

Cookbook - Image resizing

Overview

Image resizing is an out of the box feature of YC that allows on the fly image resizing to width and height specifications.

The configuration for image sizes are kept in ImageService bean. Image service will determine if there is a resized version of the image and if one does not exist it will resize the image and cache it in the image vault. Because image service can resize any image to any size there is a risk of flooding the HDD with all the different versions of resized images. This is why ImageService is restricted to use only specific dimensions (i.e. width and height) and will return full sized image for invalid parameters.

Configurations

The configurations for all the different image sizes and resize parameters are configured directly in the Spring bean in core-services.xml

There are several configuration parameters:

Parameter	Value	Notes
Image dimensions	40x40,50x50,60x60,80x80,200x200,160x160,360x360,120x120,280x280,240x240	Comma separated list of width x height of allowed sizes Allowed pictures sizes to resize. Original image will be returned if requested size not allowed. In version 3.2.0+ this can be configured in System > Preferences > SYSTEM_ALLOWED_IMAGE_SIZES
Background RGB	255 255 255	Background RGB color for re-sized picture. When picture is scaled this is the border colour around the picture where it does not fill all the space
CropToFit	false	When set to true will crop the image to proper ratio prior to scaling, so that scaled image fills all the space. This is useful for those who wish to have images that fill all space dedicated for image without having border around the image. This works best when all images are roughly same diagonal ratio. For those who wish images of products in the middle e.g. as it is in YC demo better to set this to false. False works best for non-standard image diagonal ratios (i.e. mix of panoramic, landscape and square images)
ForceCropToFitOnSize	100	Forcefully uses cropping if width or height is below given size. This option is very useful for small thumbs, as you really cannot see much with added padding for panoramic or tall portrait images.

ETag

To speed up performance ImageFilter uses ETag in order to abort requests for images that have not been modified. This allows to gain immense performance boost on page loads as image content is not streamed.

ETag timeout is set using System > Preferences > SYSTEM_ETAG_CACHE_IMAGES_TIME which is set in minutes. If the file have not been modified then ImageFilter responds with 304 and Last-Modified header only.

JPEG resizing in Oracle JDK 7+

In some cases JDK Image IO standard plugins are unable to cope with image files that have certain meta data configurations.

Typical error in log that you may see for these files will contain message:

```
IllegalArgumentException: Numbers of source Raster bands and source color  
space components do not match For a color image Exception
```

Unfortunately the only way to solve this issue is to add 3rd party plugins that are trying to be more forgiving towards the meta data inside the image files.

One such library is: <https://github.com/haraldk/TwelveMonkeys>

There are numerous plugins for various image formats. We recommend to use only the bare minimum, which is: JPEG and TIFF plugins.

In order to do this you will need the following dependencies:

- twelvemonkeys-common-lang-3.2.1.jar
- twelvemonkeys-common-io-3.2.1.jar
- twelvemonkeys-common-image-3.2.1.jar
- twelvemonkeys-imageio-core-3.2.1.jar
- twelvemonkeys-imageio-metadata-3.2.1.jar
- twelvemonkeys-imageio-jpeg-3.2.1.jar
- twelvemonkeys-imageio-tiff-3.2.1.jar

As described here <https://github.com/haraldk/TwelveMonkeys#manual-dependency-example>

Due to various reason, which are described on <https://github.com/haraldk/TwelveMonkeys> you must ensure that these plugins load in global JVM classpath scope (i.e. prior Tomcat starting).

We recommend adding above mentioned jars to standard Java 3rd party directory:

```
JAVA_HOME/jre/lib/ext
```

On Ubuntu Java home is in `/usr/lib/jvm/java-1.7.0-openjdk-amd64/jre/lib/ext`

Although some people may suggest that this is bad practice since we are introducing extra dependency which we need to manage if we say upgrade to new Java version, unfortunately this is the best way for this particular case due to how Image IO works.

Once the libraries are installed simply restart Tomcat. There are no code changes in YC required as it is fully compliant with Image IO.