

# かっこいい管理画面デザインの適用(前編)

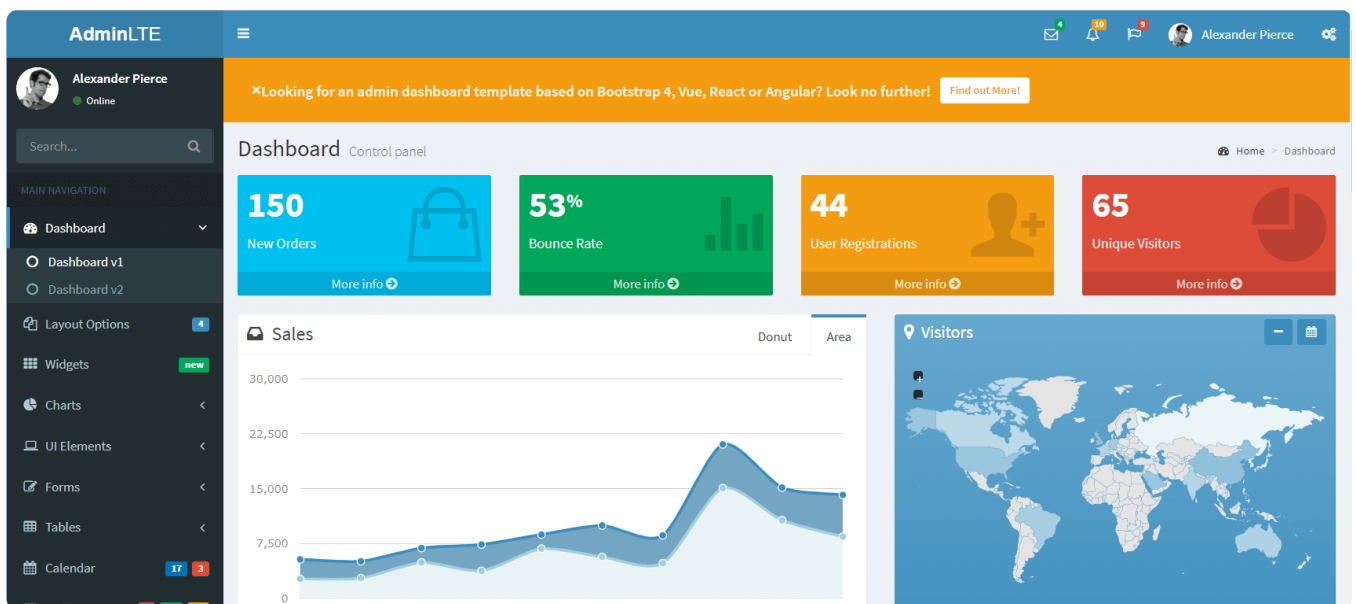
このパートでは、出来合いの無料テンプレートファイルを使ってDjangoアプリの画面をかっこいいデザインに仕上げます。

WEBアプリケーションを開発する上では、デザインに詳しい人でない限り見た目の良い画面デザインを作るのはなかなか難しいですし、デザインの勉強に時間をかけるのも大変です。

そこで、この教材ではフリーで公開されている画面テンプレートをDjangoに適用することで、最短時間でデザインの整ったWEB画面のデザインを作成します。

## 本パートの目標物

本パートでは、DjangoのWEBアプリに**AdminLTE**という無料の管理画面のテンプレートを適用します。



参考) [AdminLTE](#)

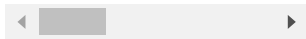
なお、「**管理テンプレート bootstrap**」といったキーワードでググってみると無料で使えるいろいろな管理テンプレートがありますので、お好みに合わせて好きなデザインを適用していただいても問題ありません。

具体的には以下の流れで設定を行います。

## 目標物を作成するまでの流れ

### 1. AdminLTEテンプレ

2. Django環境の基本
3. urls.pyの設定
4. views.pyの設定
5. TOP画面テンプレ-
6. AdminLTEのテンブ
7. AdminLTEのテンブ



それでは、AdminLTEを利用したDjangoアプリのWEB管理画面の開発手順を説明します。

## 1. AdminLTEテンプレートのダウンロード

まずは、以下のGitサイトから最新のAdminLTEのテンプレートzipファイルをダウンロードします。

<https://github.com/ColorlibHQ/AdminLTE/releases>

最新バージョン「AdminLTE **x.x.xx**」の「**Source code(zip)**」をクリックしてダウンロードしましょう。

# AdminLTE 2.4.12



REJack released this 22 days ago · 1 [commit](#) to master since this release

## Release Notes

- Fixed npm audit error
- Fixed strange navigation menu behavior + dark space on reloads
- Added height auto to .login-page & .register-page

### ▼ Assets 2



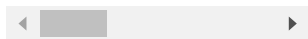
[Source code \(zip\)](#)



[Source code \(tar.gz\)](#)

※今回はバージョン2

※なお、2019年7月3日



**AdminLTE-x.x.xx.zip**というファイルがダウンロードできたら任意の場所に解凍しましょう。

解凍すると「**AdminLTE-x.x.xx**」というフォルダが解凍されます。

※ファイル数が多いため解凍には数分かかります。

## 2. Django環境の基本設定

---

つづいて、AdminLTEを利用するために必要なDjango周りの基本設定を行っていきます。

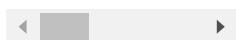
まずは、アプリケーション(pdfmr) フォルダ直下に**static**という名称のフォルダを生成します。

フォルダを作成したら、staticフォルダ直下に先ほどダウンロードして解凍した「**AdminLTE-x.x.xx**」フォルダを丸ごと配置します。

この時点のフォルダ構成は以下のようになります。

python

```
1  ../tutorial
2      |_____
3      |      |
4      |_____
5      |_____
6      |      |
7      |      |
8      |      |
9      |      |
10     |_____
11     |      |
12     |
```



## 3. urls.pyの設定

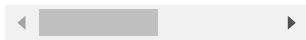
---

続いて、**urls.py**の設定を行います。

今の時点では最低限TOP画面を表示するためのURLパターンの定義を行います。

まずは、プロジェクト直下の**urls.py**を以下のように変更します。

```
tutorial
├── tutorial
│   └── urls.py
```



### ※変更前

```
python
```

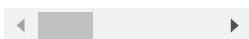
```
1 from django.conf.urls import patterns, url
2 from django.contrib import admin
3
4 urlpatterns = [
5     path('admin/', admin.site.urls),
6 ]
```



### ※変更後

```
python
```

```
1 from django.conf.urls import patterns, url
2 from django.contrib import admin
3
4 urlpatterns = [
5     path('admin/', admin.site.urls),
6     path('pdfmr/', include('pdfmr.urls')),
7 ]
8
```



django.urlsから**include**を追加して、**pdfmrアプリケーション**配下の**urls.py**をインクルードしてあげます。

続いて、アプリケーション**pdfmr**で新規に**urls.py**ファイルを作成し、以下のコードを追記します。

```
tutorial
├── pdfmr
```

urls.py

python

```
1 from django.conf.urls.defaults import *
2 from django.contrib.auth.views import login
3 from . import views
4
5 app_name = "pdfm"
6
7 urlpatterns = [
8     path('top/',
9 ]
```

ここでは、TOP画面を表示するためのURLパターンを以下の通り定義します。

URLパターン : http://

呼び出す関数 : views

URLパターン名称 : tc

また、以下の通り名前空間としてpdfmrという名称を設定しています。

python

```
1 app_name = "pdfm"
```

## 補足

名前空間は「app\_name」

名前空間を利用することで

URLを呼び出すことが

名前空間を利用してい

合に名前の衝突が

## 4. views.pyの設定

**views.py**ではTOP画面を表示するための設定を行います。  
今回は単純に関数ベースで定義します。

**views.py**に**top関数**の定義を以下の通り追加しましょう。

```
tutorial
├── pdfmr
│   └── view.py
```



```
python

1  def top(request):
2      return render
```



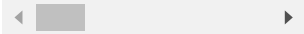
**renderメソッド**を使って単純にpdfmr/top.htmlにrequestオブジェクトをレンダリングするだけの設定ですので  
特段難しい点はないでしょう。

## 5. TOP画面テンプレートの準備

---

先ほどstaticフォルダにコピーしたAdminLTE-x.x.xxフォルダ直下の**index.html**ファイルを  
**tutorial\templates\pdfmr**フォルダ直下へコピーして、**top.html**という名前へ変更しておきます。

```
tutorial
├── templates
│   └── pdfmr
│       └── top.
```



この時点で、開発サーバを起動しTOP画面を表示してみましょう。

まずは、tutorialプロジェクトに移動して以下のコマンドで開発サーバを起動します。

```
python

1  $ cd tutorial
2  $ python manage
```



※cloud9の場合は以下の通り8080ポートを指定

```
python
```

```
1 $ cd tutorial
2 $ python manage
```



開発サーバが起動したら以下のURLにブラウザでアクセスしてみましょう。


<http://127.0.0.1:8000/pdfmr/top>

※cloud9の場合は[https://\\*\\*\\*\\*.cloud9.\\*\\*\\*\\*.amazonaws.com/pdfmr/top](https://****.cloud9.****.amazonaws.com/pdfmr/top)にアクセスしましょう。(以降cloud9の場合の説明は割愛します)

現時点では、以下のようにCSS等が未適用状態でTOP画面が表示されます。

[ALT AdminLTE](#)

[Toggle navigation](#)

- [4](#)
  - You have 4 messages
  - ■  User Image

[Support Team 5 mins](#)

[Why not buy a new awesome theme?](#)

-  User Image

[AdminLTE Design Team 2 hours](#)

[Why not buy a new awesome theme?](#)

-  User Image

[Developers Today](#)

[Why not buy a new awesome theme?](#)

-  User Image

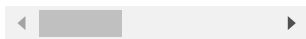
[Sales Department Yesterday](#)

このままだと使い物にならないので、AdminLTEの各種デザインが適用されるように静的ファイルの設定を行っていきます。

## 6. AdminLTEのテンプレートの適用

tutorial\templates\pdfmr\top.html内の**linkタグ**と**scriptsタグ**のパス設定をDjangoで定義した静的フォルダ（static）を参照するように以下の形式に修正していきます。

```
href="{% static '参  
src="{% static '参
```



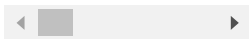
数が多いので大変ですが、linkタグとscriptsタグについて上記フォーマットに従って変更しましょう。

### [linkタグの変更方法]

※変更前

```
html
```

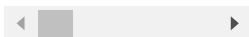
```
1 <link rel="styl
```



※変更後

```
html
```

```
1 <link rel="styl
```

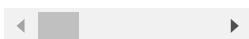


### [scriptタグの変更方法]

※変更前

```
html
```

```
1 <script src="bo
```

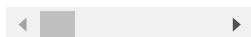


※変更後



html

```
1 <script src="{%
```



すべてのlinkタグとscriptタグの修正が終わったら、最後に**top.html**の先頭に{% load staticfiles %}を追記します。

html

```
1 # =====ここカ  
2 {% load staticf  
3 # =====ここマ  
4 <!DOCTYPE html>  
5 <html>  
6 <head>
```



これはDjangoの静的ファイルをロードするための設定です。

再度以下のURLにアクセスしてみましょう。

<http://127.0.0.1:8000/pdfmr/top>

以下のようにAdminLTEのデザインが適用されればOKです。



以上で今回のパートは終了です。

次のパートでは、AdminLTEのテンプレート内容を今回開発するWEBアプリ用にカスタマイズしていきます。

お疲れさまでした。



