

## **MOKIO GLOBAL CONFIGURATION**

Here is how you can customize the behaviour of Mokia application:

### **Default language shortname for backend.**

Set to the shortname of the Lang record with the ID specified in :backend\_default\_lang\_id.

If you wish to change it for example to english, you need to do the following:

1. Create Lang for english (with shortname en - shortname needs to be unique)
2. Set backend\_default\_lang to "en"

```
config.backend_default_lang = "pl"
```

### **How many records are displayed per page.**

Whenever there is a need to display a list of elements in a table (contents, static modules, users) this value is used for pagination. Change this if you wish to display more or less records per page.

Please be careful, performance may decrease after increasing this value.

```
config.backend_default_per_page = 10
```

### **Enable adding google maps for listed content types.**

Mentioned content types will have option to choose location. This can be displayed in frontend using Google Maps API. Available Content types: Article, PicGallery, MovGallery, Contact.

Shorthand: Mokia::CONTENT\_TYPES

```
config.backend_gmap_enabled = ["Contact"]
```

### **Enable adding meta tags for listed models.**

For mentioned classes it will be possible to set META tags (Facebook and Google ones).

By default it is set for menu and all content types. Menu meta tags are more important than content meta tags if both are defined.

```
config.backend_meta_enabled = ["Menu"] + Mokia::CONTENT_TYPES
```

### **How much records are listed in dashboard boxes.**

```
config.dashboard_size = 5
```

Once a photo is uploaded, four copies of this photo are created - using below values for width and height. Photo is scaled to match these dimensions.

### **Default photo width for whole application (in px).**

Uploaded photo will be scaled to have the following width.

Photos will also be displayed with this width in the backend (when a photo in gallery is clicked - big view) and in frontend.

Please note that after you change it, you need to upload your photos again.

Otherwise uploaded photo will have previous width, only view will be scaled accordingly, what may affect photo quality.

```
config.default_width = 500
```

**Default photo height for whole application (in px).**

Similar to default\_width.

**config.default\_height = 500**

**Width for photos thumb (in px).**

Photo thumb (either automatically scaled from original photo or manually changed to different image) will be scaled to have the following width.

Photos will also be displayed with this width in the backend (when a list of photos is displayed, when the thumb is displayed) and in frontend.

Please note that after you change it, you need to upload your photos again.

Otherwise uploaded photo thumb will have previous width, only view will be scaled accordingly, what may affect photo thumb quality.

**config.photo\_thumb\_width = 100**

**Height for photos thumb (in px).**

Similar to photo\_thumb\_width

**config.photo\_thumb\_height = 100**

**Other photo sizes** - works similarly to :default\_width and :default\_height

Medium width for scaling photos (in px)

**config.photo\_medium\_width = 400**

**Medium height for scaling photos (in px)**

**config.photo\_medium\_height = 400**

**Big width for scaling photos (in px)**

**config.photo\_big\_width = 1000**

**Big height for scaling photos (in px)**

config.photo\_big\_height = 1000

**Enable placing watermarks on photos.**

Once set to true, all the images uploaded to the site will be marked with watermark located under :watermark\_path

config.enable\_watermark = false

config.watermark\_path = ""

**Default quality for youtube movies.**

You can set it to low, medium or high. It is used when getting movies from Youtube, Dailymotion

and Vimeo - if only there are movies with given quality provided.

```
config.youtube_movie_quality = "medium"
```

### **Default lang for frontend**

Set to the shortname of the default frontend language

If you wish to change it for example to english, you need to do the following:

1. Create Lang for english (with shortname en - shortname needs to be unique) - unless already created
2. Set frontend\_default\_lang to "en"
3. Create Initial menu structure for this language (unless already created)
  - 3a. Create Menu element for lang node: en (fake: true, parent: nil, lang\_id - ID of the record from point 1)
  - 3b. Create Menu element for menu position: top (fake: true, parent - element from point 4a, lang\_id - ID of the record from point 1 )
  - 3c. (Optional) Create Menu element for other position if you wish to have more than one menu on your site
4. Add frontend\_initial\_en variable and set it to the ID of Menu record from point 4a.

In your frontend views use :frontend\_initial\_en when calling build\_menu

```
config.frontend_default_lang = "pl"
```

Id of root menu element for specific language

```
config.frontend_initial_pl = 1
```

### **Facebook app id to use in frontend.**

If you would like to have facebook comments enabled for your application you need to provide Facebook App ID - you can get it from <https://developers.facebook.com/apps>

```
config.frontend_facebook_app_id = "
```