



## Flutter Training

Network requests

*By Anus Wahaj Ali*

# Recap

- Explore different types of widgets
- Explore widget catalog
- Week #2 Presentation (Flutter Basics):  
[https://docs.google.com/presentation/d/1bL1pE-VvCrVGSTxIWPlunH7x0\\_154tl7-TZiXhBZRow/edit?usp=sharing](https://docs.google.com/presentation/d/1bL1pE-VvCrVGSTxIWPlunH7x0_154tl7-TZiXhBZRow/edit?usp=sharing)
- Week #3 Presentation (Custom widgets and responsive screen design)  
<https://docs.google.com/presentation/d/1FBDmT4vtGJSf8ljyt3-DopdE40UnuZQ8E7Ky9p5VJls/edit?usp=sharing>



# Agenda

- Add dependencies
- Serialization and deserialization of JSON
- Manual serialization
- Posting Data On Internet
- Code generation with `json_serializable`
- Fetching data from internet
- Parse JSON in background



# Add Dependencies Using Pubspec.yaml

- Pubspec.yaml and it's role
- Types of dependencies.
- Add package in pubspec.yaml.
- How to add image assets in flutter.



# Http package

- Fetching/posting data from the internet is necessary for most apps.
- Luckily, Dart and Flutter provide tools, such as the `http` package, for this type of work.
- Add package in pubspec.yaml.
- Android apps must [declare their use of the internet](#) in the Android manifest (`AndroidManifest.xml`)



# Parse JSON as Dart object

- JSON Encode and Decode Functions http packages.
- Simply converts string response in to key map.
- Set those value to your dart object field by creating your own methods.



# Json Serialization Types

- Manual serialization
- Automated serialization using code generation



# Manual Serialization

- Manual JSON decoding refers to using the built-in JSON decoder in `dart:convert`
- Serializing JSON inline

```
Map<String, dynamic> user = jsonDecode(jsonString);  
print('Howdy, ${user['name']}!');
```

- Serializing JSON inside model classes  
A `User.fromJson()` constructor, for constructing a new `User` instance from a map structure.  
A `toJson()` method, which converts a `User` instance into a map.





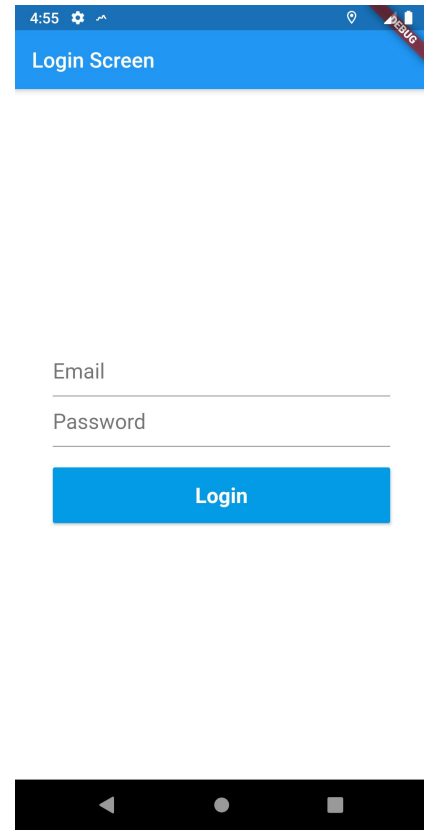
# Manual Serialization

```
class Movie {  
    String title;  
    String imageUrl;  
    String releaseDate;  
  
    Movie.fromJson(Map<String, dynamic> json)  
        : title = json['title'],  
          imageUrl = json['poster_path'],  
          releaseDate = json['release_date'];  
}
```



# Posting Data On Internet

- Post API Implementation
- Login Flow



A mobile application login screen mockup. At the top, a blue header bar contains the text "Login Screen" in white. Below the header, there are two input fields: "Email" and "Password", each with a light gray border. A blue "Login" button is positioned below the password field. The screen is framed by a dark blue border at the top and bottom, with a black navigation bar at the very bottom containing three white icons: a back arrow, a home circle, and a recent apps square. A red "Pineapp" logo is visible in the top right corner of the blue header bar.

4:55

Login Screen

Email

Password

Login



VentureDive

# Serializing JSON using code generation libraries

- Add Json Serializable package.
- Add Build runner
- Add annotation on your class
- Add factory methods
- Execute command on your terminal “flutter pub run build\_runner build”



# Serializing JSON using code generation Sample

```
part 'movie.g.dart';

@JsonSerializable()
class Movie {
  String title;
  @JsonKey(name: 'poster_path')
  String imageUrl;
  @JsonKey(name: 'release_date')
  String releaseDate;

  Movie(this.title, this.imageUrl, this.releaseDate);
  factory Movie.fromJson(Map<String, dynamic> json) => _$MovieFromJson(json);
```



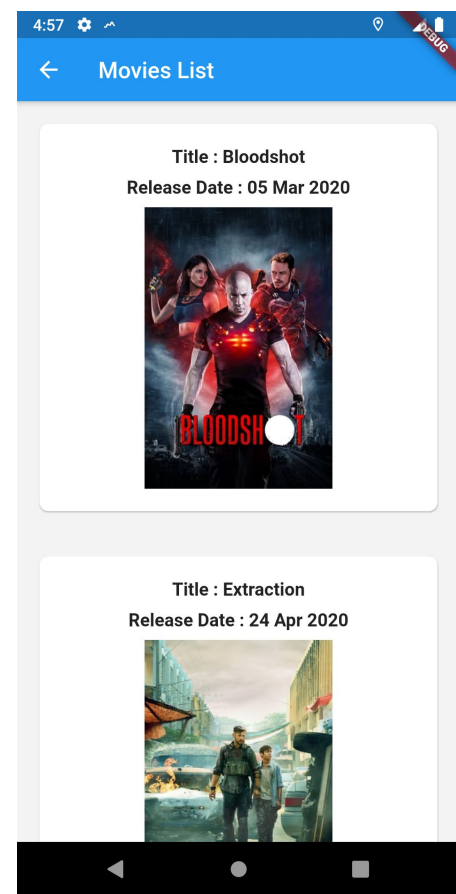
# Json Serializable Support

- `@JsonKey(defaultValue: false)`
- `@JsonKey(required: true)`
- `@JsonKey(ignore: true)`
- `@JsonKey(name: 'keyName')`



# Fetching Data From Internet

- GEt API Implementation
- Movie Listing Flow



VentureDive

# Json Parse In Background

- If this work takes more than 16 milliseconds, your users experience jank
- Isolate
- Thread based concept in dart
- `compute()` : function runs expensive functions in a background isolate and returns the result



# Take Home Assignment

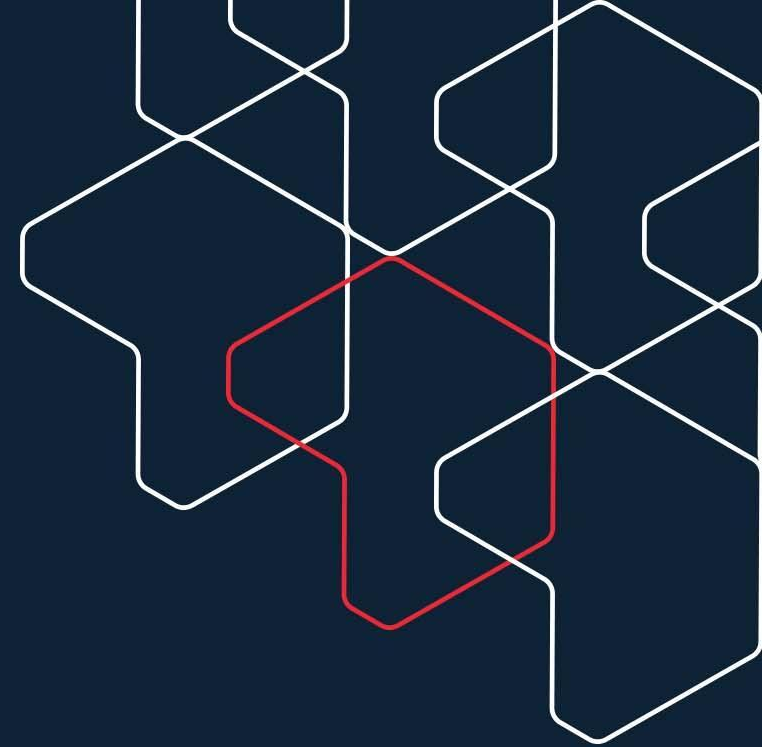
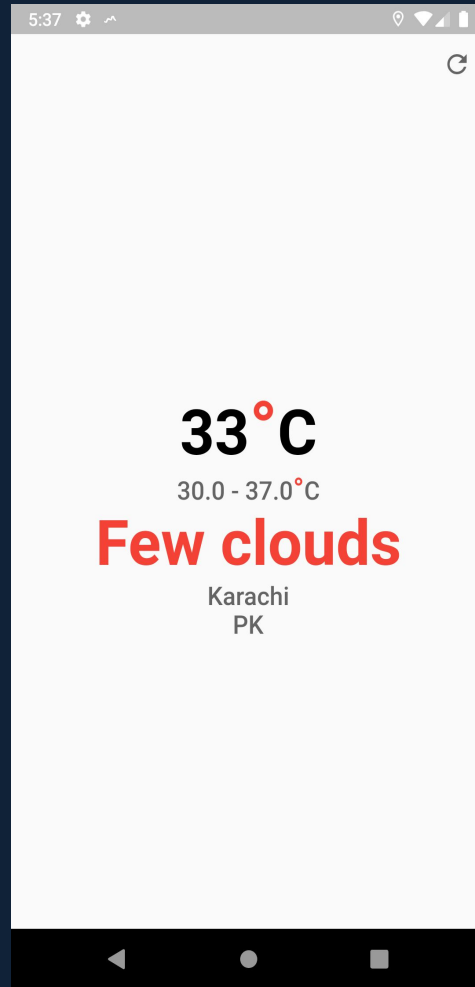
- Create a weather application which fetches current temperature of any specific city.
- Bonus Point : Add a city selection dropdown to view city wise temperature.
- You can use Open Weather for fetch weather updates.

<https://openweathermap.org/api>



**VentureDive**





Take Home Assignment

# Assignment submission

Upload your code on github and submit it's link on the Google chat group.



**VentureDive**



# Thank you

Upcoming: Persistent storage and unit tests

