

一、课前准备

- Flutter环境

二、课堂目标

- 完成掘金APP首页剩余部分

三、知识要点

- Image
- BottomNavigationBar (Tab路由)
- Navigator路由
- Icon/ImageIcon
- 在flutter中使用iconfont(神奇的一段js代码帮助Flutter使用iconfont)
- 保持BottomNavigationBar页的状态
 - AutomaticKeepAliveClientMixin
 - BottomNavigationBar+PageView
 - BottomNavigationBar+IndexedStack (不推荐)
- WebView

四、总结

- Flutter框架在使用上和前端开发习惯还是有很大不同的，但理解规则后也不难上手

五、作业

- 实现一个BottomNavigationBar，有三个Tab分别为“首页”，“发现”，“我的”，实现点击Tab切换currentIndex。

使用第三方网络请求库

```
http: ^0.12.2  
dio: 3.0.10
```

FadeInImage

transparent_image: 1.0.0

```
FadeInImage.memoryNetwork(  
  placeholder: kTransparentImage,  
  image: 'https://github.com/flutter/website/blob/master/_includes/code/layout/lakes/images/lake.jpg?raw=true',  
)
```

Image

```
// 加载网络图片  
Image.network(String src, {  
  Key key,  
  double scale = 1.0,  
  this.semanticLabel,  
  this.excludeFromSemantics = false,  
  this.width,  
  this.height,  
  this.color,  
  this.colorBlendMode,  
  this.fit,  
  this.alignment = Alignment.center,  
  this.repeat = ImageRepeat.noRepeat,  
  this.centerSlice,  
  this.matchTextDirection = false,  
  this.gaplessPlayback = false,  
  this.filterQuality = FilterQuality.low,  
  Map<String, String> headers,
```

```
} )
```

```
// 从项目资源中加载
```

```
Image.asset(String name, {  
    Key key,  
    AssetBundle bundle,  
    this.semanticLabel,  
    this.excludeFromSemantics = false,  
    double scale,  
    this.width,  
    this.height,  
    this.color,  
    this.colorBlendMode,  
    this.fit,  
    this.alignment = Alignment.center,  
    this.repeat = ImageRepeat.noRepeat,  
    this.centerSlice,  
    this.matchTextDirection = false,  
    this.gaplessPlayback = false,  
    String package,  
    this.filterQuality = FilterQuality.low,  
})
```

```
// 加载本地文件
```

```
Image.file  
Image.memory
```

BottomNavigationBar

```
required this.items,  
this.onTap,  
this.currentIndex = 0,  
this.elevation,  
this.type,  
Color? fixedColor,  
this.backgroundColor,  
this.iconSize = 24.0,  
Color? selectedItemColor,  
this.unselectedItemColor,  
this.selectedIconTheme,  
this.unselectedIconTheme,  
this.selectedFontSize = 14.0,  
this.unselectedFontSize = 12.0,  
this.selectedLabelStyle,  
this.unselectedLabelStyle,  
this.showSelectedLabels,  
this.showUnselectedLabels,  
this.mouseCursor
```

Icon (MaterialDesigns)

```
Icon(Icons.home)
```

Navigator路由

```
Navigator.push(context, MaterialPageRoute(builder:
(BuildContext context){
  return PageSearch();
}));
```

- 命名导航路由

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new MaterialApp(
      title: 'Navigation',
      initialRoute: '/',
      routes: <String, WidgetBuilder>{
        '/': (BuildContext context) => new
PageHome(),
        '/search': (BuildContext context) => new
PageSearch(),
      },
    );
  }
}
```

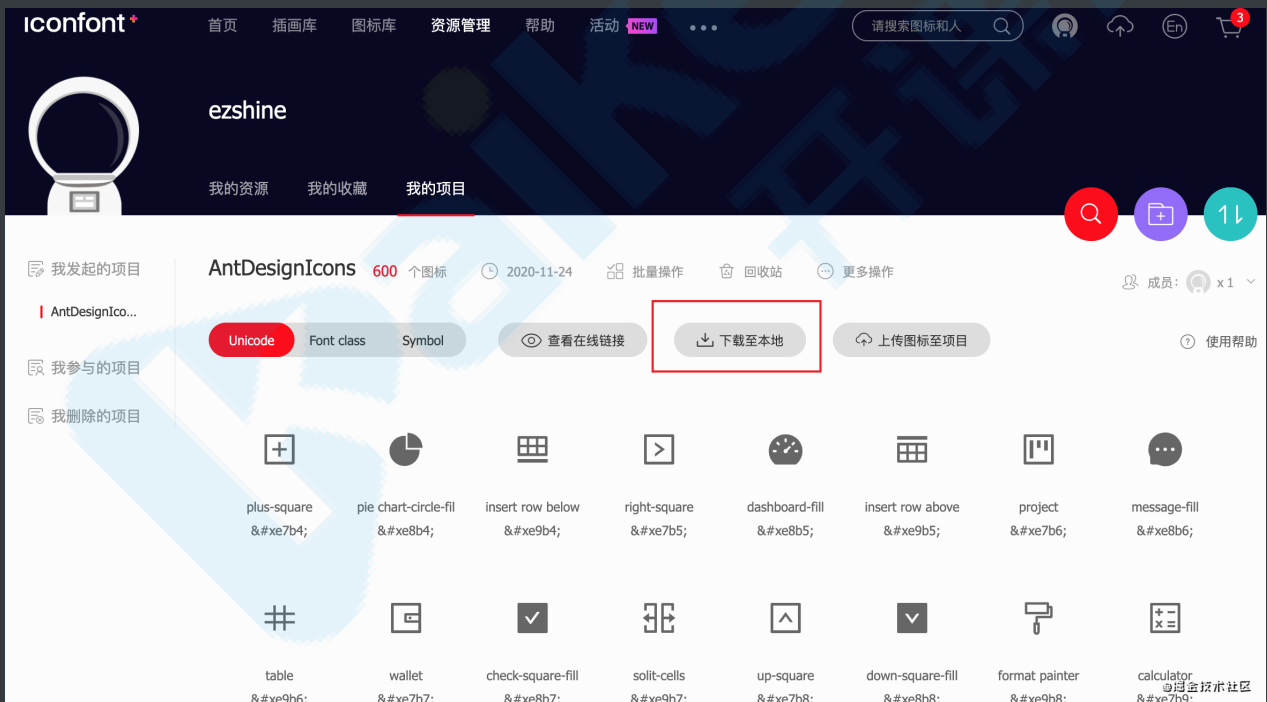
```
Navigator.of(context).pushNamed("/search");
```

ImageIcon

```
ImageIcon(AssetImage("assets/images/dashboard.png"))
```

使用IconFont

从我的项目中进入该项目，并选择下载至本地



将下载好的zip包解压缩，复制其中的iconfont.ttf文件至你的flutter项目中，比如 flutter项目名称/assets/fonts/ 内



font_2226161_wyss8445ws



名称



修改日期



demo_index.html

今天 下午 8:49



demo.css

今天 下午 8:49



iconfont.css

今天 下午 8:49



iconfont.eot

今天 下午 8:49



iconfont.js

今天 下午 8:49



iconfont.json

今天 下午 8:49



iconfont.svg

今天 下午 8:49



iconfont.ttf

今天 下午 8:49



iconfont.woff

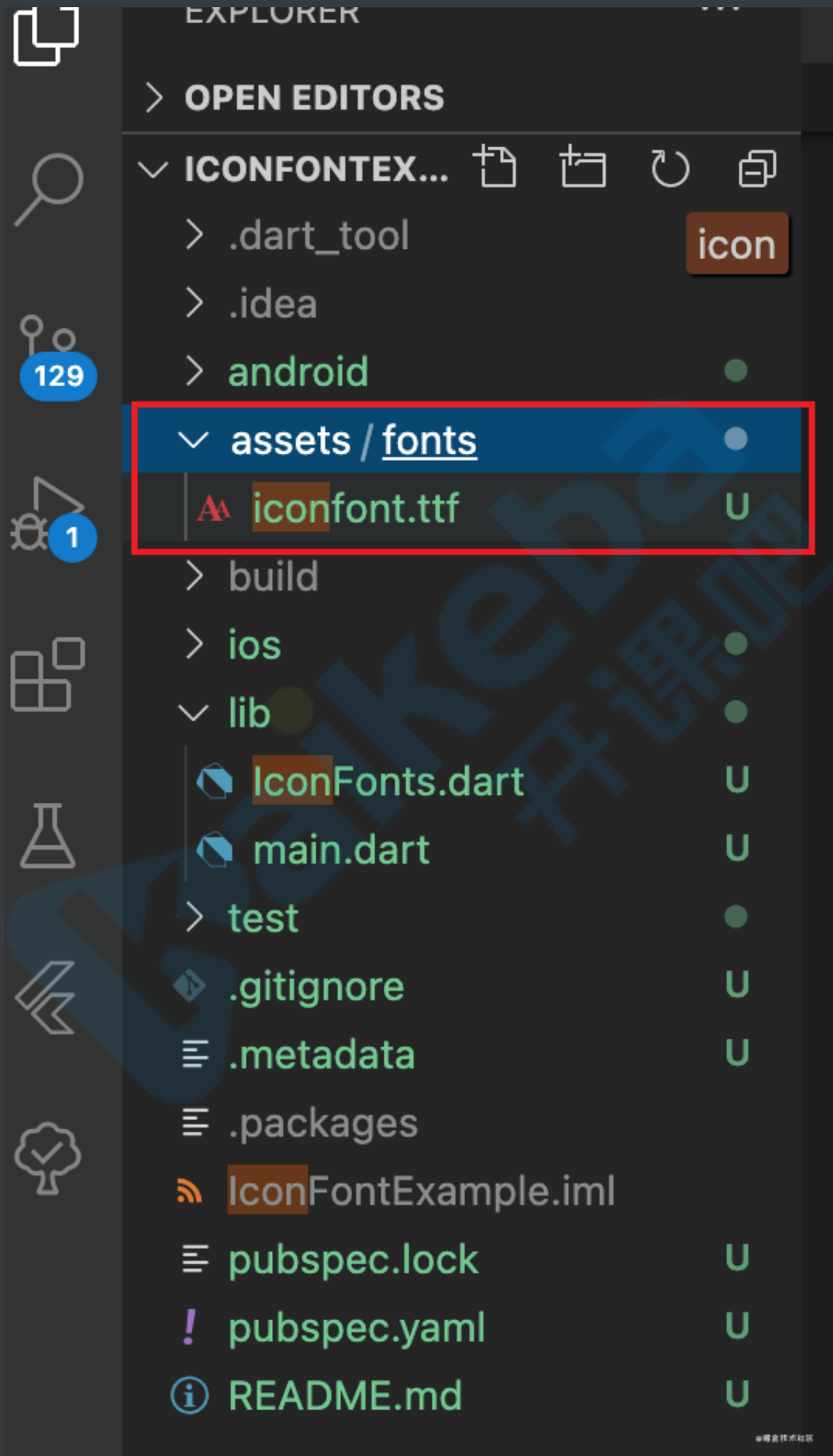
今天 下午 8:49



iconfont.woff2

今天 下午 8:49

掘金技术社区



在Flutter中要使用自定义字体，我们需要在 `pubspec.yaml` 文件中添加以下内容

```
fonts:
  - family: IconFont
    fonts:
      - asset: assets/fonts/iconfont.ttf
```

复制代码

请注意这个结构，不要写错了。通常情况下，在 `pubspec.yaml` 中有相关的注释，按照注释的结构写即可。

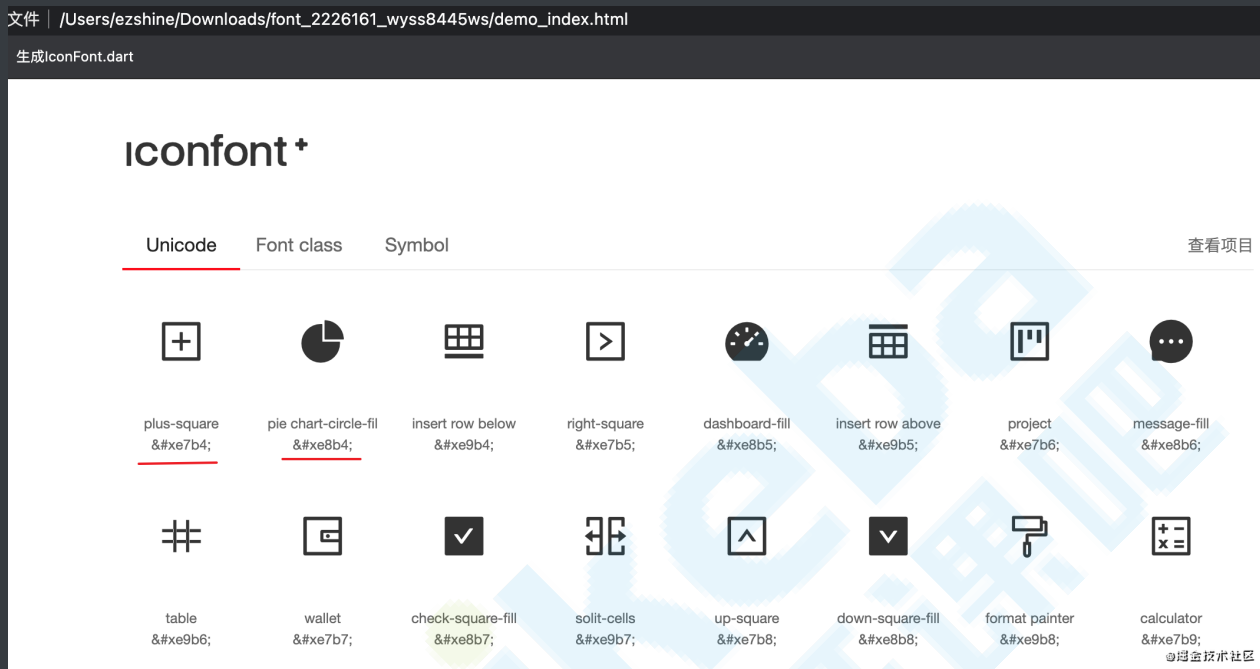
使用IconData

现在我们已经可以正常使用自定义的iconfont了，用法如下

```
Icon(
  IconData(0xe7b4, fontFamily: 'IconFont'),
  size: 20,
  color: Colors.black
)
```

复制代码

其中fontFamily的值'IconFont'就是我们刚才在 pubspec.yaml 中声明的新字体，但是代码中的0xe7b4是指什么呢？回到之前下载解压zip包的文件夹，双击demo_index.html文件在浏览器中打开后，我们可以看到下面的画面



每个图标下面都标记出了这个图标对应的unicode编码，所以我们想用哪个图标，只要copy它的unicode编码到代码里就行了

自定义Icon类

可如果我们有多个Icon，使用unicode非常不友好...读到代码的时候根本不知道是什么吧！所以我们需要把这个东西变得更 优雅 一些。把它们统统装进一个dart类集中管理，并通过图标的名称来使用。

```
class IconFontIcons {  
    static const IconData iconRotateRight =  
IconData(0xe9b3, fontFamily: 'IconFont');  
    static const IconData iconHeartFill =  
IconData(0xe8b3, fontFamily: 'IconFont');  
    static const IconData iconMinusSquare =  
IconData(0xe7b3, fontFamily: 'IconFont');  
    ...  
}
```

复制代码

再使用的话，就非常简单直观了

```
Icon(IconFontIcons.iconRotateRight)
```

复制代码

浏览器插件：一键下载 IconFont.dart

<https://juejin.cn/post/6898690760892710926>

使用 IndexedStack 配合 BottomNavigationBar

```
IndexedStack(  
  index: bottomNavigationBarIndex,  
  children: [  
    PageIndex(),  
    PageMy()  
  ],  
)
```

使用 PageView 配合BottomNavigationBar

PageView有点像是

```
PageView(  
  physics:NeverScrollableScrollPhysics(),//禁止滚动  
  controller: _pageController,  
  children: [  
    PageMy(),  
    PageIndex()  
  ],  
)
```

AutomaticKeepAliveClientMixin

在需要缓存的页面使用

```
with AutomaticKeepAliveClientMixin
```

```
@override
```

```
bool get wantKeepAlive => true;
```

WebView

webview_flutter: 1.0.7

```
<key>io.flutter.embedded_views_preview</key>  
<string>YES</string>
```

```
import 'package:flutter/material.dart';  
import 'package:flutter/widgets.dart';  
import 'dart:io';  
import  
'package:webview_flutter/webview_flutter.dart';
```

```
class PageArticle extends StatefulWidget {  
  final String url;  
  PageArticle(this.url, {Key key}) : super(key:  
key);
```

```
@override
```

```
    _PageArticleState createState() =>
    _PageArticleState();
  }

  class _PageArticleState extends State<PageArticle>
  {
    @override
    Widget build(BuildContext context) {
      return Scaffold(
        appBar: AppBar(title:Text("webview")),
        body: WebView(
          initialUrl: widget.url,
        ),
      );
    }
  }
}
```

作业

实现一个BottomNavigationBar，有三个Tab分别为“首页”，“发现”，“我的”，实现点击Tab切换currentIndex。

暗号：我要涨薪