

Mobile Apps Workshop

Flutter: UI Basics 2

Navigation

```
1 Navigator.push(  
2     context,  
3     MaterialPageRoute(builder: (context) => SecondRoute()),  
4     );
```

- Problem: duplizierter Code, Navigationslogik überall im Code verteilt
- Lösung: Navigationslogik zusammenfassen → **Named Routes**

Navigation

- Route-Mapping in der *routes* Property definieren
- *initialRoute* setzen anstatt *home*
- Mit *Navigator.pushNamed()* zu den definierten Routes navigieren

```
1 MaterialApp(  
2   initialRoute: '/',  
3   routes: {  
4     // When navigating to the "/" route, build the FirstScreen widget.  
5     '/': (context) => FirstScreen(),  
6     // When navigating to the "/second" route, build the SecondScreen widget.  
7     '/second': (context) => SecondScreen(),  
8   },  
9 );
```

```
1 // Within the 'FirstScreen' widget  
2 onPressed: () {  
3   Navigator.pushNamed(context, '/second');  
4 }
```

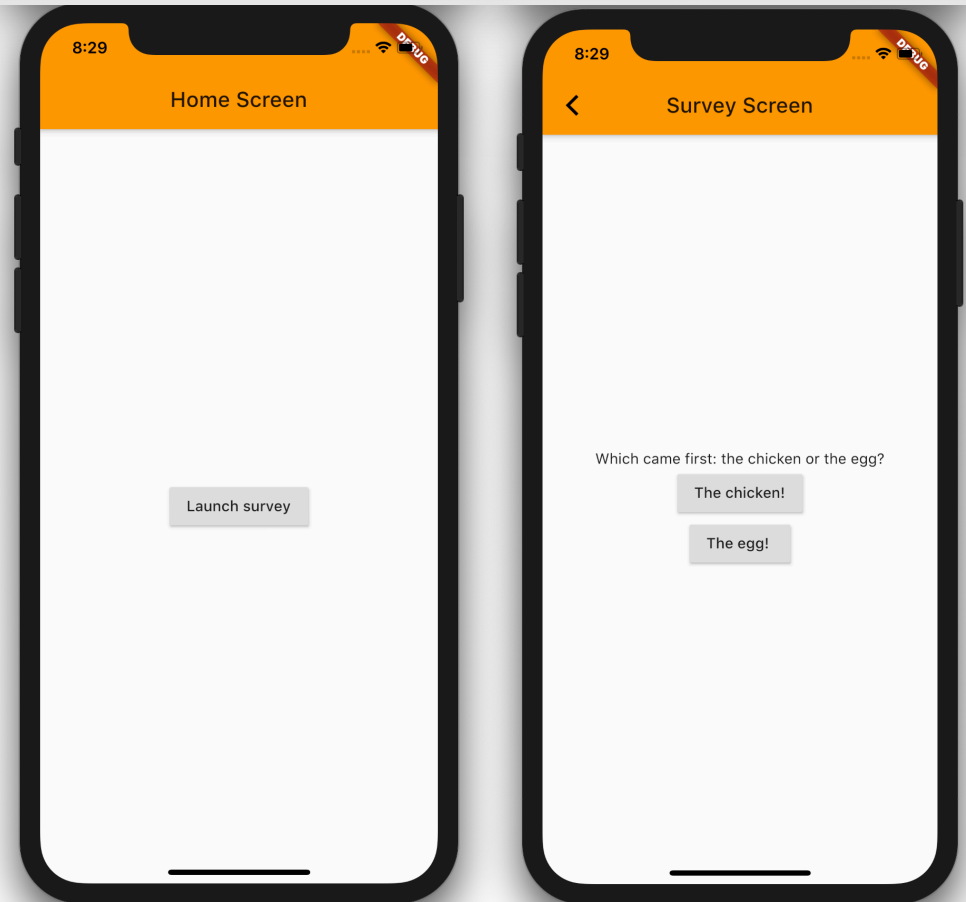
Navigation

- Um den Screens Argumente mitzugeben, kann auch *onGenerateRoute* von *MaterialApp* verwendet werden
- Innerhalb der *generateRoute()* Funktion können dann anhand der gegebenen *RouteSettings* die richtigen Screens erstellt und mit nötigen Argumenten befüllt werden

```
1 MaterialApp(  
2   title: 'Navigation Basics',  
3   theme: ThemeData(  
4     primarySwatch: Colors.orange,  
5   ),  
6   initialRoute: HomeScreen.routeName,  
7   onGenerateRoute: RouteGenerator.generateRoute,  
8 );
```

Übung

- Code auschecken
 - > [NavigationExercise2](#)
- TODOs bearbeiten, um mit Named Routes zwischen den Screens zu navigieren



Übung

- Code auschecken
 - > [MovieListExercise](#)
- TODOs bearbeiten, um MovieList mit statischen Daten anzuzeigen
- [flutter_launcher_icons](#) verwenden, um aus `img/icon.png` ein App Icon zu erzeugen

