

# Uniformity Guidelines

Starting in FreePBX 13 the development team made the decision to apply a consistent look and feel to all of FreePBX. The below guidelines should be taken into account when you are creating or editing your module for FreePBX 13.

- Getting Started
- Using the Grid
- Adding "Bootnav"
- Adding Tabable Pages
- Add Form Elements
- Special Classes
- Standard Icons
- View Functions

## Getting Started

Starting with FreePBX 12 we now incorporate Twitter Bootstrap into the project. Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web. You should give it a once over before continuing with this article at: <http://getbootstrap.com/>

Since most of FreePBX is user driven "form" based input you'll want to focus on the "form" elements <http://getbootstrap.com/css/#forms>

We also use bootstrap's "grid" system <http://getbootstrap.com/css/#grid>

## Using the Grid

When we use the "grid" to create a module page we are trying to achieve two goals. First when we are in a desktop environment we want the page to be divided with 66% of the space given to the form elements while 33% is given to what we now call "bootnav" (formerly called rnav).

**Extension: 1000**

General Voicemail Find Me/Follow Me Advanced

— Edit Extension

Display Name ?

Add Extension

Tango Frog <1000>

Digit Robot <1001>

Delete Reset Submit

When the page shrinks to less than 768px width we hide our "bootnav". Since bootstrap "stacks" when the width is less than 768px by default the nav will jump to the bottom of the page, however since it is hidden it is not seen and our primary form display becomes full width.

### Extension: 1000


General Voicemail Find Me/Follow Me Advanced

— Edit Extension

Display Name ?

Delete Reset Submit

You may be asking as this point. "Well if I'm on a small device and have no rnav or 'bootnav' how can I navigate?". On smaller devices users are encouraged to use the new search box on the upper nav bar to find what they want.

Search 

---

Bootstrap itself works with columns of 12. So every row **must** have 12 columns total. To get our desired 66% to 33% ratio we are going to use 9 for the left half and 3 for the right half.

So for our page layout (usually in page.<modulename>.php or under the "views" folder we want to start off like so:

```
<div class="container-fluid">
  <div class="row">
    <div class="col-sm-12">
      Form Elements here!
    </div>
  </div>
</div>
```

## Adding "Bootnav"

[Adding Floating Right Nav to Your Module](#)

## Adding Tabable Pages

We are going to throw a lot of html in your face right now but quickly review the below html:

```

<div class="container-fluid">
  <div class="row">
    <div class="col-sm-12">
      <div class="fpbx-container">
        <form class="fpbx-submit" name="
frm_extensions" action="config.php?display=extensions" method="post" data-
fpbx-delete="config.php?display=extensions&extdisplay=1000&
action=del" role="form">
          <div class="nav-container">
            <div class="scroller
scroller-left"><i class="glyphicon glyphicon-chevron-left"></i></div>
            <div class="scroller
scroller-right"><i class="glyphicon glyphicon-chevron-right"></i></div>
            <div class="wrapper">
              <ul class="nav nav-
tabs list" role="tablist">
                <li data-
name="tab1" class="change-tab active"><a href="#tab1" aria-controls="tab1"
role="tab" data-toggle="tab">Tab1 (active!)</a></li>
                <li data-
name="tab2" class="change-tab"><a href="#tab2" aria-controls="tab2" role="
tab" data-toggle="tab">Tab2</a></li>
              </ul>
            </div>
          </div>
          <div class="tab-content display">
            <div id="tab1" class="tab-
pane active">
              Tab 1 Data (Shown)
            </div>
            <div id="tab2" class="tab-
pane">
              Tab 2 Data (Hidden)
            </div>
          </div>
        </form>
      </div>
    </div>
  </div>
</div>

```

You'll notice we've now added a div with a class of "freepbx-container". This is important as it helps our CSS and Javascript match to your page and automatically perform all of the beautifulness that needs to happen. You don't have to do a thing (well besides any special tricks you'd like to perform)

The above HTML will produce output looking like below



You can click on tab2 and watch as javascript hides the first page and shows the second (hidden) page without refreshing the entire page. Everything is still stored on the page so that when the "form" is submitted everything will go along with it. No need to keep track of multiple pages or tags



So lets break apart the HTML used to create this

```

<div class="fpbx-container">
  <form class="fpbx-submit" name="frm_extensions" action="config.php?
display=extensions" method="post" data-fpbx-delete="config.php?
display=extensions&extdisplay=1000&action=del" role="form">
    <div class="nav-container">
      <div class="scroller scroller-left"><i class="
glyphicon glyphicon-chevron-left"></i></div>
      <div class="scroller scroller-right"><i class="
glyphicon glyphicon-chevron-right"></i></div>
      <div class="wrapper">
        <ul class="nav nav-tabs list" role="
tablist">
          <li data-name="tab1" class="change-
tab active"><a href="#tab1" aria-controls="tab1" role="tab" data-toggle="
tab">Tab1 (active!)</a></li>
          <li data-name="tab2" class="change-
tab"><a href="#tab2" aria-controls="tab2" role="tab" data-toggle="tab"
>Tab2</a></li>
        </ul>
      </div>
    </div>
    <div class="tab-content display">
      <div id="tab1" class="tab-pane active">Tab 1 Data
(Shown)</div>
      <div id="tab2" class="tab-pane">Tab 2 Data (Hidden)
</div>
    </div>
  </form>
</div>

```

First and foremost we've got the "freepbx-container" div. Remember that if you want to use tabbing pages you need to wrap the tabs inside of this so that the javascript and css will be applied.

Next up we've got the form element. Make sure you've added all necessary data tags that are needed by the traveling page buttons (such as data-fpbx-delete, the class "fpbx-submit" and role="form"). For more information on this please see [Adding Form Buttons to Your Module \(Action Bar\)](#)

```

<form
  method="post"
  action="config.php?display=extensions"
  class="fpbx-submit"
  name="frm_extensions"
  data-fpbx-delete="config.php?display=modeule&extdisplay=1000&

```

```

action=del "
  role="form"
>

```

Beyond that we have our "tablist" in which we can see our active tab.

```

<div class="nav-container">
  <div class="scroller scroller-left"><i class="glyphicon glyphicon-
chevron-left"></i></div>
  <div class="scroller scroller-right"><i class="glyphicon glyphicon-
chevron-right"></i></div>
  <div class="wrapper">
    <ul class="nav nav-tabs list" role="tablist">
      <li data-name="tab1" class="change-tab active"><a href="#"
href="#tab1" aria-controls="tab1" role="tab" data-toggle="tab">Tab1
(active!)</a></li>
      <li data-name="tab2" class="change-tab"><a href="#"
href="#tab2" aria-controls="tab2" role="tab" data-toggle="tab">Tab2</a></li>
    </ul>
  </div>
</div>

```

Then we have the wrapper div called "display". Your pages **must** be wrapped in this div to have the border surrounding them. Each tab is contained in an "info-pane" div with an id that matches the tab's data-name. Make sure to add the "hidden" class to any tab page that should not be shown on initial page load (usually everything but your active tab).

```

<div class="tab-content display">
  <div id="tab1" class="info-pane active">Tab 1 Data (Shown)</div>
  <div id="tab2" class="info-pane">Tab 2 Data (Hidden)</div>
</div>

```

## Add Form Elements

Now that we've got a nice tabbed page going on we should add our form elements to each page along with the optional auto hiding help text.

```

<div class="container-fluid">
  <div class="row">
    <div class="col-sm-12">
      <div class="fpbx-container">
        <form class="fpbx-submit" name="
frm_extensions" action="config.php?display=extensions" method="post" data-
fpbx-delete="config.php?display=extensions&extdisplay=1000&
action=del" role="form">
          <div class="nav-container">
            <div class="scroller
scroller-left"><i class="glyphicon glyphicon-chevron-left"></i></div>
            <div class="scroller
scroller-right"><i class="glyphicon glyphicon-chevron-right"></i></div>
            <div class="wrapper">

```

```

        <ul class="nav nav-
tabs list" role="tablist">
            <li data-
name="tab1" class="change-tab active"><a href="#tab1" aria-controls="tab1"
role="tab" data-toggle="tab">Tab1 (active!)</a></li>
            <li data-
name="tab2" class="change-tab"><a href="#tab2" aria-controls="tab2" role="
tab" data-toggle="tab">Tab2</a></li>
        </ul>
    </div>
</div>
<div class="tab-content display">
    <div id="tab1" class="tab-
pane active">
        <div class="
container-fluid">
            <div
class="section-title" data-for="section1"><h3><i class="fa fa-minus"></i>
Section 1</h3></div>
            <div
class="section" data-id="section1">
<div class="element-container">
<div class="row">
<div class="form-group form-horizontal">
<div class="col-md-3">
<label class="control-label" for="element1">Prompt Text</label>
<i class="fa fa-question-circle fpbx-help-icon" data-for="element1"></i>
</div>
<div class="col-md-9">
<input type="text" class="form-control" id="element1">
</div>
<div class="col-md-12">
<span id="element1-help" class="help-block fpbx-help-block">Help Text<
/span>
</div>
</div>
</div>
</div>

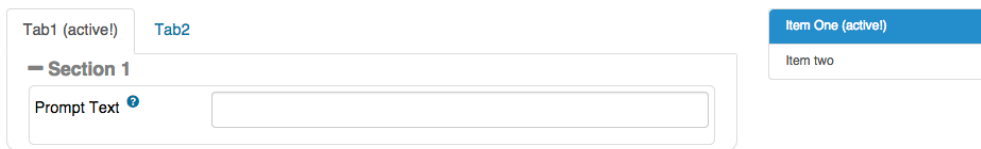
```

```

/div>
                                </div>
                                </div>
                                </div>
                                <div id="tab2" class="tab-
pane">
                                </div>
                                </div>
                                </form>
                                </div>
                                </div>
                                </div>
                                </div>
                                </div>

```

The above page will output the below result. Let's spend some time dissecting each bit.



Here is the only part we are caring about at this point, which is everything inside our first tab.

```

<div class="container-fluid">
  <div class="section-title" data-for="section1"><h3><i class="fa fa-
minus"></i> Section 1</h3></div>
  <div class="section" data-id="section1">
    <div class="element-container">
      <div class="row">
        <div class="form-group form-horizontal">
          <div class="col-md-3">
            <label class="control-
label" for="element1">Prompt Text</label>
            <i class="fa fa-question-
circle fpbx-help-icon" data-for="element1"></i>
          </div>
          <div class="col-md-9">
            <input type="text" class="
form-control" id="element1">
          </div>
        </div>
        <div class="col-md-12">
          <span id="element1-help" class="
help-block fpbx-help-block">Help Text</span>
        </div>
      </div>
    </div>
  </div>
</div>

```

First off make sure everything is in a "container-fluid" element which gives us a little more padding

```
<div class="container-fluid">
```

The next part is our section. Note the usage of data-id for the section instead of just id. This is because many of the section ids have spaces, which are not valid.

```
<div class="section-title" data-for="section1"><h3><i class="fa fa-minus"><
/i> Section 1</h3></div>
<div class="section" data-id="section1">
  Text for inside my section!
</div>
```

Now we start building the first form element. Make sure to use the wrapper class "element-container" so that everything works as expected!

```
<div class="element-container">
  It'll go here!
</div>
```

Lets add some rows and columns to separate everything out. Use the class "control-label" for each label and make sure that the for points to the id of the form element.

Warning: If you don't add 'form-horizontal' to the form-group class, your labels will be too high!

```
<div class="row">
  <div class="form-group form-horizontal">
    <div class="col-md-3">
      <label class="control-label" for="element1">Prompt
Text</label>
    </div>
    <div class="col-md-9">
      <input type="text" class="form-control" id="
element1" placeholder="This gives an example of what to put in this field">
    </div>
  </div>
</div>
```

If your form input needs an explanation, you lay it out slightly differently, by adding a 'fpbx-help-icon' and a new full width row. Before adding a help row, think if you could rephrase the name, or add better placeholder text instead!

```
<div class="row">
  <div class="form-group form-horizontal">
    <div class="col-md-3">
      <label class="control-label" for="element1">Prompt
Text</label>
      <i class="fa fa-question-circle fpbx-help-icon"
data-for="element1"></i>
    </div>
    <div class="col-md-9">
      <input type="text" class="form-control" id="
```



```

element1" placeholder="This gives an example of what to put in this field">
    </div>
</div>
<!-- This line is normally hidden because our help block itself is
hidden -->
<div class="col-md-12">
    <span id="element1-help" class="help-block fpbx-help-block"
>Help Text</span>
    </div>
</div>

```

Remember that any form element you want to error out you simply need to call "warnInvalid" like so (against any jqueryed element). This will error out the element until input has changed:

```
warnInvalid($("#element1"), "Message about why the data is/was wrong");
```

Validation Errors

```
<input type="text" class="form-control" name="name" id="name" data-
invalid="Error Message on Invalid required">
```

## Special Classes

btn-file

Used to wrap file uploads.

```

<span class="btn btn-default btn-file">
    Browse <input type="file" class="form-control" name="elementname"
id="elementname">
</span>
<span class="filename"></span>

```

delAction

Adds Confirmation to delete links.

```

<a href="?display=directory&action=delete&id='+value['id']+'" class="
delAction"><i class="fa fa-trash"></i></a>

```

## password-meter and toggle-password

Adds a password strength meter and a show hide toggle to password fields.

```
<div class="input-group">
  <input type="password" class="form-control password-meter" id="
secret" name="secret" value="<?php echo isset($secret)?$secret:''?>">
  <span class="input-group-addon toggle-password" id="pwtoggle" data-
id="secret"><i class="fa fa-eye"></i></a></span>
</div>
```

## clicktoedit

Framework 13.0.7.4+

Makes the field read only until the user clicks on it. This is used in password fields to prevent browsers from auto-populating inappropriately

```
<div class="input-group">
  <input type="password" class="form-control password-meter
clicktoedit" id="secret" name="secret" value="<?php echo isset
($secret)?$secret:''?>">
  <span class="input-group-addon toggle-password" id="pwtoggle" data-
id="secret"><i class="fa fa-eye"></i></a></span>
</div>
```

## Standard Icons

<http://fontawesome.github.io/Font-Awesome/icons/>

Purpose	Icon to use
Delete	<code>fa fa-trash</code>
List	<code>fa fa-list</code>
Add	<code>fa fa-plus</code>
Report	<code>fa fa-area-chart</code>
Close	<code>fa fa-times</code>
Settings	<code>fa fa-cog</code>

## View Functions

Available Framework 13.0.1alpha67

**⚠ This Module has been deprecated**

This module has been deprecated and may not work properly in this or future releases. This module is no longer maintained



**📘 title open** —

Help Text

**📘 title collapsed** +

**📘 title collapsed** —

Help Text

```
//show_deprecated($message="", $dismissible = true, $logit = false);  
//show_help($message, $title='', $collapse=false, $collapseable=true,  
$class='default')  
echo show_deprecated(_("This module has been deprecated and may not work  
properly in this or future releases. This module is no longer  
maintained"))?>  
echo show_help('Help Text', 'title open')?>  
echo show_help('Help Text', 'title collapsed',true)?>  
echo show_help('Help Text', 'title collapsed',true, true,"info")?>
```