



## 02-05 flutter provider的使用

Posted on 2020-11-23 23:34 肖无情 阅读(145) 评论(0) 编辑 收藏 举报

```
class StateManagerByProviderModel with ChangeNotifier{
  int count = 0;
  void increment(){
    count ++;
    notifyListeners();
  }
}
```

```
var pmodel = StateManagerByProviderModel();

var providers = Providers();
providers..provide(Provider<StateManagerByProviderModel>.value(pmodel));
runApp(ProviderNode(child: MyApp(), providers: providers));
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: WQAppBar(
      titleName: ""
    ).appBar(context),
    body: Center(
      child: GestureDetector(
        onTap: (){

        },
        child: Provide(
          builder: (context,child,StateManagerByProviderModel scope){
            return Text("${scope.count}");
          },
        ),
      ),
    ),
  );
}
```

```
child: GestureDetector(
  onTap: (){
    Provide.value<StateManagerByProviderModel>(context).increment();
    Navigator.of(context).push(CupertinoPageRoute(builder: (context){
      return StateManagerShowPage();
    }));
  },
  child: Text("点击"),
),
```

### 封装之后

```
class ThemeConfigInfo {
  Color themeColor = color00CCBB;
}

class ThemeConfigModel extends ThemeConfigInfo with ChangeNotifier{

  Future setThemeColor(themeColor) async{
    this.themeColor = themeColor;
    notifyListeners();
  }
}
```

```
class ProvierHelper{

  static var providers = Providers();
  static var themeConfigModel = ThemeConfigModel();

  static init({model,child,dispose = true}){

    providers = Providers()
      ..provide(Provider<ThemeConfigModel>.value(themeConfigModel));

    return ProviderNode(child: child, providers: providers,dispose: dispose);
  }

  static connect<T>({builder, child, scope}) {
    return Provide<T>(builder: builder, child: child, scope: scope);
  }

  static T value<T>(context, {scope}){
    return Provide.value<T>(context,scope: scope);
  }
}
```

### main文件中

```
runApp(ProvierHelper.init(child: MyApp()));
```

```
Widget build(BuildContext context) {  
  return ProvierHelper.connect<ThemeConfigModel>(  
    builder: (context, child, ThemeConfigModel model) {  
      return MaterialApp(  
        title: "app",  
        localizationsDelegates: [  
          // 本地化的代理类  
          GlobalMaterialLocalizations.delegate,  
          GlobalWidgetsLocalizations.delegate,  
          WQCommonLocalizationsDelegate(),  
          WQLocalDelegate(),  
        ],  
        supportedLocales: [  
          const Locale('en', 'US'), // 美国英语  
          const Locale('zh', 'CN'), // 中文简体  
          //其它Locales  
        ],  
        //这句话是直接设置中文, 国际化配置也会失效  
        locale: const Locale('zh', 'CN'), // 中文简体  
  
        theme: ThemeData(  
          primaryColor: model.themeColor,  
          highlightColor: Colors.transparent,  
          splashColor: Colors.transparent,  
        ),  
        routes: WQRoute.routes,  
        home: Builder(  
          builder: (context) {  
            return TabBarController();  
          },  
        ),  
      );  
    },  
  );  
}
```

#### 改变一个按钮的颜色

```
Widget build(BuildContext context) {  
  return Scaffold(  
    body: ListView(  
      children: <Widget>[  
        ProvierHelper.connect<ThemeConfigModel>(  
          builder: (context, child, ThemeConfigModel model) {  
            return RaisedButton(  
              color: model.themeColor,  
              onPressed: () {  
                Navigator.push(context, CupertinoPageRoute(builder: (context) {  
                  return SettingPage();  
                }));  
              },  
              child: Text("设置"),  
            );  
          },  
        ),  
      ],  
    ),  
  );  
};
```

#### 改变主题颜色

```
void changeTheme() {  
  ProvierHelper.value<ThemeConfigModel>(context).setThemeColor(colorList[selectedIndex]);  
}
```

分类: flutter

标签: flutter



肖无情  
粉丝 - 2 关注 - 5

+加关注

« 上一篇: 02-04 flutter 构造函数总结

» 下一篇: 02-06 flutter 依赖管理 sdk 介绍

0 推荐  
0 反对

刷新评论 刷新页面 返回顶部

登录后才能查看或发表评论, 立即 登录 或者 逛逛 博客园首页

【推荐】阿里云新人特惠, 爆款云服务器2核4G低至0.46元/天

编辑推荐:

- gRPC 入门与实操 (.NET 篇)
- dotnet 代码优化 聊聊逻辑圈复杂度
- 一个棘手的生产问题，但是我写出来之后，就是你的了
- 你可能不知道的容器镜像安全实践
- .Net 6 使用 Consul 实现服务注册与发现

阅读排行:

- Redux与前端表格施展“组合拳”，实现大屏展示应用的交互增强
- gRPC入门与实操(.NET篇)
- 博客园主题修改分享 - 过年篇
- 如何优雅地校验后端接口数据，不做前端背锅侠
- 产品与研发相处之道