Style Guide

# Fonts

For body text, we use a font-size of 84%, which corresponds to roughly 10 pixels. Though we target a size of 10 pixels, we use a percent based unit because some browsers have problems scaling layouts with pixel specified text.

Note that the font size and style does not cascade into inputs, buttons, or select in CSS, so we must set it manually. We also increase the font size to 100% or 12 pixels in order to make the options more readable.

## CSS

**body,** **table** **td** {

font-family: *Arial* *Unicode* *MS,* *Arial,* *sans-serif*;

font-size: *84%*;

}

**input,** **button,** **select** {

font-family: *Arial* *Unicode* *MS,* *Arial,* *sans-serif*;

font-size: *100%*;

}

**pre** {

font-family: *"courier new",* *courier*;

font-size: *84%*;

}

## Some useful links:

Describes the tra

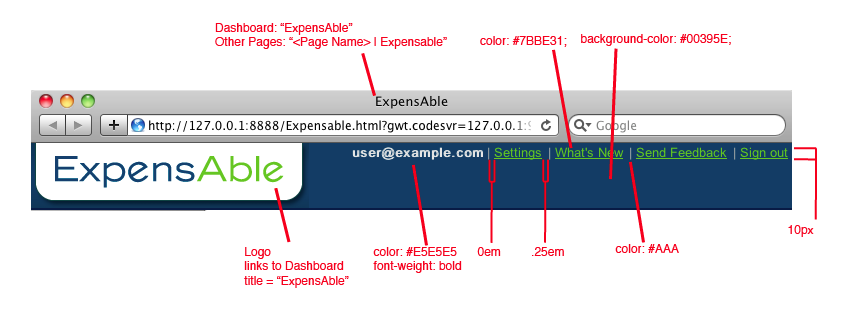
<http://kyleschaeffer.com/best-practices/css-font-size-em-vs-px-vs-pt-vs/>

Describes the tradeoffs for different sizing techniques.

<http://www.awayback.com/revised-font-stack/>

Suggests different “font stacks” or font choices based on the fonts that clients are likely to have.

# Header



## Features

1. **Logo:** The logo in the top-left corner should link to the main page. In the case of the ExpensAble site, this should link to the Dashboard.
2. **Email address:** Instead of displaying the user’s name, use the user’s email address. This way, if single user “Tony the reporter” has accounts for multiple companies on ExpensAble, he knows which account he is logged into.
3. **Settings:** links to the user settings page. This page is application dependent, but common features on this page are changing his name, password, and privacy settings.
4. **What’s New:** Provides ExpensAble with a means by which to notify users of new features. For more important changes, consider changing the color of this to red, or another color that will stand out.
5. **Send Feedback:** Provides an email address for customers to quickly and easily provide feedback on the application.
6. **Sign out:** Logs the user out of the application.

## HTML

<div class=*"{style.header}"*>

<span class=*"{style.user-info}"*>

<!-- Note: formating is intentional: span should start at EOL -->

<span class=*"{style.username}"* ui:field=*"username"*>user@example.com</span>|<span>

<a href=*"#"*>Settings</a></span>|<span>

<a href=*"#"*>What's New</a></span>|<span>

<a href=*"mailto:example@example.com"*>Send Feedback</a></span>|<span>

<a href=*"#"*>Sign out</a></span>

</span>

<a href=*"#"*><img title=*"ExpensAble"*

src=*"https://server.expensable.com  
 /EComImages/menu/EXP\_UI\_logo.gif"* border=*"0"*/></a>

</div>

## CSS

*.header* {

background: *#00395E*;

border-bottom: *2px solid #001E44*;

}

*.user-info* {

color: *#AAA*;

float: *right*;

}

*.username* {

font-weight: *bold*;

}

*.user-info* **a,** *.user-info* **span,** *.footer* **span** {

color: *#E5E5E5*;

padding: *0 0.25em 0 0*;

}

*.user-info* **a,** *.user-info* **a***:visited*{

color: *#7BBE31*;

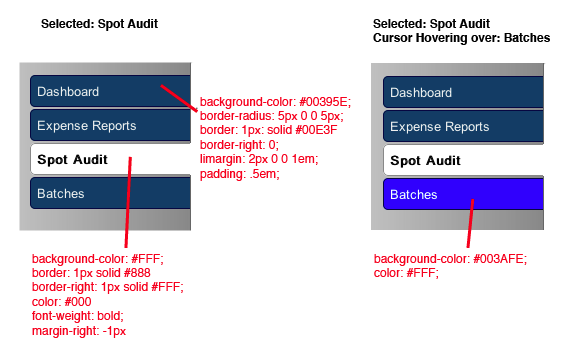
}

# Footer



For the page footer, we opted to use a hanging footer (a footer that always remains visible, even if the page requires scrolling). The benefit of this is that users can quickly find the Privacy Policy and Terms of Service. However, since these links are likely not to be commonly used by a user, it is acceptable put the footer at the bottom of the scrolled page.

# Vertical Tab Bar



In order to navigate through the web site, we use a vertical tab bar. This provides two major benefits. First, if we consider the navigation of the site as being a “left to right” process, the vertical tab bar shows the origin of the navigation process. Furthermore, the selected tab provides the user with a clear indication of where he is, so that he knows how to get back to it in the future.

# Uniform Resource Locator (URL)

## Each page should have its own URL

URLs should follow the rules of the Representational State Transfer (REST) architectural style. Accordingly, each page should have its own unique URL. By following this standard, pages can be bookmarked, and, if exposed to search engines, crawl-able.

Consider a site with a root url of: https://server.expensable.com

|  |  |  |
| --- | --- | --- |
|  | HTTP Method | URL |
| BAD | GET | /esweb/757234393406461292104544352/index.htm |
| GOOD | GET | /reports/1233 |
| GOOD | GET | /batches/1235 |
| GOOD | GET |  |

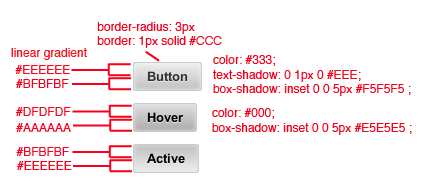
## Browser History should be managed by the browser

The application should **NEVER** manage its own history. By managing its own history, the application must provide its own back and forward buttons. Since a user is accustomed to using the back and forward buttons on the browser, having its own back and forward buttons (or even browser shortcuts) can confuse the user. Furthermore, without storing resource state on the URL when the user clicks the back and forward button, all state is lost. Instead, the application should use the DOM’s API for manipulating browser history.

## Relevant Material

1. Fielding, Roy. Chapter 5: Representational State Transfer (REST) <<http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm> >
2. Richardson, L, and S Ruby. *RESTful Web Services*. O’Reilly 2007.
3. Mozilla. “Manipulating the Browser History.” <<https://developer.mozilla.org/en/DOM/Manipulating_the_browser_history>>

# Buttons



## Features

Buttons should be consistent across all properties. Although color changes are valid for special cases, such as a call to action, disabling a button, or providing a sense of warning.

Buttons should provide a sense of feedback to the user. In the example above, when the user hovers over a button, the background and text color get darker, while when the user clicks on the button, the button reverses the original color gradient.

## HTML

<a class=*"button"*/>

## CSS

*.button* {

background: *#eee*;

background: *-moz-linear-gradient(0%* *100%* *90deg,* *#bfbfbf,* *#eee)*;

background: *-webkit-gradient(linear,* *0%* *0%,* *0%* *100%,* *from(#eee),* *to(#bfbfbf))*;

border: *1px* *solid* *#ccc*;

-moz-border-radius: *3px*;

-webkit-border-radius: *3px*;

border-radius: *3px*;

-moz-box-shadow: *inset* *0* *0* *5px* *#f5f5f5*;

-webkit-box-shadow: *inset* *0* *0* *5px* *#f5f5f5*;

box-shadow: *inset* *0* *0* *5px* *#f5f5f5*;

color: *#333*;

font-size: *100%*;

font-weight: *bold*;

line-height: *1*;

margin: *0.5em*;

padding: *.5em* *1em*;

text-align: *center*;

text-shadow: *0* *1px* *0px* *#eee*;

text-decoration: *none*;

}

*.button:hover* {

background: *#dfdfdf*;

background: *-moz-linear-gradient(0%* *100%* *90deg,* *#aaa,* *#dfdfdf)*;

background: *-webkit-gradient(linear,* *0%* *0%,* *0%* *100%,* *from(#dfdfdf),* *to(#aaa))*;

-moz-box-shadow: *inset* *0* *0* *5px* *#e5e5e5*;

-webkit-box-shadow: *inset* *0* *0* *5px* *#e5e5e5*;

box-shadow: *inset* *0* *0* *5px* *#e5e5e5*;

color: *#000*;

}

*.button:active* {

background: *#bfbfbf*;

background: *-moz-linear-gradient(0%* *100%* *90deg,* *#eee,* *#bfbfbf)*;

background: *-webkit-gradient(linear,* *0%* *0%,* *0%* *100%,* *from(#bfbfbf),* *to(#eee))*;

}

# Button Groups



If a button corresponds to an unrevertable action, it should be offset from the other buttons in the group in order to prevent the user from accidently clicking that button. In the figure above, notice how the “Delete” button is further from the “Export” button than the “Export” button is from the “Open” button.

Additionally, it may be wise to confirm such a change with a confirmation dialog.

# Table

# Macintosh HD:Users:dpurpura:Desktop:Screen shot 2010-12-11 at 11.36.28 AM.png

## More Information

The table provided is a GWT Cell Table widget, more information can be found about it from the links below.

<http://code.google.com/webtoolkit/doc/latest/DevGuideUiCellWidgets.html#celltable>

<http://gwt.google.com/samples/Showcase/Showcase.html#!CwCellTable>