Colorful Flutter Logs

Friday, December 9, 2022 3:35 PM

Logging - Logging is the process printing information about your code.

You can install Logging package in flutter then create a function where you can define your logs .

1.How to use Logging?

```
import 'dart:developer' as developer;

import 'package:logging/logging.dart';



void main() async {

  //logging in flutter
  initRootLogger();
  runApp(App());
}
```

2.Create function

```
void initRootLogger() {
  Logger.root.level = Level.ALL;
  Logger.root.onRecord.listen((record) {
    // set ansi color code
    var start = '\x1b[90m';
    const end = '\x1b[0m';
    const white = '\x1b[37m';
    switch (record.level.name) {
      case 'INFO':
        start = '\x1b[32m';
        // start = '\x1b[37m';
        break;
      case 'WARNING':
        start = '\x1b[93m';
        break;
```

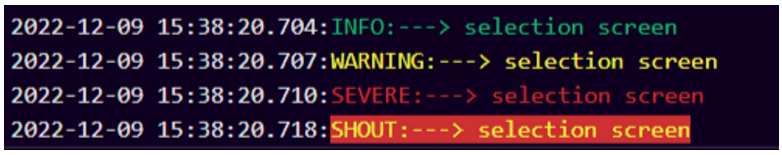
```
    case 'SEVERE':
      start = '\x1b[31m';
      break;
    case 'SHOUT':
      start = '\x1b[41m\x1b[93m';
      break;
  }
  final message =
      '$white${record.time}:$end$start${record.level.name}:---> ${record.message}$end ';
  developer.log(
    message,
    name: record.loggerName.padRight(12),
    level: record.level.value,
    time: record.time,
  );
 });
}
```

3.Then create instance of Logging and use log variable to print logs -

```
final log = Logger('LEAP INDIA');
```

4.You can use now

```
  log.info('selection screen');
   log.warning('selection screen');
   log.severe('selection screen');
   log.shout('selection screen');
```




```
// if only enable logging for debug mode
 if (kDebugMode) {
   Logger.root.level = Level.ALL;
 } else {
   Logger.root.level = Level.OFF;
 }
```

ANSI CODE---

```
Black("30"),
 Red("31"),
 Green("32"),
 Yellow("33"),
 Blue("34"),
 Magenta("35"),
 Cyan("36"),
 White("37");
```

2. You can Print logs in different color using custom function

```
voidmain() {
  print('This is a normal message.');
  printWarning('This is a warning.');
  printError('This is an error.');
}
voidprintWarning(Stringtext) {
  print('\x1B[33m$text\x1B[0m');
}
voidprintError(Stringtext) {
  print('\x1B[31m$text\x1B[0m');
}
```