Image alias handling

Version compatibility
This feature is available since **eZ Publish** 5.4 **/** 2014.11

Description

Image aliases are generated with LiipImagineBundle, using the underlying Imagine library from avalanche123. This bundle supports GD, Imagick or Gmagick PHP extensions, and allows to define flexible filters in PHP. Imagefiles are stored using the the `IOService,` and are completely independent from `ezimage` FieldType. They are generated only once and cleared on demand (e.g. content removal).

Configuration

Image variation (aka "Image alias") definition follows the same format as before, in `ezpublish.yml` or any imported semantic configuration file. It's dynamic, so can be configured per site access and all the other scopes.

```
# Example
ezpublish:
    system:
        my_siteaccess:
            image_variations:
                small:
                    reference: null
                    filters:
                        - { name: geometry/scaledownonly, params: [100, 160] }
                medium:
                    reference: null
                    filters:
                        - { name: geometry/scaledownonly, params: [200, 290] }
                listitem:
                    reference: null
                    filters:
                        - { name: geometry/scaledownonly, params: [130, 190] }
                articleimage:
                    reference: null
                    filters:
                        - { name: geometry/scalewidth, params: [770] }
```

Important: Each variation name **must be unique**. It may contain _ or - or numbers, but no space.

- `reference`: Name of a reference variation to base the variation on. If `null` (or ~, which means `null` un YAML), the variation will take the original image for reference. It can be any available variation configured in `ezpublish` namespace, or a `filter_set` defined in `liip_imagine` namespace.
- `filters`: array of filter definitions (hashes containing `name` and `params` keys). See possible values below.

Available filters

In addition to filters exposed by LiipImagineBundle, the following are available:

Filter name	Parameters	Description
geometry/scaledownonly	[width, height]	Generates a thumbnail that will not exceed width/height.
geometry/scalewidthdownonly	[width]	Generates a thumbnail that will not exceed width.
geometry/scaleheightdownonly	[height]	Generates a thumbnail that will not exceed height.
geometry/scalewidth	[width]	Alters image width. Proportion will be kept.
geometry/scaleheight	[height]	Alters image height. Proportion will be kept.
geometry/scale	[width, height]	Alters image size, not exceeding provided width and height. Proportion will be kept.
geometry/scaleexact	[width, height]	Alters image size to fit exactly provided width and height. Proportion will not be kept.
geometry/scalepercent	[widthPercent, heightPercent]	Scales width and height with provided percent values. Proportion will not be kept.
geometry/crop	[width, height, startX, startY]	Crops the image. Result will have provided width/height, starting at provided startX/startY
border	[thickBorderX, thickBorderY, color=#000]	Adds a border around the image. Thickness is defined in px. Color is "#000" by default.
filter/noise	[radius=0]	Smooths the contours of an image (`imagick`/`gmagick` only). `radius` is in pixel.
filter/swirl	[degrees=60]	Swirls the pixels of the center of an image (`imagick`/`gmagick` only). `degrees` defaults to 60°.
resize	{size: [width, height]}	Simple resize filter (provided by LiipImagineBundle).
colorspace/gray	N/A	Converts an image to grayscale.

> Tip: *LiipImagineBundle supports additional settings, it is possible to combine filters from the list above to the ones provided in LiipImagineBundle or custom ones.*

Examples with filters:

With liip filter

It is now possible to define the `jpeg_quality` setting. This configuration adds a limit to the image quality using a liip filter.

```
ezpublish:
    system:
        my_siteaccess:
            image_variations:
                reduced_jpeg:
                    reference: null
                    filters:
liip_imagine:
    driver: imagick
    filter_sets:
        mediumimage:
            jpeg_quality: 50
```

With ezpublish filter

Using the geometry/scalewidth filter

```
ezpublish:
    system:
        my_siteaccess:
            image_variations:
                mediumimage:
                    reference: null
                    filters:
                       - geometry/scalewidth:
            params: [770]
```

Discarded filters

The following filters have been discarded due to incompatibility:

- `flatten`. Obsolete, images are automatically flattened.
- `bordercolor`
- `border/width`
- `colorspace/transparent`
- `colorspace`

Custom filters

Please refer to LiipImagineBundle documentation on custom filters. Imagine library documentation may also be useful.

Post-Processors

LiipImagineBundle supports post-processors on image aliases. It is possible to specify them in image alias configuration:

```
ezpublish:
    system:
        my_siteaccess:
            image_variations:
                articleimage:
                    reference: null
                    filters:
                        - { name: geometry/scalewidth, params: [770] }
                    post_processors:
                        jpegoptim: {}
```

Please refer to post-processors documentation in LiipImagineBundle for details.

Drivers

LiipImagineBundle supports GD (default), Imagick and GMagick PHP extensions and only work on image blobs (no command line tool is needed). See the bundle's documentation to learn more on that topic.

Upgrade

- Instantiate `LiipImagineBundle` in your kernel class

If you were using ImageMagick, please install Imagick or Gmagick PHP extensions and activate the driver in `liip_imagine` (see LiipImagineBundle configuration documentation for more information):

```
# ezpublish.yml or config.yml
liip_imagine:
    # Driver can either "imagick", "gmagick" or "gd", depending on the PHP extension
you're using.
    driver: imagick
```

GD will be used by default if no driver is specified.

Purging aliases

ADDED IN 5.4.3 / 2015.05

Starting from the 5.4.3 and 2015.05, eZ Platform handles aliases on its own. It is possible to use the Liip Imagine console tool to clear generated aliases.

```
$ php ezpublish/console liip:imagine:cache:remove --filters=large
$ php ezpublish/console liip:imagine:cache:remove -v
```

The first example will clear the image files for the `large` alias. The second will clear all the generated aliases (be careful), and list the removed files (`-v`).

Note that the naming scheme change introduced by this feature isn't enabled by default on 5.4.x. An alternative purge process has been implemented, that takes much more time, but is compatible with the existing format. More technical information can be found on the pull-request.

Code injection in image EXIF

EXIF metadata of an image may contain e.g. HTML, JavaScript, or PHP code. eZ Platform is itself does not parse EXIF metadata, but third-party bundles need to be secured against this eventuality. Images should be treated like any other user-submitted data - make sure the metadata is properly escaped before use.