

앱 개발 입문

성균관대학교 컬처애편테크놀로지융합전공 하계 부트캠프

8일차 - 2022. 08. 05 (금)

강의 슬라이드 링크

https://github.com/kunny/skku-bootcamp-2022-summer/blob/main/_slides/day8.pdf

오늘 강의에서 다룰 내용

- 사진 검색 API를 구현한 PhotoService 클래스를 다시 살펴봅니다.
- AnalyticsService와 PhotoService의 인스턴스를 관리하는 `providers.dart` 파일을 작성합니다.
- 검색어 입력 화면과 검색 결과 조회 화면을 작성한 후, 두 화면을 연결해줍니다.
- 완성된 앱을 크롬에서 실행해봅니다.
- 앱 광고를 소개합니다.
- 다양한 포맷의 광고를 앱에 적용하는 과정을 살펴봅니다. (라이브 데모)

PhotoService 구현 리뷰

PhotoService 구현 리뷰

```
const _accessKey = '[개발자 계정에서 발급받은 Access Key]';

class PhotoService {
  Future<SearchResult> search(String searchTerm, int page) async {
    try {
      final response = await Dio().get(
        'https://api.unsplash.com/search/photos',
        queryParameters: {
          'query': searchTerm,
          'page': page,
        },
        options: Options(
          headers: {
            'Accept-Version': 'v1',
            'Authorization': 'Client-ID $_accessKey',
          },
        ),
      );
      return SearchResult.from(response.data);
    } catch (e) {
      return Future.error(e);
    }
  }
}
```

photo_service.dart

providers.dart 작성하기

providers.dart 작성하기

```
AnalyticsService? _analyticsService;

PhotoService? _photoService;

AnalyticsService analyticsService() {
  if (_analyticsService == null) {
    _analyticsService = AnalyticsService();
  }
  return _analyticsService!;
}

PhotoService photoService() {
  if (_photoService == null) {
    _photoService = PhotoService();
  }
  return _photoService!;
}
```

providers.dart 작성하기

```
AnalyticsService? _analyticsService;
```

```
PhotoService? _photoService;
```

```
AnalyticsService analyticsService() {  
  return _analyticsService ??= AnalyticsService();  
}
```

```
PhotoService photoService() {  
  return _photoService ??= PhotoService();  
}
```

if (something == null) something = Something() 의 축약 표현

providers.dart 작성하기

```
AnalyticsService? _analyticsService;
```

```
PhotoService? _photoService;
```

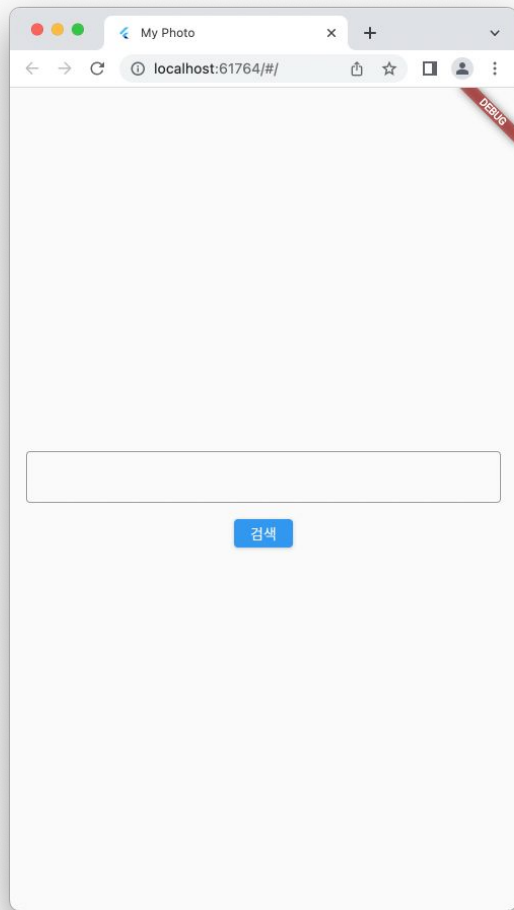
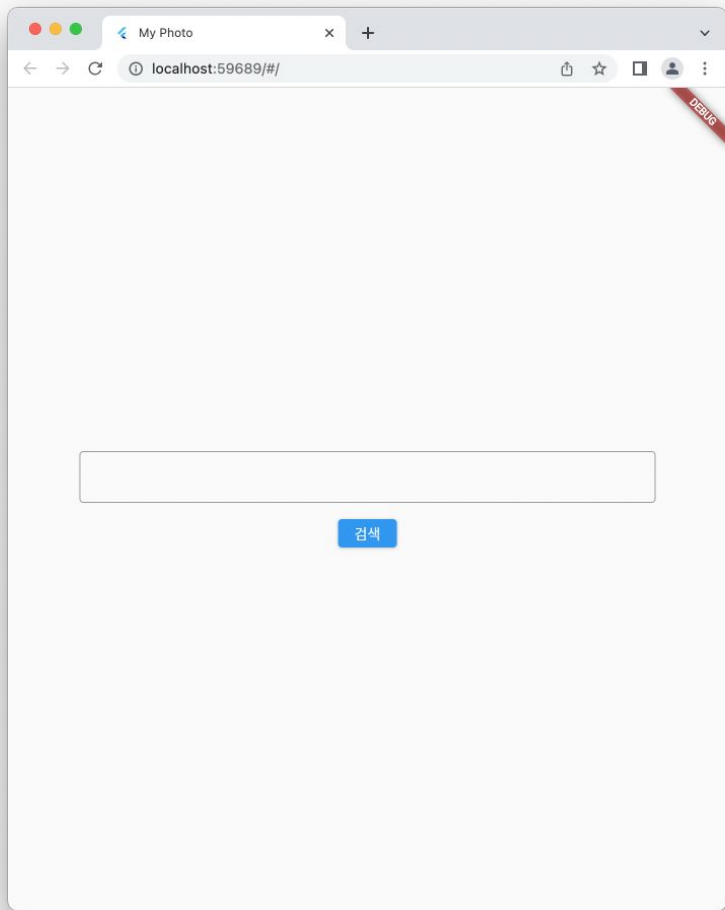
```
AnalyticsService analyticsService() => _analyticsService ??= AnalyticsService();
```

```
PhotoService photoService() => _photoService ??= PhotoService();
```

검색어 입력 화면 작성하기

검색어 입력 화면 작성하기

1. `page` 폴더 아래에 `home_page.dart` 파일을 추가합니다.
2. 검색어 입력 페이지를 표시하는 위젯인 `HomePage`를 구현합니다.



검색어 입력 화면 작성하기

```
class HomePage extends StatelessWidget {  
  static const routeName = '/';  
  
  final _searchTermController = TextEditingController();  
  
  HomePage({Key? key}) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) { ... }  
}
```

검색어 입력 화면 작성하기

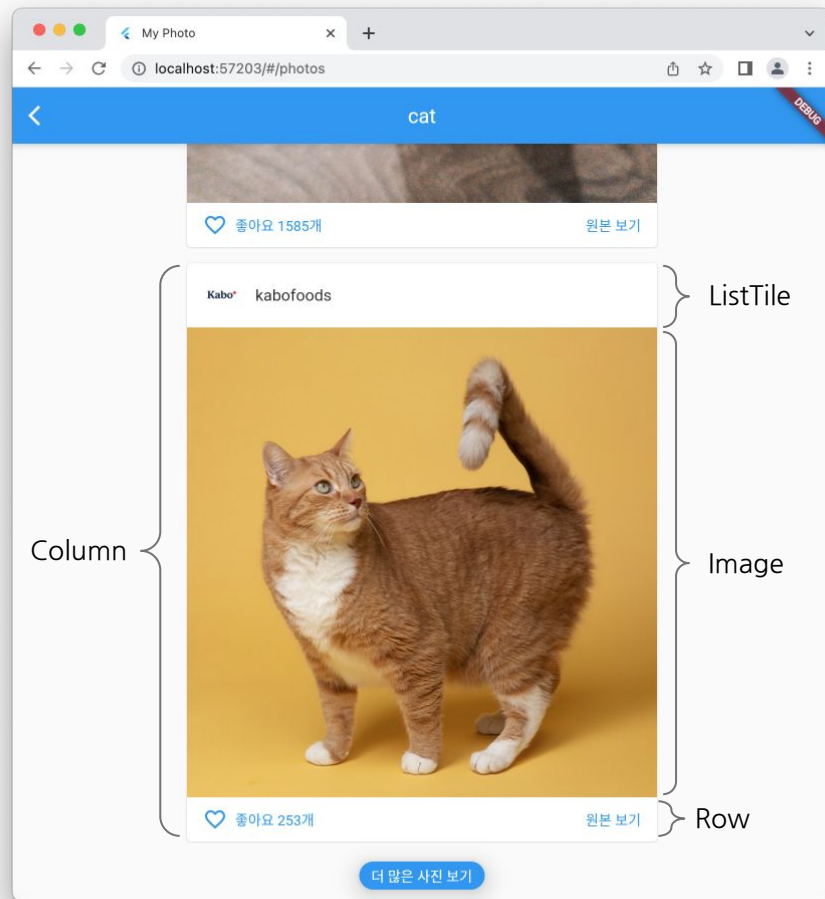
```
class HomePage extends StatelessWidget {  
  ...  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      body: Center(  
        child: ConstrainedBox(  
          constraints: const BoxConstraints(  
            minWidth: 240,  
            maxWidth: 600,  
          ),  
        child: Padding(  
          padding: const EdgeInsets.symmetric(horizontal: 16.0),  
          child: Column(  
            mainAxisAlignment: MainAxisAlignment.min,  
            children: [  
              TextField(  
                controller: _searchTermController,  
                decoration: const InputDecoration(  
                  border: OutlineInputBorder(),  
                ),  
              ),  
              const SizedBox(height: 16.0),  
              ElevatedButton(  
                onPressed: () {  
                  final searchTerm = _searchTermController.text.trim();  
                  analyticsService().search(searchTerm);  
                  _searchTermController.text = '';  
                },  
                child: const Text('검색'),  
              ),  
            ],  
          ),  
        ),  
      ),  
    );  
  }  
}
```

home_page.dart

검색 결과 조회 화면 작성하기

검색 결과 조회 화면 작성하기

1. 외부 링크를 열 수 있게 해주는 `url_launcher` 패키지 추가하기
2. `page` 폴더 아래에 `search_result_page.dart` 파일을 추가합니다.
3. 검색 결과를 표시하는 위젯인 `SearchResultPage`를 구현합니다.



외부 링크를 열 수 있게 도와주는 패키지 설치하기

pubspec.yaml 파일을 연 후, dependencies 섹션에 url_launcher 패키지를 추가합니다.

```
dependencies:  
  flutter:  
    sdk: flutter  
  url_launcher: ^6.1.4
```

검색 결과 조회 화면 작성하기

```
class SearchResultPage extends StatefulWidget {  
  static const routeName = '/photos';  
  
  final String searchTerm;  
  
  const SearchResultPage(this.searchTerm, {Key? key}) : super(key: key);  
  
  @override  
  State createState() => _SearchResultPageState();  
}
```

검색 결과 조회 화면 작성하기

```
class _SearchResultPageState extends State<SearchResultPage> {  
  final photos = <Photo>[];  
  
  int currentPage = 1;  
  
  int? totalPages;  
  
  @override  
  Widget build(BuildContext context) { ... }  
}
```

검색 결과 조회 화면 작성하기

```
class _SearchResultPageState extends State<SearchResultPage> {  
  ...  
  @override  
  void initState() {  
    super.initState();  
    _fetchPhotos();  
  }  
  
  ...  
  void _fetchPhotos() {  
    photoService().search(widget.searchTerm, currentPage).then((result) {  
      setState(() {  
        currentPage++;  
        totalPages = result.totalPages;  
        photos.addAll(result.results);  
      });  
    }, onError: (e) {  
      debugPrint(e.toString());  
    });  
  }  
}
```

search_result_page.dart

검색 결과 조회 화면 작성하기

```
class _SearchResultPageState extends State<SearchResultPage> {  
  ...  
  Widget _buildCard(Photo photo) {  
    return Align(  
      alignment: Alignment.center,  
      child: SizedBox(  
        width: 500,  
        child: Card(  
          margin: const EdgeInsets.symmetric(vertical: 8.0, horizontal: 16.0),  
          child: Column(  
            children: [ ... ],  
          ),  
        ),  
      ),  
    );  
  }  
}
```

[전체 소스코드 보기](#)

검색 결과 조회 화면 작성하기

```
class _SearchResultPageState extends State<SearchResultPage> {
  ...
  Widget _buildList() {
    final hasNextPage = totalPages != null && currentPage < totalPages!;
    return ListView.builder(
      itemCount: hasNextPage ? photos.length + 1 : photos.length,
      itemBuilder: (context, index) {
        if (index == photos.length) {
          return Padding(
            padding: const EdgeInsets.symmetric(vertical: 12.0),
            child: ActionChip(
              backgroundColor: Theme.of(context).primaryColor,
              label: const Text(
                '더 많은 사진 보기',
                style: TextStyle(color: Colors.white),
              ),
              elevation: 12.0,
              onPressed: () {
                _fetchPhotos();
              },
            ),
          );
        }
        return _buildCard(photos[index]);
      },
    );
  }
}
```

search_result_page.dart

검색 결과 조회 화면 작성하기

```
class _SearchResultPageState extends State<SearchResultPage> {  
  ...  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text(widget.searchTerm),  
      ),  
      body: _buildList(),  
    );  
  }  
}
```


검색어 입력 화면과 조회 화면 연결하기

검색어 입력 화면과 조회 화면 연결하기

```
class MyApp extends StatelessWidget {  
  const MyApp({Key? key}) : super(key: key);  
  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'My Photo',  
      theme: ThemeData(  
        primarySwatch: Colors.blue,  
      ),  
      routes: {  
        HomePage.routeName: (context) => HomePage(),  
        SearchResultPage.routeName: (context) {  
          final searchTerm =  
            ModalRoute.of(context)!.settings.arguments as String;  
          return SearchResultPage(searchTerm);  
        },  
      },  
    );  
  }  
}
```

main.dart

검색어 입력 화면과 조회 화면 연결하기

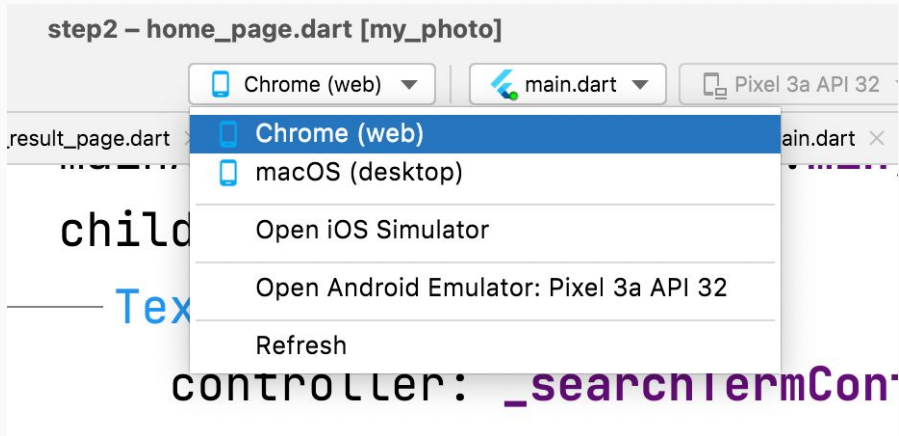
```
class HomePage extends StatelessWidget {  
  ...  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      body: Center(  
        child: ConstrainedBox(  
          ...  
          child: Padding(  
            padding: const EdgeInsets.symmetric(horizontal: 16.0),  
            child: Column(  
              mainAxisAlignment: MainAxisAlignment.min,  
              children: [  
                ...  
                ElevatedButton(  
                  onPressed: () {  
                    final searchTerm = _searchTermController.text.trim();  
                    Navigator.of(context).pushNamed(  
                      SearchResultPage.routeName,  
                      arguments: searchTerm,  
                    );  
                    analyticsService().search(searchTerm);  
                    _searchTermController.text = '';  
                  },  
                  child: const Text('검색'),  
                ),  
              ],  
            ),  
          ),  
        ),  
      );  
    }  
  }  
}
```

home_page.dart

완성된 앱 크롬에서 실행해보기

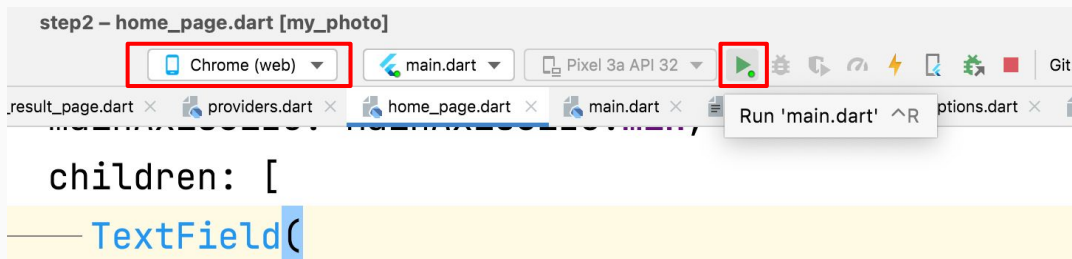
완성된 앱 크롬에서 실행해보기

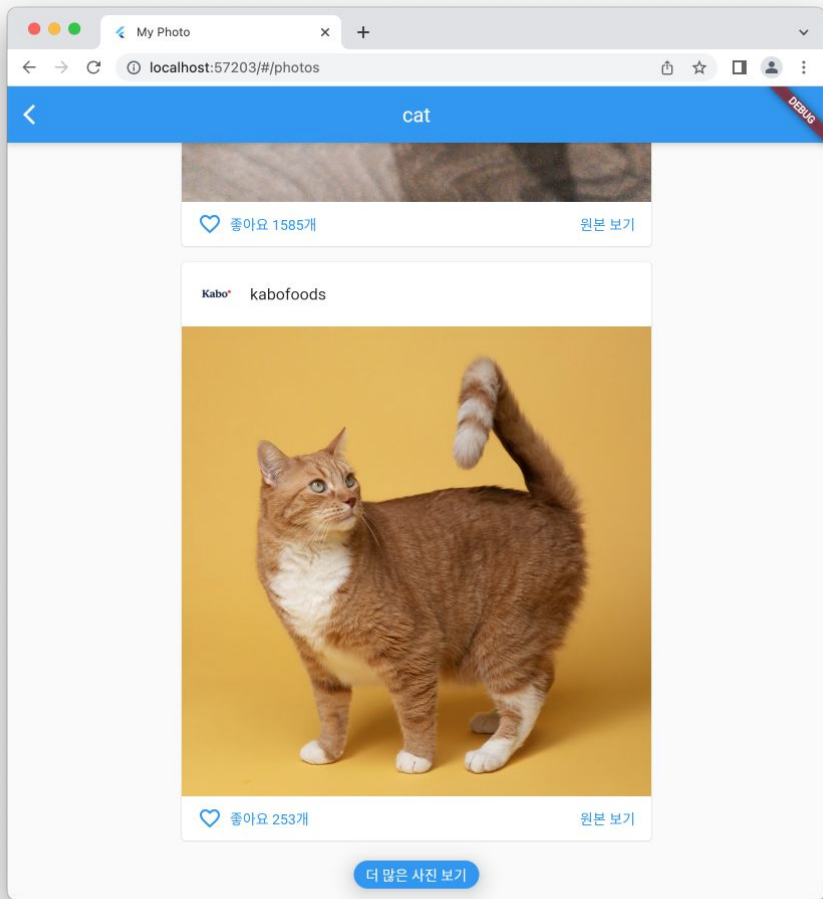
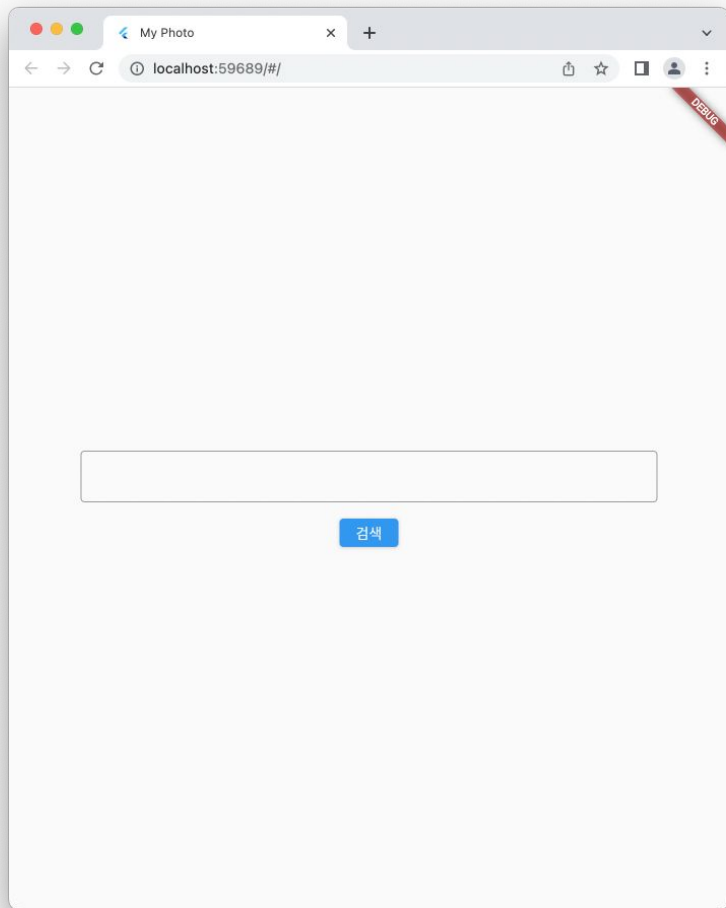
Flutter Device Selection 콤보박스를 클릭한 다음, Chrome (web)을 선택합니다.



완성된 앱 크롬에서 실행해보기

Run 'main.dart' 를 선택하여 앱을 크롬에서 실행합니다.





완성된 코드

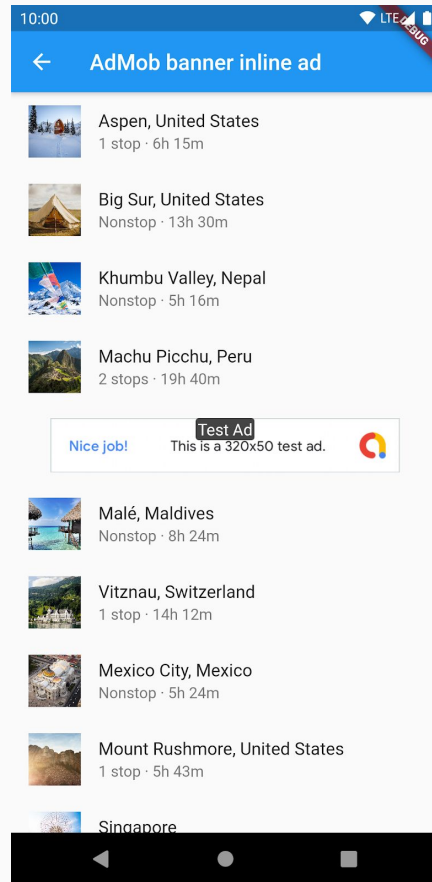
https://github.com/kunny/skku-bootcamp-2022-summer/tree/main/my_photo/step2

앱 광고 소개

앱 광고 소개

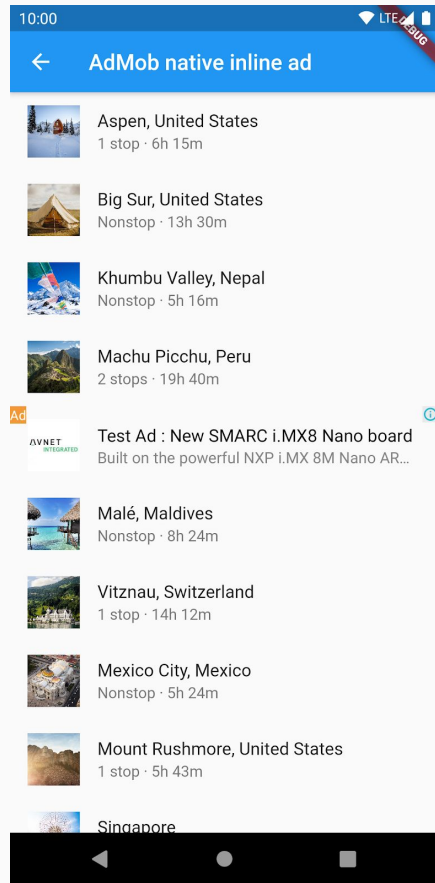
- 앱 개발자는 광고주가 집행하는 광고 캠페인을 게재할 수 있는 지면을 제공합니다.
- 앱의 유형, 사용자 수, 사용자 성향 등에 따라 지면의 가치가 달라지므로, 광고를 통해 얻을 수 있는 수익도 달라집니다.
- 구글 애드몹을 비롯하여 다양한 광고 네트워크가 있으며, 여러 광고 네트워크를 동시에 사용하여 수익을 최대화 할 수도 있습니다.
- 다양한 광고 포맷을 제공하므로, 앱의 디자인 및 사용자 성향에 적합한 광고를 게재할 수 있습니다.

- 배너
 - 가장 간단한 포맷으로, 텍스트/이미지/동영상 광고 소재를 지원합니다.
 - 다양한 형태 (띠, 정방형 등)을 제공합니다.



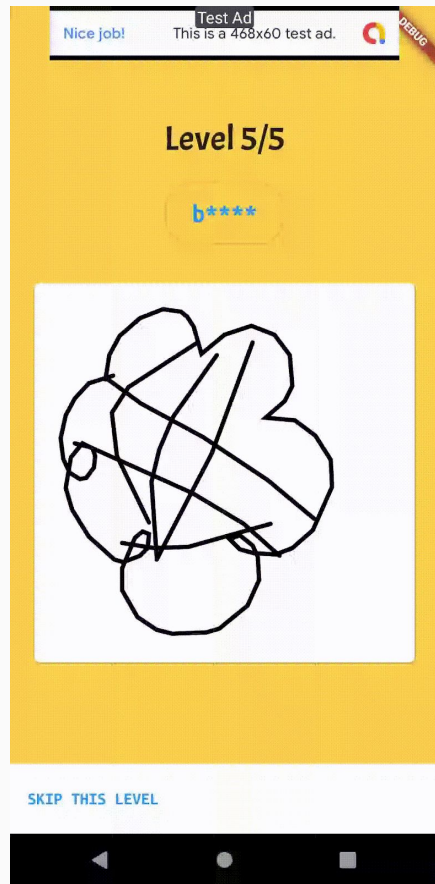
앱 광고 소개

- 네이티브
 - 광고 레이아웃을 앱의 UI와 유사하게 구성할 수 있으므로 광고를 자연스럽게 배치할 수 있습니다.
 - 네이티브(안드로이드, iOS)단에서 광고를 렌더링하므로, 네이티브 코드를 별도로 작성해야 합니다. (예: 안드로이드용으로 Java 코드, iOS용으로 Swift 코드)



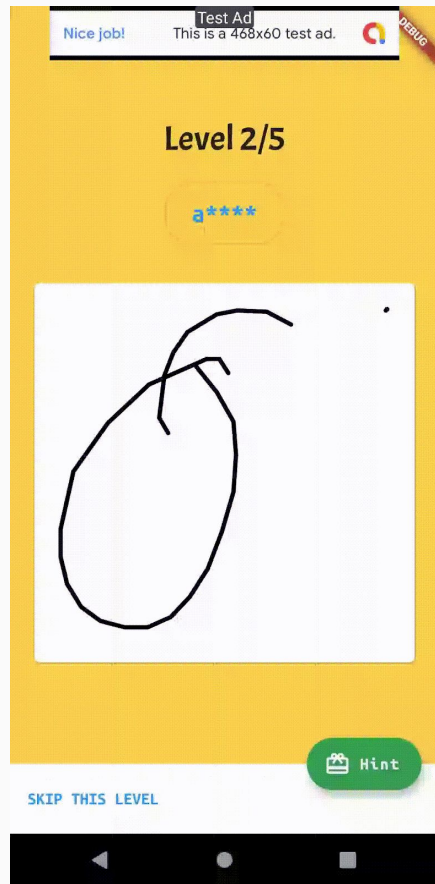
앱 광고 소개

- 전면
 - 앱 화면이 전환되는 시점에 전체 화면으로 광고를 보여줍니다.
 - 텍스트/이미지/동영상 광고 소재를 지원하며, 배너 혹은 네이티브 대비 광고 단가가 높습니다.



앱 광고 소개

- 리워드
 - 사용자가 광고를 보면 앱 내에서 사용할 수 있는 재화를 줄 수 있는 포맷입니다.
 - 광고를 볼지 여부를 사용자가 결정하므로 거부감이 적고, 단가도 상대적으로 높습니다.



앱 광고 구현 라이브 데모

**1 Introduction****2** Set up your Flutter development environment**3** Set up the AdMob app and ad units**4** Add the Google Mobile Ads Flutter plugin**5** Add a helper class for ads**6** Initialize the Google Mobile Ads SDK**7** Add a banner ad**8** Add an interstitial ad**9** Add a rewarded ad**10** All done!

Report a mistake

Adding AdMob ads to a Flutter app

About this codelab

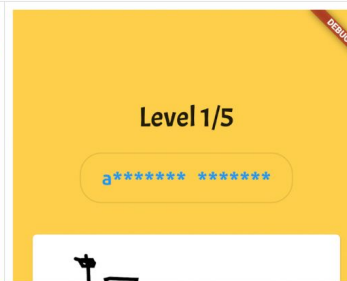
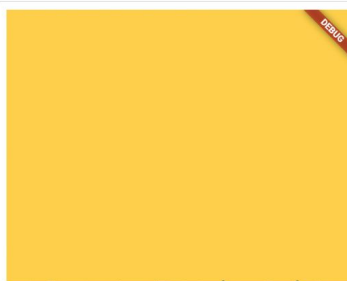
Last updated Jul 4, 2022

Written by Taeho Kim

1. Introduction

What you'll build

This codelab guides you through adding a banner, interstitial, and a rewarded ad to an app called **Awesome Drawing Quiz**, a game that lets players guess the name of the drawing.

**Next**

<https://codelabs.developers.google.com/codelabs/admob-ads-in-flutter>

감사합니다!

김태호 (jyte82@gmail.com)

fb.com/androidhuman

instagram.com/androidhuman

androidhuman.com