как сделать dropdownbutton в flutter

Вы можете создать DropdownButton в Flutter, используя следующий код:

typescriptCopy code

import 'package:flutter/material.dart'; class MyDropdownButton extends StatefulWidget { @override \_MyDropdownButtonState createState() => \_MyDropdownButtonState(); } class \_MyDropdownButtonState extends State<MyDropdownButton> { String dropdownValue = 'One'; @override Widget build(BuildContext context) { return DropdownButton<String>( value: dropdownValue, icon: const Icon(Icons.arrow\_drop\_down), iconSize: 24, elevation: 16, style: const TextStyle(color: Colors.deepPurple), onChanged: (String newValue) { setState(() { dropdownValue = newValue; }); }, items: <String>['One', 'Two', 'Three', 'Four'] .map<DropdownMenuItem<String>>((String value) { return DropdownMenuItem<String>( value: value, child: Text(value), ); }).toList(), ); } }

Здесь мы создаем StatefulWidget, чтобы динамически изменять значение выбранного элемента. Мы также используем строковую переменную **dropdownValue**, чтобы хранить выбранный элемент.

**DropdownButton** принимает множество параметров, которые определяют его внешний вид и поведение. В данном случае мы используем значок внизу для указания, что это выпадающее меню. Элементы выпадающего меню определяются в **items** как список строк. Мы используем метод **map** для преобразования списка строк в список виджетов **DropdownMenuItem**.

Когда пользователь выбирает элемент, выполняется функция **onChanged**, которая устанавливает новое значение **dropdownValue** и вызывает метод **setState**, чтобы обновить пользовательский интерфейс и показать новое значение выбранного элемента.

в чем ошибка drawer: Drawer( child: ListView( padding: EdgeInsets.zero, children: const <Widget>[ DrawerHeader( decoration: BoxDecoration( color: Colors.blue, ), child: Text( 'Drawer Header', style: TextStyle( color: Colors.white, fontSize: 24, ), ), ), // ignore: unnecessary\_const DropdownButton(items: [ DropdownMenuItem(child: Text('Profile')), DropdownMenuItem(child: Text('Profile')) ], value: \_dropdownValue, onChanged: dropdownCallback), ], ), ),

Ошибка в коде заключается в том, что вы не указали список значений **items** для **DropdownButton**. Попробуйте изменить свой код следующим образом:

lessCopy code

drawer: Drawer( child: ListView( padding: EdgeInsets.zero, children: <Widget>[ DrawerHeader( decoration: BoxDecoration( color: Colors.blue, ), child: Text( 'Drawer Header', style: TextStyle( color: Colors.white, fontSize: 24, ), ), ), DropdownButton( items: [ DropdownMenuItem( child: Text('Profile 1'), value: 'profile1', ), DropdownMenuItem( child: Text('Profile 2'), value: 'profile2', ), ], value: \_dropdownValue, onChanged: dropdownCallback, ), ], ), ),

Здесь мы добавляем список значений **items**, включающий два элемента **DropdownMenuItem**, каждый из которых имеет свое значение и свой текст. Также вам нужно обновить **value** в **DropdownButton** для использования **\_dropdownValue**, который должен быть определен в вашем коде.

как определить \_dropdownValue

Для того, чтобы определить **\_dropdownValue** в вашем коде, вы можете объявить его как переменную состояния внутри вашего StatefulWidget. Вот пример:

javaCopy code

class MyHomePage extends StatefulWidget { const MyHomePage({Key? key}) : super(key: key); @override \_MyHomePageState createState() => \_MyHomePageState(); } class \_MyHomePageState extends State<MyHomePage> { String \_dropdownValue = 'profile1'; // задаем значение по умолчанию void dropdownCallback(String? newValue) { setState(() { \_dropdownValue = newValue ?? 'profile1'; // обновляем значение переменной состояния }); } @override Widget build(BuildContext context) { // ваш код, включая Drawer с DropdownButton } }

Здесь мы объявляем **\_dropdownValue** как переменную состояния внутри нашего StatefulWidget. Мы также задаем ему значение по умолчанию **'profile1'**. Затем мы объявляем функцию **dropdownCallback**, которая будет вызываться при изменении значения в **DropdownButton**. Внутри этой функции мы используем **setState**, чтобы обновить значение **\_dropdownValue**. Также мы добавляем проверку на **null** в **newValue**, чтобы избежать ошибок, если значение не установлено.

у меня код ругается на setState

Ошибка может возникать, если вы пытаетесь вызвать метод **setState** вне класса **State** или если этот метод вызывается внутри метода, не являющегося методом **build**. Например, если у вас есть следующий код:

lessCopy code

class MyHomePage extends StatefulWidget { const MyHomePage({Key? key}) : super(key: key); @override \_MyHomePageState createState() => \_MyHomePageState(); } class \_MyHomePageState extends State<MyHomePage> { String \_dropdownValue = 'profile1'; void dropdownCallback(String? newValue) { \_dropdownValue = newValue ?? 'profile1'; // ОШИБКА: setState должен использоваться для обновления переменных состояния } @override Widget build(BuildContext context) { return Scaffold( drawer: Drawer( child: ListView( padding: EdgeInsets.zero, children: <Widget>[ DrawerHeader( decoration: BoxDecoration( color: Colors.blue, ), child: Text( 'Drawer Header', style: TextStyle( color: Colors.white, fontSize: 24, ), ), ), DropdownButton( items: [ DropdownMenuItem( child: Text('Profile 1'), value: 'profile1', ), DropdownMenuItem( child: Text('Profile 2'), value: 'profile2', ), ], value: \_dropdownValue, onChanged: dropdownCallback, ), ], ), ), body: Container(), ); } }

Здесь метод **dropdownCallback** не использует метод **setState** для обновления переменной состояния **\_dropdownValue**. Вместо этого он напрямую обновляет значение переменной, что приводит к ошибке.

Чтобы исправить ошибку, вы должны использовать метод **setState** для обновления переменной состояния внутри **dropdownCallback**. Вот пример исправленного кода:

lessCopy code

class MyHomePage extends StatefulWidget { const MyHomePage({Key? key}) : super(key: key); @override \_MyHomePageState createState() => \_MyHomePageState(); } class \_MyHomePageState extends State<MyHomePage> { String \_dropdownValue = 'profile1'; void dropdownCallback(String? newValue) { setState(() { \_dropdownValue = newValue ?? 'profile1'; // ИСПРАВЛЕНИЕ: использование setState для обновления переменной состояния }); } @override Widget build(BuildContext context) { return Scaffold( drawer: Drawer( child: ListView( padding: EdgeInsets.zero, children: <Widget>[ DrawerHeader( decoration: BoxDecoration( color: Colors.blue, ), child: Text( 'Drawer Header', style: TextStyle( color: Colors.white, fontSize: 24, ), ), ), DropdownButton( items: [ DropdownMenuItem( child: Text('Profile 1'), value: 'profile1', ), DropdownMenuItem( child: Text('Profile 2'), value: 'profile2', ), ], value: \_dropdownValue, onChanged: dropdownCallback, ), ], ), ), body: Container(), ); } }

Здесь мы добавляем метод **setState** в **dropdownCallback**, чтобы обновить

исправь ошибки class MainPage extends StatelessWidget { const MainPage({super.key}); String \_dropdownValue = 'profile1'; // задаем значение по умолчанию @override Widget build(BuildContext context) { const dropdownMenuItem = DropdownMenuItem(child: Text('Profile')); return Scaffold( backgroundColor: Color.fromARGB(248, 248, 248, 248), appBar: AppBar( iconTheme: IconThemeData( color: Color.fromARGB(255, 74, 81, 0), ), backgroundColor: Color.fromARGB(248, 248, 248, 248), title: const Text('Octopus', textAlign: TextAlign.center, textDirection: TextDirection.ltr, style: TextStyle( color: Color.fromARGB(255, 74, 81, 0), fontSize: 32, fontFamily: 'Nunito', fontWeight: FontWeight.w900, )), ), drawer: Drawer( child: ListView( padding: EdgeInsets.zero, children: <Widget>[ DrawerHeader( decoration: BoxDecoration( color: Colors.blue, ), child: Text( 'Drawer Header', style: TextStyle( color: Colors.white, fontSize: 24, ), ), ), DropdownButton( items: [ DropdownMenuItem( child: Text('Profile 1'), value: 'profile1', ), DropdownMenuItem( child: Text('Profile 2'), value: 'profile2', ), ], value: \_dropdownValue, onChanged: dropdownCallback, ), ], ), ), body: SafeArea( child: Column( crossAxisAlignment: CrossAxisAlignment.center, mainAxisAlignment: MainAxisAlignment.center, children: [], ))); } }

Ошибки в вашем коде связаны с использованием конструктора **const** и отсутствием метода **dropdownCallback**.

Исправленный код:

lessCopy code

class MainPage extends StatefulWidget { const MainPage({Key key}) : super(key: key); @override \_MainPageState createState() => \_MainPageState(); } class \_MainPageState extends State<MainPage> { String \_dropdownValue = 'profile1'; void dropdownCallback(String newValue) { setState(() { \_dropdownValue = newValue; }); } @override Widget build(BuildContext context) { return Scaffold( backgroundColor: Color.fromARGB(248, 248, 248, 248), appBar: AppBar( iconTheme: IconThemeData( color: Color.fromARGB(255, 74, 81, 0), ), backgroundColor: Color.fromARGB(248, 248, 248, 248), title: const Text( 'Octopus', textAlign: TextAlign.center, textDirection: TextDirection.ltr, style: TextStyle( color: Color.fromARGB(255, 74, 81, 0), fontSize: 32, fontFamily: 'Nunito', fontWeight: FontWeight.w900, ), ), ), drawer: Drawer( child: ListView( padding: EdgeInsets.zero, children: <Widget>[ DrawerHeader( decoration: BoxDecoration( color: Colors.blue, ), child: Text( 'Drawer Header', style: TextStyle( color: Colors.white, fontSize: 24, ), ), ), DropdownButton( items: [ DropdownMenuItem( child: Text('Profile 1'), value: 'profile1', ), DropdownMenuItem( child: Text('Profile 2'), value: 'profile2', ), ], value: \_dropdownValue, onChanged: dropdownCallback, ), ], ), ), body: SafeArea( child: Column( crossAxisAlignment: CrossAxisAlignment.center, mainAxisAlignment: MainAxisAlignment.center, children: [], ), ), ); } }

В этом коде используется **StatefulWidget**, так как мы меняем значение переменной **\_dropdownValue** и должны использовать **setState()**. Метод **dropdownCallback()** определен в этом коде как пустая функция, поэтому вам нужно добавить свою логику для обработки изменения значения выпадающего списка.

напиши логику для обработки изменения значения выпадающего списка.

Конечная логика обработки изменения значения выпадающего списка зависит от того, что вы хотите сделать с этим значением. Однако я могу показать вам пример того, как использовать выбранное значение в выпадающем списке для отображения соответствующего виджета на экране.

Допустим, у нас есть два профиля - "Профиль 1" и "Профиль 2". Мы хотим отображать разные виджеты на экране в зависимости от выбранного профиля.

Ниже приведен обновленный код с логикой обработки изменения значения выпадающего списка:

lessCopy code

class MainPage extends StatefulWidget { const MainPage({Key key}) : super(key: key); @override \_MainPageState createState() => \_MainPageState(); } class \_MainPageState extends State<MainPage> { String \_dropdownValue = 'Профиль 1'; Widget \_selectedProfileWidget = const Text('Это профиль 1'); void dropdownCallback(String newValue) { setState(() { \_dropdownValue = newValue; if (newValue == 'Профиль 1') { \_selectedProfileWidget = const Text('Это профиль 1'); } else if (newValue == 'Профиль 2') { \_selectedProfileWidget = const Text('Это профиль 2'); } }); } @override Widget build(BuildContext context) { return Scaffold( backgroundColor: Color.fromARGB(248, 248, 248, 248), appBar: AppBar( iconTheme: IconThemeData( color: Color.fromARGB(255, 74, 81, 0), ), backgroundColor: Color.fromARGB(248, 248, 248, 248), title: const Text( 'Octopus', textAlign: TextAlign.center, textDirection: TextDirection.ltr, style: TextStyle( color: Color.fromARGB(255, 74, 81, 0), fontSize: 32, fontFamily: 'Nunito', fontWeight: FontWeight.w900, ), ), ), drawer: Drawer( child: ListView( padding: EdgeInsets.zero, children: <Widget>[ DrawerHeader( decoration: BoxDecoration( color: Colors.blue, ), child: Text( 'Drawer Header', style: TextStyle( color: Colors.white, fontSize: 24, ), ), ), DropdownButton( items: [ DropdownMenuItem( child: Text('Профиль 1'), value: 'Профиль 1', ), DropdownMenuItem( child: Text('Профиль 2'), value: 'Профиль 2', ), ], value: \_dropdownValue, onChanged: dropdownCallback, ), ], ), ), body: SafeArea( child: Column( crossAxisAlignment: CrossAxisAlignment.center, mainAxisAlignment: MainAxisAlignment.center, children: [ \_selectedProfileWidget, ], ), ), ); } }

В этом примере мы определяем переменную **\_selectedProfileWidget**, которая будет хранить выбранный виджет профиля.

мне выдает ошибку The argument type 'void Function(String)' can't be assigned to the parameter type 'void Function(String?)?'.

Эта ошибка возникает потому, что функция **dropdownCallback** принимает аргумент типа **String**, тогда как функция **onChanged** принимает аргумент типа **String?**.

Вы можете исправить это, добавив знак вопроса после типа **String** в сигнатуре функции **dropdownCallback**. Таким образом, функция будет принимать аргумент типа **String?**, который можно передать функции **onChanged**.

Вот исправленный код:

javascriptCopy code

void dropdownCallback(String? newValue) { setState(() { \_dropdownValue = newValue!; }); }

Обратите внимание на знак вопроса после типа **String**.

мне выдает ошибку The parameter 'key' can't have a value of 'null' because of its type, but the implicit default value is 'null'. Try adding either an explicit non-'null' default value or the 'required' modifier.

Эта ошибка возникает потому, что вы забыли указать имя параметра для конструктора **MainPage** в строке **const MainPage({super.key});**, а параметр **key** объявлен как обязательный, но не имеет значения по умолчанию.

Вы можете исправить это, добавив имя параметра **key** после двоеточия, чтобы он соответствовал сигнатуре конструктора **StatelessWidget**, а также установить значение по умолчанию для параметра **key**, например, **Key('mainPage')**.

Вот исправленный код:

typescriptCopy code

class MainPage extends StatelessWidget { const MainPage({Key? key}) : super(key: key); String \_dropdownValue = 'profile1'; void dropdownCallback(String? newValue) { setState(() { \_dropdownValue = newValue!; }); } @override Widget build(BuildContext context) { return Scaffold( ... ); } }

Обратите внимание на добавление имени параметра **key** после двоеточия и использование **Key('mainPage')** для установки значения по умолчанию.

всё работает, спасибо

Рад, что помог! Если у вас будут еще какие-то вопросы, не стесняйтесь спрашивать.

мне нужно чтобы при выборе одного из значений кнопки, меня пересылало на советующую страницу

Для того, чтобы при выборе значения из выпадающего списка перейти на соответствующую страницу, вам необходимо сначала определить, какие страницы должны открываться при выборе каждого значения в списке.

Затем вы можете использовать метод **Navigator.push()** для перехода на нужную страницу. Например, если у вас есть страница **Profile1Page**, которая должна открываться при выборе значения **'Profile 1'** в списке, то вы можете добавить следующий код в метод **dropdownCallback**:

scssCopy code

void dropdownCallback(String? newValue) { setState(() { \_dropdownValue = newValue!; }); if (\_dropdownValue == 'profile1') { Navigator.push( context, MaterialPageRoute(builder: (context) => Profile1Page()), ); } }

Здесь мы проверяем, равно ли выбранное значение **\_dropdownValue** значению **'profile1'**, и если да, то вызываем **Navigator.push()** с помощью **MaterialPageRoute** для открытия страницы **Profile1Page**.

Аналогично можно добавить код для открытия других страниц, которые соответствуют другим значениям в списке.

Обратите внимание, что для использования **Navigator** вам необходимо импортировать пакет **package:flutter/material.dart**.

как задать не изменяемы заголовок кнопки

Если вы хотите задать неизменяемый заголовок для кнопки **DropdownButton** в Flutter, то можно вместо параметра **value** использовать параметр **hint**. Параметр **hint** задает текст, который будет отображаться на кнопке, когда никакое значение не выбрано.

Например, если вы хотите задать неизменяемый заголовок для кнопки "Выберите профиль", то можете использовать следующий код:

lessCopy code

DropdownButton( items: [ DropdownMenuItem( child: Text('Profile 1'), value: 'profile1', ), DropdownMenuItem( child: Text('Profile 2'), value: 'profile2', ), ], hint: Text('Выберите профиль'), value: \_dropdownValue, onChanged: dropdownCallback, ),

Здесь мы задаем текст **"Выберите профиль"** для параметра **hint** и не указываем значение для параметра **value**, так как мы хотим, чтобы заголовок кнопки был неизменяемым.