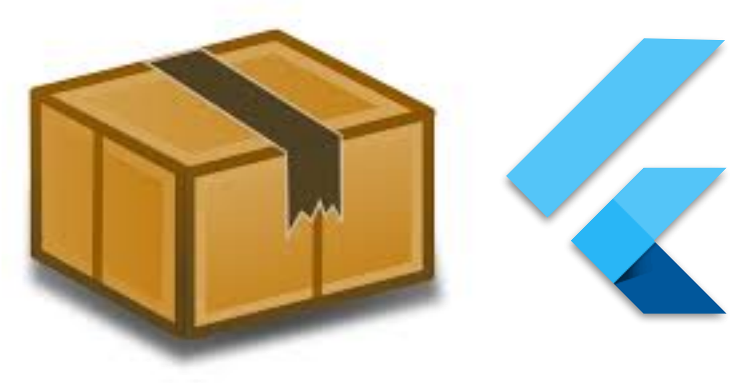
Chapter V

Third Party Flutter Packages



1.Shared Preferences
2.Cached Network Image
3.Image Picker
4.Image Cropper
5.Multi Image Picker
6.Image Gallery Saver
7.Carousel Pro
8.Permission Handler
9.Location
10.Flutter Swipper
11.Url Launcher
12.Connectivity
13.Toast

1.Shared Preferences

Add plugin name in pubspec.yaml as:

shared_preferences

Usage:

```
var prefs = await SharedPreferences.getInstance();
int runTimeCount = (prefs.getInt('runtime') ?? 0);
 ++runTimeCount;
prefs.setInt('runtime', runTimeCount);
print(runTimeCount)
```

2.Cached Network Image

cached_network_image (Declaration in pubspec.yaml)

Usage:

```
CachedNetworkImage(
       imageUrl: part.image,

       fit: BoxFit.cover,

       width: double.maxFinite,

       height: size.height / 5,

     ),

   ), //As a widget


 CachedNetworkImageProvider(
       part.image,

       fit: BoxFit.cover,

       width: double.maxFinite,

       height: size.height / 5,

     ),

   ),//ImageProvider instead allows you to identify an image
```

3.Image Picker

image_picker (Declaration in pubspec.yaml)

```
List<File> imageFiles = []; //For use as array if need
File imageFile
File picture = await ImagePicker.pickImage(source:
        ImageSource.gallery);

this.setState(() {

        imageFile = picture;

});
```

Note:Add Privacy key in info.plist for iOS

```
<key>NSCameraUsageDescription</key>

<string>Camera Access Warning</string>

<key>NSMicrophoneUsageDescription</key>

<string>Microphone Access Warning</string>

<key>NSPhotoLibraryUsageDescription</key>

<string>Photo Library Access Warning</string>
```

And, in Android declared in AndroidManifest.xml

```
<activity

        android:name="com.yalantis.ucrop.UCropActivity"

        android:screenOrientation="portrait"

        android:theme="@style/Theme.AppCompat.Light.NoActionBar"/

        >
```

Full Example

```
    _openGalleryForImage(BuildContext context) async {

            var picture = await ImagePicker.pickImage(source:
                            ImageSource.gallery);
                    this.setState(() {
                    File imageFile = picture;

                    });
                    Navigator.of(context).pop();
             }

     _openCameraForImage(BuildContext context) async {
            var picture = await ImagePicker.pickImage(source:
            ImageSource.camera);
             this.setState(() {
                    File imageFile = picture;
                    imageFiles.add(imageFile);
                    });
                    Navigator.of(context).pop();
            }
Future<void> _showChoiceDialogForImage(BuildContext context) {
  return showDialog(
      context: context,
      builder: (BuildContext context) {
        return CupertinoAlertDialog(
          title: new Text("Make a choice"),
          content: new Text("You can upload image using..."),
          actions: <Widget>[
            CupertinoDialogAction(
              isDefaultAction: true,
              child: Text("Camera"),
              onPressed: () {
                _openCameraForImage(context);
              },
            ),
            CupertinoDialogAction(
              isDefaultAction: true,
              child: Text("Gallery"),
              onPressed: () {
                _openGalleryForImage(context);
              },
            )
          ],
        );
      });
}
```

4.Image Cropper

image_cropper: ^0.0.4

```
File croppedFile = await ImageCropper.cropImage(
  sourcePath: image.path,
  // ratioX: 1.0,
  // ratioY: 1.0,
  // maxWidth: 512,
  // maxHeight: 512,
);
}
```

5.Multi Image Picker

multi_image_picker:4.0.3

```
    List<UploadFileInfo> files;


    List<Asset> resultList = List<Asset>( );
resultList = await MultiImagePicker.pickImages(
  maxImages: 10,
  enableCamera: false,
);
```

Full Example:

```
List<UploadFileInfo> files;


Future getImage() async {


    List<Asset> resultList = List<Asset>();
    resultList = await MultiImagePicker.pickImages(
      maxImages: 10,
      enableCamera: false,
    );

    if (resultList.length == 0) return;

    for (var asset in resultList) {
      int MAX_WIDTH = 300; //keep ratio
      int height =
          ((MAX_WIDTH * asset.originalHeight) / asset.originalWidth).round();

      ByteData byteData =
          await asset.requestThumbnail(MAX_WIDTH, 300, quality: 80);

      if (byteData != null) {
        List<int> imageData = byteData.buffer.asUint8List();
        UploadFileInfo u = UploadFileInfo.fromBytes(imageData, asset.name);

        files.add(u )
      }
    }

    setState(() {

    });
  }
```

6.Image Gallery Saver

image_gallery_saver

Usage:

```
var result = await ImageGallerySaver.saveImage(imageData);
```

Full Example

```
import 'dart:convert';
import 'dart:typed_data';
import 'dart:ui';
import 'package:image_gallery_saver/image_gallery_saver.dart';


Future<Uint8List> _capturePng() async {
    try {
      RenderRepaintBoundary boundary =
          _globalKey.currentContext.findRenderObject();
      var image = await boundary.toImage(pixelRatio: 3.0);
      ByteData byteData = await image.toByteData(format:
ImageByteFormat.png);
      var pngBytes = byteData.buffer.asUint8List();
      var bs64 = base64Encode(pngBytes);

      setState(() {});
      return pngBytes;
    } catch (e) {
      print(e);
    }


var imageData = await _capturePng();
                var result = await ImageGallerySaver.saveImage(imageData);
```

7.Carousel Pro

carousel_pro

Usage

```
Carousel(
 boxFit: BoxFit.cover,

 images: //Image here,

autoplay: true,

 // animationCurve: Curves.fastOutSlowIn,

 // animationDuration: Duration(milliseconds: 1),

 dotSize: 5.0,

 dotColor: Colors.white,

 showIndicator: true,

 indicatorBgPadding: 8.0,

 dotBgColor: Colors.transparent,

),
```

8.Permission Handler

Usage

```
PermissionHandler()
.requestPermissions(<PermissionGroup>[PermissionGroup.storage]);
```

Note

For iOS

```
<key>NSLocationAlwaysUsageDescription</key>
        <string>Always</string>
<key>NSLocationWhenInUseUsageDescription</key>
        <string>To provide location based services</string>
```

For android to access location. Add following line in AndroidManifest.xml

```
<uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION"/>
    <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

9.Location

location: ^2.3.5

Usage

```
        StreamSubscription<LocationData> _locationSubscription;


         Location _locationService = new Location();
        bool _permission = false;
        String error;

        initPlatformState() async {
        await _locationService.changeSettings(
       accuracy: LocationAccuracy.HIGH, interval: 1000);

    LocationData location;

    try {
      bool serviceStatus = await _locationService.serviceEnabled();
      print("Service status: $serviceStatus");
      if (serviceStatus) {
        _permission = await _locationService.requestPermission();

        if (_permission) {
```

```
        location = await _locationService.getLocation();

        CurrentLocation = location;
      }
    } else {
      bool serviceStatusResult = await _locationService.requestService();
      print("Service status activated after request: $serviceStatusResult");
      if (serviceStatusResult) {
        initPlatformState();
      }
    }
  } on PlatformException catch (e) {
    print(e);
    if (e.code == 'PERMISSION_DENIED') {
      error = e.message;
    } else if (e.code == 'SERVICE_STATUS_ERROR') {
      error = e.message;
    }
    location = null;
  }
}
```

10.Flutter Swipper

flutter_swiper

Usage

```
Swiper.children(
autoplay: false,

index: 0,

loop: false,

pagination: SwiperPagination(

  margin: EdgeInsets.fromLTRB(0.0, 0.0, 0.0, 40.0),

  builder: DotSwiperPaginationBuilder(

      color: Colors.grey,

      activeColor: appColor,

      size: 6.5,

      activeSize: 8.0),

),
    control: SwiperControl(
      iconPrevious: null,
      iconNext: null,
    ),
    children: _getPages(context),
  ),
```

11.Url Launcher

<Refer to package link>

12.Connectivity

<Refer to package link>

13.Toast

<Refer to package link>