
gs.content.layout Documentation

Release 3.1.0

GroupServer.org

January 28, 2016

1	gs.content.form.base Templates	3
1.1	Layouts	3
1.2	Full-Page Layout	3
1.3	Layout with a Menu	3
1.4	Slots	3
2	Page layout examples	5
2.1	A Simple Page	5
2.2	Using the SiteInfo	5
2.3	Page-Specific Style and JavaScript	6
3	Changelog	7
3.1	3.1.12 (2016-01-28)	7
3.2	3.1.11 (2016-01-25)	7
3.3	3.1.10 (2016-01-15)	7
3.4	3.1.9 (2015-12-09)	7
3.5	3.1.8 (2015-11-13)	7
3.6	3.1.7 (2015-11-09)	7
3.7	3.1.6 (2015-11-06)	8
3.8	3.1.5 (2015-10-27)	8
3.9	3.1.4 (2015-10-15)	8
3.10	3.1.3 (2015-03-20)	8
3.11	3.1.2 (2015-03-10)	8
3.12	3.1.1 (2015-03-03)	8
3.13	3.1.0 (2014-09-29)	8
3.14	3.0.0 (2014-09-09)	8
3.15	2.5.0 (2014-06-27)	9
3.16	2.4.3 (2014-06-18)	9
3.17	2.4.2 (2014-05-30)	9
3.18	2.4.1 (2014-04-03)	9
3.19	2.4.0 (2014-03-19)	9
3.20	2.3.6 (2014-03-05)	9
3.21	2.3.5 (2014-02-17)	9
3.22	2.3.4 (2014-01-30)	9
3.23	2.3.3 (2013-11-26)	10
3.24	2.3.2 (2013-11-20)	10
3.25	2.3.1 (2013-11-05)	10
3.26	2.3.0 (2013-10-17)	10

3.27	2.2.1 (2013-10-01)	10
3.28	2.2.0 (2013-05-31)	10
3.29	2.1.0 (2013-04-04)	10
3.30	2.0.0 (2013-01-29)	11
3.31	1.2.1 (2013-01-16)	11
3.32	1.2.0 (2012-12-12)	11
3.33	1.1.0 (2012-11-29)	11
3.34	1.0.0 (2012-08-02)	11
4	Indices and tables	13
5	Resources	15

Contents:

gs.content.form.base Templates

1.1 Layouts

There are two standard *layouts* for pages on GroupServer:

- A *full-page layout*, and
- A *layout with a menu*.

1.2 Full-Page Layout

The full-page layout is the standard layout for GroupServer. To use the full-page layout with your page first add a `metal:use-macro` call to the `<html>` element.

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:tal="http://xml.zope.org/namespaces/tal"
      xmlns:metal="http://xml.zope.org/namespaces/metal"
      metal:use-macro="context/@@groupserver_full_layout/page">
```

Then fill the *slots*.

1.3 Layout with a Menu

The layout with a menu has a *context menu* running down the left-hand side of the page. Otherwise it is exactly the same as the *full-page layout*. To use the layout with a menu use the `groupserver_menu_layout` macro.

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:tal="http://xml.zope.org/namespaces/tal"
      xmlns:metal="http://xml.zope.org/namespaces/metal"
      metal:use-macro="context/@@groupserver_menu_layout/page">
```

Then fill the *slots*.

1.4 Slots

There are nine slots defined by the two layouts.

metal:fill-slot="title": The compulsory title of the page. It is **always** provided by pages that use the standard layouts, and **always** contains a `<title>` element.

```
<title metal:fill-slot="title">This is the title</title>
```

metal:fill-slot="metadata": Optional extra metadata for the page. While `<meta>` elements can be added they almost never are. However, `<link>` elements are often added (especially in pages that would use a breadcrumb trail, such as within groups). Normally a `<metal:block>` element has the `fill-slot` attribute, and the metadata to add is placed inside the block.

metal:fill-slot="style": Optional extra style (CSS) information for the page. Normally a `<style>` element has the `fill-slot` macro. When rendered the style slot appears after the standard GroupServer CSS ¹, so the page can overwrite the general style.

metal:fill-slot="utilitylinks": The optional *utility* menu. The only time this is ever set is to ensure the is *no* menu. By default the utility links show either:

- A *Login* link, or
- A *Log Out* link and a *Profile* link.

metal:fill-slot="breadcrumb": The optional *breadcrumb trail*. It is normally an unordered list, with the first item a link to the site-homepage.

```
<ul metal:fill-slot="breadcrumb">
  <li>
    <a href="/" class="icon-alone">
      <span aria-hidden="true" data-icon="#x2302;"></span>
      <span class="screen-reader-text">Site home page</span>
    </a>
  </li>
  <li>
    <a href="#"><strong>Important</strong></a>
  </li>
  <li>
    A page.
  </li>
</ul>
```

metal:fill-slot="messages": Feedback messages for the form. This is almost only ever filled by the content-provider supplied by the `gs.content.form.egg` ².

```
<tal:block
  content="structure provider:groupserver.FormStatusMessage"
  define="errors view/errors; status view/status; widgets view/widgets"
  metal:fill-slot="messages">&#160;</tal:block>
```

metal:fill-slot="body": The **compulsory** body of the page.

metal:fill-slot="footer": The optional footer of the page. It appears after the body. By default it contains the contents of the `Templates/output/footerlinks.xml` instance in the ZML.

metal:fill-slot="javascript": The JavaScript (technically the ECMAScript) for the page. The page-specific scripts appear *after* the standard JQuery code ³ has been loaded, and the rest of the page has been rendered.

¹ See the `gs.content.css` product <<https://github.com/groupserver/gs.content.css/>>

² See the `gs.content.form.base` product <<https://github.com/groupserver/gs.content.form.base/>>

³ See the `gs.content.js.jquery.base` product <<https://github.com/groupserver/gs.content.js.jquery.base/>>

Page layout examples

Below are some examples for using the pay layouts.

2.1 A Simple Page

Most pages only have to fill three slots: the title, breadcrumb and the body

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:tal="http://xml.zope.org/namespaces/tal"
      xmlns:metal="http://xml.zope.org/namespaces/metal"
      metal:use-macro="context/@@groupserver_full_layout/page">
  <head>
    <title metal:fill-slot="title">I am a page: Example</title>
  </head>
  <body>
    <ul metal:fill-slot="breadcrumb">
      <li>
        <a href="/" class="icon-alone">
          <span aria-hidden="true" data-icon="&#x2302;"></span>
          <span class="screen-reader-text">Site home page</span>
        </a>
      </li>
    </ul>
    <div id="a-page" metal:fill-slot="body">
      <p>I am a page, honest.</p>
    </div><!--a-page-->
  </body>
</html>
```

The rest of the slots will be filled by the defaults.

2.2 Using the SiteInfo

Because *most* pages have `gs.content.base.SitePage`¹ as the base view-class, there is always a `siteInfo` available to use in the title slot and the rest of the page:

¹ See the `gs.content.base` product <<http://github.com/groupserver/gs.content.base/>>

```
<head>
  <title metal:fill-slot="title">I am a page:
    <span tal:replace="view/siteInfo/name">wibble</span></title>
</head>
```

The `tal:replace` attribute is used because a `` element is not actually allowed to appear within a `<title>`. A `<tal:block>` could also be used.

2.3 Page-Specific Style and JavaScript

Some pages have some page-specific CSS styling and JavaScript.

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:tal="http://xml.zope.org/namespaces/tal"
      xmlns:metal="http://xml.zope.org/namespaces/metal"
      metal:use-macro="context/@@groupserver_full_layout/page">
  <head>
    <title metal:fill-slot="title">I am a page: Example</title>
    <style type="text/css" metal:fill-slot="style">
      .wibble {font-variant: small-caps; font-weight: bold;}
    </style>
  </head>
  <body>
    <metal:block fill-slot="body">
      <p>I am a <span class="wibble">page,</span> honest.</p>
    </metal:block>
    <script type="text/javascript" metal:fill-slot="javascript"
      defer="true" src="/++resource++my.js"> </script>
  </body>
</html>
```

The `defer="true"` is important: while both jQuery and Bootstrap are loaded by default, the loading of both is deferred until *after* the page has loaded. Anything that wants to use jQuery has to have `defer="true"` set.

Changelog

3.1 3.1.12 (2016-01-28)

- Following the changes to the global CSS

3.2 3.1.11 (2016-01-25)

- Following the update to `gs.content.js.disclosure`
- Following the update to `gs.content.js.loader`
- Following the update to `' gs.content.js.submit`

3.3 3.1.10 (2016-01-15)

- Following the updated jQuery JavaScript

3.4 3.1.9 (2015-12-09)

- Following the changes to the global CSS

3.5 3.1.8 (2015-11-13)

- Following the changes to the global CSS

3.6 3.1.7 (2015-11-09)

- Following the updated `required widgets` JavaScript

3.7 3.1.6 (2015-11-06)

- Following the updated [global CSS](#)

3.8 3.1.5 (2015-10-27)

- Following the updated site CSS

3.9 3.1.4 (2015-10-15)

- Following the updated [global CSS](#)

3.10 3.1.3 (2015-03-20)

- Following the updated [global CSS](#)

3.11 3.1.2 (2015-03-10)

- Following the updated [global CSS](#)

3.12 3.1.1 (2015-03-03)

- Pointing at [Transifex](#)
- Added a German translation (thanks, [Alice](#))
- Added a French translation (thanks, [Mohamed-Younès](#))
- Following the updated [global CSS](#)

3.13 3.1.0 (2014-09-29)

- Added the product to [GitHub](#)
- Renaming the `txt` files `rst` files
- Added Sphinx documentation

3.14 3.0.0 (2014-09-09)

- Added internationalisation

3.15 2.5.0 (2014-06-27)

- Moved the shortcut icon to its own product: `gs.content.favicon`.

3.16 2.4.3 (2014-06-18)

- Updating the shortcut icon (favicon), with [thanks to Jon Richter](#)

3.17 2.4.2 (2014-05-30)

- Following the new CSS version
- Updating the documentation for the breadcrumb slot

3.18 2.4.1 (2014-04-03)

- Dealing with some base-URL corner cases

3.19 2.4.0 (2014-03-19)

- Handling a base URL that includes a port
- Not deferring the loading of jQuery, so IE8 will work
- Following the changes to some JavaScript resources

3.20 2.3.6 (2014-03-05)

- Following the new CSS version, closing [Bug 4082](#)

3.21 2.3.5 (2014-02-17)

- Following the updated [global CSS](#)

3.22 2.3.4 (2014-01-30)

- Following the updated [global CSS](#)
- Explicitly not deferring the loading of the JavaScript loader

3.23 2.3.3 (2013-11-26)

- Following the updated `global CSS`

3.24 2.3.2 (2013-11-20)

- Following the updated `global CSS`

3.25 2.3.1 (2013-11-05)

- Following the updated `global CSS`

3.26 2.3.0 (2013-10-17)

- Switching to a simpler `<base>` element that works better
- Reporting the skin, if set, as a `data` attribute in the `<html>` element

3.27 2.2.1 (2013-10-01)

- Added the viewport meta-tag to the page header so the pages look better on small-screen devices
- Cleanup of the product metadata

3.28 2.2.0 (2013-05-31)

- Following the update to jQuery
- Adding the required-widgets JavaScript to all pages

3.29 2.1.0 (2013-04-04)

- Using the new icon-font from `gs.content.css`
- Switching some links to buttons
- Allowing infinite footers
- Switching to minified versions of the standard JavaScript
- Deferring the loading of the JS
- Added WAI-ARIA roles to the page

3.30 2.0.0 (2013-01-29)

- Switching to HTML5 and Twitter Bootstrap

3.31 1.2.1 (2013-01-16)

- Updating the jQuery links.

3.32 1.2.0 (2012-12-12)

- Adding an image to the profile link

3.33 1.1.0 (2012-11-29)

- Dropping the external-bar
- Dropping the site navigation
- Adding breadcrumbs

3.34 1.0.0 (2012-08-02)

- Initial import from `gs.content.base`

This product defines the standard layouts for pages on [GroupServer](#). The layouts provide slots that are filled by the pages that provide the content. By using these standard layouts less code is required to get all the pages to look consistent.

Indices and tables

- `genindex`
- `modindex`
- `search`

Resources

- Documentation: <http://groupserver.rtf.d.org/projects/gscontentlayout/>
- Code repository: <https://github.com/groupserver/gs.content.layout>
- Translations: <https://www.transifex.com/projects/p/gs-content-layout/>
- Questions and comments to <http://groupserver.org/groups/development>
- Report bugs at <https://redmine.iopen.net/projects/groupserver>