



Adding Syntax Highlighting in Flutter Cookbook App

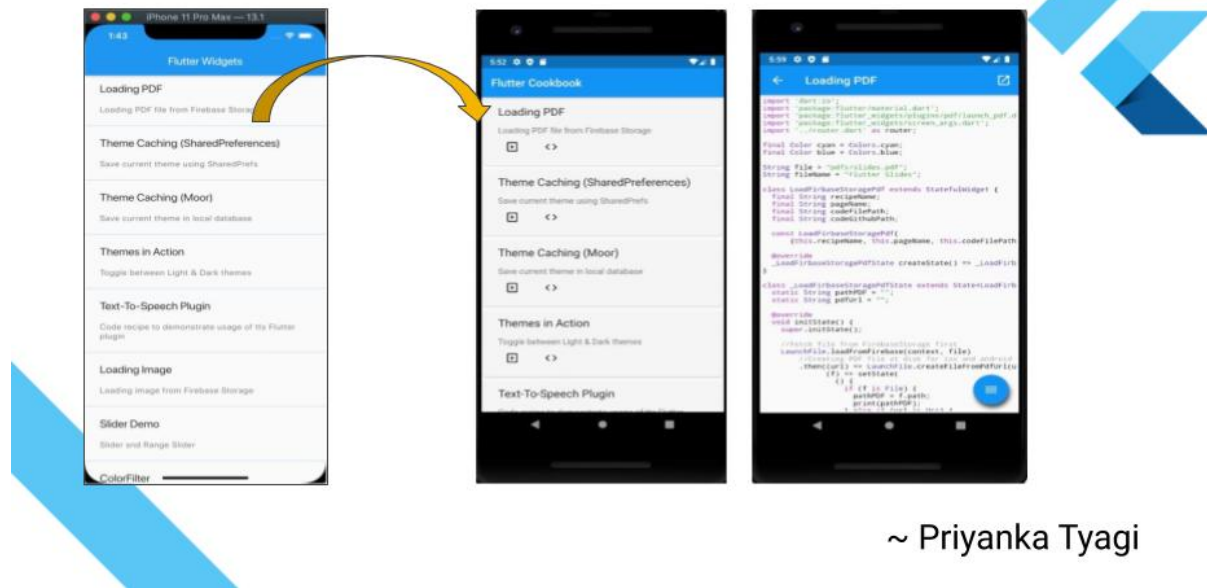
Posted on February 06, 2020 in [Flutter](#)

Always Use Protection

Watch Online Content in Private with Namecheap VPN.

[Namecheap.com](#)

Syntax Highlighting in Flutter Cookbook App



~ Priyanka Tyagi

Target Audience: Beginner

Recipe: Displaying code snippets for Flutter Cookbook code recipes.

Focus Widget: `syntax_highlighter` plugin

Checkout the companion video tutorial:



Lets's go!

`pubspec.yaml` dependency

Add `syntax_highlighter` plugin dependency as below:

```
dependencies:
  flutter:
    sdk: flutter

  #Highlighting syntax in code snippets for code recipes
  syntax_highlighter: ^0.1.0

  #To provide Floating Action widget to zoom in and out code snippet text
  animated_floatactionbuttons: ^0.1.0
```

- `syntax_highlighter` plugin: This plugin is used to highlight syntax in code snippets for code recipes.
- `animated_floatactionbuttons` plugin: This plugin is used to provide Floating Action widget to zoom in and out code snippet text.

CodeFileWidget: A `StatefulWidget` to display Code snippet

Class `CodeFileWidget` is a `StatefulWidget`. It's responsible to display underlying code for a Flutter code recipe.

This class takes four parameters to populate appropriate code snippet in `CodeFileWidget`.

- `recipeName`: Code recipe's name to display in the appBar at the top.
- `pageName`: This is used to routing purposes. In this particular app, this is a required parameter to navigate the recipe's code and its preview and vice versa.
- `codeFilePath`: This is the location of code file in the code. For example: `lib/pdf/`. Don't forget to add this path in `assets` section of `pubspec.yaml` file like below:

```
assets:
  - lib/pdf/
```

- `codeGithubPath`: This is the external `https://` link to code's file in Github repo.

```
class CodeFileWidget extends StatefulWidget {
  final String recipeName;
```

```

final String pageName;
final String codeFilePath;
final String codeGithubPath;

CodeFileWidget(
  {this.recipeName, this.pageName, this.codeFilePath, this.codeGithubPath});

@override
CodeFileWidgetState createState() {
  return CodeFileState();
}

class CodeFileWidgetState extends State<CodeFileWidget> {
  double scaleFactorText = 1.0;

  @override
  Widget build(BuildContext context) {
    ...
  };

  List<Widget> codepreviewActions(String codeContent) {
    ...
  }

  Widget highlightCodeSyntax(String codeContent, BuildContext context) {
    ...
  }
}

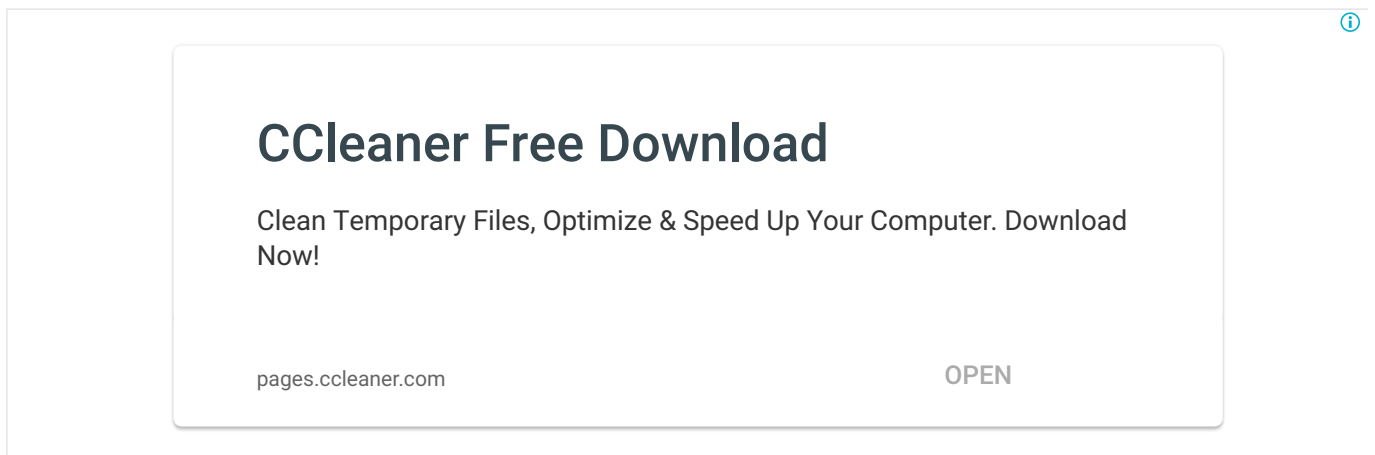
```

Next, we'll go over the each of the three methods of `CodeFileWidgetState`. Please note that default scale factor value for text is set to 1: `double scaleFactorText = 1.0;`

Building screen

First, we'll build the main interface to display code snippet.

AppBar: The App Bar has an action to open the external link to Github repository file. Title displays the recipe name.



FutureBuilder: `FutureBuilder` loads the file contents of the passed `codeFilePath` file. It shows the `CircularProgressIndicator` widget while file contents are loading. Once the data is available, `highlightCodeSyntax(...)` method takes the file contents and highlights syntax.

`AnimatedFloatingActionButton`: `AnimatedFloatingActionButton` is used to show actions to zoom_in and zoom_out code, and to navigate to the demo view.

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      leading: IconButton(
        icon: Icon(Icons.arrow_back),
        onPressed: () => Navigator.pop(context),
      ),
      title: Text(widget.recipeName),
      actions: <Widget>[
        IconButton(
          icon: Icon(Icons.open_in_new),
          onPressed: () => UrlUtils.open(this.widget.codeGithubPath),
        )
      ],
    ),
    body: FutureBuilder(
      future: rootBundle.loadString(widget.codeFilePath) ??
        'Error loading code file ${this.widget.codeFilePath}',
      builder: (BuildContext context, AsyncSnapshot<String> snapshot) {
        if (snapshot.hasData) {
          return Scaffold(
            body: Padding(
              padding: EdgeInsets.all(4.0),
              child: highlightCodeSyntax(snapshot.data, context),
            ),
            floatingActionButton: AnimatedFloatingActionButton(
              fabButtons: codepreviewActions(),
              colorStartAnimation: Colors.blue,
              colorEndAnimation: Colors.cyan,
              animatedIconData: AnimatedIcons.menu_close,
            ),
          );
        } else {
          return Center(child: CircularProgressIndicator());
        }
      },
    ),
  );
}
```

Adding actions

Now, add actions like zoom_in, zoom_out, and navigating to code recipe demonstration.

- **Zooming Out:** `scaleFactorText`'s value is decreased by `0.1` every time 'Zoom out' widget is tapped. That means text is made smaller.
- **Zooming In:** `scaleFactorText`'s value is increased by `0.1` every time 'Zoom out' widget is tapped. That means text is made bigger. You can see more text at a time.
- **Navigating to Code demo:** Tapping on this action opens the code recipe demonstration screen.

```
List<Widget> codepreviewActions() {
  return <Widget>[
    //making text smaller
    FloatingActionButton(
      heroTag: "zoom_out",
      child: Icon(Icons.zoom_out),
    ),
  ];
}
```

```

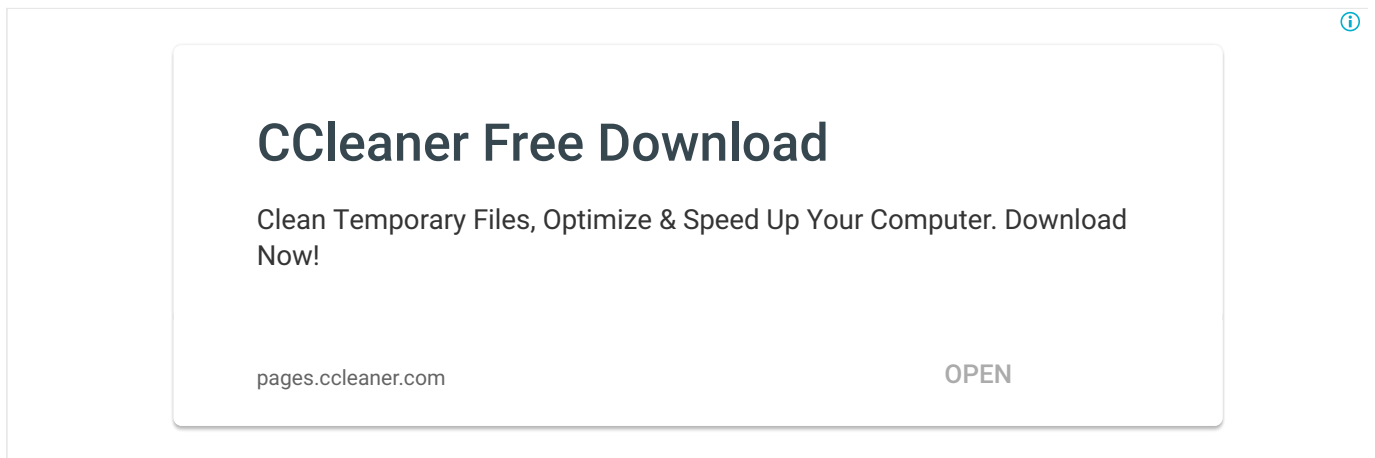
        tooltip: 'Zoom out',
        onPressed: () => setState(() {
          this.scaleFactorText = max(0.8, this.scaleFactorText - 0.1);
        }),
      ),
      //making text bigger
      FloatingActionButton(
        heroTag: "zoom_in",
        child: Icon(Icons.zoom_in),
        tooltip: 'Zoom in',
        onPressed: () => setState(() {
          this.scaleFactorText += 0.1;
        }),
      ),
      FloatingActionButton(
        heroTag: "open_page",
        child: Icon(Icons.slideshow),
        tooltip: 'See Demo',
        onPressed: () => Navigator.popAndPushNamed(
          context,
          widget.pageName,
          arguments: ScreenArguments(widget.recipeName, widget.pageName,
            widget.codeFilePath, widget.codeGithubPath),
        ),
      ),
    ],
  );
}

```

Highlighting code syntax

This is the main method which is responsible to highlight syntax for the code file. Plugin provides `SyntaxHighlighterStyle` class to help provide the correct theme for the current platform's theme (light vs dark).

`Container` widget is used to display the code snippet using `BoxConstraints.expand()` constraints.



`Scrollbar` widget uses two nested `SingleChildScrollView` widgets. First one provides the vertical scrolling by default. Second `SingleChildScrollView` sets its `scrollDirection` property to `Axis.horizontal` to provide horizontal scrolling.

`RichText` widget uses `textScaleFactor` property to adapt the variable value for `scaleFactorText`.

`TextSpan` widget uses `monospace` fontFamily to show code in a readable view.

Plugin's `DartSyntaxHighlighter` class uses the chosen `SyntaxHighlighterStyle` to render file content using proper syntax highlighting.

```
Widget highlightCodeSyntax(String codeContent, BuildContext context) {
  final SyntaxHighlighterStyle style =
    Theme.of(context).brightness == Brightness.dark
      ? SyntaxHighlighterStyle.darkThemeStyle()
      : SyntaxHighlighterStyle.lightThemeStyle();
  return Container(
    constraints: BoxConstraints.expand(),
    child: Scrollbar(
      child: SingleChildScrollView(
        child: SingleChildScrollView(
          scrollDirection: Axis.horizontal,
          child: RichText(
            textScaleFactor: this.scaleFactorText,
            text: TextSpan(
              style: TextStyle(fontFamily: 'monospace', fontSize: 12.0),
              children: <TextSpan>[
                DartSyntaxHighlighter(style).format(codeContent)
              ],
            ),
          ),
        ),
      ),
    ),
  );
}
```

Source code repo

- Recipe source code for `CodeFileWidget` class is available [here](#)
- Flutter Cookbook project's source code is available [here](#)

References:

1. [animated_floatactionbuttons](#) plugin
2. [syntax_highlighter](#) plugin

Happy cooking with Flutter :)

*Liked the article ? Couldn't find a topic of your interest ? Please leave comments or [email me](#) about topics you would like me to write ! *BTW I love cupcakes and coffee both :)**

Follow me at [twitter](#)

SyntaxHighlighting

Cross-platform

Flutter

Code-recipes

Android

Android Studio

iOS

development

Like this article? Share it with your friends!

[Comments](#) [Community](#) [Privacy Policy](#) [Login](#) 1 ▼[Recommend](#) [Tweet](#) [Share](#) [Sort by Best](#) ▼

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Be the first to comment.

[✉ Subscribe](#) [Add Disqus to your site](#) [Add Disqus](#) [Add](#)[Do Not Sell My Data](#)© 2020 - This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#)Powered by [Pelican](#) - [Flex](#) theme by [Alexandre Vicenzi](#)