

Using the Contentstack Images API

Welcome, developers, to the Contentstack Images API! As you may know, images are an integral part of any digital experience, and we understand the importance of delivering optimized images quickly and efficiently. That's why we've created the Contentstack Images API, a powerful tool that allows you to manipulate, optimize, and deliver images to your web and mobile applications. Whether you're new to the platform or a seasoned developer, this guide will provide you with everything you need to know to get started with the Contentstack Images API. So let's dive in and explore the possibilities of this powerful tool together!

Here are three advantages to using the Contentstack Images API:

1. **Performance:** The Contentstack Images API allows you to deliver optimized images at scale, reducing load times and improving the overall performance of your web and mobile applications.
2. **Flexibility:** With the Contentstack Images API, you have the flexibility to resize, and transform images in real-time, giving you the ability to customize images to fit any layout or design.

Trying out the Image Delivery API

The quickest way to get started with the Image Delivery API is to try it out with the [online documentation](#).

Resize an image on the fly

Contentstack's Images API resizes images on the server and provides the resized file in the response. This is highly performant as the resized image is stored in the CDN cache on the Contentstack side.

Exercise: Resize an image with the Contentstack Images API

1. Go to the following URL, <https://www.contentstack.com/docs/developers/apis/image-delivery-api/#resize-images>
2. Update the width of the image and see the image size and dimensions being changed

GET: `https://images.contentstack.io/v3/assets/{stack_api_key}/{asset_uid}/{version_uid}/filename.jpg?width={width_value}`

Sample Request

Show Parameters



GET: `https://images.contentstack.io/v3/assets/blteae40eb499811073/bltc5064f36b5855343/59e0c41ac0eddd140d5a8e3e/owl.jpg?width=100`

Query Parameters

width(optional)

100

Try

Response 200



Response Headers

- content-length: 12.692KB
- content-type: image/jpeg

Exercise: Auto-optimizing images with webp

1. Navigate to the Automate Optimization section of the documentation, <https://www.contentstack.com/docs/developers/apis/image-delivery-api/#enable-auto-optimization>
2. Leave the default as webp in the query parameters

Automate Optimization / Auto / Enable auto optimization



GET: `https://images.contentstack.io/v3/assets/{stack_api_key}/{asset_uid}/{version_uid}/filename.jpg?auto={auto_value}`

Sample Request

Show Parameters



GET: `https://images.contentstack.io/v3/assets/blteae40eb499811073/bltc5064f36b5855343/59e0c41ac0eddd140d5a8e3e/owl.jpg?auto=webp`

Query Parameters

auto(optional)

webp

Try

Using the Images API in your NextJS Applications

Pass the width, height, or auto attributes to the src property in the img tag

```

<img
  className='img-fluid'
  src={postData?.image?.url + '?height=800'}
  alt={postData?.image?.filename}
  {...postData?.image.$?.url as {}}
/>
</div>

```

Exercise: Use the Images API to resize the home world image

1. Update the NextJS code to apply a width param of 150px to the home world image. This resizes the image on the server-side and improves the performance of the page load.

```

<div key={indx.toString()} className="mb-3">
  {homeWorld?.title ? <p><strong>{homeWorld?.title}</strong></p> : ''}
  {homeWorld?.image?.url ?
    <img
      className='superHero-logo-img img-fluid mb-3'
      src={homeWorld?.image?.url+ '?width=150'}
      alt={homeWorld?.image?.filename}
      {...homeWorld?.image.$?.url as {}}
    />
    : ''}
  <hr />
</div>

```