C🍪🍪KIE-APP

Convergent Capstone Design 2
Week 10 - Progress Report

Group 6 Park JongBeum & Baek SeungHeon, November 6th 2023

Recall

Recall

What we've done last week

- Location Sharing through socket.io
- ChatRoom Creation & Join Implementation
- iPhone Notification Test
- Refactor Widget Tree - Scoping Provider
- MUI3 Adoption

Progress

Maps Tab

Maps Tab
Marker

- We will display the location information of our friends with a "marker"
- Currently, we can share locations through "socket.io"
- So, We use "custom_marker" package

Maps Tab
custom_marker

- Manage Image Cache
- Support various formats
- Easy image processing

Maps Tab

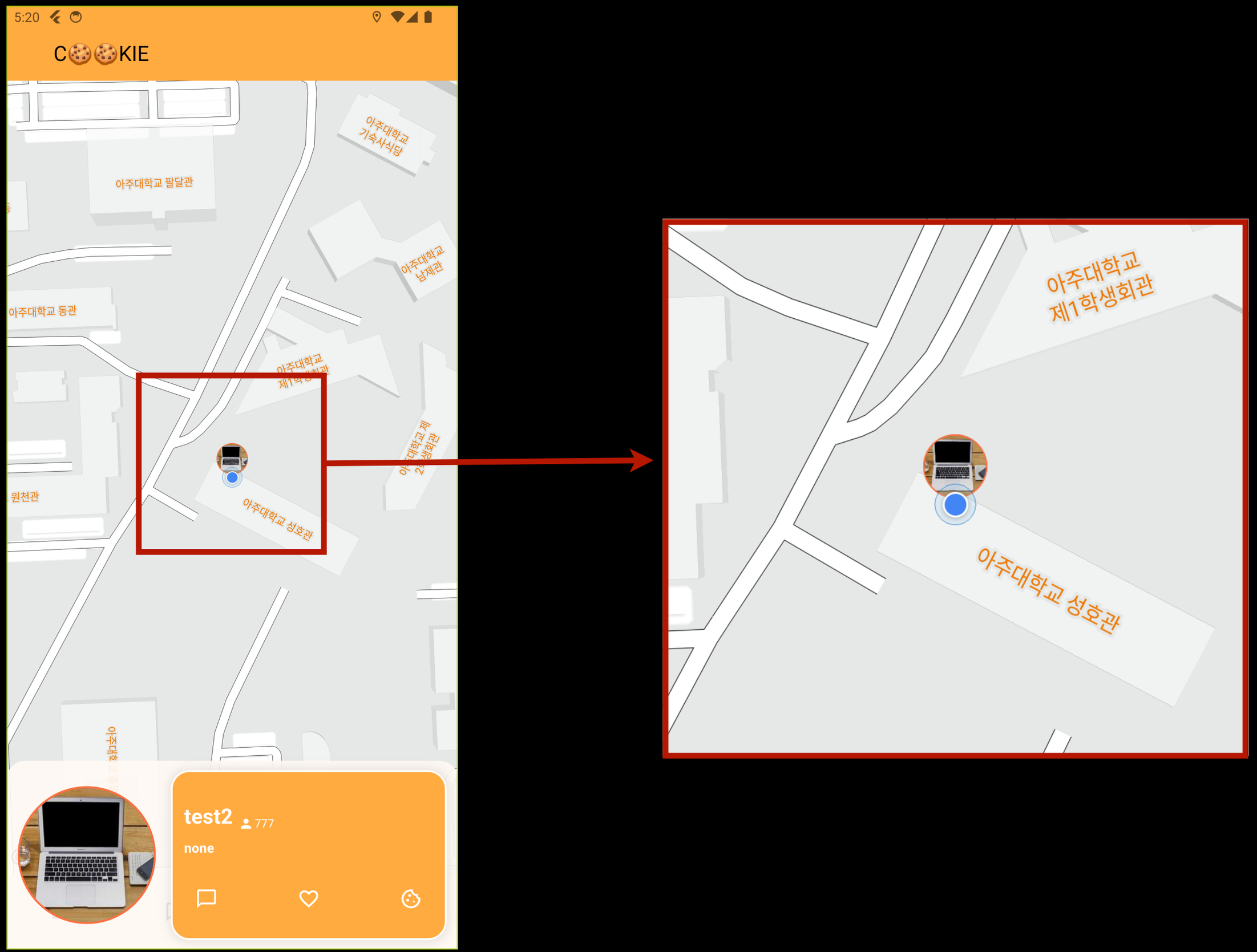
custom_marker

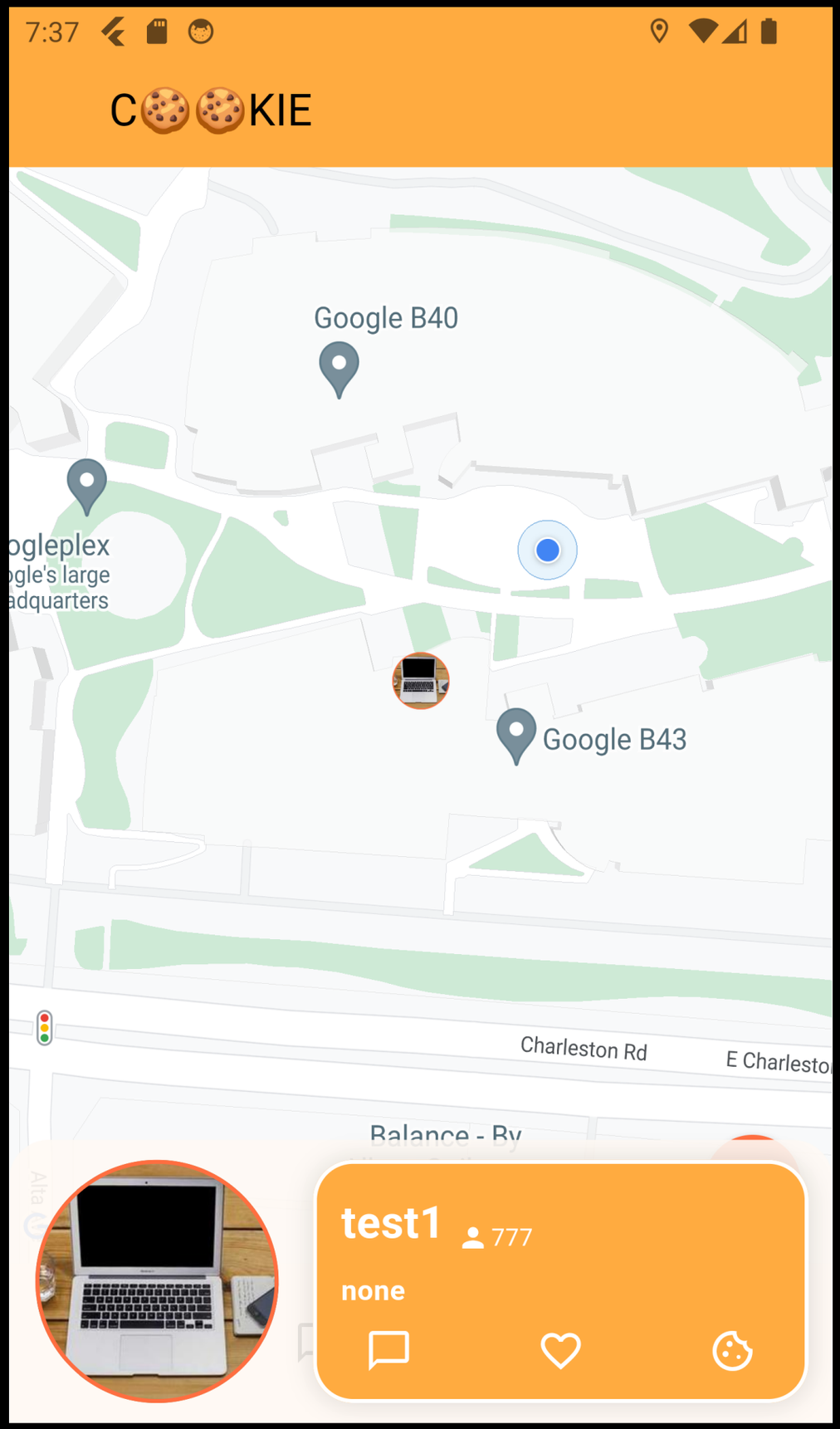
```dart
Future<Uint8List> getRoundedImage(
  ImageProvider image, {
  int width = 100,
  Color borderColor = Colors.deepOrangeAccent,
  double borderWidth = 4,
}) async {
  final imageStream = image.resolve(ImageConfiguration.empty);
  final completer = Completer<Uint8List>();

  imageStream.addListener(
    ImageStreamListener((ImageInfo info, bool _) async {
      final roundedImage = await _createRoundedImage(
        info.image,
        width,
        borderColor,
        borderWidth,
      );

      completer.complete(roundedImage);
    }),
  );

  return completer.future;
}

Future<Uint8List> _createRoundedImage(
  ui.Image image,
  int width,
  Color borderColor,
  double borderWidth,
) async {
  final paintRect = Offset.zero & Size(width.toDouble(), width.toDouble());
  final pictureRecorder = ui.PictureRecorder();
  final canvas = Canvas(pictureRecorder);

  final paint = Paint()..isAntiAlias = true;

  final double radius = width.toDouble() / 2;

  final borderPaint = Paint()
    ..color = borderColor
    ..style = PaintingStyle.stroke
    ..strokeWidth = borderWidth;

  canvas.saveLayer(paintRect, Paint()); // Create a layer

  // Draw circular image
  canvas.drawCircle(Offset(radius, radius), radius, paint);

  // Clip the image
  canvas.clipPath(Path()..addOval(paintRect));

  // Draw the image
  canvas.drawImageRect(
    image,
    Rect.fromLTRB(0, 0, image.width.toDouble(), image.height.toDouble()),
    paintRect,
    paint,
  );

  // Draw the border
  canvas.drawCircle(
    Offset(radius, radius),
    radius - borderWidth / 2,
    borderPaint,
  );

  canvas.restore(); // Restore the layer

  final picture = pictureRecorder.endRecording();
  final roundedImage = await picture.toImage(width, width);
  final byteData =
      await roundedImage.toByteData(format: ui.ImageByteFormat.png);

  if (byteData != null) {
    return byteData.buffer.asUint8List();
  } else {
    throw Exception('Failed to convert image to byte data.');
  }
}
```

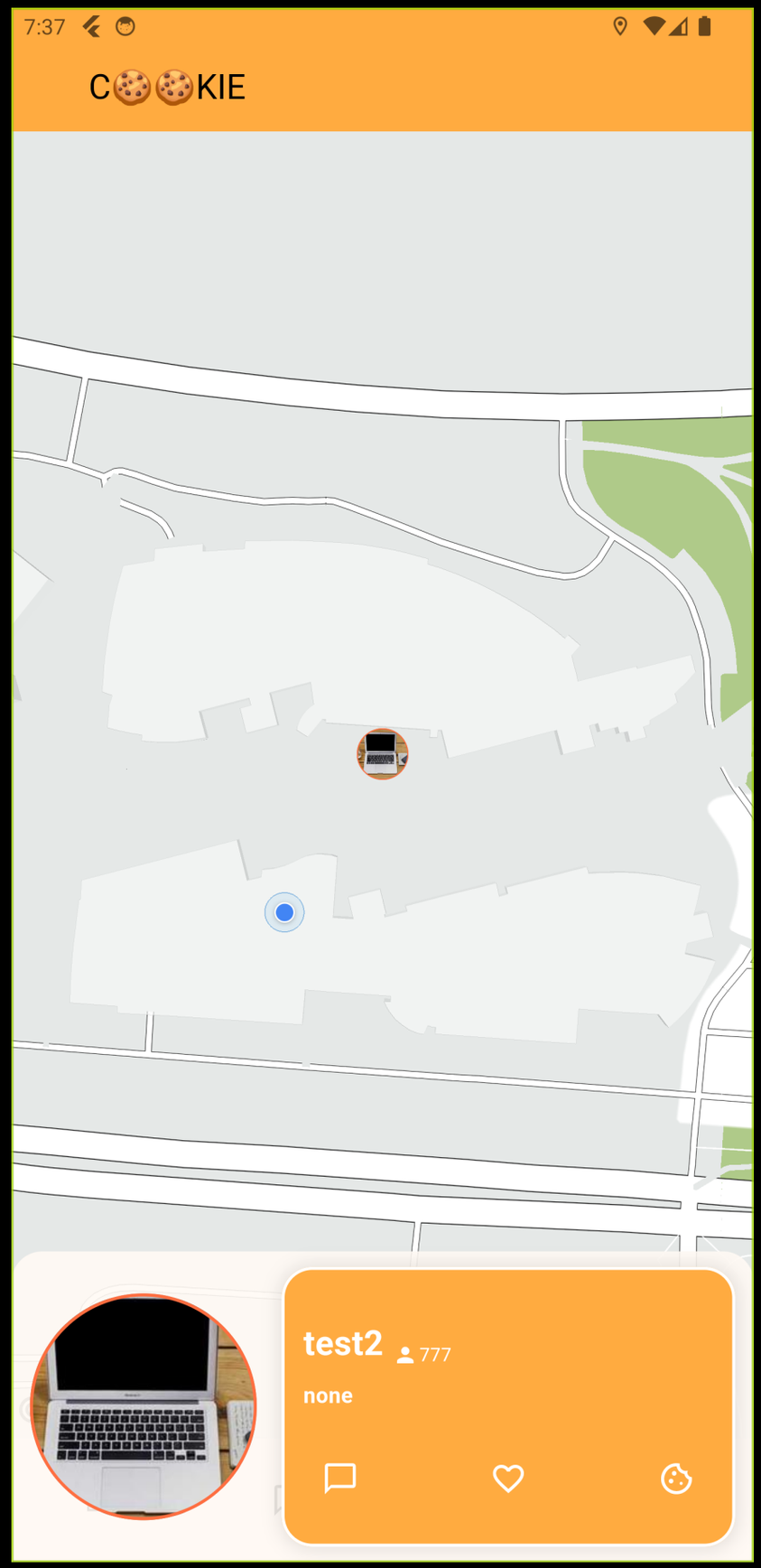
```dart
Uint8List markIcons = await getRoundedImage(
  user.info.profile.image as ImageProvider<Object>,
  width: size,
  borderColor: color,
  borderWidth: width,
);
return Marker(
  markerId: MarkerId(user.info.name.toString()),
  icon: BitmapDescriptor.fromBytes(markIcons),
```

Maps Tab
custom_marker

```
icon: await MarkerIcon.downloadResizePictureCircle(
    friendInfo.profile.image.toString(),
    size: size,
    addBorder: true,
    borderColor: color,
    borderSize: width,
)
```


5:20
C🍪🍪KIE
아주대학교 팔달관
아주대학교 기숙사식당
아주대학교 남제관
아주대학교 동관
아주대학교 제1학생회관
원천관
아주대학교 성호관
test2
777
none
아주대학교 제1학생회관
아주대학교 성호관


7:37
C🍪🍪KIE
Google B40
ogleplex
gle's large
adquarters
Google B43
Charleston Rd
E Charleston
Balance - By
test1
777
none


7:37
C🍪🍪KIE
test2
777
none

API Call

Retrofit + Dio

RetrofitGenerator

```dart
@RestApi(baseUrl: "http://localhost:3000")
abstract class RestClient {
  factory RestClient(Dio dio, {String baseUrl}) = _RestClient;

  @GET("/account/info")
  Future<InfoResponse> getInfo({
    @Query("fields") List<String>? fields,
  });
}
```

Retrofit Generator

Automatically generates code to request

```dart
// **************************************************************************
// RetrofitGenerator
// **************************************************************************

// ignore_for_file: unnecessary_brace_in_string_interps,no_leading_underscores_for_local_identifiers

class _RestClient implements RestClient {
  _RestClient(
    this._dio, {
    this.baseUrl,
  }) {
    baseUrl ??= 'http://localhost:3000';
  }

  final Dio _dio;

  String? baseUrl;

  @override
  Future<InfoResponse> getInfo({List<String>? fields}) async {
    const _extra = <String, dynamic>{};
    final queryParameters = <String, dynamic>{r'fields': fields};
    queryParameters.removeWhere((k, v) => v == null);
    final _headers = <String, dynamic>{};
    final Map<String, dynamic>? _data = null;
    final _result = await _dio
        .fetch<Map<String, dynamic>>(_setStreamType<InfoResponse>(Options(
      method: 'GET',
      headers: _headers,
      extra: _extra,
    )
            .compose(
              _dio.options,
              '/account/info',
              queryParameters: queryParameters,
              data: _data,
            )
            .copyWith(
                baseUrl: _combineBaseUrls(
              _dio.options.baseUrl,
              baseUrl,
            ))));
    final value = InfoResponse.fromJson(_result.data!);
    return value;
  }

  RequestOptions _setStreamType<T>(RequestOptions requestOptions) {
    if (T != dynamic &&
        !(requestOptions.responseType == ResponseType.bytes ||
            requestOptions.responseType == ResponseType.stream)) {
      if (T == String) {
        requestOptions.responseType = ResponseType.plain;
      } else {
        requestOptions.responseType = ResponseType.json;
      }
    }
    return requestOptions;
  }

  String _combineBaseUrls(
    String dioBaseUrl,
    String? baseUrl,
  ) {
    if (baseUrl == null || baseUrl.trim().isEmpty) {
      return dioBaseUrl;
    }

    final url = Uri.parse(baseUrl);

    if (url.isAbsolute) {
      return url.toString();
    }

    return Uri.parse(dioBaseUrl).resolveUri(url).toString();
  }
}
```

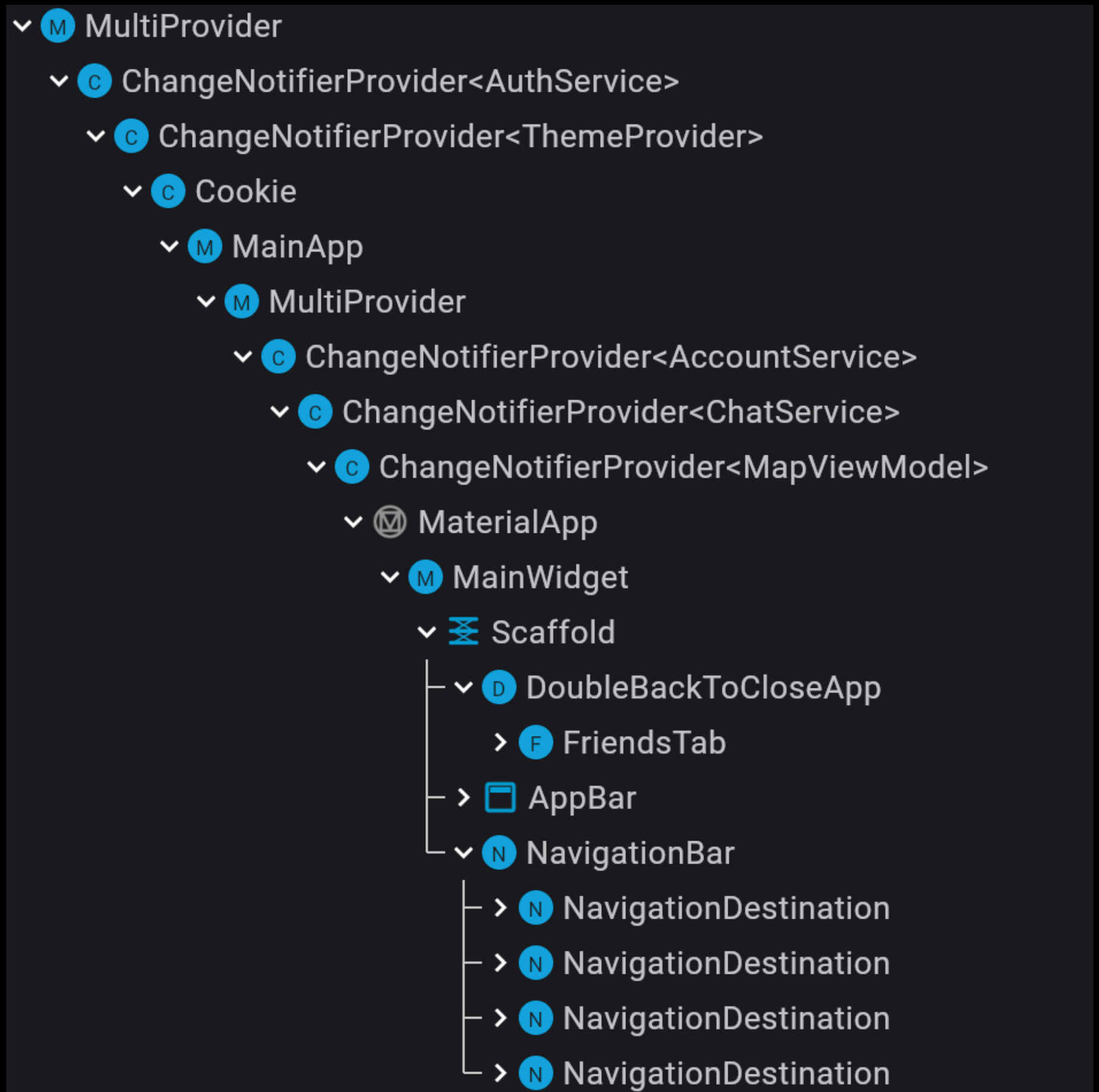
Interceptor

```dart
_dio.interceptors.addAll([
  responseLogger,
  ErrorInterceptor(),
  accessTokenInterceptor,
  refreshTokenInterceptor,
  updateAccessTokenInterceptor,
  updateRefreshTokenInterceptor,
]);

_api = AuthRestClient(_dio, baseUrl: dotenv.env['BASE_URI']!);
```

```dart
InterceptorsWrapper get accessTokenInterceptor {
  return InterceptorsWrapper(
    onRequest: (options, handler) {
      if (this._accessToken != null) {
        options.headers[ACCESS_TOKEN_HEADER] = this._accessToken;
        logger.i(
          'Interceptor injects "AccessToken" header: ${this._accessToken}',
        );
      }
      return handler.next(options);
    },
  );
}
```

AuthViewModel

AuthService


MultiProvider
ChangeNotifierProvider<AuthService>
ChangeNotifierProvider<ThemeProvider>
Cookie
MainApp
MultiProvider
ChangeNotifierProvider<AccountService>
ChangeNotifierProvider<ChatService>
ChangeNotifierProvider<MapViewModel>
MaterialApp
MainWidget
Scaffold
DoubleBackToCloseApp
FriendsTab
AppBar
NavigationBar
NavigationDestination
NavigationDestination
NavigationDestination
NavigationDestination

ViewModel?
Service?

Too much Business Logics in ViewModel! Doesn’t it rather looks like a Service?

Refactor Architecture MVVM+S

View

ViewModel

Model

View

Service

ViewModel

Model

Simplified ViewModels

abstract AccountViewModel

PublicAccountViewModel

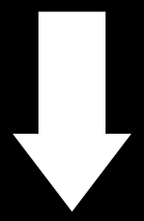
PrivateAccountViewModel

FriendsViewModel

PublicAccountViewModel

+

PrivateAccountViewModel



AccountViewModel

AccountService

```
AccountViewModel my;
Map<String, AccountViewModel> friends;
Future<void> update();
```

View

AccountService

AccountViewModel

AccountModel

Git Branching..

main / dev

Thank You

Questions are welcome