/products/{gtin}/{angle}/image>

The request to the standard API needs to include OAuth2 bearer token authentication and can be fetched like this:

POST <https://gatekeeper.tradesolution.no/connect/token>

### Headers

Content-Type: application/x-www-form-urlencoded

### Body

client\_id=<insert\_client\_id>&client\_secret=<insert\_secret>&grant\_type=client\_credentials&resource=ht  
tps://trades.no/TradesolutionApi

### Response

```
{
  "access_token": "eyJhbG*****",
  "expires_in": 36000,
  "token_type": "Bearer",
  "scope": "openid profile tradesolutionApi"
}
```

The token is valid for 10 hours.

## C# implementation example:

```
var client = new System.Net.Http.HttpClient();

var response = await client.PostAsync("https://gatekeeper.tradesolution.no/connect/token",
    new System.Net.Http.FormUrlEncodedContent(new List<KeyValuePair<string, string>>
    {
        new KeyValuePair<string, string>("client_id", "${Provided_By_Tradesolution}"),
        new KeyValuePair<string, string>("client_secret", "${Provided_By_Tradesolution}"),
        new KeyValuePair<string, string>("grant_type", "client_credentials"),
        new KeyValuePair<string, string>("resource", "https://trades.no/TradesolutionApi")
    }));

var json = await response.Content.ReadAsStringAsync();
var jobject = Newtonsoft.Json.Linq.JObject.Parse(json);
var accessToken = (string)jobject["access_token"];

client = new System.Net.Http.HttpClient();
client.DefaultRequestHeaders.Add("Cache-Control", "no-cache");
client.SetBearerToken(accessToken);

response = await
client.GetAsync("https://mediastore.tradesolution.no/api/products/123456789/1C/image");

if (response.IsSuccessStatusCode)
{
    var filename = response.Content.Headers.ContentDisposition.FileName;
    var bytes = await response.Content.ReadAsByteArrayAsync();
    await System.IO.File.WriteAllBytesAsync($"{filename}", bytes);
}
```

## cURL implementation example:

```
curl -X GET -H "Authorization: Bearer ey....." -H
https://mediastore.tradesolution.no/api/products/123456789/1C/image
```

## CDN Operations

The CDN Request URL structure to get the binary image data of a product looks like this:

<https://tradesolutions.akamaized.net/api/products/{userHash}/{gtin}/{angle}/image>

The userHash value is returned by the API operation /products/getUserHash. Any image metadata request made to the standard API will also reveal the userHash for the calling user in the CDNUrl returned. The userHash is different for every user but does not change for a particular user.

The request needs some request headers:

Cache-control: no-cache

Host: tradesolutions.akamaized.net

\_\_token\_\_ (Value to be used is to be taken from the /products/gettoken operation)

### C# implementation example:

```
var client = new System.Net.Http.HttpClient();
client.DefaultRequestHeaders.Add("Cache-Control", "no-cache");
client.DefaultRequestHeaders.Add("host", "tradesolutions.akamaized.net");
client.DefaultRequestHeaders.Add("__token__", "exp=1476265823~acl=/*/dsf7239sdfsd98723cds
/*!/*~hmac=a96d6f4853079197f9f0ba523f591c003c7e2058a81674eece9c3739fcafea53");

var response = await client.GetAsync
("https://tradesolutions.akamaized.net/api/products/dsf7239sdfsd98723cds/123456789/1C/image");

if (response.IsSuccessStatusCode)
{
    var filename = response.Content.Headers.ContentDisposition.FileName;
    var bytes = await response.Content.ReadAsByteArrayAsync();
    await System.IO.File.WriteAllBytesAsync($"{filename}", bytes);
}
```

### cURL example:

```
curl -X GET -H "host: tradesolutions.akamaized.net" -H "__token__:
exp=1476265823~acl=/*/dsf7239sdfsd98723cds/*!/*~hmac=a96d6f4853079197f9f0ba523f591c003
c7e2058a81674eece9c3739fcafea53" -H "Cache-Control: no-cache"
"https://tradesolutions.akamaized.net/api/products/123456789/1C/image"
```