

🌟 Plotly Studio: the AI-native desktop app for creating data apps and visualizing data. [Sign up for early access!](#)

Show and Tell - Dash Bootstrap Components

[show-and-tell](#) · [community-components](#)

tcbelevy 1 December 7, 2018, 2:34pm

Hi everyone,

We'd like to present our library of Bootstrap themed components for Dash.

We have documentation available [here](#), which we're still adding to and improving, but there should already be enough there to give you a flavour of what the library can do and to get you started. We're also working on some proper example apps in addition to the documentation.

The source code is all available [on Github](#). We welcome bug reports and pull requests from anyone who would like to contribute!

[@chriddy](#) if you could add us to the meta thread that would be great!

39 Likes

[Dash Bootstrap and DataGrid components/contrib?](#)

[:mega: Community Components Index](#)

yantaka2 2 December 7, 2018, 4:38pm

WOW, this is very useful!

As a developer who only knows the basics of front end development, something like this allows me to do a lot more with dash!

Thanks for contributing!

4 Likes

Damian 3 December 7, 2018, 4:45pm

Agreed, this is pretty amazing. I totally see integrating this in to a few of our apps, thanks!

4 Likes

ndbcorp 4 December 10, 2018, 9:31pm

[@tcbelevy](#) Just started with Dash and happened to come across this.

I would like to begin my testing of dash while using your bootstrap components. I saw the documentation and was curious if you have any built-out examples that go a little deeper to use as a reference when learning/testing.

Many thanks for building this!

1 Like

tcbelevy 5 December 10, 2018, 10:33pm

Hey [@ndbcorp](#),

We don't really have many built-out examples right now unfortunately, but it's very much on our to-do list and hopefully we'll get a few out there in the next few weeks.

In the meantime if you have specific questions about dash-bootstrap-components I'm happy to answer them here.

ndbcorp 6 December 11, 2018, 3:55pm

First question i have is about the navbar and how it can be used to navigate a multi page app. I will refer to the demo layout you have in the documentation.

Lets say in the menu drop down, clicking Entry 1 would display what your current demo layout shows (Heading + Graph). Now clicking Entry 2 would show lets says the dcc upload component and a corresponding dash datatable. Can you show how to code this? How about having tabs within entry 2...is it possible?

The dash Tabs + Subtabs doesnt really work well for what i am looking for. See thread on this here: [Dash Tabs and SubTabs](#)

tcbelevy 7 December 11, 2018, 5:27pm

The `NavLink` and `DropdownItem` components can work just like a `dash-core-components` style `Link`, which you can use in conjunction with `dcc.Location` to re-render the layout based on the current url. Checkout the [Dash docs](#) for more detail on that.

Here's a version of the app you described. It's definitely possible to have multiple pages and then have a set of tabs on one of the pages. I used some simple cards for the different layouts so as to not complicate things, but you can replace them with whatever you like, including the upload component, or graphs, or whatever else you like.

```
import dash
import dash_bootstrap_components as dbc
import dash_core_components as dcc
import dash_html_components as html
from dash.dependencies import Input, Output

app = dash.Dash(external_stylesheets=[dbc.themes.BOOTSTRAP])

navbar_children = [
    dbc.NavItem(dbc.NavLink("Page 1", href="/page-1")),
    dbc.NavItem(dbc.NavLink("Page 2", href="/page-2")),
]

navbar = dbc.Navbar(navbar_children, brand="Example", sticky="top")
```

```
# define content for page 1
page1 = dbc.Card(
    [
        dbc.CardTitle("Page 1 contents"),
        dbc.CardText("You can replace this with whatever you like"),
    ],
    body=True,
)

# define content for page 2

tab1 = dbc.Card(
    [
```

You can link between pages from a `DropdownMenu` embedded in the `NavBar` too, just set the `href` property of each `DropdownMenuItem` component and use it in the same way in the callback. Hope that makes sense, let me know if you're having problems.

As your app grows, you should probably also split the above across multiple files. See the "Structuring a Multi-Page App" section of [the Dash docs](#).

2 Likes

ndbcorp 8 December 11, 2018, 9:44pm

Thanks for the quick reply. I tried modifying the code you provided, adding the dcc upload + datatable from the first example at <https://dash.plot.ly/dash-core-components/upload>.

I added/updated the following packages:

```
import dash_table
from dash.dependencies import Input, Output, State
```

In your code, i replaced the `page1` variable with

```
page1 = html.Div([
    dcc.Upload(
        id='upload-data',
        children=html.Div([
            'Drag and Drop or ',
            html.A('Select Files')
        ]),
        style={
            'width': '100%',
            'height': '60px',
            'lineHeight': '60px',
            'borderWidth': '1px',
            'borderStyle': 'dashed',
            'borderRadius': '5px',
            'textAlign': 'center',
            'margin': '10px'
        },
        # Allow multiple files to be uploaded
        multiple=True
    ),
    html.Div(id='output-data-upload'),
])
```

and i added the following at the bottom for functionality:

```
def parse_contents(contents, filename, date):
    content_type, content_string = contents.split(',')

    decoded = base64.b64decode(content_string)
    try:
        if 'csv' in filename:
            # Assume that the user uploaded a CSV file
            df = pd.read_csv(
                io.StringIO(decoded.decode('utf-8')))
        elif 'xls' in filename:
            # Assume that the user uploaded an excel file
            df = pd.read_excel(io.BytesIO(decoded))
    except Exception as e:
        print(e)
        return html.Div([
            'There was an error processing this file.'
        ])

    return html.Div([
        html.H5(filename),
        html.H6(datetime.datetime.fromtimestamp(date)),

        dash_table.DataTable(
            data=df.to_dict('rows'),
            columns=[{'name': i, 'id': i} for i in df.columns]
        ),

        html.Hr(), # horizontal line
```

It didn't work. Am i doing something wrong here? I am getting the following error:

```
dash.exceptions.NonExistantIdException:
Attempting to assign a callback to the
component with the id "output-data-upload" but no
components with id "output-data-upload" exist in the
```

app's layout.

Here is a list of IDs in layout:
`['url', 'content']`

If you are assigning callbacks to components that are generated by other callbacks

(and therefore not in the initial layout), then you can suppress this exception by setting ``app.config['suppress_callback_exceptions']=True``.

tcbeley 9 December 11, 2018, 11:22pm

You're creating a callback that references a component that is not present in the layout when the app loads (specifically 'output-data-upload' is on the second page, so Dash doesn't know about it). It throws the error to warn you in case you might be doing something wrong.

You can configure Dash to ignore when this happens, and not complain when you reference elements in callbacks that aren't present in the layout. This is essential if you are assigning callbacks to components that are added to the layout at runtime, such as in a multi-page app.

TL;DR - Just add `app.config.suppress_callback_exceptions = True` after creating the `app` object and it should work (or at least throw a different error).

ndbcorp 10 December 12, 2018, 2:49am

Got it, thanks. I was able to get it to work...however the uploaded table disappears if i navigate to another page and then come back. Is this a due to the dbc library or dash?

tcbeley 11 December 12, 2018, 9:47am

That's to do with how Dash works rather than specific to `dash-bootstrap-components`. If you want the table to persist after being uploaded, you'll have to cache the data somehow, then get Dash to retrieve it every time you visit Page 1 again.

Here's an example using `dcc.Store`. I did this pretty quickly, so there might be a better way to set it up, but it works:

Modify `app.layout` to

```
app.layout = html.Div(
    [
        dcc.Location(id="url", pathname="/page-1"),
        dcc.Store(id="upload-storage"),
        navbar,
        dbc.Container(id="content", style={"padding": "20px"}),
    ]
)
```

When a file is uploaded, we'll serialise it and save it to `upload-storage`. Modify `parse_contents` to the following:

```
def parse_contents(contents, filename, date):
    content_type, content_string = contents.split(",")
    decoded = base64.b64decode(content_string)
    try:
        if "csv" in filename:
            # Assume that the user uploaded a CSV file
            df = pd.read_csv(io.StringIO(decoded.decode("utf-8")))
        elif "xls" in filename:
            # Assume that the user uploaded an excel file
            df = pd.read_excel(io.BytesIO(decoded))
    except Exception as e:
        print(e)
        return html.Div(["There was an error processing this file."])

    return {
        "filename": filename,
        "df": df.to_dict(orient="rows"),
        "date": date,
    }
```

Then create a callback to populate `upload-storage` when a file or files are uploaded:

```
@app.callback(
    Output("upload-storage", "data"),
    [Input("upload-data", "contents")],
    [State("upload-data", "filename"), State("upload-data", "last_modified")],
)
def store_upload(list_of_contents, list_of_names, list_of_dates):
    if list_of_contents is not None:
        children = [
            parse_contents(c, n, d)
            for c, n, d in zip(list_of_contents, list_of_names, list_of_dates)
        ]
        return children
```

Finally we want to use data from `upload-storage` when the page is loaded. Let's first make a function that generates page 1 from the data instead of having it defined as a global variable, so replace `page1 = ...` with

```
def make_page1(data):
    if data is not None:
        uploaded_data = []
        for d in data:
            filename = d["filename"]
            df = pd.DataFrame(d["df"])
            date = d["date"]
            uploaded_data.append(
                html.Div(
                    [
                        html.H5(filename),
                        html.H6(datetime.datetime.fromtimestamp(date)),
                        dash_table.DataTable(
                            data=df.to_dict("rows"),
                            columns=[{"name": i, "id": i} for i in df.columns],
                        ),
                    ]
                )
            )
    else:
        uploaded_data = None
    return html.Div(
        [
            dcc.Upload(
                id="upload-data",
                children=html.Div(
                    ["Drag and Drop or ", html.A("Select Files")]
                ),
            ),

```

Finally we want this to get called when the pathname changes, so change the url callback to

```
@app.callback(
    Output("content", "children"),
    [Input("url", "pathname"), Input("upload-storage", "data")],
)
def display_page(pathname, data):
    if pathname == "/page-1":
        return make_page1(data)
    if pathname == "/page-2":
        return page2
    # if not recognised, return 404 message
    return html.P("404 - page not found")
```

Hope that all makes sense. Let me know if you can't get it to work.

3 Likes

ndbcorn 12 December 12, 2018, 4:55pm

At first when uploading a file on page 1, no table shows. The only way to get the table to display is to navigate to page 2 and then return to page 1.

```
import dash
import dash_bootstrap_components as dbc
import dash_core_components as dcc
import dash_html_components as html
import dash_table
from dash.dependencies import Input, Output, State

import base64
import datetime
import io
import pandas as pd

app = dash.Dash(external_stylesheets=[dbc.themes.BOOTSTRAP])
app.config.suppress_callback_exceptions = True

navbar_children = [
    dbc.NavItem(dbc.NavLink("Page 1", href="/page-1")),
    dbc.NavItem(dbc.NavLink("Page 2", href="/page-2")),
    html.Div(html.Img(src='DeepDive_Logo_V1_Horizontal.png', height="100%"))
]

navbar = dbc.Navbar(navbar_children, brand="Example", sticky="top")

# define content for page 1
def make_page1(data):
    if data is not None:
        uploaded_data = []
        for d in data:
```

tchegley 13 December 12, 2018, 5:21pm

Delete this function and it should work

```
@app.callback(Output('output-data-upload', 'children'),
    [Input('upload-data', 'contents'),
     [State('upload-data', 'filename'),
      State('upload-data', 'last_modified')]])
def update_output(list_of_contents, list_of_names, list_of_dates):
    if list_of_contents is not None:
```

```
children = [
    parse_contents(c, n, d) for c, n, d in
    zip(list_of_contents, list_of_names, list_of_dates)]
return children
```

It seems to be because both `update_output` and `make_page1` are trying to modify `output-data-upload`.

sriotti 14 December 12, 2018, 6:37pm

@tcbegley This is pretty cool. I will continue to test this over the next couple weeks.

Question about customizing the navbar. In the example in your documentation, i tried adding a logo to the top left of the navbar (a .png file located in assets folder) & move the name of the app to the center of the navbar. Cant seem to get this to work properly. Could you show how to do this?

tcbegley 15 December 12, 2018, 8:43pm

Hey **@sriotti**

Unfortunately the navbar is not super customisable. We had to make some trade-offs between customisability and simplicity in the interface.

You could manually construct what you're describing though using html components and applying the relevant Bootstrap classes. For example, I downloaded [this logo](#) and saved it to `assets/logo.png` then built the following app:

```
import dash
import dash_bootstrap_components as dbc
import dash_core_components as dcc
import dash_html_components as html

app = dash.Dash(external_stylesheets=[dbc.themes.BOOTSTRAP])

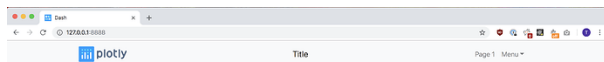
dropdown = dbc.DropdownMenu(
    nav=True,
    in_navbar=True,
    label="Menu",
    children=[
        dbc.DropdownMenuItem("Entry 1"),
        dbc.DropdownMenuItem("Entry 2"),
        dbc.DropdownMenuItem(divider=True),
        dbc.DropdownMenuItem("Entry 3"),
    ],
)

logo = html.Img(src="/assets/logo.png", height="50px")

title = dcc.Link("Title", href="/", className="navbar-brand")

nav_items = html.Ul(
    [dbc.NavItem(dbc.NavLink("Page 1", href="/page-1")), dropdown],
    className="navbar-nav",
)
```

which gives me this



1 Like

sriotti 16 December 12, 2018, 9:17pm

Thanks for sharing. How about a sidebar like the one requested in the following post in [Dash Tabs and SubTabs](#) ? The linked post mentions the Shiny Server sidebar, which currently can't be replicated using dash tabs + sub tabs in a visually appealing way.

tcbegley 17 December 12, 2018, 9:45pm

I've managed to do something like that in the past, but it's a little too involved to reply with here, I'll add one to the examples when we sort them out soon.

sriotti 18 December 13, 2018, 3:16pm

Cool, I look forward to seeing it and the rest of your examples. Thanks again for working on this.

1 Like

ndbcorp 19 December 13, 2018, 10:20pm

Thanks, that ended up working. So this approach would be necessary for all pages if i would like to keep the previous state while navigating?

I have a question about the *customizing style sheet* section of documentation... I am trying to follow the hypothetical scenario you proposed of using the *Cerulean* theme. I added a bootstrap.css file to my /assets folder that includes the Cerulean code from bootswatch. How do i select the different options within the theme? Do i have to add a className to my components in app.layout specifying? For instance, how do i get the navy blue navbar from the Cerulean theme as opposed the silver navbar?

tcbepley 20 December 13, 2018, 10:45pm

[@ndbcorp](#) yes, afaiw when you modify the props of components in the layout it's getting modified in the browser, I don't believe that the modifications are being sent back to the Python object that created them on the server. Therefore when you remove these components from the layout, the modified props get lost. If you want them to persist you should cache them as above.

The navy navbar is Cerulean's "dark" navbar. In `dash-bootstrap-components` you can do

```
dbc.Navbar(..., color="dark", dark=True)
```

The syntax looks a little redundant in this case, the `dark=True` tells `dash-bootstrap-components` that we've chosen a dark color (namely "dark" in this case!) and will choose a light text color to maintain a good contrast.

Doing that I see something like this:



Note: in case it wasn't clear in the docs, the Cerulean `bootstrap.css` replaces the regular `bootstrap.css`. Reading the docs back I thought there could be potential for confusion there, but perhaps it's fine.

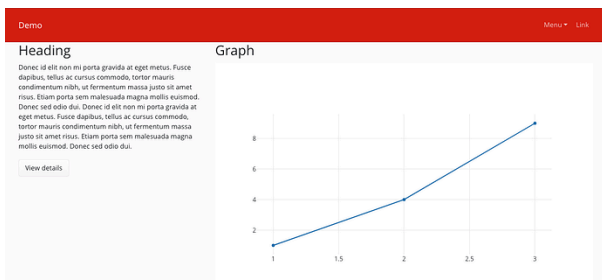
ndbcorp 22 December 14, 2018, 3:33am

Sticking with the documentation example, i changed the css theme to Simplex, and noticed an issue with the components spacing/alignment after the change. Why is that?

```
from dash import Dash
import dash_bootstrap_components as dbc
import dash_core_components as dcc
import dash_html_components as html

_navbar = dbc.Navbar(
    brand="Demo",
    brand_href="#",
    sticky="top",
    color="primary",
    dark=True,
    children=[
        dbc.DropdownMenu(
            nav=True,
            in_navbar=True,
            label="Menu",
            children=[
                dbc.DropdownMenuItem("Entry 1"),
                dbc.DropdownMenuItem("Entry 2"),
                dbc.DropdownMenuItem(divider=True),
                dbc.DropdownMenuItem("Entry 3"),
            ],
        ),
        dbc.NavItem(dbc.NavLink("Link", href="#")),
    ],
)


```



tcbepley 23 December 14, 2018, 8:08am

Good catch [@ndbcorp](#), the top margin is being applied by the `docs-content` class on the `Container` in `_body`, which is defined in our docs stylesheet, and so if you just copy paste the code on the docs homepage (which you definitely ought to be able to do without issue!) you won't have that defined anywhere. You'll actually see the same issue with regular bootstrap styling.

I've raised [an issue on our repo](#), and will update the docs. In the mean time just change `className="docs-content"` to `className="mt-4"` (the classes `mt-*` are bootstrap classes for applying top margins easily) and things should look better, or if you prefer add a top margin with the `style` argument of `Container`. Note though that using the `mt-*` classes will mean that if you switch to a different theme it should adjust the margin appropriately for that theme, so it's probably preferable.

EDIT - I've updated the example in the docs to use Bootstrap classes. Hopefully if you run that now you should see a margin at the top of the demo app.

ndbcorp 24 December 14, 2018, 4:43pm

Ok got it fixed on my end, thanks.

I am no css expert, so excuse the following question if its an obvious answer . Is it possible to have 2 style sheets to use in my app. Lets say i want to use the SIMPLEX theme from dbc to style certain components like the navbar, but i also want to use <https://codepen.io/chriddyp/pen/bWLwgP120> to style other components in my app. Can i use both? Do i need to create my own style sheet combining the desired styling from each? What do you recommend?

tcbelev 25 December 14, 2018, 11:12pm

You can definitely use multiple Stylesheets together. Just be aware that if they haven't been designed to be used together they may clash on some things and give weird behaviour.

sriotti 26 January 4, 2019, 11:06am

Hi [@tcbelev](#)

Any updates on the examples that will be added to the documentation as mentioned above?

Any sample or guideline to use Bootstrap 4 with Dash?

tcbelev 27 January 4, 2019, 11:18am

Hey [@sriotti](#)

Thanks for your interest and patience! I don't have the sidebar example for you yet, but that's next on my list of things to do.

We've most recently been working on a more customisable **Navbar** component as discussed above (check [this pull request if you're interested](#)), that will be released pretty soon.

I've also started work on cleaning out the **examples directory** of our repo. There's a few ports of RShiny examples in there at the moment as well as demos of the components. We'll continue adding examples there, including the multi-page examples.

In the meantime, you could check out [@nedned](#)'s **slapdash** repo, which has a sidebar layout that uses **dash-bootstrap-components**.

3 Likes

Chris369 28 January 9, 2019, 7:27pm

Really loving dash-bootstraps-components and the ease of implementing themes from Bootswatch.

What is the best approach to styling tables? I'm using the dcc.DataTable component, but it doesn't have a className attribute.

2 Likes

tcbelev 29 January 9, 2019, 11:29pm

Hey [@Chris369](#),

I don't know if there's a great way to style the **DataTable** using Bootstrap CSS. The component uses mostly inline styles which can be a little bit tricky to override with CSS classes.

If you're willing to compile your own Bootstrap stylesheet using sass, here's an approach that worked for me. The basic idea is that since we can't apply the **.table** class to the **DataTable**, we instead wrap the **DataTable** in an element with a particular **id** like **table-wrapper**, then add selectors like the following to your stylesheet:

```
#table-wrapper table {
  /* styles from .table class */
}
```

You could write this stylesheet manually, but that would be really tedious. Sass lets you inherit / extend existing selectors. Here's a bit more detail:

1. Download Bootstrap source from [Bootstrap website](#).
2. In the source folder there's a directory called **scss**, that's the one we want, copy that to your working directory.
3. Wrap your table in a div with a particular **id** like this:

```
app.layout = html.Div(dash_table.DataTable(...), id="table-wrapper")
```

4. Create your scss file, I called mine **table-shim.scss**, and write something like this:

```
@import "scss/bootstrap.scss";

#table-wrapper table{
  @extend .table;
  @extend .table-striped;
  @extend .table-hover;
  @extend .table-bordered;
  font-family: inherit;
  --border: none;
}
```

This basically says we want to create a new selector that will select any **table** that is a child of an element with id **table-wrapper**, and apply all the styles of the classes **.table**, **.table-striped** etc. to it. I also added the **font-family** and **--border** arguments as they weren't specified by Bootstrap so it defaulted to the **DataTable** defaults.

5. Compile with sass **sass table-shim.scss assets/my-bootstrap.css** or similar. See [here](#) for how to install sass.

That's it, run your app and hopefully it should work! As a final note, this worked for me with Bootswatch styles too. Just grab the **_variables.scss** and **_bootswatch.scss** files for the theme you're interested in from the [Bootswatch website](#) (I tried it with the "Lux" theme), and change the imports in your scss file to something like

```
@import "lux/_variables.scss";
@import "scss/bootstrap.scss";
@import "lux/_bootswatch.scss";
```

Depending on the theme you choose, you might have to make additional adjustments if the styles clash with the styles set by **DataTable**. I don't know for example how the dark Bootswatch themes would work with this approach.

Let me know how you get on if you try it, and definitely let me know how you get on if you find an easier way!

2 Likes

How to add className to DataTable?**Chris369** 30 January 10, 2019, 6:01pm

Thank you very much for your detailed instructions 😊 I'm only just getting started with Bootstrap/CSS and so this sounds rather complicated at the moment! But is definitely something I'll work towards! Thanks again.

1 Like

Chris369 31 January 10, 2019, 6:39pm

So I've been giving this ago, using Koala to compile my custom scss file [table-custom.scss]:

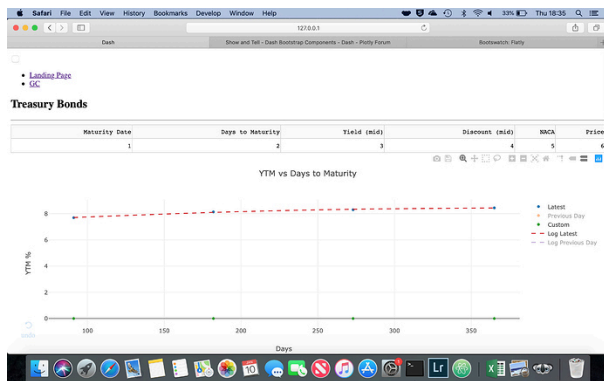
```
@import "../flatly/_variables.scss.css";
@import "bootstrap.scss";
@import "../flatly/_bootswatch.scss.css";

#table-wrapper table{
  @extend .table;
  @extend .table-striped;
  @extend .table-hover;
  @extend .table-bordered;
  font-family: inherit;
  --border: none;
}
```

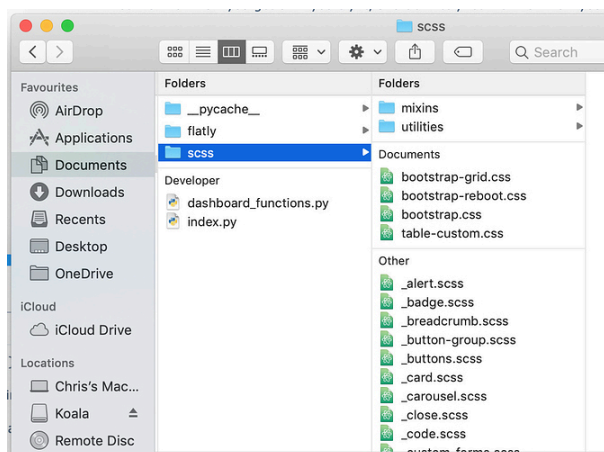
Using this code in my dash app:

```
app = dash.Dash(__name__, external_stylesheets=['/scss/table-custom.css'])
```

But when I run my app I seem to have lost all the flatly styling?



This is my folder structure:



Where've I gone wrong?

tcbeley 32 January 10, 2019, 6:53pm

The process for linking local CSS is a bit different to linking to external CSS via a CDN. Check out the [Dash docs](#) for detailed instructions, but the easiest way is to create a folder called `assets` in the same folder as your main app script, and just place the CSS (i.e. the compiled `table-custom.css`) in there. Dash will automatically detect and link it for you.

The local CSS you compiled replaces the need for any external CSS, so just ignore the `external_stylesheets` argument now. Sorry for the confusion!

EDIT - Also, I just noticed that your imports `@import "../flatly/_variables.scss.css";` look like they have an unneeded `.css` in there, change it to `@import "../flatly/_variables.scss";` (make sure the file name matches obviously). Same for `_bootswatch.scss`.

1 Like

Chris369 33 January 10, 2019, 7:10pm

Thank you so much! I have got it to work 😊 At first I was getting sort of a flaty theme with slightly different colours, but I've removed those additional .css's as advised and it looks as expected now. (I'm not sure why the extra .css's were there - when I downloaded the scss files my computer seemed to add .css automatically!).

Now if I have more than one table I can't use the same id='table-wrapper'. So am I write in thinking I just create a 'table-wrapper2' in my scss, recompile, and use the new id for table 2?

tcbelev 34 January 10, 2019, 7:24pm

Great! Glad it's working 😊

The table doesn't need to be an immediate child of `table-wrapper`, it just needs to be contained inside one, so if you like you could just give your top-level component the id `table-wrapper`, like `app.layout = html.Div(..., id="table-wrapper")`, and the styles will be applied to every table in your app. That also means your CSS / Sass doesn't have to change when you change the layout of your app and should be a lot more maintainable.

Chris36q 35 January 10, 2019, 7:30pm

Amazing - thank you! 😊

1 Like

Damian 36 March 11, 2019, 3:27pm

@tcbelev I just wanted to say thanks, I finally got around to really trying to implement this on my apps last week and after a day of getting to understand Bootstrap a little bit it's made the job of laying out my sites much simpler and more flexible.

One thing I was trying and failing to do, which I now suspect was just the wrong approach, was getting animations to work with real layout changes.

E.g. with 2 columns I wanted 1 to disappear, I tried using collapse and setting the width to zero but it made the the 2nd column fall below the 1st column and then the collapse occurred causing the 2nd column to then jump up. I assume this was just not how this was conceived to be used?

1 Like

tcbelev 37 March 11, 2019, 4:01pm

Glad you've enjoyed using it 😊

Could you describe a bit more the effect you're going for? Also any kind of code snippet that shows what you tried in a little bit more detail. You're right that the collapse is not really designed for this, but there might be a way to achieve the effect you want with a little bit of hacking.

`Col` doesn't do any input validation (perhaps it should), so if you set `width=0` the `Col` will try to apply a class called `col-0` which doesn't exist, and it will call back on the default behaviour specified in the `col` class which is probably why you saw the column expand and wrap onto a new line.

sclavijo 38 March 29, 2019, 7:54pm

So this has been really useful, but I was wondering how I would go about partitioning and customizing the 'brand' attribute?

Let's say I wanted my 'brand' in my SimpleNav be 'ABC' where each letter was a different color, how would I go about doing that?

Thanks,

tcbelev 39 March 29, 2019, 10:21pm

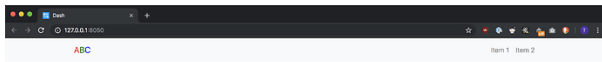
Hey @sclavijo,

To achieve that sort of thing you will probably need to use `Navbar` instead of `NavbarSimple`, then wrap each character in a `html.Span` and set the colour. Here's a simple example

```
import dash
import dash_bootstrap_components as dbc
import dash_html_components as html

app = dash.Dash(external_stylesheets=[dbc.themes.BOOTSTRAP])

app.layout = dbc.Navbar(
    dbc.Container(
        [
            dbc.NavbarBrand(
                [
                    html.Span("A", style={"color": "red"}),
                    html.Span("B", style={"color": "green"}),
                    html.Span("C", style={"color": "blue"}),
                ]
            ),
            dbc.Nav(
                [
                    dbc.NavItem(dbc.NavLink("Item 1", href="#")),
                    dbc.NavItem(dbc.NavLink("Item 2", href="#")),
                ],
                navbar=True,
            ),
        ],
    ),
)
```



There's a bunch more **Navbar** examples [here](#).

The reason you'll struggle to do the same with **NavbarSimple** is that (currently at least), you can only pass dash components to the **children** prop of other dash components, so you wouldn't be able to use **html.Span** like above with the **brand** prop. The only alternative would be if it's possible to write CSS to cycle through colours each character, but that sounds pretty nasty to me and might not be possible anyway.

Hope that helps.

2 Likes

tcbegeley 41 April 19, 2019, 7:22pm

[@sriotti](#) and anyone else interested: I finally added some simple multi-page examples to our repo like I've been promising for a long time. Check them out [here](#). They include a navbar example, plus multiple sidebar examples, including collapsible submenus and responsive designs. Try running them or check out some screenshots on the [pull request](#).

Also more generally, the release of Dash 0.41.0 let us add some cool new features. Personally I'm pretty excited about the new **Modal component**, which lets you easily add modals to your apps without having to fiddle around with CSS and z-indexes. Check out the steadily improving documentation or our repo to see the new stuff we're adding.

6 Likes

Dash Tabs and SubTabs

sriotti 42 April 20, 2019, 5:31pm

[@tcbegeley](#) Great work Tom. Really appreciate the time and effort you have put into dbc. I look forward to using these new components and am excited to see what's next!

1 Like

dayxx369 43 June 5, 2019, 2:55pm

Hello,

I have extensively read over the multi-app program, and I finally understand it. I tested other apps and it works. I am however stuck at how to create a callback for connecting the different DropdownMenuItem instances I have to pull out a new page (i.e. I am confused with how to make inputs and outputs of the callback). I only need help with the index page. Here is my structure:

```
from dash.dependencies import Input, Output
import dash_bootstrap_components as dbc
import dash_core_components as dcc
import dash_html_components as html
import socket
from app import app
import page_1, page_2, page_3
import base64

image_filename = 'file/path.png'
encoded_image = base64.b64encode(open(image_filename, 'rb').read())

app.layout = html.Div([
    html.Img(
        id = 'image',
        alt='Image',
        src='data:image/png;base64,{}'.format(encoded_image.decode()),
        style= {
            'width': '100%',
            'height': '100px'
        }
    ),
    dbc.NavbarSimple([
        dbc.DropdownMenu(
            id = 'Menu',
            nav=True,
            in_navbar=True,
            label="Menu",
```

tcbegeley 44 June 5, 2019, 4:11pm

Hey [@dayxx369](#)

You need to add the **dcc.Location** component to your layout, then use it in a callback to monitor the pathname and render the layout accordingly. When you set **href** on each of the **DropdownMenuItem** components, they behave just like **dcc.Link**, so you don't need to use either the **DropdownMenu** or **DropdownMenuItem** as inputs to any callbacks.

Something like this:

```
# add dcc.Location to your layout
app.layout = html.Div([
    dcc.Location(id="url"),
    ...
])

# use Location in the display_page callback
@app.callback(Output('page-content', 'children'),
              [Input('url', 'pathname')])
def display_page(pathname):
    if pathname in ('/page_1', '', None, '/'):
        return page_1.layout
```

```
elif pathname == '/page_2':
    return page_2.layout
elif pathname == '/page_3':
    return page_3.layout
else:
    return '404'
```

2 Likes

dayxx369 45 June 5, 2019, 4:57pm[@tcbegley](#)

Great. I understand. Now, should I have the NavbarSimple in the index or should it be redefined in the other programs? I want to set the dropdown item to disabled if it's currently on that page (i.e. other apps). Thanks again

lahashh 46 June 5, 2019, 8:26pm

really nice work can't wait to test it

1 Like

dayxx369 47 June 5, 2019, 8:40pm[@tcbegley](#)

Nevermind, I answered my own question. The NavbarSimple should be redefined in the other programs (apps). I am really looking forward to how dash evolves!

tcbegley 48 June 5, 2019, 8:41pm[@dayxx369](#)

Correct, put the `NavbarSimple` in the index. The top level structure of the app should be something like

```
app.layout = html.Div([
    dcc.Location(...),
    dbc.NavbarSimple(...),
    html.Div(id="page-content"),
])
```

EDIT - You got in just before me, but I'll leave my answer in case it helps someone else 😊

2 Likes

tcbegley 49 June 5, 2019, 8:47pm[@dayxx369](#), sorry for the spam...

On the question of disabling links in the navbar depending on which page you're on, you can write a callback that takes the location as an input, and sets the `disabled` prop of the relevant `DropdownMenuItem` accordingly. Though I would have thought that setting the `active` prop rather than `disabled` might be more natural?

Check out [this example](#) which does something similar with `NavLink` rather than `DropdownMenuItem`. I think you should be able to adapt it to your case pretty easily.

1 Like

dayxx369 50 June 5, 2019, 9:00pm

Oh yes, I figured that out as well. thanks again!

lahashh 51 June 6, 2019, 5:00pm

Question. Can i still override css classes with my custom CSS file in the assets folder if i use your components??

tcbegley 52 June 6, 2019, 6:26pm

Yes absolutely! All of the styles are applied through CSS anyway, so you can modify or add to them as you please.

`dbc.themes.BOOTSTRAP` is just a link to BootstrapCDN, if you prefer you can have a local copy of Bootstrap CSS, or a modified copy of it, or you can complement it with your own CSS.

2 Likes

lahashh 53 June 6, 2019, 7:58pm

thanks a lot it was my mistake i was messing with the wrong class

1 Like

Chris369 54 February 16, 2020, 4:08pm

Does anyone know how to use Sliders and Dropdown with dbc?

They seem to have been added here - <https://github.com/facultyai/dash-bootstrap-components/pull/297> - but when I try to use `dbc.Slider` / `dbc.Dropdown` I get this:

```
Traceback (most recent call last):
  File "index.py", line 334, in <module>
    dbc.Slider(
AttributeError: module 'dash_bootstrap_components' has no attribute 'Slider'
```

My dbc version is up to date

tebeley 55 February 16, 2020, 4:47pm

Hey [@Chris369](#),

`dash-bootstrap-components` doesn't have it's own versions of those components (basically because we wanted to minimise unnecessary duplication of functionality). The pull request you linked to is related to custom CSS that will restyle the `Slider` or `Dropdown` from `dash-core-components` to make it look more "bootstrapy".

The use of this custom CSS hasn't been documented yet, in the future we'd like to host our own Bootstrap stylesheets with some Dash specific features on a CDN and make it easy to access through `dbc.themes` or similar, but I haven't got around to it.

In the meantime however, you can check out my [dash-bootstrap-css repo](#) which has compiled versions of the styles that were added in that pull request, ready for download.

To use them, just download one of the compiled stylesheets in `dist/`, put it in your app's `assets/` folder, and make sure that any component you want to apply the styles to is a child of a component with `className="dash-bootstrap"`. For example I downloaded the [minty stylesheet](#), added it to `assets/` then ran this app

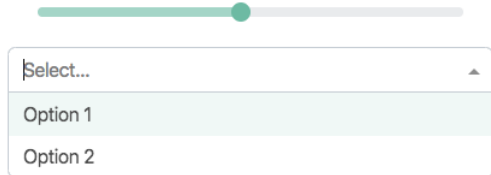
```
import dash
import dash_bootstrap_components as dbc
import dash_core_components as dcc
import dash_html_components as html

app = dash.Dash()

app.layout = dbc.Container(
    children=[
        dcc.Slider(min=0, max=10, step=1),
        dcc.Dropdown(options=[
            {"label": "Option 1", "value": 1},
            {"label": "Option 2", "value": 2},
        ]),
    ],
    className="dash-bootstrap p-5",
)

if __name__ == "__main__":
    app.run_server(debug=True)
```

and with that code I see these "minty" version of the `Slider` and `Dropdown`.



Note that I no longer need to use `external_stylesheets=[dbc.themes.MINTY]` or anything like that, these are full Bootstrap stylesheets with a few extra features that can be used as a drop-in replacement for other Bootstrap stylesheets.

2 Likes

Chris369 56 February 16, 2020, 6:05pm

Thank you so much for the quick reply 😊 And this looks perfect - just what I was after and looks easy enough to use - thank you!

1 Like

tebeley 57 February 16, 2020, 7:26pm

Great! The custom CSS still isn't properly battle-tested, so if you do try it out and have any issues please let me know.

andrewmarco 58 April 12, 2020, 7:56am

Hi [@tebeley](#). I would like to apply dash-bootstrap css to the modal component I am using, but not the other components of my app which I have styled manually.

For some reason when I add dash-bootstrap css file to my `./assets` folder, the dash css is loading on my dash table, even though I have not declared the `className='dash-bootstrap'` to the div. Oddly, it is not impacting other divs in my app (just the modal which has the `className`, and the table Div for some odd reason).

Could you look at my code below? Any idea why this is happening?

```
from app import app
import dash
import dash_table
from dash_table.Format import Format, Scheme, Sign, Symbol
import dash_html_components as html
import dash_core_components as dcc
import pandas as pd
```

```
import python.extract_portfolio
from datetime import date as Date
from dash.dependencies import Input, Output, State
import dash_bootstrap_components as dbc
import dash_table.FormatTemplate as FormatTemplate

# load dataframe from python logic
df = python.extract_portfolio.trade_list.sort_index(ascending=False)

# reformat index / trade date
df['Trade_Date'] = pd.DatetimeIndex(df['Trade_Date']).strftime("%b, %d %Y")

# add external css (note: style tags override the external css)
#external_stylesheets = ['https://codepen.io/chridhyp/pen/bWLwgP.css']

# serve dash app
app3 = dash.Dash(
    __name__,
    server=app,
    routes_pathname_prefix='/dash3/',
)
```

You can see in the picture below: The dash bootstrap css is working on the modal, but also impacting my table (using dash bootstrap fonts and stretching the columns). It is not the divs 'Filter by Ticker' or 'Filter by Transaction Type'.

tcbelevy 59 April 12, 2020, 11:21am

Hey [@andrewmarco](#)

The dash-bootstrap-css stylesheets are just the regular Bootstrap stylesheets + some additional styles that only get applied to descendants of an element with class `dash-bootstrap`, so all the regular Bootstrap styles get applied everywhere which is why your table is being affected.

If you only want to style the modal with Bootstrap, I would suggest copying **just the modal styles** from regular Bootstrap and putting them in a stylesheet inside your `assets/` folder.

1 Like

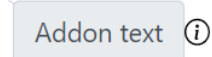
jbehic 60 September 30, 2020, 7:33pm

Hi there, this is awesome, thank you for sharing!

I would like to add an svg icon inside a `dbc.InputGroupAddon`. Something along the lines of:

```
dbc.InputGroupAddon([
    "Add-on text",
    html.Img(src=app.get_asset_url('icons/info_circle.svg'))
],
    addon_type="append"
)
```

It works, but it does not apply the background styling of the addon. See image:



I got the icon svg from:



Info circle

Official open source SVG icon library for Bootstrap

saved it into `info_circle.svg` and removed this part `class="bi bi-info-circle" fill="currentColor"` from the definition.

What am I doing wrong? All help is welcome!

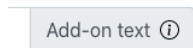
tcbelevy 61 September 30, 2020, 7:55pm

Hey [@jbehic](#), looks like there's something a bit funny going on with the `InputGroupAddon` component. You can work around it by wrapping the text and image in a span and adding the relevant class name yourself

```
dbc.InputGroupAddon(
    html.Span(
        [
            "Add-on text",
            html.Img(
                src=app.get_asset_url("icons/info_circle.svg"),
                className="ml-2", # bootstrap utility class, sets left margin
            ),
        ],
    ),
)
```

```
        className="input-group-text",
    ),
    addon_type="append",
)
```

Looks like this for me



1 Like

jhehic 62 October 4, 2020, 2:28am

Thank you for the fast and complete response, really appreciate it!

DiepNguyen 63 April 3, 2021, 8:40pm

Hi [@tchegley](#) thanks for your answer, it's great. I just wonder why it is so hard to make the table persist in this case, why "persistence = True" doesn't work in this case?

DiepNguyen 64 April 3, 2021, 9:57pm

We should add a "raise PreventUpdate" in the 2nd callback. If not, the table will be lost after the second page refresh.

```
def store_upload(list_of_contents, list_of_names, list_of_dates):
    if list_of_contents is not None:
        children = [
            parse_contents(c, n, d)
            for c, n, d in zip(list_of_contents, list_of_names, list_of_dates)
        ]
        return children
    else:
        raise PreventUpdate
```

1 Like

tchegley 65 April 4, 2021, 7:26am

It's quite possible you can simplify my example a lot using persistence (I think it will depend a bit on exactly what you want to do with the persisted data), but when I wrote the post you're responding to the feature didn't exist in Dash yet.

AnnMarieW 66 April 14, 2021, 5:22pm

Hey Everyone,

Be sure to upgrade to the latest release of dash-bootstrap-components and check out these new features:

- In release 0.12.0, it's now possible to trigger a dbc.Popover without any callbacks 🥳
See the first example in the updated docs here [Popover - dbc docs](#)
- There is also a new **download** prop for the dbc.Button. This functions basically like html.A, but it allows you to style the link like a Bootstrap Button. The new **download** prop allows you to specify a different file name on download than the filename specified in the **href**.
Here's a styled download button - it also includes a Font Awesome icon. See the code at the end of this post for an example that uses this button and downloads a file from the assets folder.



- See also a new Theme Explorer section of the docs that allows you to change the Bootstrap theme and see how the components respond with different colors, fonts etc. Note that if you use dbc.RadioItems, dbc.Checklist, dbc.Spinner, rather than the dash-core-component equivalents, that the style of the components updates automatically with the selected Bootstrap theme colors. You can see this in action here <https://dash-bootstrap-components.opensource.faculty.ai/docs/themes/explorer/>
- As of release 0.11.2 Table.from_dataframe for the html.Table supports multiple column header levels. Also, active_tab_style and activeTabClassName was added to enhance the styling of the dcc.Tabs component.

Please join me in saying thanks to [@tchegley](#) for maintaining this library – which he does in his spare time, and for all of the great support he provides on the forum.

```
import dash
import dash_html_components as html
import dash_bootstrap_components as dbc

FONT_AWESOME = (
    "https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css"
)

external_stylesheets = [dbc.themes.BOOTSTRAP, FONT_AWESOME]
app = dash.Dash(__name__, external_stylesheets=external_stylesheets)

app.layout = dbc.Container(
    [
        dbc.Button(
            [html.I(className="fa fa-download mr-1"), "Download"],
            href="/assets/data_file.txt",
```

```

        download="my_data.txt",
        external_link=True,
        color="info"
    ),
],
fluid=True,
className='m-4'
)

if name == "main":

```

5 Likes

bgivens33 67 May 14, 2021, 8:17pm

I can't seem to access the style of a dbc.button in a callback, is there anyway to do that?

tcbelev 68 May 16, 2021, 6:55amHey [@bgivens33](#)

What are you trying to do exactly? It seems to work for me, check out the following example

```

import dash
import dash_bootstrap_components as dbc
from dash.dependencies import Input, Output

COLORS = ["red", "orange", "yellow", "green", "blue", "indigo", "violet"]

app = dash.Dash(external_stylesheets=[dbc.themes.BOOTSTRAP])

app.layout = dbc.Container(
    dbc.Button("Click me", id="button", className="p-5")
)

@app.callback(Output("button", "style"), Input("button", "n_clicks"))
def toggle_color(n_clicks):
    if n_clicks:
        return {"backgroundColor": COLORS[(n_clicks - 1) % 7]}
    return {}

if __name__ == "__main__":
    app.run_server(debug=True)

```

bgivens33 69 May 18, 2021, 3:22am

Thanks for the tip... I was trying to bring in the style via a State, which wasn't working (might not be exposed?),but your method is working just fine. Thanks!

Amox 70 July 28, 2021, 12:53pmHey [@tcbelev](#),

It is possible to work with external stylesheets (CDN) and local CSS file in the same time ?
I just want to apply the compiled file only to table.

tcbelev 71 July 28, 2021, 1:12pm

Sure, you can have a mix of local and externally hosted stylesheets. The main thing to watch out for is any duplication. Anything that is repeated you are paying for with bandwidth and load time.

Amox 72 July 29, 2021, 12:02pmHey [@tcbelev](#),

Thank you for the response, but when i follow your instructions for styling Data Table it override all my external stylesheets (Bootstrap theme and font awesome CDN).

tcbelev 73 July 29, 2021, 12:36pm

Could you say a bit more about what stylesheets you are loading and how you are linking each of them. Is it `external_stylesheets` for external and `assets/` for local?
If your stylesheets have a clash (e.g. both of them define font-size for a particular class), then the local will override the external I believe. But it shouldn't stop it from loading entirely.




Amox 74 July 29, 2021, 1:06pm

Yes `external_stylesheets` for external and `assets/` for local.

```
FONT_AWESOME = 'https://pro.fontawesome.com/releases/v5.8.1/css/all.css'

app = dash.Dash(__name__, external_stylesheets=[dbc.themes.LITERA_FONT_AWESOME],
               suppress_callback_exceptions=True)
```

My-bootstrap.css file is in `assets/`

| Name | Date | Type | Size | Tags |
|---|------------------|--------------------|--------|------|
|  my-bootstrap.css | 29/07/2021 13:39 | CSS Source File | 192 KB | |
|  my-bootstrap.css.m... | 29/07/2021 13:39 | Linker Address Map | 49 KB | |
|  typography.css | 06/05/2021 10:29 | CSS Source File | 1 KB | |

My problem is that when i compile the my-bootstrap.css file with `sass table-shim.scss assets/my-bootstrap.css` i loose all Font awesome icons.

Select...

tcbelegy 75 July 29, 2021, 8:17pm

Strange. Can you use the developer tools in your browser to check if the stylesheet is being linked correctly and fetched from the CDN successfully?

Related topics

| Topic | Replies | Activity |
|---|---------|------------------|
| Display tables in Dash | 103 | November 2, 2018 |
| Dash Blueprint Components show-and-tell community-components | 22 | August 22, 2024 |
| Plotly Dash Dropdown is asymmetric question | 18 | July 7, 2024 |
| mega: Dash 2.5 Released - Easier Multi-Page Apps, Component Properties announcements | 31 | July 26, 2022 |
| Dash Bootstrap Templates V1.0.0 New: Theme Switch Components show-and-tell | 20 | June 11, 2024 |