

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
```

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

```
flutter:  
  generate: true
```

4. Get the dependencies.
5. Go to main.dart.
6. Add supportedLocales: L10n.all to the MaterialApp

```
supportedLocales: L10n.all,
```

7. Create a folder called l10n
8. Create an l10n.dart file.
9. Follow this syntax:

```
import 'package:flutter/material.dart';  
  
class L10n {  
  static final all = [  
    const Locale('en'),  
    const Locale('es'),  
    const Locale('bg'),  
  ];  
}
```

10. Add an l10n.yaml file in the root directory
11. Add these 3 lines of code

```
arb-dir: lib/l10n  
template-arb-file: app_en.arb  
output-localization-file: app_localizations.dart
```

12. Create app_en.arb file in the l10n folder.
13. Follow this pattern and make sure you don't add a trailing comma.

```
{  
  "date": "Date",  
  "@date": {  
    "description": "Date"  
  },  
  "cancel": "Cancel",  
  "@cancel": {  
    "description": "Cancel"  
  }  
}
```

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

```
{
    "date": "Fecha",
    "cancel": "Cancelar"
}
```

15. 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 += f'    },\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 += f'    },\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)
```

16. Add this import statement in main.dart:

```
import 'package:flutter_gen/gen_l10n/app_localizations.dart';
```

17. Add this property to the MaterialApp

```
localizationsDelegates: const [  
    AppLocalizations.delegate,  
    GlobalMaterialLocalizations.delegate,  
    GlobalCupertinoLocalizations.delegate,  
    GlobalWidgetsLocalizations.delegate,  
],
```

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