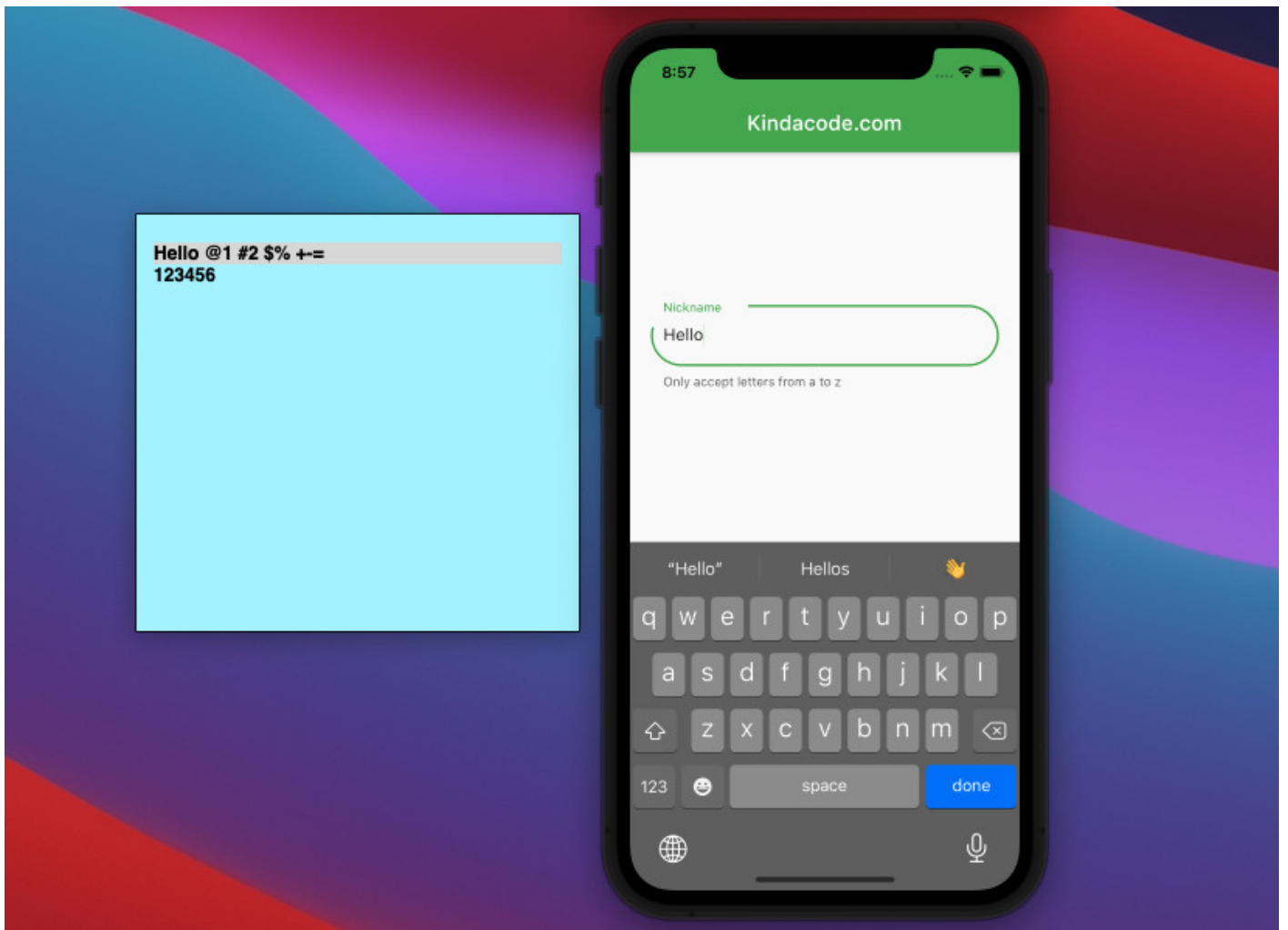




[Home](#) » [Mobile](#) » [Flutter](#) » Flutter: FilteringTextInputFormatter Examples

Flutter: FilteringTextInputFormatter Examples

🕒 Last updated on August 31, 2021 👤 A Goodman 👁 1382 💬 [Post a comment](#)



This article walks you through a few practical examples of using **FilteringTextInputFormatter** to blacklist or whitelist certain characters when the user

enters text into the TextField (or TextFormField). Without any further ado, let's dive into the code.

Table of Contents



1. Example 1: Only Allowing Alphabet Letters

1.1. Preview

1.2. The code

2. Example 2: Preventing Offensive Words from Being Displayed

2.1. Preview

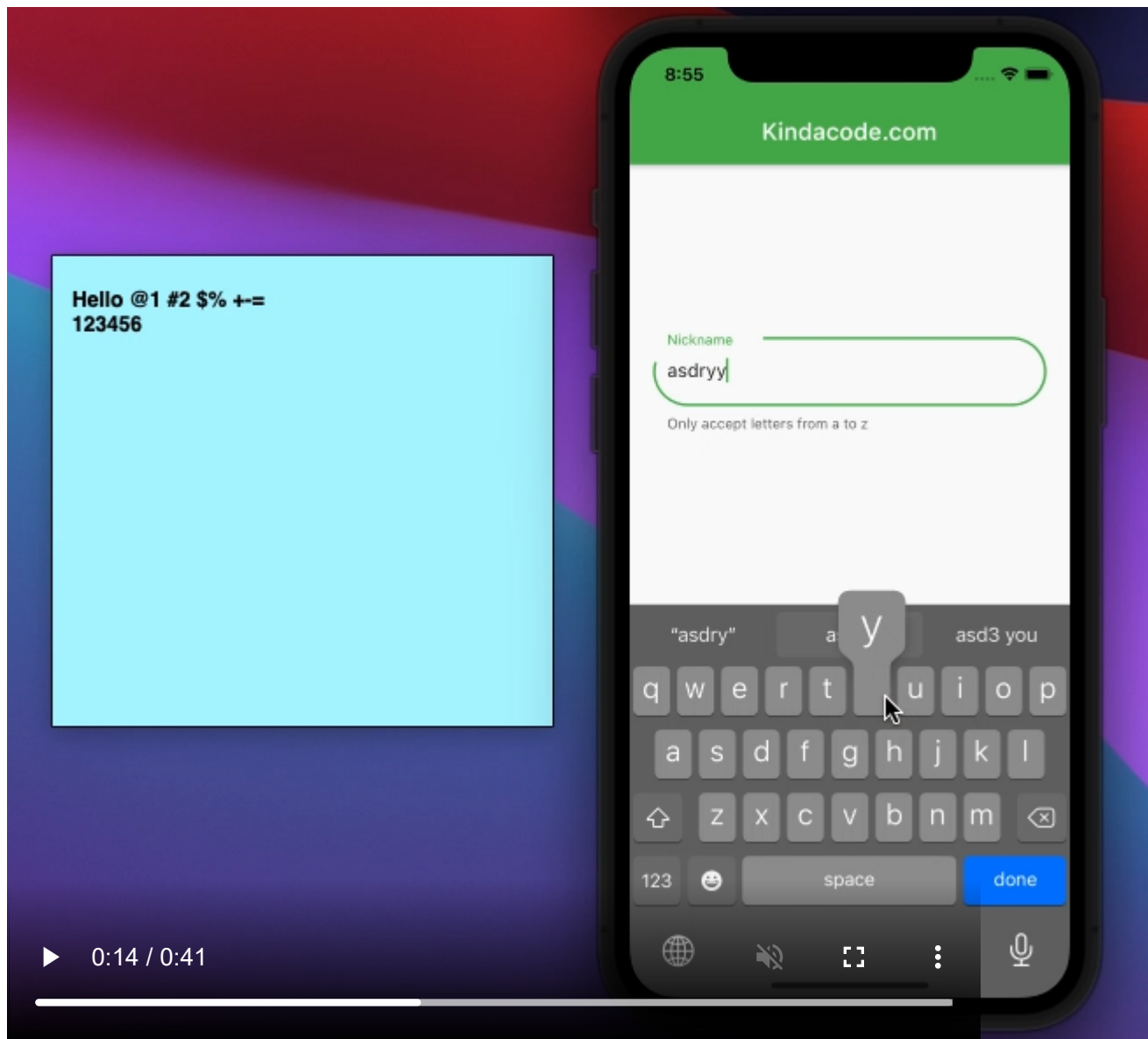
2.2. The Code

3. Conclusion

Example 1: Only Allowing Alphabet Letters

Preview

Let's say we have a text field that lets the user enter his or her nickname. Only letters in the alphabet are allowed (both lowercase and uppercase). No numbers, symbols, emojis are accepted regardless of whether they are entered from the keyboard or copied/pasted from somewhere else.



The code

Here's the complete code that produces the app shown in the video above:

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

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
```

```
@override
Widget build(BuildContext context) {
  return MaterialApp(
    // Remove the debug banner
    debugShowCheckedModeBanner: false,
    title: 'Kindacode.com',
    theme: ThemeData(
      primarySwatch: Colors.green,
    ),
    home: HomePage());
}

class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Kindacode.com'),
      ),
      body: Center(
        child: Padding(
          padding: const EdgeInsets.all(20),
          child: TextField(
            inputFormatters: [
              // only accept letters from a to z
              FilteringTextInputFormatter(RegExp(r'[a-zA-Z]'), allow: true)
            ],
            decoration: InputDecoration(
              labelText: 'Nickname',
              helperText: 'Only accept letters from a to z',
              border: OutlineInputBorder(
                borderRadius: BorderRadius.circular(50))),
          ),
        ),
      ),
    );
  }
}
```

```
    ),  
    ),  
    ),  
  );  
}  
}
```

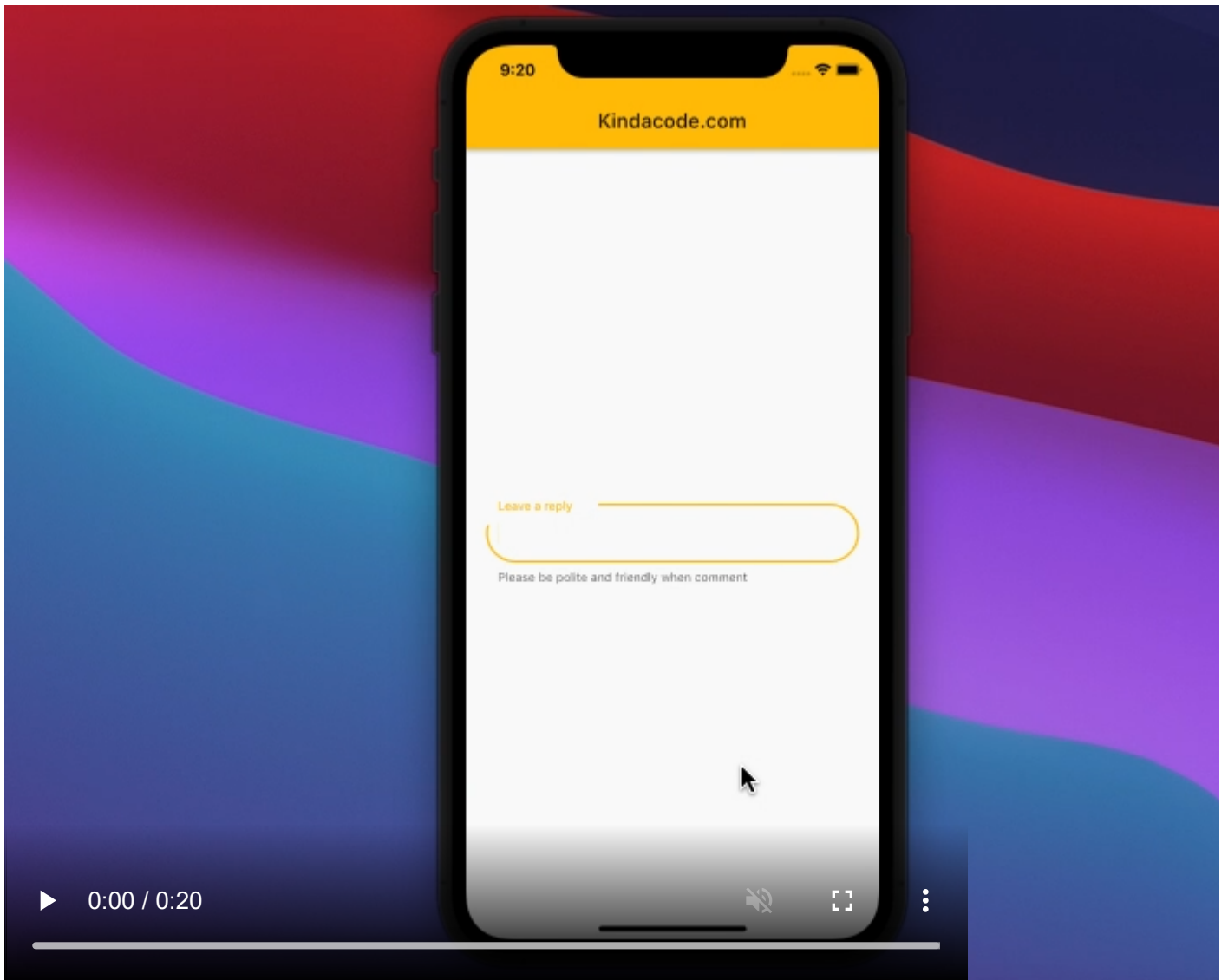
Example 2: Preventing Offensive Words from Being Displayed

In this example, we'll use the **FilteringTextInputFormatter.deny** constructor to blacklist some words.

Preview

Let's say we have a social app where users can make comments. Because we don't want toxic people to say offensive words on our platform, we'll define a list of banned words. When a user types a word that exists in the blacklist, it will be replaced with "***".

In the demo, we'll prevent only 3 words: "fool", "donkey", "idiot" from being displayed. However, you can add as many as you want.



The Code

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

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
```

```
// Remove the debug banner
debugShowCheckedModeBanner: false,
title: 'Kindacode.com',
theme: ThemeData(
  primarySwatch: Colors.amber,
),
home: HomePage());
}
}

class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  _HomePageState createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Kindacode.com'),
      ),
      body: Center(
        child: Padding(
          padding: const EdgeInsets.all(20),
          child: TextField(
            inputFormatters: [
              // Everything is allowed except banned words
              FilteringTextInputFormatter.deny(
                RegExp(r'(idiot|donkey|fool)', caseSensitive: false),
                replacementString: '***')
            ],
            decoration: InputDecoration(
              labelText: 'Leave a reply',
              helperText: 'Please be polite and friendly when comment',
              border: OutlineInputBorder(
                borderRadius: BorderRadius.circular(50))),
          ),
        ),
      ),
    );
  }
}
```

```
    ),  
  ),  
);  
}  
}
```

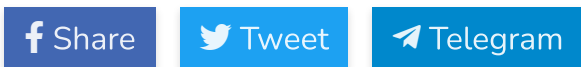
Conclusion

We've gone through 2 examples of implementing **FilteringTextInputFormatter** in an app. You can find more information about that class on flutter.dev.

If you'd like to explore more new and fascinating stuff about Flutter and Dart, take a look at the following articles:

- [Flutter & Dart: Regular Expression Examples](#)
- [Flutter form validation example](#)
- [Flutter: Show/Hide Password in TextField/TextFormField](#)
- [Working with dynamic Checkboxes in Flutter](#)
- [Flutter: Caching Network Images for Big Performance gains](#)

You can also take a tour around our [Flutter topic page](#) and [Dart topic page](#) to see the latest tutorials and examples.



✉ Subscribe ▼



Be the First to Comment!

1000

B *I* U