

Web Based User Interface Framework

Saroja Rani Polamarasetty

*Chaitanya Engineering College,
Visakhapatnam*

Abstract— The development time of creating Web Applications has decreased a lot compared with the situation a decade before. But still creating Web interface that is cross browser compatible and having same behaviour across browsers is a challenge. Using Class-UI we can create web applications easily at the same time preserving flexibility of even creating facebook like user interface without worrying about cross browser issues.

Keywords— Cross Browser, User Interface. GRID layout, Framework

I. INTRODUCTION

Framework is an abstraction in which software providing generic functionality can be selectively changed by additional user-written code, thus providing application-specific software. A good framework makes the life of a programmer easy by providing him/her a utility to focus on specific problem than to get bogged up with the details of internals.

Class-UI is a framework for Web User Interface written in HTML, CSS, JavaScript using SASS (CSS Compiler), Compass, and JQuery. Our project aims at facilitating the creation of Web based User Interface with ease and at the same time retaining flexibility to change any aspect of default behaviour.

Another important feature of Class-UI is support for font-awesome. We made it a default integration into Class-UI. Through this many standard symbols could be inserted.

It was mainly aimed at minimizing the use of JavaScript for user interface rendering, and make use of CSS wherever possible. We ensured the flexibility of creating one's own interface using Class-UI, at the same time providing utilities to reuse standard functionalities and hence decreasing one's work time and preserving flexibility at the same time.

II. TECHNOLOGIES

Technologies used for the creation of Class-UI framework are :

- SASS (CSS compiler)
- Compass (SASS project manager)
- JQuery (Javascript and DOM framework)

A. SASS

SASS is a CSS Compiler that overcomes the limitations of CSS like support for variables, nested styles, functions, reusable components etc. SASS facilitates the creation of calculated layouts. Without SASS creation of calculated layouts becomes very cumbersome.

SASS provides functionality which is impossible to do with CSS alone.

SASS Features includes :

- Variables
- Nesting Styles
- Partials
- Import
- Mixins
- Extend/Inheritance
- Operators

Files with scss extension are sass files.

B. Compass

Compass is a SASS project manager that automates the task of compilation of SASS files to CSS. It also provides default modules that could be reused in any project if required like cross browser CSS3 extensions etc.

Highlights of COMPASS are :

- Experience cleaner markup without presentational classes.
- It's chock full of the web's best reusable patterns.
- Compass mixins make CSS3 easy.
- Create beautiful typographic rhythms.
- Download and create extensions with ease.

C. JQuery

JQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

Simply JQuery is type less do more framework for javascript and DOM functionality of Web pages.

D. Reset.css

The goal of a reset stylesheet is to reduce browser inconsistencies in things like default line heights, margins and font sizes of headings, and so on. The general reasoning behind this was discussed in a May 2007 post, if you're interested. Reset styles quite often appear in CSS frameworks, and the original "meyerweb reset" found its way into Blueprint, among others.

The reset styles given here are intentionally very generic. There isn't any default colour or background set for the body element, for example it should be tweaked, edited, extended, and otherwise tuned to match your specific reset baseline. Filled in our preferred colours for the page, links, and so on.

In other words, this is a starting point, not a self-contained black box of no-touchiness.

III. SECTIONS

Sections define the different parts of the framework that the project is divided into.

A. Scaffolding

Global styles for the webpage to reset styles and background, add default customized styles for links, grid system, etc.

B. Base CSS

Styles for common HTML elements like typography, code, tables, forms, and buttons. Also includes Glyphicons, a great little icon set.

C. Components

Basic styles for common interface components like tabs and pills, navbar, alerts, page headers, and more.

D. JavaScript Plugins

Similar to Components, these JavaScript plugins are interactive components for things like tooltips, popovers, modals, and more. It also includes extended form validation providing powerful way of validating forms.

IV. FEATURES

Class-UI by default provides many functionalities that simplify the life of web developer. Here is the comprehensive list of features currently supported by Class-UI. This isn't final list. Features can be added to it at any time retaining the flexibility, the design principle of Class-UI.

- Button
- Alert
- Dropdown
- Well
- Navbar
- Grid layout
- Flash
- Form (with automated form validation)
- Tables
- Alerts
- Popover
- Utilities

A) Dropdowns

Configurable dropdown functionality that changes its behaviour according to the context where it is defined. Here implies changes its render behaviour according to the markup on fly.

B) Utilities

Utilities provide a way to change the behaviour of an element by just adding classes to an element. Classes in utilities include most common recurring patterns in a web page design like floats, clear, alignment, font, etc.

C) Wells

Wells are easy way of beautifying the Elements by having configurable background and colour by just changing of classes. Wells are mainly used to take up more space like tiles in webpages hence highlighting an element in webpage.

D) Navbars

Navbar in Class-UI provides innate flexibility by applying the semantics of a table internally. Navbars in Class-UI also provide many components like dark cells, light cells etc.

E) Tables

Tables are an excellent way of representing structured data. HTML has its own markup of defining tables and its styles. Class-UI defines default styles that defines design for different types of tables.

F) Grid layout

Grid Layout provides innate flexibility of defining layouts in Class-UI. Its 940px grid system is unique to classUI where the columns are divided into 20 columns.

The main idea behind grid-based designs is a solid visual and structural balance of web-sites you can create with them. Sophisticated layout structures offer more flexibility and enhance the visual experience of visitors. In fact, users can easier follow the consistency of the page, while developers can update the layout in a well thought-out, consistent way. However, it's quite hard to find your way through all the theory behind grid systems: it isn't easy at all. Some important notions and related key-facts can help to learn basics and keep essential techniques in mind.

G) Flash

Flash component provides the illusion of a window inside webpage. It also adds controls like close, move etc by just adding classes configurably...

H) Forms

Form validation is the most important part of any web site. Validating forms include lots of javascript with conditional checks and notifying user about errors if any or success if valid. Class-UI has dedicated an extensive library for form validation itself where a user can validate forms by just adding the validation criteria to the form element. And rest is taken care by Class-UI.

V. PROJECT LAYOUT

The file structure of Class-UI is very simple. It comprises of one main.css file which is the main Cascading Style sheet that beautifies the web application. It is the compressed one built using SASS compiler.

“js/” folder consists of components of Class-UI like flash, dropdown, form etc. that adds dynamic behavior to the web page.

Class-UI/

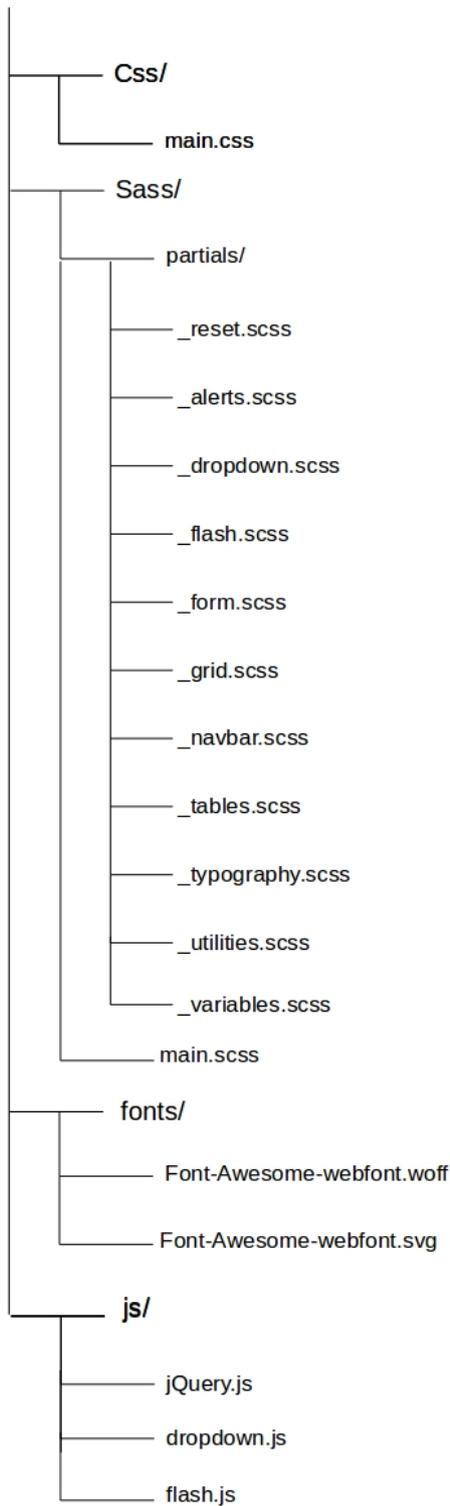


Fig 1. File Structure of the project.

The files of real concern in production environment are the folders css, js, and fonts. Sass folder includes the development files of cascading style sheets. In production environment there's no need of having sass folder, only compiled css files are needed.

VI. EXAMPLES

Here the examples and sample screenshots of our work is shown.

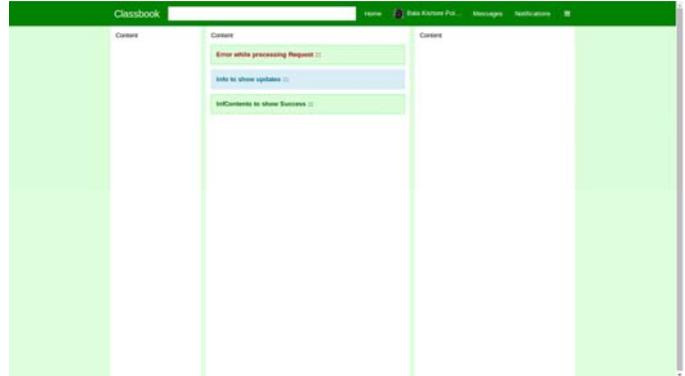


Fig 2 Sample webpage created with Class-UI framework.

Even a very complicated layouts of webpages could be made easy using this framework.



Fig 3. Changed look by just configuring the framework.

By configuring the colours of the framework and recompile the project, we can even change the look of the whole website!!!

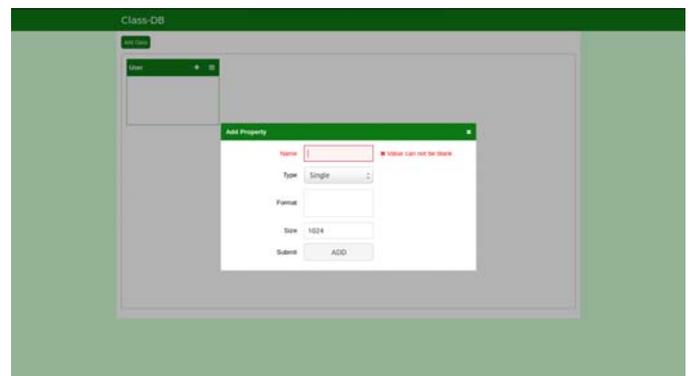


Fig 4. Form validation using Class-UI framework.

Form validation is made easy by just adding constraints to attributes where our framework takes care of every other thing.

No need of having sass folder, only compiled css files are needed.

CONCLUSION

Class-UI framework provides a very robust way of creating webpages that works identically in all browser. It also reduces the time for creating reusable components again and again. It also provides many utilities that could be used for any type of project irrespective of the framework. It is also very configurable that helps in changing the behaviour of the framework as per the needs.

REFERENCES

- [1] Reset CSS for cross browser elements normalization : <http://www.cssreset.com/>
- [2] Sass for CSS styling : <http://sass-lang.com/guide>
- [3] Compass for creating SASS projecsts : <http://leveluptuts.com/tutorials/compass-tutorials>
- [4] Grid layout : <http://css-tricks.com/almanac/properties/g/grid/>
- [5] Grid layout framework sample : <http://960.gs/>
- [6] JQuery : <http://jquery.com/>
- [7] JQuery UI : <http://jqueryui.com/>