# 13.4 国际化常见问题

本节主要解答一下在国际化中常见的问题。

### 默认语言区域不对

在一些非大陆行货渠道买的一些Android和iOS设备，会出现默认的Locale不是中文简体的情况。这属于正常现象，但是为了防止设备获取的Locale与实际的地区不一致，所有的支持多语言的APP都必须提供一个手动选择语言的入口。

### 如何对应用标题进行国际化

MaterialApp有一个title属性，用于指定APP的标题。在Android系统中，APP的标题会出现在任务管理器中。所以也需要对title进行国际化。但是问题是很多国际化的配置都是在MaterialApp上设置的，我们无法在构建MaterialApp时通过Localizations.of来获取本地化资源，如：

MaterialApp(  
 title: DemoLocalizations.of(context).title, //不能正常工作！  
 localizationsDelegates: [  
 // 本地化的代理类  
 GlobalMaterialLocalizations.delegate,  
 GlobalWidgetsLocalizations.delegate,  
 DemoLocalizationsDelegate() // 设置Delegate  
 ],  
);

上面代码运行后，DemoLocalizations.of(context).title 是会报错的，原因是Localizations.of会从当前的context沿着widget树向顶部查找DemoLocalizations，但是我们在MaterialApp中设置完DemoLocalizationsDelegate后，实际上DemoLocalizations是在当前context的子树中的，所以DemoLocalizations.of(context)会返回null，报错。那么我们该如何处理这种情况呢？其实很简单，我们只需要设置一个onGenerateTitle回调即可：

MaterialApp(  
 onGenerateTitle: (context){  
 // 此时context在Localizations的子树中  
 return DemoLocalizations.of(context).title;  
 },  
 localizationsDelegates: [  
 DemoLocalizationsDelegate(),  
 ...  
 ],  
);

### 如何为英语系的国家指定同一个locale

英语系的国家非常多，如美国、英国、澳大利亚等，这些英语系国家虽然说的都是英语，但也会有一些区别。如果我们的APP只想提供一种英语（如美国英语）供所有英语系国家使用，我们可以在前面介绍的localeListResolutionCallback中来做兼容：

localeListResolutionCallback:  
 (List<Locale> locales, Iterable<Locale> supportedLocales) {  
 // 判断当前locale是否为英语系国家，如果是直接返回Locale('en', 'US')   
}