



BRAND RESOURCES

A style and design guide for Centurylink® Model Controller *(for use by developers)*



CenturyLink®
Model Controller

TABLE OF CONTENTS

(click to go directly to each assigned page)

MODEL CONTROLLER FONT

TEXT CLASS MODIFIERS

CARDS

HEADERS

2ND-LEVEL NAV

BACKGROUNDS

BUTTONS

MODALS

Model Controller uses the open source Google font [Open Sans](#). Open Sans provides a clean, modern look for Model Controller with a wide variety of thicknesses. Open Sans is both a traditional print and web font rendered well in most modern browsers.

a few examples:

This is an example of Open Sans

* Style: Regular , Size: 24

This is an example of Open Sans

Style: Semibold , Size: 24

This is an example of Open Sans

Style: Bold , Size: 24

This is an example of Open Sans

Style: Extrabold , Size: 24

This is an example of Open Sans

Style: Italic , Size: 24

This is an example of Open Sans

Style: Lite , Size: 24

** Please note the styles listed are from www.google.com and not necessarily reflective of Model Controller's specific style class modifiers (see page 4).*



Model Controller has a specified set of text styles. Attach any of these css class modifiers to your text to achieve the desired affect.

examples

(all have Open Sans Size 12 as base style)

code

(Click to Copy & Paste)

This is muted text

THIS IS UPPERCASE TEXT

THIS IS MUTED UPPERCASE TEXT

This is thin text

This is standard text

This is medium text

This is bold text

This is xx-small text

This is x-small text

This is small text

This is regular text

This is large text

This is x-large text

This is xx-large text

`.text-muted`

`.text-uppercase`

`.text-muted-upper`

`.text-thin`

`.text-standard`

`.text-medium`

`.text-bold`

`.text-xxsmall`

`.text-xsmall`

`.text-small`

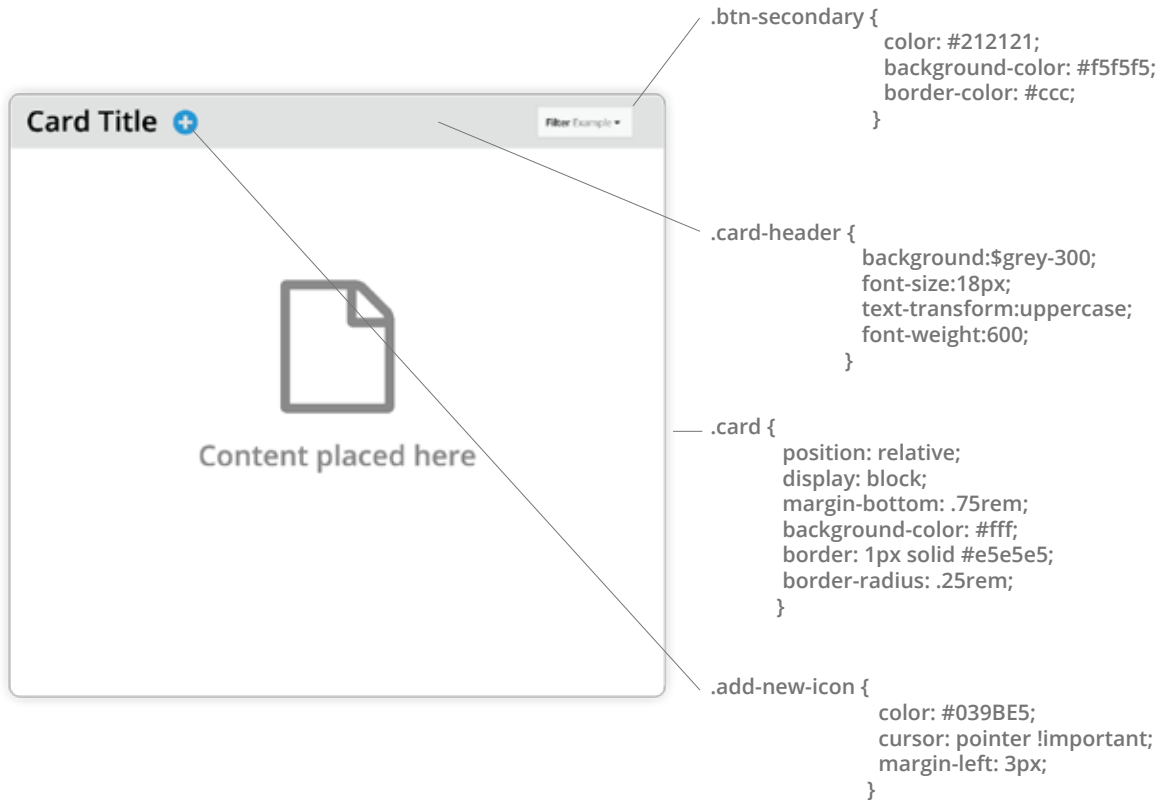
`.text-regular`

`.text-large`

`.text-xlarge`

`.text-xxlarge`

Model Controller uses cards to display varying types of information. Cards are a simple, packaged container with a specified set of styles almost always containing a header with a formatted title and sometimes other small action-oriented buttons and filters.

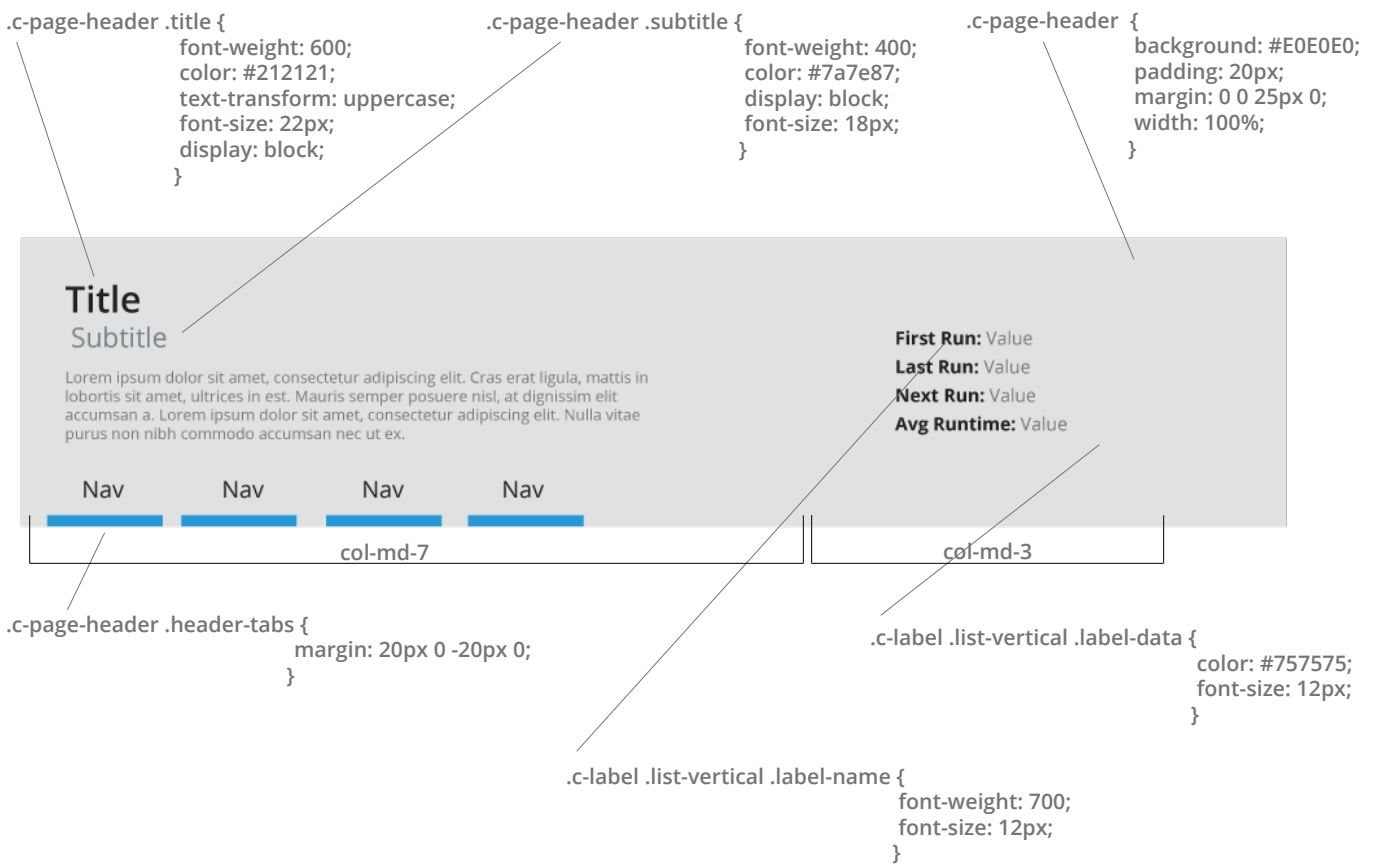


HTML CODE:

```

<div class="card">
  <div class="card-header">
    Card Title
    <i class="fa fa-plus-circle add-new-icon"></i>
    <button class="btn btn-secondary btn-sm dropdown-toggle" type="button" data-toggle="dropdown">
    <strong>Filter</strong>
    <em class="ng-binding">Example</em>
  </div>
  <i class="fa fa-file-o"></i>Content placed here
</div>
  
```

Model Controller uses a custom header found throughout the application, but most notably on the home dashboard, project, model and version specific pages. The header has a specified set of styles with varying information being displayed (execution stats, bio information) and can contain specially formatted navigation tabs.

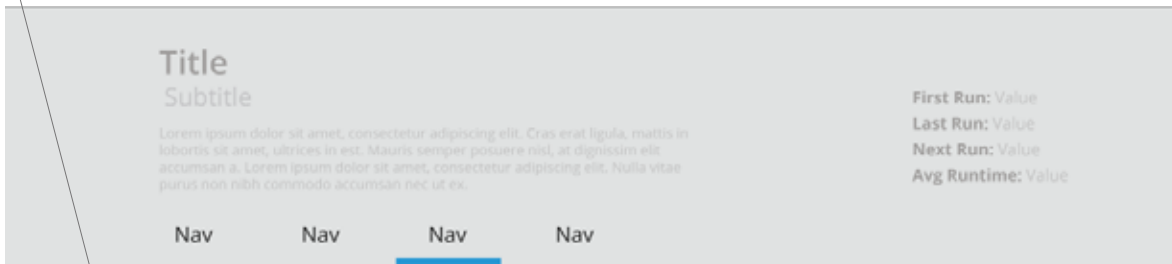


HTML CODE:

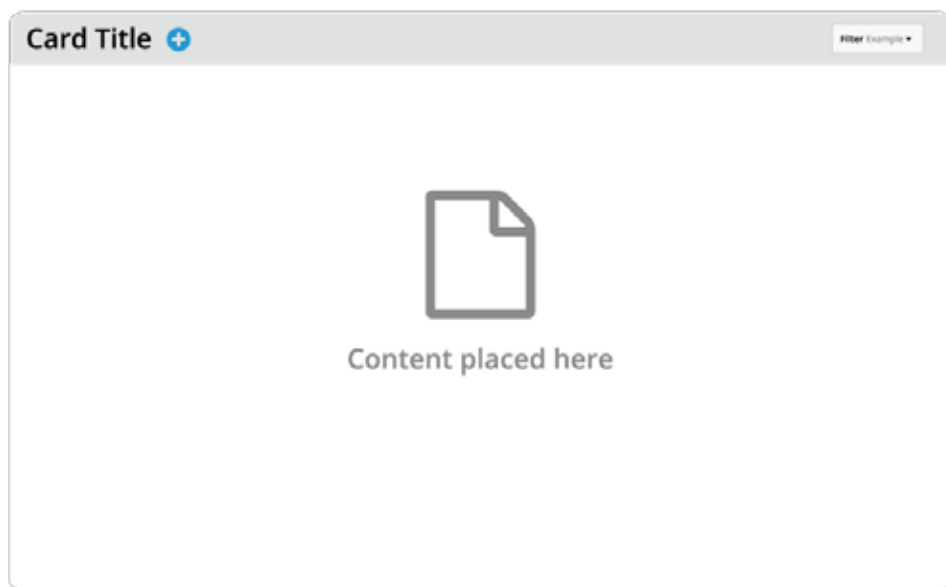
```
<div class="c-page-header-wrapper">
  <div class="c-page-header">
    <div class="row">
      <div class="col-md-7">
        <div class="title">Title</div>
        <div class="subtitle">Subtitle</div>
        <div class="bio">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras erat ligula, mattis in lobortis sit amet,
          ultrices in est. Mauris semper posuere nisl, at dignissim elit accumsan a. Lorem ipsum dolor sit
          amet, consectetur adipiscing elit. Nulla vitae purus non nibh commodo accumsan nec ut ex.</div>
      </div>
      <div class="col-md-3 c-label">
        <ul class="list-vertical">
          <li>
            <span class="label-name">First Run:</span><span class="label-data">Value</span>
          </li>
          <li>
            <span class="label-name">Last Run:</span><span class="label-data">Value</span>
          </li>
          <li>
            <span class="label-name">Next Run:</span><span class="label-data">Value</span>
          </li>
          <li>
            <span class="label-name">Avg Runtime Run:</span><span class="label-data">Value</span>
          </li>
        </ul>
      </div>
    </div>
    <div class="nav nav-pills header-tabs">
      <li class="nav-item">
        <a class="nav-link" ui-sref="" ui-sref-active="active">
          Nav
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" ui-sref="" ui-sref-active="active">
          Nav
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" ui-sref="" ui-sref-active="active">
          Nav
        </a>
      </li>
      <li class="nav-item">
        <a class="nav-link" ui-sref="" ui-sref-active="active">
          Nav
        </a>
      </li>
    </div>
  </div>
</div>
```

In the event of a second-level navigation beyond the formatted horizontal navigation tabs (see page 6), Model Controller will use a formatted vertical left-justified navigation.

```
.nav-stacked .nav-item {  
  display:block;  
  float:none;  
}
```



- Nav
- Nav
- Nav
- Nav










```
.nav-pills.nav-stacked .nav-link.active, .nav-pills.nav-stacked .nav-item.open .nav-link {  
  border-bottom:0;  
  border-left:5px solid #039BE5;  
}
```



Model Controller uses a custom set of primary and secondary colors from [Google's Material Design Color Guidelines](#). These specific colors and their variables can always be found in *colors.scss* (*NOTE: Always use the variables and not the hex values.*)

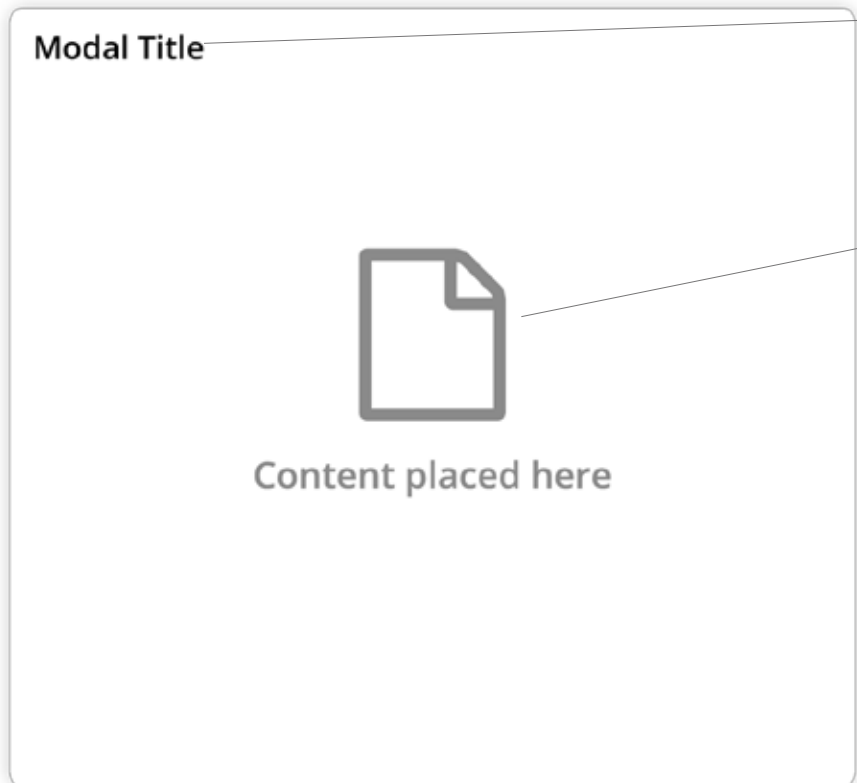
PRIMARY COLORS	SECONDARY HIGHLIGHTS
<p>HEX - 000000 RGB - 0 0 0 CMYK - 75 68 67 90 Variable - \$black</p> <p><i>examples: top bar navigation, left side navigation, text, icons</i></p>	<p>HEX - 2196F3 RGB - 33 150 243 CMYK - 71 34 0 0 Variable - \$blue-500</p> <p><i>examples: "running" status</i></p>
<p>HEX - 424242 RGB - 66 66 66 CMYK - 67 60 59 45 Variable - \$grey-800</p> <p><i>examples: breadcrumb</i></p>	<p>HEX - 4caf50 RGB - 76 175 80 CMYK - 72 5 95 0 Variable - \$green-500</p> <p><i>examples: "completed" status</i></p>
<p>HEX - E0E0E0 RGB - 224 224 224 CMYK - 11 8 9 0 Variable - \$grey-300</p> <p><i>examples: page headers, card headers</i></p>	<p>HEX - F44336 RGB - 244 67 54 CMYK - 0 89 84 0 Variable - \$red-500</p> <p><i>examples: "failed" status</i></p>
<p>Hex - f5f5f5 RGB - 245 245 245 CMYK - 3 2 2 0 Variable - \$grey-100</p> <p><i>examples: body background</i></p>	<p>HEX - 9e9e9e RGB - 158 158 158 CMYK - 41 33 33 1 Variable - \$grey-500</p> <p><i>examples: "stopped" status</i></p>

Model Controller uses a bootstrap's default button classes with the addition of a set of custom colors. In most instances, using **.btn-primary** is sufficient.

	✓	Background: #039BE5 (\$light-blue-600) Font: #ffffff
	✓	Background: #f5f5f5 (\$grey-100) Font: #212121 (\$grey-900)
	✓	Background: #009688 (\$teal-500) Font: #ffffff
	✓	Background: #00BCD4 (\$cyan-500) Font: #ffffff
	✓	Background: #ff9800 (\$orange-500) Font: #ffffff
	✓	Background: #f44336 (\$red-500) Font: #ffffff
	✗	Do not place any font awesome and/or custom icons next to the type.

Model Controller uses some very basic rules for using bootstrap's modal system:

1. A modal is appropriate whenever a  is present (*ie new project, new model...*)
2. A user should never have to scroll down the page. Keep the modal **short and simple**, otherwise, create and direct the user to a new page/view.



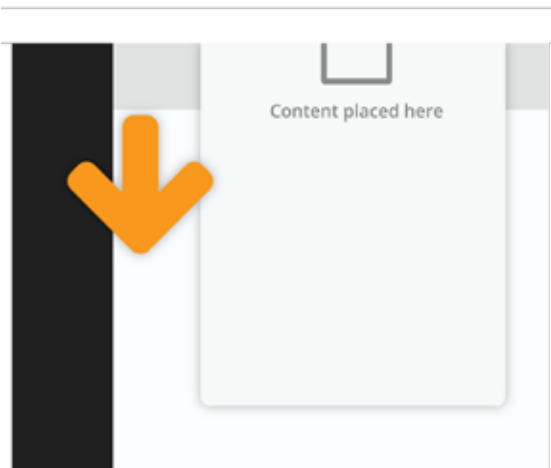
Modal Title

```
.modal-title {  
  font-weight:600;  
  font-size:24px;  
}
```

Some examples of *content*:

1. Input fields
2. File uploads
3. Radio buttons
4. Action buttons

Content placed here



Never make a modal long enough to where a user will have to scroll down the page.