

はじめに

※この記事は、「RUNTEQ Advent Calendar 2019」、19日目の記事です。

先日、社内のRailsエンジニアと話していたときに、「AdminLTEなんかもっといい感じに使えたいんだよね」というお悩み相談を受けました。

普段、Webサービス・アプリのUIデザインを仕事としてやっているのですが、管理画面の制作は、エンジニアの誰かが良しなに対応してくれているような状況が多いです。

この記事では、AdminLTEを使用する際に持っていると助かりそうな知識と、ちょっとしたテクニックをまとめてみました。

デザイナーがいないプロジェクトでも、CSSフレームワーク等を用いてせっせと管理画面をつくってくれている、最高のエンジニアの方々のお役に立てたら最高です。

AdminLTEとはなにか

AdminLTEは、Bootstrapベースでつくられた、管理画面作成などに特化したCSSフレームワークです。

これから使うならバージョン3以降がおすすめです。

骨組みはBootstrapで、それにスタイリングを加えて、サンプルも提供してくれているものがAdminLTEと考えてもらえると良いかなと思います。

AdminLTE 3

Alexander Pierce

Dashboard

Widgets

Layout Options

Charts

UI Elements

Forms

Tables

Simple Tables

DataTables

jsGrid

EXAMPLES

Calendar

Gallery

Mailbox

Pages

Extras

MISCELLANEOUS

HomeContactSearch

Simple Tables

Home / Simple Tables

Bordered Table

#	Task	Progress	Label
1.	Update software	<div></div>	55%
2.	Clean database	<div></div>	70%
3.	Cron job running	<div></div>	30%
4.	Fix and squish bugs	<div></div>	90%

Simple Full Width Table

#	Task	Progress	Label
1.	Update software	<div></div>	55%
2.	Clean database	<div></div>	70%
3.	Cron job running	<div></div>	30%
4.	Fix and squish bugs	<div></div>	90%

Striped Full Width Table

#	Task	Progress	Label
1.	Update software	<div></div>	55%
2.	Clean database	<div></div>	70%
3.	Cron job running	<div></div>	30%
4.	Fix and squish bugs	<div></div>	90%

Condensed Full Width Table

#	Task	Progress	Label
1.	Update software	<div></div>	55%
2.	Clean database	<div></div>	70%
3.	Cron job running	<div></div>	30%
4.	Fix and squish bugs	<div></div>	90%

Responsive Hover Table

ID	User	Date	Status	Reason
----	------	------	--------	--------

AdminLTE 3の画面サンプルはこちら。

<https://adminlte.io/themes/dev/AdminLTE/index.html>

これはv3.0.0の状態で作られていて、Bootstrapはv4.3.1、FontAwesomeはv5.11.2が内包されています。

コード一式を手元にダウンロードしたい場合は、GitHubリポジトリからも可能です。

<https://github.com/ColorlibHQ/AdminLTE/releases/tag/v3.0.0>

こんなお悩みありませんか？

いい感じといわれても、ズバッと一言でお答えするのが難しかったので、いったん悩みを集めるところからはじめてみました。

飲み会や休憩中を狙ってヒアリングした中で出てきたものや、コードをのぞいて困ってそうだなあと思った事柄は大体こんな感じでした。

お悩み

- いっぱいサンプルがあるけど、どれを使えばいいかわからない
- とりあえずスマホの表示が崩れない状態になりたい
- フッターが欲しいとか途中で言われてかなしい
- ページのタイトルをどうつければいいかわからない
- サイドメニューがなんかいい感じにならない
- テーブルにめっちゃボタンが並んでてやばい
- 入力項目が多すぎてめっちゃ大変に見える
- たまに要素間の余白がなくなったりする
- 数字がみづらい
- ちょっとだけカスタマイズしたいんだけどどうしよう

個別に答えていくのもできるかもしれませんが、方針やテクニックを知っていれば、解決できそうな予感がします。

あわよくば応用もできるのではと思い、いったんまとめ直してみました。

お答え

- 基本構造を知る
- 便利なclassを使いこなす
- 導線のつくりかたを統一する
- そのページでやれることを絞る
- 項目のグループをつくる
- 上書き用のCSSファイルをつくる

解説

順番にみていきます。

基本構造を知る

AdminLTEの構造

AdminLTEのデフォルトサンプルは、ヘッダー、サイドバー、コンテンツ、フッターで構成されています。

```
body
  .wrapper
    nav.main-header
    aside.main-sidebar
    .content-wrapper
      section.content-header
        .container-fluid
      section.content
        .container-fluid # ここらへんにグリッドレイアウトの骨組みが入っていく
    aside.control-sidebar # サンプルではカスタマイズ用のサイドバー
    footer.main-footer
    #sidebar-overlay # スマホサイズでサイドメニュー閉じる用のオーバーレイ
```

この形を崩さなければ、基本構成はほぼ完成です。

Bootstrapのグリッドレイアウト

コンテンツ部分はBootstrapのグリッドレイアウトで組みます。グリッドは12分割です。rowとcolの組み合わせでレイアウトをつくっていくのですが、入れ子にすることもできます。

```
.container-fluid
  .row
    .col-md-6 # 全体の50%
    .row
```

```
.col-md-6 # 全体の50%の50%=25%
.row
.col-md-6
.col-md-6
```

いったん `.col-md-xx` だけ使えば大丈夫です。が、横幅によって組み方を変えたい場合はオプションがあったりもします。

-	Extra small	Small	Medium	Large	Extra large
Breakpoint	<576px	≥576px	≥768px	≥992px	≥1200px
Class	<code>.col-xs-xx</code>	<code>.col-sm-xx</code>	<code>.col-md-xx</code>	<code>.col-lg-xx</code>	<code>.col-xl-xx</code>

その場合は連結させてく感じになります。

```
.col-md-6.col-lg-4
```

使い所としては、カード型UIの一覧部分などで、列を増減させたい場合がわかりやすいかなと思います。

便利なclassを使いこなす

AdminLTE自体にも様々なコンポーネントが用意されています。

ただ、骨組みであるBootstrapには、borderをつけたり余白をつけたりするための便利classが存在して、細かい調整も可能です。

例えば、borderだったら `.border-xx` 。余白だったら `.mt-xx` や `.pr-sm-xx` のようなものです。他には、`text-right` なんかもよく使います。

このあたりを組み合わせると、めっちゃよく使うテーブルのレイアウトを、サンプルよりもうちょっと良い感じにできます。

テーブルレイアウト

Bordered Table

実案件でよくみかけるのは、ボーダー付きのテーブルです。

Bordered Table

#	Task	Progress	Label
1.	Update software	<div><div></div></div>	55%
2.	Clean database	<div><div></div></div>	70%
3.	Cron job running	<div><div></div></div>	30%
4.	Fix and squish bugs	<div><div></div></div>	90%

«

1

2

3

»

```
.card
  .card-header
    .card-title
    .card-tools
  .card-body
    table.table.table-bordered
      thead
      tbody
  .card-footer
```

ただ、このままだとレスポンス対応してません。

そんなときは `.card-body` に `.table-responsive` をつけるといい感じになるかと思いきや、上部の線がいなくなってしまう。

Bordered Table

#	Task	Progress	Label
1.	Update software	<div><div></div></div>	55%
2.	Clean database	<div><div></div></div>	70%
3.	Cron job running	<div><div></div></div>	30%
4.	Fix and squish bugs	<div><div></div></div>	90%

«

1

2

3

»

```
.card
  .card-header
    .card-title
    .card-tools
  .card-body.table-responsive
    table.table.table-bordered
      thead
      tbody
  .card-footer
```

そんなときは、こうするといい感じです。

```
.card
  .card-header
    .card-title
    .card-tools
  .card-body.table-responsive
    table.table.table-bordered.border-top
      thead
      tbody
  .card-footer
```

良い感じのテーブルができました。

Bordered Table			
#	Task	Progress	Label
1.	Update software	<div><div></div></div>	55%
2.	Clean database	<div><div></div></div>	70%
3.	Cron job running	<div><div></div></div>	30%
4.	Fix and squish bugs	<div><div></div></div>	90%

« 1 2 3 »

これを鬼のようにコピペすると、管理画面はほぼ完成です。

あとはちょっとしたテクニックですが、数値を表示するときは、小数点以下の桁数を揃えて `.text-right` をつけると見やすくなります。

Bordered Table		
#	Item	Price
1.	Update software	150.00
2.	Clean database	2999.99
3.	Cron job running	300.00
4.	Fix and squish bugs	1000.00

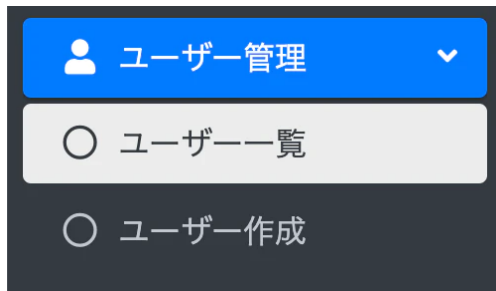
« 1 2 3 »

```
table.table.table-bordered.border-top
thead
  tr
    th.text-right
tbody
  tr
    td.text-right
```

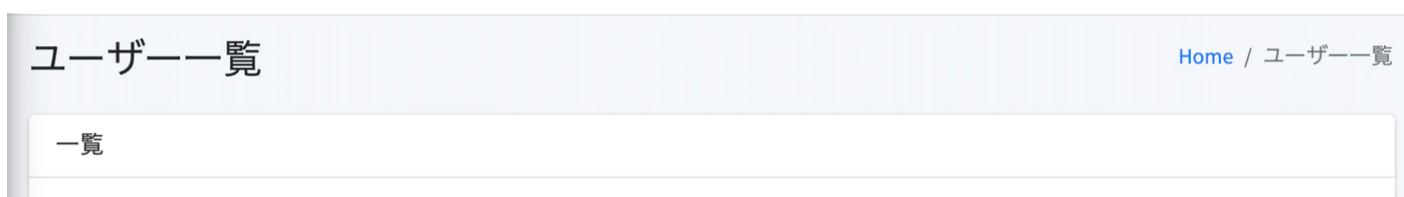
導線のつくりかたを統一する

いったんCRUDどおりにやって悩まないのが良いと思います。

サイドメニューに、ユーザー管理という区分を設けて、一覧・作成を配置します。



コンテンツ部分はサンプル通りに、タイトル、パンくず、コンテンツを並べて、このレイアウトを使い回すつもりでやりましょう。



基本の骨組みを省こうとすると、別の調整が必要になるので、いったんは教科書どおりに使うべきです。

そのページでやれることを絞る

一覧ページじゃなくて、詳細ページにだけあればいいものをトルという方針でやるのが定石です。

一覧ページでは詳細・編集ボタン、詳細ページでは編集・複製・削除ボタンのような区別をします。

IDやタイトル名にリンクを貼ると、詳細ボタンを省略できて、さらにアクション部分がスッキリします。

※あくまでも指針なので、複製・削除をすることがものすごく多いのであれば、一覧ページにも置くべきではあります

一覧

#	氏名	進捗	アクション
1.	内田 勲	<div><div></div></div>	<button>編集</button>
2.	菅原 英一	<div><div></div></div>	<button>編集</button>
3.	Yukihiro Matsumoto	<div><div></div></div>	<button>編集</button>

[«](#) [1](#) [2](#) [3](#) [»](#)

もしリンクってことがわからない。。。などと言われたら、`style="text-decoration: underline;"` と無理やり書いたり、`.text-decoration-underline` というclassを自作したりしましょう。（`.text-decoration-none` は存在する）

ボタンが増えすぎてデータ表示領域がなくなるよりはだいぶマシです。

項目のグループをつくる

例えば、ユーザー新規登録ページがあったとします。単純に項目を並べると、こんな感じになります。

ユーザー新規登録

氏名

氏名

ニックネーム

ニックネーム

Githubアカウント名

Githubアカウント名

メールアドレス

メールアドレス

パスワード

パスワード

権限

権限

受講期

受講期

一日当たりのコミット時間

一日当たりのコミット時間

備考

備考

Submit

入力項目などが多数ある場合は、似たものを近づけてグループ化してあげると、心理的な負担が和らぎます。

そのグループごとに、見出しをつけるといい感じです。ページ自体は長くなってしまうのですが、小さいゴールがあると、ツラさが少し和らぎます。

プロフィール

氏名

ニックネーム

ページが長くなるのがどうしてもツライ場合は、レイアウトを変えると、多少短くはできます。

プロフィール

氏名

ニックネーム

もう少し使いやすさも意識するなら、入力エリアは入力量に近い見た目にするといいです。ゴールや負荷が見えたほうが、タスクには取り組みやすくなるためです。

プロフィール

氏名

ニックネーム

上書き用のcssファイルをつくる

フォントの指定を変えたり、ちょっとした色味の変更、新しいパーツの追加がしたくなるときがあったりします。

そんなときは、大元のcssをいじるのではなく、新規で上書き用のファイルをつくるといいです。

assets内に `_override.scss` のようなファイルをつくるのがおすすめです。

bodyに `.override` というclassをつけておくと、セレクトの詳細度があがるので、上書きがしやすくなる利点もあります。

※ `.override` じゃなくて、`.custom` とか `.patch` とかでもいいかも。

```
body.override
```

```
...
```

```
.container
```

```
.override {
```

```
  .container {
```

```
    ...
```

```
  }
```

```
}
```

AdminLTEやBootstrapのドキュメントをみて、使いたいコンポーネントがなかなか見つからなかった場合は、最悪自分でHTML&CSSを書いちゃっていいかなとも思います。

その場合、プロジェクトの頭文字などを接頭辞としてつけておくと、class名かぶりを避けやすくなります。（qiitaならqtなど。）

```
.qt-container {
```

```
...
```

```
}
```

まとめ

というわけで、AdminLTEを使うときに知っていると良さそうな事柄をまとめてみました。

- 基本構造を知る
- 便利なclassを使いこなす
- 導線のつくりかたを統一する

- そのページでやれることを絞る
- 項目のグループをつくる
- 上書き用のcssファイルをつくる

「俺たちは雰囲気AdminLTEを使っている」とは思いますが、ちょっとした知識を武器にして、もう少しだけでも良い雰囲気です。

その他

他にも何か質問がきたり、思いついたりしたら、ここに簡易的なお答えを追記していきます。

- ダッシュボードにはとりあえずサイトマップ的なものを置いておくとも便利かも
 - 不要ならそもそもつukらないのが吉
- layoutファイルにはどこまで記述するか
 - .content-wrapper までで、その先は個別のviewファイルに分けるのがいいかなあと思います

おまけ

なぜAdminLTEを使うかの補足

いろいろ理由は並べられるけど、とにかく秒でつukり切ることが大事です。

サービスやプロダクトは、管理画面自体をつukりこんでも、ユーザーへの提供価値に直接つなげづらいことが多いです。単純に、エンジニアやデザイナーがじっくりと時間をかけてつukる余裕がなかったりもします。

少ない時間でなんとか最低限の使い勝手も担保したいけれど、バックエンド・サーバーサイドのエンジニアは他にもやることがたくさんあって、CSS、JavaScriptの習得まで手が回りづらいという状況もある気がします。

そんなときにBootstrapやAdminLTEのように、パーツの組み合わせだけでつくりきれるフレームワークが重宝されるのだと思います。

ただ、管理担当者が、素早いオペレーションをできることには十分に価値があるので、いい感じに手軽に管理画面をつくりたい俺たちにとって、AdminLTEのようなものは最高のCSSフレームワークです。

レイアウトの指針

レイアウトの4つの基本原則だけでも知っていると、さらに応用がききそうなので、少しだけふれておきます。

原則は、近接、整列、反復、対比の4つです。

- ①近接→関連があるものは近づける、関連がないものは余白をつける
- ②整列→端をそろえる
- ③反復→繰り返す
- ④対比（コントラスト）→違うものには差をつける

きちんと知りたい方は、ノンデザイナーズデザインブックという書籍を読むと最高です。