Report layout and content customization guide

Version 0.1

Date 15 Oct 2022

Roborigger Dashboard

1. Summary

The report page is generated based on the HTML, CSS and JavaScript codes. All these three parts are integrated together into the template fill called template.html. The directory to access the template.html is:

*/Backend/setup/templates/template.html*

Additional templates can be added to this directory for easier access. Regarding the HTML template file, it has the standard HTML5 formatted code by passing the W3C HTML5 Validator. The structure of the template.html is:

*-HTML heading*

*-CSS style.*

*-JavaScript.*

*-HTML body.*

The template.html consists of two parts, the first part is the overlay page promoting user to enter the site details including site name, address, device number and contact information. Once the user clicks the “view PDF” button, the final report will be shown in the second part. In the second part, the user is able to export the report to pdf format by clicking the “save PDF” button. The saving process of the report is achieved by the pop-up dialog box.

1. JS script explanation

JavaScript is to help the HTML file to render charts correctly. This is because HTML files do not support using variables in the code. JavaScript helps the HTML to load the part C data individual records by using AJAX call. Moreover, in order to correctly link and load charts from the server, the system uses two main prefixes to differentiate charts and static images such as logo and background images:

*chart\_DIR is used for directory storing charts*

*Icon\_DIR is used for directory storing static images*

* 1. Loading charts

The following table shows file names, display names, and variable names for individual charts:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index: | Chart name (display) | ChartName (fileName) | VariableName | W:H ratio |
| 1 | Lifts Per Day | lifts\_per\_day.png | lifts\_per\_day | 12:5 |
| 2 | Lifts Per Week Last 8 Weeks | lifts\_per\_week\_last\_8\_weeks.png | lifts\_per\_week\_last\_8\_weeks | 7:5 |
| 3 | Average Time Per Lift | average\_time\_per\_lift.png | average\_time\_per\_lift | 12:5 |
| 4 | Average Time Per Lift Last 8 Weeks | average\_timep\_per\_lift\_last\_8\_weeks.png | average\_timep\_per\_lift\_last\_8\_weeks | 7:5 |
| 5 | Lift Time Per Day | lift\_time\_per\_day.png | lift\_time\_per\_day | 12:5 |
| 6 | Lift Time Per Week Last 8 Weeks | lift\_time\_per\_week\_last\_8\_weeks.png | lift\_time\_per\_week\_last\_8\_weeks | 7:5 |
| 7 | Temperature / Wind Speed / Rainfall | temperature\_wind\_speed\_rainfall.png | temperature\_wind\_speed\_rainfall | 12:5 |
| 8 | Mass Lifted Per Hour Last 8 Weeks | mass\_lifted\_per\_hour\_last\_8\_weeks.png | mass\_lifted\_per\_hour\_last\_8\_weeks | 7:5 |
| 9 | Average Mass Per Lift Last 8 Weeks | average\_aass\_per\_lift\_last\_8\_weeks.png | average\_mass\_per\_lift\_last\_8\_weeks | 8:5 |
| 10 | Mass Lifted Per Week Last 8 Weeks | mass\_lifted\_per\_week\_last\_8\_weeks.png | mass\_lifted\_per\_week\_last\_8\_weeks | 8:5 |
| 11 | Lifts Per Month | lifts\_per\_month.png | lifts\_per\_month | 6:5 |
| 12 | Lift Time Per Month | lift\_time\_per\_month.png | lift\_time\_per\_month | 6:5 |
| 13 | Mass Lifted Per Month | mass\_lifted\_per\_month.png | mass\_lifted\_per\_month | 6:5 |
| 14 | Loading Level Percentage | loading\_level\_percentage.png | loading\_level\_percentage | 18:5 |
| 15 | Percentage of Time with Load on Hook | percentage\_of\_time\_with\_load\_on\_hook.png | percentage\_of\_time\_with\_load\_on\_hook | 12:5 |
| 16 | Operating Vs. Idle Last 8 Weeks | operating\_vs\_idle\_last\_8\_weeks.png | operating\_vs\_idle\_last\_8\_weeks | 7:5 |
| 17 | Time Per Lift Exceedance | time\_per\_lift\_exceedance.png | time\_per\_lift\_exceedance | 12:5 |
| 18 | Time Per Lift Histogram | time\_per\_lift\_histogram.png | time\_per\_lift\_histogram | 7:5 |
| 19 | Time Since Last Lift Exceedance | time\_since\_last\_lift\_exceedance.png | time\_since\_last\_lift\_exceedance | 12:5 |
| 20 | Time Since Last Lift Histogram | time\_since\_last\_lift\_histogram.png | time\_since\_last\_lift\_histogram | 7:5 |

Once the page is loaded correctly, JavaScript will load images based on details from the above table. Every image container in the html body will have its unique id, these images will be added to the corresponding containers.

To get the best looking of the report, it is highly recommended to generate charts based on the correct width: height ratio values listed on the right-hand side of the above table.

* 1. Loading dynamic data from JSON

JSON file is generated by “plotgenerator.py”, it consists of two parts: part C data and overview data.

* + 1. Part C data

In the actual production environment, the records included in part C may be varied based on different devices. Therefore, the framework of part C has not coded as the static code in the HTML body part. The system used XML call to get the pre-generated JSON file from the server and load the data accordingly.

The JSON file for part C contains the following values:

|  |  |
| --- | --- |
| Index | DataName |
| 0 | Timestamp |
| 1 | Battery |
| 2 | Control Mode |
| 3 | Coolant Temperature |
| 4 | Event |
| 5 | Load |
| 6 | Location |
| 7 | Motor Temp |
| 8 | Thumbnail |
| 9 | Load\_col\_int |
| 10 | Time |
| 11 | Month |
| 12 | Date |
| 13 | Week |
| 14 | Day |
| 15 | LoadStatus |
| 16 | TimeBefore |
| 17 | TimeAfter |
| 18 | mode |
| 19 | LiftTime |
| 20 | TimePerLift |
| 21 | WaitingTime |
| 22 | Time since Last Lift |
| 23 | InAir |

All of the above values can be displayed based on the actual code. To update the displaying details in part C, go to “Part C Script XML” part in JavaScript part. “title 1-8” inside the if statement are used to display data. To make sure data are displayed correctly, please update the display label for the data correspondingly.

If the JSON file cannot be loaded correctly due to network or reference errors, the error message will be displayed as an alert dialog box.

* + 1. Overall data

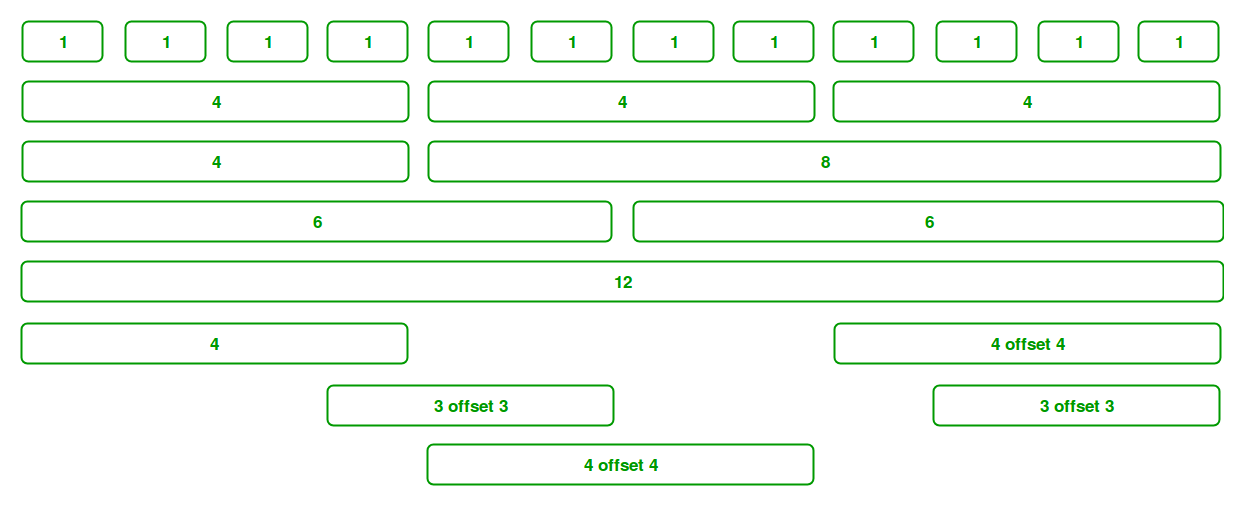
The JSON file for this part contains the following data:

|  |  |
| --- | --- |
|  | DataName |
| 1 | Total\_lift\_weight |
| 2 | Total\_lift\_count |
| 3 | Total\_lift\_time |
| 4 | Total\_days |
| 5 | RDO\_sunday\_holiday |
| 6 | Weather |
| 7 | Days\_available |
| 8 | Day\_used |
| 9 | weekends |
| 10 | Start\_date |
| 11 | End\_date |

Part of these data are displayed on the top of the first page. As the same as the part C data, this part of the report can be modified with different data names.

1. CSS style explanation

To keep the simplicity of the template.html file, the styling of the report is mainly achieved by using the existing w3 CSS file. The CDN link is included in the HTML head part. All containers in the HTML body part are formatted by using bootstrap library. Bootstrap library divides the whole width into 12 columns like this:



All charts across the full width of the page uses “col-xs-12” class, the following table shows the details of classes used to achieve different layouts:

|  |  |  |
| --- | --- | --- |
| Layout | Bootstrap class name | Sample |
| Full width | col-xs-12 |  |
| 3/3/3 | Col-xs-3 |  |
| 7/5 | col-xs-7, col-xs-5 |  |
| 6/6 | col-xs-6 |  |

The additional styling in the CSS part is used to apply minor adjustments to the layout of the elements. The most important CSS style in the template.html is:

*.reduced\_img\_height{*

*height: 180px}*

This is used to adjust the height of all charts in the report. You can update this value if you want to scale up the charts size.

1. To change layout

You can change the order of containers inside the HTML body. To rearrange charts consistently, you must move the container with class=”row” and its content at once to keep the correct formatting.