**L10N Flutter Cheat Sheet**

1. Add the *flutter\_localizations* package to the dependencies in the pubspec.yaml file.
2. Add the *intl* package as well

dependencies:

  flutter:

    sdk: flutter

  flutter\_localizations:

    sdk: flutter

  intl: any

1. Go down in the *pubspec.yaml* and put *generate: true*

flutter:

generate: true

1. Get the dependencies.
2. Go to *main.dart*.
3. Add *supportedLocales: L10n.all* to the *MaterialApp*

supportedLocales: L10n.all,

1. Create a folder called *l10n*
2. Create an *l10n.dart* file.
3. Follow this syntax:

import 'package:flutter/material.dart';

class L10n {

  static final all = [

    const Locale('en'),

    const Locale('es'),

    const Locale('bg'),

  ];

}

1. Add an *l10n.yaml* file in the root directory
2. Add these 3 lines of code

arb-dir: lib/l10n

template-arb-file: app\_en.arb

output-localization-file: app\_localizations.dart

1. Create *app\_en.arb* file in the *l10n* folder.
2. Follow this pattern and make sure you don't add a trailing comma.

{

"date": "Date",

"@date": {

"description": "Date"

},

"cancel": "Cancel",

"@cancel": {

"description": "Cancel"

}

}

1. For all the other languages, follow this pattern:

{

"date": "Fecha",

"cancel": "Cancelar"

}

1. You might want to make use of this Python script:

data = {

    "en": [

        ["user\_language", "English"],

        "Select date",

        "Cancel",

    ],

    "es": [

        ["user\_language", "Español"],

        "Seleccionar fecha",

        "Cancelar",

    ],

    "bg": [

        ["user\_language", "Български"],

        "Избор на дата",

        "Отказ",

    ],

}

def string\_to\_camel\_case(string):

    result = "".join([word.capitalize() for word in string.split(" ")])

    return result[0].lower() + result[1:]

for i in range(len(data.items())):

    result = ""

    language = list(data.items())[i][0]

    language\_data = list(data.items())[i][1]

    english\_language\_data = list(data.items())[0][1]

    for text, english\_text in zip(language\_data, english\_language\_data):

        if type(text) == list:

            if language == "en":

                result += f'    "{string\_to\_camel\_case(english\_text[0])}": "{text[1]}",\n'

                result += f'    "@{string\_to\_camel\_case(english\_text[0])}": {{\n'

                result += f'        "description": "{text[1]}"\n'

                result += "    },\n"

            else:

                result += f'    "{string\_to\_camel\_case(english\_text[0])}": "{text[1]}",\n'

        else:

            if language == "en":

                result += f'    "{string\_to\_camel\_case(english\_text)}": "{text}",\n'

                result += f'    "@{string\_to\_camel\_case(english\_text)}": {{\n'

                result += f'        "description": "{text}"\n'

                result += "    },\n"

            else:

                result += f'    "{string\_to\_camel\_case(english\_text)}": "{text}",\n'

    result = result[:-2]

    result = "{\n" + result + "\n}"

    with open(f"app\_{language}.arb", "w", encoding="utf-8") as file:

        file.write(result)

1. Add this import statement in *main.dart*:

import 'package:flutter\_gen/gen\_l10n/app\_localizations.dart';

1. Add this property to the *MaterialApp*

      localizationsDelegates: const [

        AppLocalizations.delegate,

        GlobalMaterialLocalizations.delegate,

        GlobalCupertinoLocalizations.delegate,

        GlobalWidgetsLocalizations.delegate,

      ],

1. Use *AppLocalization.of(context)!* To access your translations.