

## Flutter Demo

Notizbuch: Mobile Usergroup Zentralschweiz  
Erstellt: 11.11.2018 11:33  
Autor: loana.albisser@bluewin.ch

Geändert: 17.11.2018 11:10  
Ort: 47°1'25 N 8°17'31 E

<https://flutter.io/docs/get-started/codelab>

## Vorbereitung

- Flutter Applikation in Android Studio erstellen
- Gesamten Code löschen
- Leere MyApp Klasse erstellen

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Welcome to Flutter',
      home: Scaffold(
        appBar: AppBar(
          title: Text('Welcome to Flutter'),
        ),
        body: Center(
          child: Text('Hello World'),
        ),
      ),
    );
  }
}
```

## Erklärung Vorbereitung

- Build Methode beschreibt, wie ein Widget dargestellt wird, indem es Low-Level Widgets verwendet

## Verwendung von externen Paketen

pubspec.yaml verwaltet die Abhängigkeiten. Um verschiedene zufällige Englische Wörter zu verwenden soll das Package english\_words hinzugefügt werden.

```
dependencies:
  flutter:
    sdk: flutter

  cupertino_icons: ^0.1.0
  english_words: ^3.1.0
```

Um die Pakete zu laden muss folgender Konsolen-Befehl ausgeführt werden

```
$ flutter packages get
```

Import-Statement im main.dart ergänzen

```
import 'package:english_words/english_words.dart';
```

wordPair in build-Methode ergänzen

```
final wordPair = new WordPair.random();
```

Body ergänzen, [Hot-Reload](#)

```
body: new Center(  
  child: new Text(wordPair.asPascalCase),  
),
```

## Stateful Widget hinzufügen

---

State-Klasse ausserhalb der App-Klasse ergänzen

```
class RandomWordsState extends State<RandomWords> {  
  
}
```

Stateful-Widget hinzufügen

```
class RandomWords extends StatefulWidget {  
  @override  
  RandomWordsState createState() => new RandomWordsState();  
}
```

Build-Methode ergänzen

```
Widget build(BuildContext context) {  
  final WordPair wordPair = new WordPair.random();  
  return new Text(wordPair.asPascalCase);  
}
```

Body-Methode im MyApp-Klasse ergänzen

```
body: new Center(  
  child: new RandomWords(),  
),
```

## ListView erstellen

---

RandomWordState ergänzen

```
final List<WordPair> _suggestions = <WordPair>[];  
final TextStyle _biggerFont = const TextStyle(fontSize: 18.0);
```

Widget builder erstellen

```
Widget _buildSuggestions() {  
  return new ListView.builder(  
    padding: const EdgeInsets.all(16.0),  
    itemBuilder: (BuildContext _context, int i) {  
      if (i.isOdd) {  
        return new Divider();  
      }  
      final int index = i ~/ 2;  
      if (index >= _suggestions.length) {  
        _suggestions.addAll(generateWordPairs().take(10));  
      }  
      return _buildRow(_suggestions[index]);  
    }  
  );  
}
```

```
);  
}
```

### \_buildRow-Methode ergänzen

```
Widget _buildRow(WordPair pair) {  
  return new ListTile(  
    title: new Text(  
      pair.asPascalCase,  
      style: _biggerFont,  
    ),  
  );  
}
```

### build-Methode ersetzen

```
return new Scaffold (  
  appBar: new AppBar(  
    title: new Text('Startup Name Generator'),  
  ),  
  body: _buildSuggestions(),  
);
```

## Icon zur Liste hinzufügen

---

### \_saved zu RandomWordState hinzufügen

```
final Set<WordPair> _saved = new Set<WordPair>();
```

### \_buildRow-Methode ergänzen

```
final bool alreadySaved = _saved.contains(pair);
```

### ListTile in \_buildRow ergänzen

```
trailing: new Icon(  
  alreadySaved ? Icons.favorite : Icons.favorite_border,  
  color: alreadySaved ? Colors.red : null,  
),
```

## Interaktivität hinzufügen

---

### ListTile in \_buildRow ergänzen

```
onTap: () {  
  setState(() {  
    if (alreadySaved) {  
      _saved.remove(pair);  
    } else {  
      _saved.add(pair);  
    }  
  });  
},
```

## Seitennavigation

---

### AppBar im RandomState ergänzen

```
actions: <Widget>[  
  // Add 3 lines from here...  
  new IconButton(icon: const Icon(Icons.list), onPressed: _pushSaved),  
],
```

pushedsaved-ethode ergänzen, hot-reload zeigen

```
void _pushSaved() {  
}
```

\_pushSaved ergänzen

```
void _pushSaved() {  
  Navigator.of(context).push();  
}
```

Navigator ergänzen

```
void _pushSaved() {  
  Navigator.of(context).push(  
    new MaterialPageRoute<void>(  
      builder: (BuildContext context) {  
        final Iterable<ListTile> tiles = _saved.map(  
          (WordPair pair) {  
            return new ListTile(  
              title: new Text(  
                pair.asPascalCase,  
                style: _biggerFont,  
              ),  
            );  
          },  
        );  
        final List<Widget> divided = ListTile  
          .divideTiles(  
            context: context,  
            tiles: tiles,  
          )  
          .toList();  
      },  
    ),  
  );  
}
```

Divider hinzufügen unter divided

```
return new Scaffold(  
  appBar: new AppBar(  
    title: const Text('Saved Suggestions'),  
  ),  
  body: new ListView(children: divided),  
);
```