



International ALA workshop
3-5 October 2016

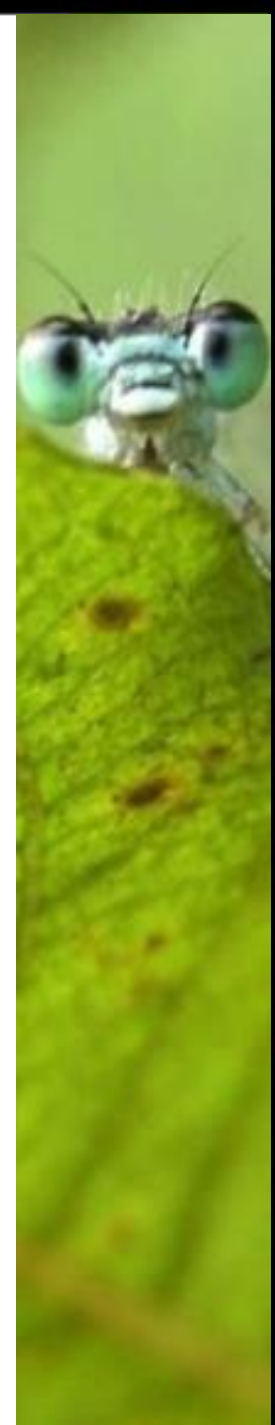
Skinning your ALA instance

Marie-Elise Lecoq
dev@gbif.fr



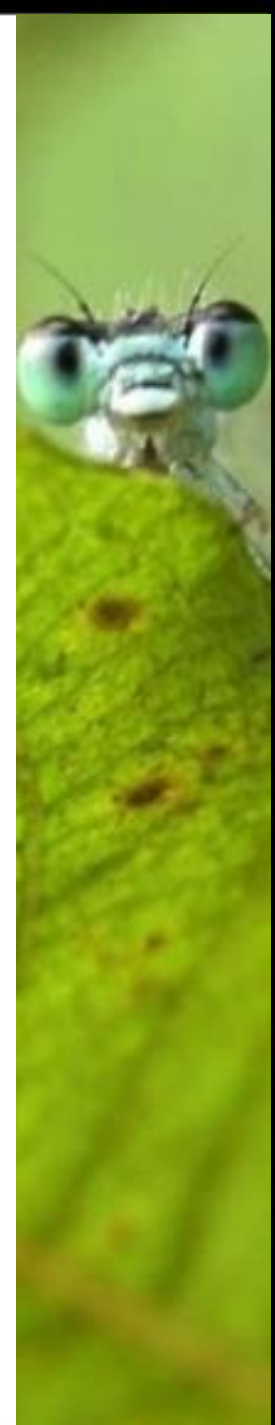
Example on ALF

[portail.gbif.fr](http://portail.gbif.org)



Installation

- Fork the module and **rename it** to reflect your organization (ex: gbiffrance-hub).
 - generic-hub for the biocache-hub module
 - generic-collectory for the collectory module
- Clone / checkout the project to your local PC
- Move the file that you want to change from the plugin to your grails-app folder



Layout

- Copy the `grails-app/views/layouts/generic.gsp` file to a new file (ex: `yourOrg.gsp`) in the same location
- Configure the web-app using one of this mechanism:
 - On the external file `/data/appName/conf/appName-config.properties`
 - On the internal Grails config file `/grails-app/conf/config.groovy`
- Change these variables :
 - `skin.layout = 'generic'` to `skin.layout = 'yourOrg'`
 - `skin.orgNameLong='Generic Data Portal'` to `skin.orgNameLong='Your Org Name'`
- Edit your `yourOrg.gsp` file and have fun !

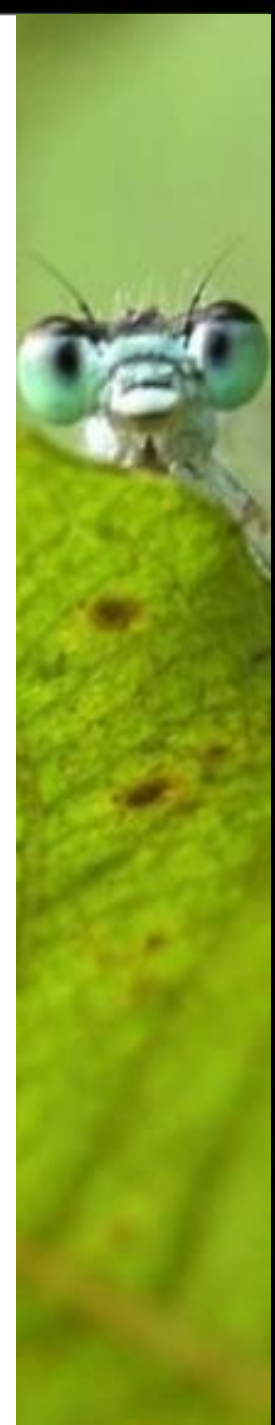


Create custom CSS and JS file (1)

- Tool to manage static resource : Resources plugin
- Declare in the file your module your Org:
`grails-app/conf/ApplicationResources.groovy`

```
alf {  
  dependsOn 'bootstrap3, collectory'  
  resource url: [dir:'js', file:'application.js']  
  resource url: [dir:'css', file:'generic.css']  
  resource url: [dir:'images', file:'rss.png']  
  resource url: [dir:'css/images', file:'arrow_state_grey_right.png']  
  resource url: [dir:'images/skin', file:'ExpandArrow.png']  
  resource url: [dir:'images/ala', file:'fauna_button_bgwd.png']  
  [..]  
}
```

- In your layout file, you need to add the module
`<r:require modules="jquery, jquery_migration, bootstrap3, application, collectory, alf" />`



Bootstrap3

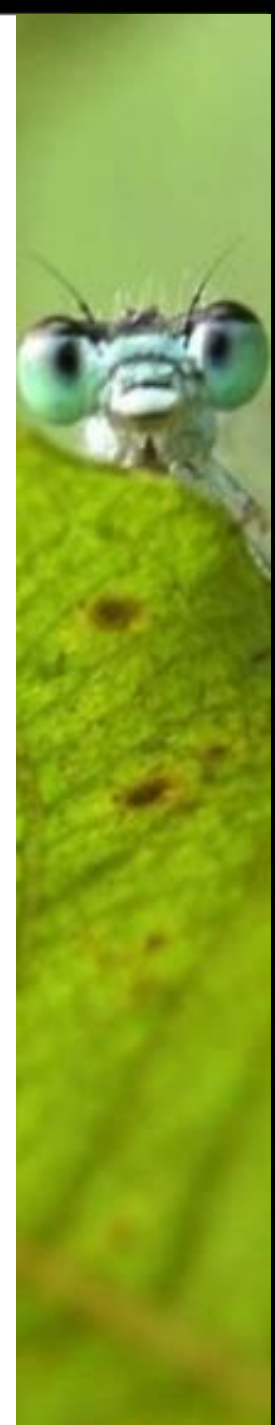
- Add the module `bootstrap3` in your `applicationResource.groovy` file using these commands:

```
bootstrap3 {  
    resource url: [dir: 'bootstrap3/js', file: 'bootstrap.js', disposition:  
    'head']  
    resource url: [dir: 'bootstrap3/css', file: 'bootstrap.css', attrs: [media:  
'screen, projection, print']]  
    resource url: [dir: 'bootstrap3/css', file: 'bootstrap-theme.css', attrs:  
    [media: 'screen, projection, print']]  
}
```

- Download Bootstrap3:

<http://getbootstrap.com/getting-started/#download>

- Add this folder in the web-app folder
- Add `bootstrap3` to your `require module` tag in your layout file

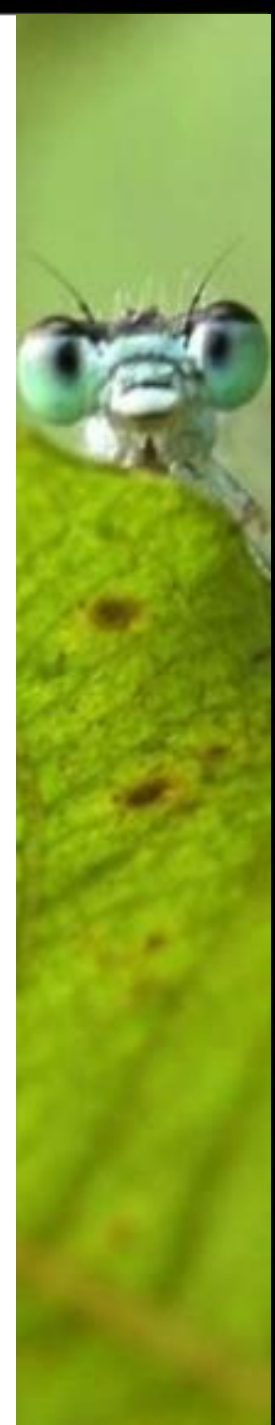


Bootstrap3 (2)

- You need to upgrade JQuery 1.8 to JQuery 1.11
 - **Be careful : Some functions are depreciated**
- For example : `.live()` need to be change to the `.on()`
- On the Config file, you need to add the `/bootstrap3/` folder to have access to image

```
grails.resources.adhoc.patterns = ['/img/**', '/images/*', '/data/*', '/css/*', '/js/**',  
'/plugins/**', '/bootstrap3/css/**', '/bootstrap3/js/**', '/bootstrap3/fonts/**']
```

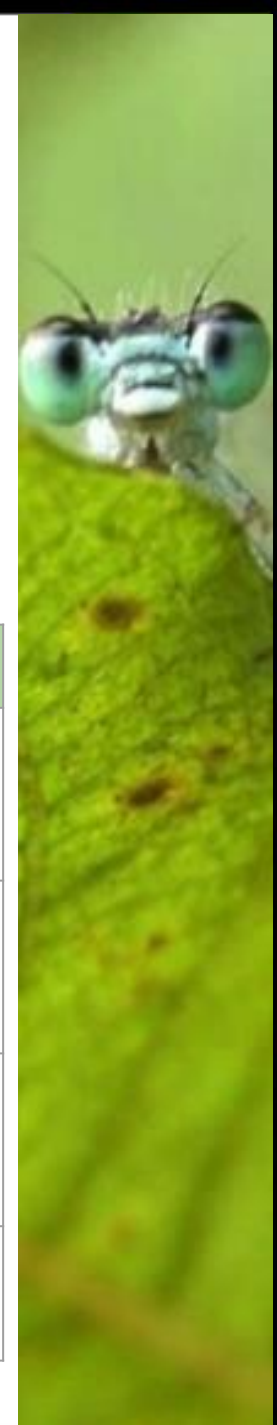
```
grails.resources.adhoc.includes = ['/img/**', '/images/*', '/data/*', '/css/*', '/js/**',  
'/plugins/**', '/bootstrap3/css/**', '/bootstrap3/js/**', '/bootstrap3/fonts/**']
```



Grails tag (1)

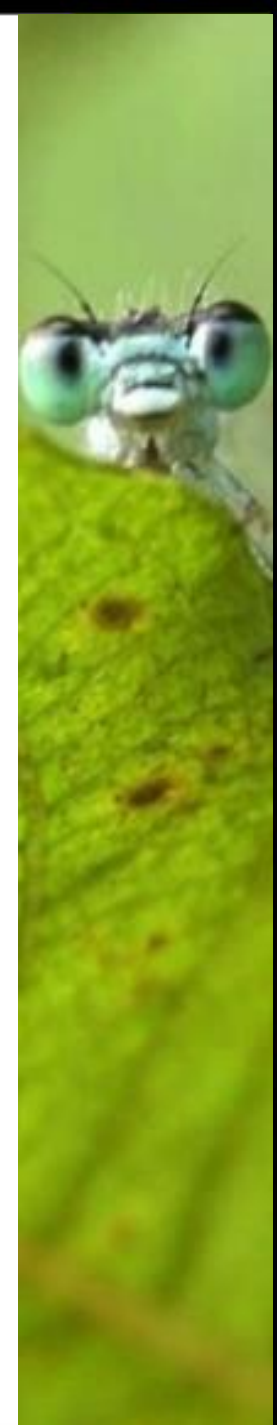
- You will find below, a non-exhaustive list of Grails tags with some explanations about them.
- Doc: <http://docs.grails.org/latest/ref/Tags/actionSubmit.html>
- format : `<g:tag name option="value_of_option" />`

Tag name	Description
message	Use for the internationalization, the code properties will be in the message.properties file: http://docs.grails.org/latest/ref/Tags/message.html
link	Anchor that creates a HTML link. More information on this page: http://docs.grails.org/latest/ref/Tags/link.html
if/else	The logical if/else tag : http://docs.grails.org/latest/ref/Tags/if.html http://docs.grails.org/latest/ref/Tags/else.html
actionSubmitImage	Generates a submit action with input type = "image": http://docs.grails.org/latest/ref/Tags/actionSubmitImage.html

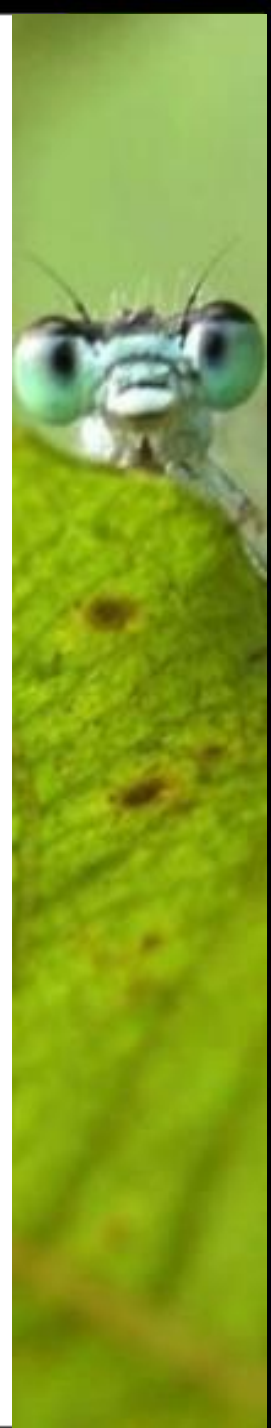


Grails tag (2)

Tag name	Description
actionSubmit	Anchor that will create a button with a submit action. In the action part, you will be able to change the action of the button (eg: delete, submit, etc.) and you can have several submit actions in one form: http://docs.grails.org/latest/ref/Tags/actionSubmit.html
select	Generates a select field in a form: http://docs.grails.org/latest/ref/Tags/select.html
checkbox	Creates a checkbox field in a form: http://docs.grails.org/latest/ref/Tags/checkBox.html
hiddenField	Adds a hidden field on your form: http://docs.grails.org/latest/ref/Tags/hiddenField.html
textArea	Creates and fills a text area field on a form: http://docs.grails.org/latest/ref/Tags/textField.html
form	Anchor that creates and manages a form that submits to a controller: http://docs.grails.org/latest/ref/Tags/form.html
textField	Creates and fills a text field of a form: http://docs.grails.org/latest/ref/Tags/textField.html



Practical work



Merci !

GBIF France's IT Team
dev@gbif.fr

Fabien Caviere
caviere@gbif.fr
<https://github.com/CaviereFabien>

Marie-Elise Lecoq
melecoq@gbif.fr
[@bouhoub](https://github.com/melecoq)

Github account
<https://github.com/gbiffrance>

