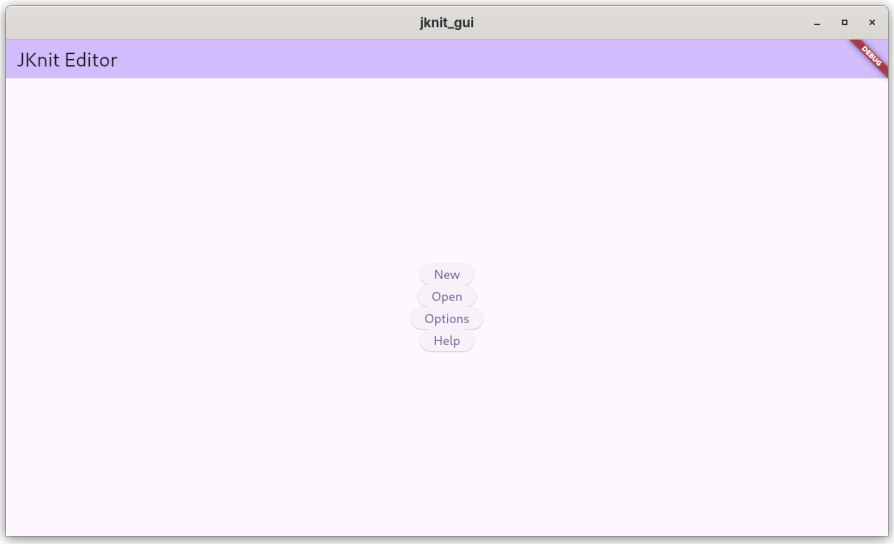
Dart/Flutter Assignment

GUI Overview Screenshots



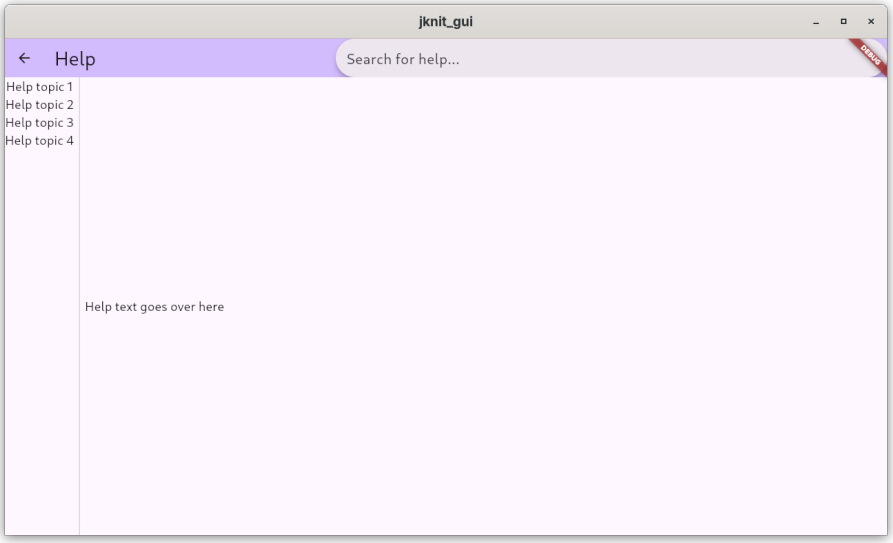
jknit_gui
Open Project
DEBUG
foo.jmd
Select file
Create Project

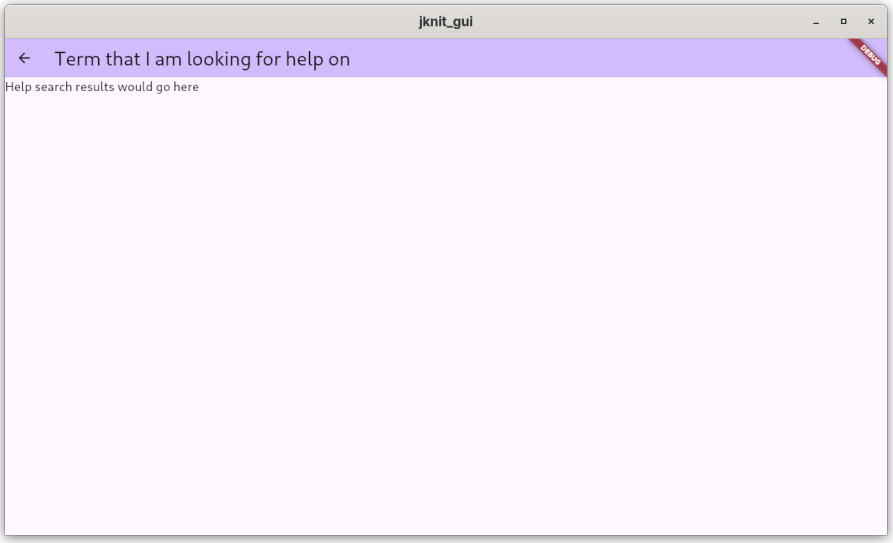
jknit_gui
JKnit Project Editor
DEBUG
Files go over here
Files go over here
Files go over here
Files go over here
Currently open file goes here
Export

jknit_gui
Export Document
DEBUG
Options
Export

jknit_gui
Options
DEBUG
Treat all Warnings as Errors
Enable Log
Enable Timer
Force Fancy Typesetting
Output File Type
LaTeX (.tex)

jknit_gui
Export ERROR
DEBUG
ERROR: Failed to open source ''
(knitting halted)
2


jknit_gui
Help
Search for help...
DEBUG
Help topic 1
Help topic 2
Help topic 3
Help topic 4
Help text goes over here


jknit_gui
Term that I am looking for help on
DEBUG
Help search results would go here

Passing Test Cases

widget_test.dart | new.dart | open.dart

test > widget_test.dart > main > testWidgets("New project navigation test")

```dart
void main() {
  Run | Debug
  testWidgets('New project navigation test', (WidgetTester tester) async {
    WidgetController.hitTestWarningShouldBeFatal = true;

    await tester.pumpWidget(const JKnitGUIApp());

    // We should be on the 'main' page
    expect(find.widgetWithText(AppBar, 'JKnit Editor'), findsOneWidget);

    // Click 'New'
    await tester.tap(find.byKey(ValueKey('main.new')));
    await tester.pumpAndSettle();

    // Ensure we made it to 'New' widget
    expect(find.widgetWithText(AppBar, 'New Project'), findsOneWidget);

    // Get to editor
    await tester.tap(find.byKey(ValueKey('new.editor')));
    await tester.pumpAndSettle();

    expect(find.widgetWithText(AppBar, 'JKnit Project Editor'), findsOneWidget);

    // Go to export page
    await tester.tap(find.byKey(ValueKey('editor.export')));
    await tester.pumpAndSettle();
    expect(find.widgetWithText(AppBar, 'Export Document'), findsOneWidget);

    // Export
    await tester.tap(find.byKey(ValueKey('export.export')));
    await tester.pumpAndSettle();
  });
}
```

```dart
  Run | Debug
  testWidgets('Help navigation test', (WidgetTester tester) async {
    WidgetController.hitTestWarningShouldBeFatal = true;

    await tester.pumpWidget(const JKnitGUIApp());

    // We should be on the 'main' page
    expect(find.widgetWithText(AppBar, 'JKnit Editor'), findsOneWidget);

    // Click 'Help'
    await tester.tap(find.byKey(ValueKey('main.help')));
    await tester.pumpAndSettle();
    expect(find.widgetWithText(AppBar, 'Help'), findsOneWidget);

    // Find and use the search bar
    await tester.enterText(find.byKey(ValueKey('help.search')), 'search text');
    expect(find.widgetWithText(AppBar, 'search text'), findsOneWidget);
  });
```

```
jorb@archlinux ~/Programs/jknit_gui> flutter test
00:04 +2: All tests passed!
jorb@archlinux ~/Programs/jknit_gui>
```