Flutter and Dart

Part 3

Web API

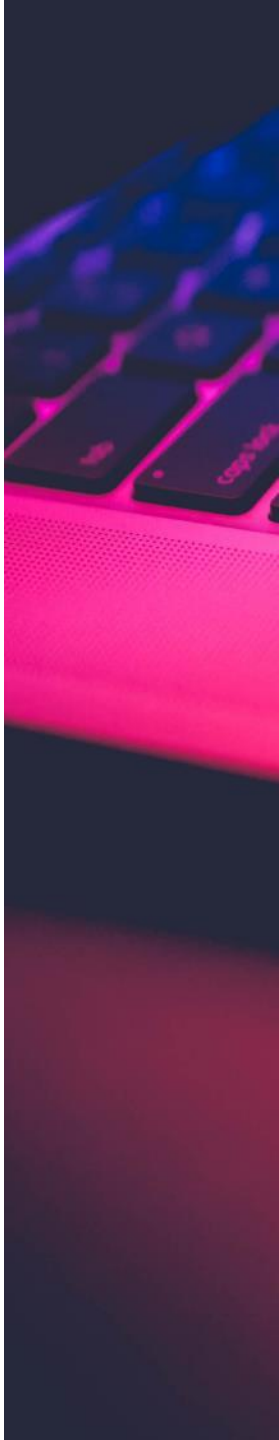
OUM SAOKOSAL

Master of Engineering in Information Systems,
Jeonju University, South Korea

012252752

Oum Saokosal

- MSc. In Information Systems, South Korea‘2010
- Dean of Faculty of Computer Science, NPIC since 2010
- 15 Years' Experience in Master Programs and Bachelor Programs
- Lecturer at NU, BBU, UP, NIPTICT, and Instinct Institute
- Former Lecturer at UC, RULE, SETEC, AEU
- 7 Years' Experience in Internet Marketing
- Tel/Telegram: 012 252 752



Fetch Data from Internet

- Add http in pubspect.yaml:, check https://pub.dev/packages/http

```
dependencies:
  http: ^0.12.0+1
```

- Import http:

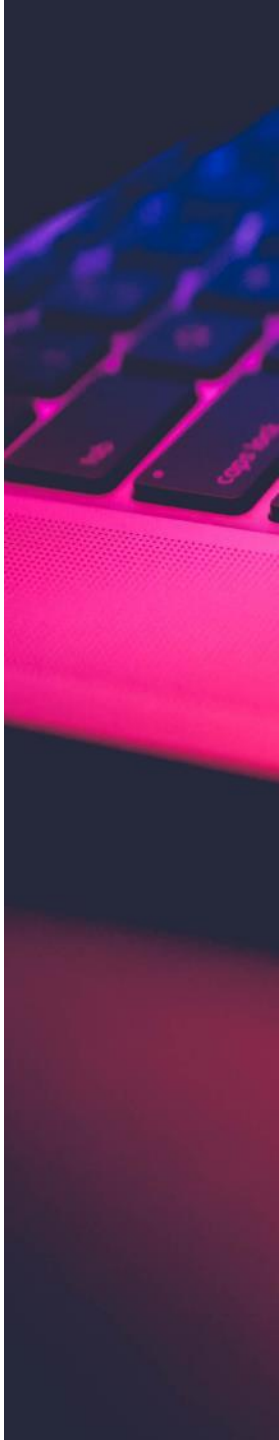
```
import 'package:http/http.dart' as http;
```

- Parse data:

```
List<Photo> parseData(String responseBody) {
  //convert JSON text to List<Map<String, dynamic>>
  List<Map<String, dynamic>> parsed =
                  json.decode(responseBody).cast<Map<String, dynamic>>();
  //convert from List<Map<String, dynamic>> to List<Photo>
  List<Photo> list = parsed.map((m) => Photo.fromJson(m)).toList();
  return list;
}
```

- Fetch data:

```
Future<List<Photo>> fetchData(http.Client client) async {
  final response =
      await client.get('https://jsonplaceholder.typicode.com/photos');

  return compute(parseData, response.body);
}
```

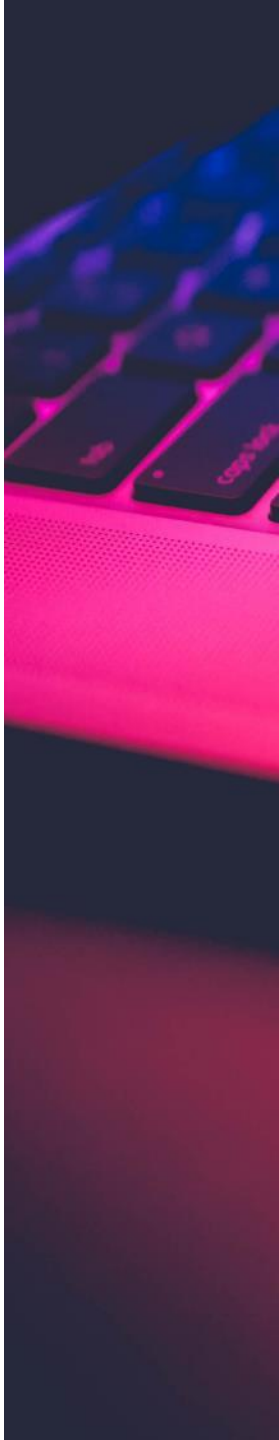


Create Photo class:

```
class Photo {
      final int id;
      final String title;
      final String thumbnailUrl;

      Photo({this.id, this.title, this.thumbnailUrl});

      factory Photo.fromJson(Map<String, dynamic> map) {
            return Photo(
                  id: map['id'] ?? 0,
                  title: map['title'] ?? "notitle",
                  thumbnailUrl: map['thumbnailUrl'] ?? "nourl",
            );
      }
}
```

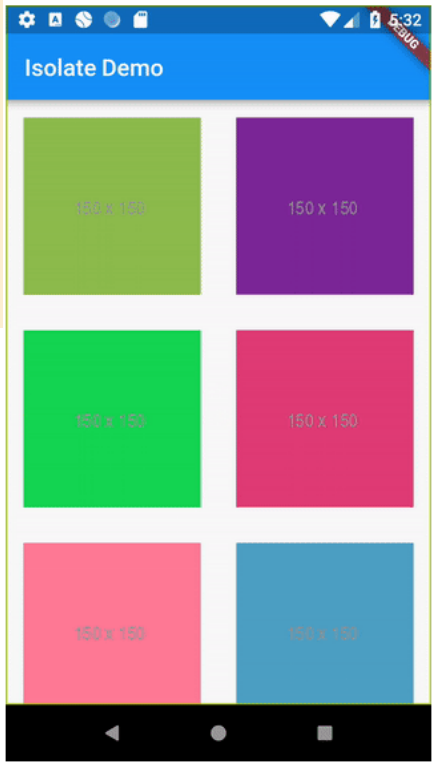
Read data with FutureBuilder:

```
FutureBuilder<List<Photo>>(
      future: fetchData(http.Client()),
      builder: (context, snapshot) {
            if (snapshot.hasError) print(snapshot.error);

            return snapshot.hasData
                  ? PhotosList(photos: snapshot.data)
                  : Center(child: CircularProgressIndicator());
      },
),
```



- Then use ListView.builder or GridView.builder:

```
class PhotosList extends StatelessWidget {
  final List<Photo> photos;

  PhotosList({Key key, this.photos}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return GridView.builder(
      gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
          crossAxisCount: 2,
        ),
      itemCount: photos.length,
      itemBuilder: (context, index) {
          return Image.network(photos[index].thumbnailUrl);
        },
    );
  }
}
```



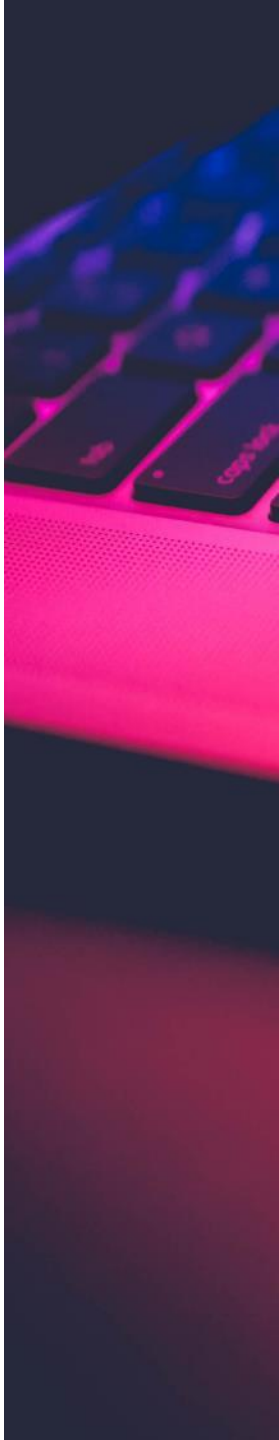
More details at:
https://flutter.dev/docs/cookbook/networking/background-parsing

Complex JSON Parsing

Structure #1: **Simple map**

```
{
    "id":"487349",
    "name":"Pooja Bhaumik",
    "score" : 1000
}
```

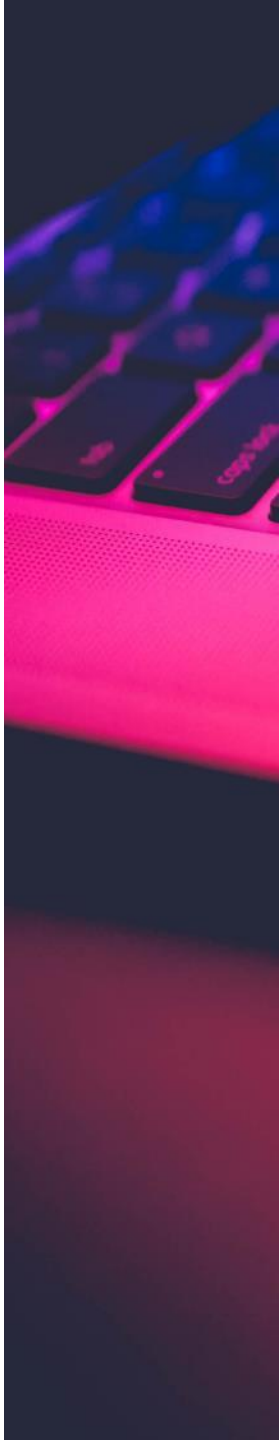
```
class Student{
  String studentId;
  String studentName;
  int studentScores;

  Student({
    this.studentId,
    this.studentName,
    this.studentScores,
  });

 factory Student.fromJson(Map<String, dynamic> parsedJson){
    return Student(
      studentId: parsedJson['id'],
      studentName : parsedJson['name'],
      studentScores : parsedJson ['score'],
    );
  }
}
```



Structure #2: **Simple structure with arrays**

```
{
    "city": "Mumbai",
    "streets": [
      "address1",
      "address2",
    ]
}
```

```
class Address {
  final String city;
  final List<String> streets;

  Address({
    this.city,
    this.streets
  });

  factory Address.fromJson(Map<String, dynamic> parsedJson) {
     var streetsFromJson = parsedJson['streets'];
     List<String> streetsList = new List<String>.from(streetsFromJson);
     return new Address(
        city: parsedJson['city'],
        streets: streetsList ,
     );
  }
}
```

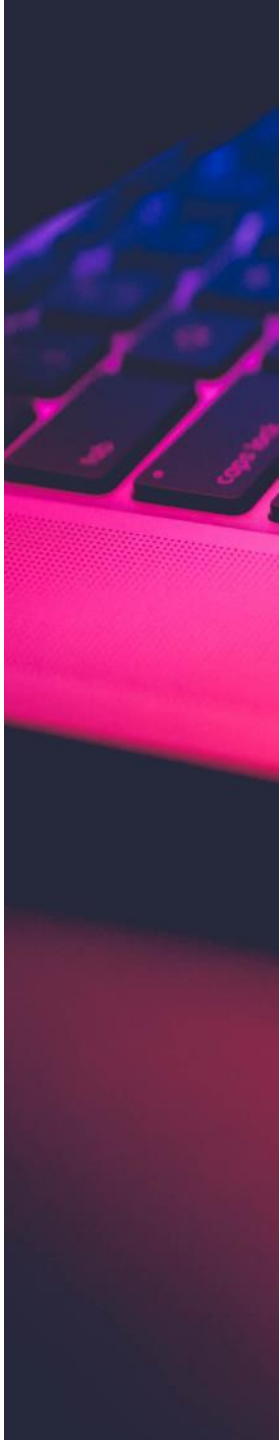


Structure #3: **Simple Nested structures**

```
{
  "shape_name":"rectangle",
  "property":{
    "width":5.0,
    "breadth":10.0
   }
}
```

```
class Property{
  double width;
  double breadth;
  Property({this.width, this.breadth});
  factory Property.fromJson(Map<String, dynamic> map){
      return Property(width: map['width'], breadth: map[' breadth']);
  }
}
```

```
class Shape{
  String shapeName;
  Property property;

  Shape({this.shapeName, this.property});

  factory Property.fromJson(Map<String, dynamic> json){
    return Property(
      width: json['width'],
      breadth: Property.fromJson(parsedJson['property'])
    );
   }
}
```

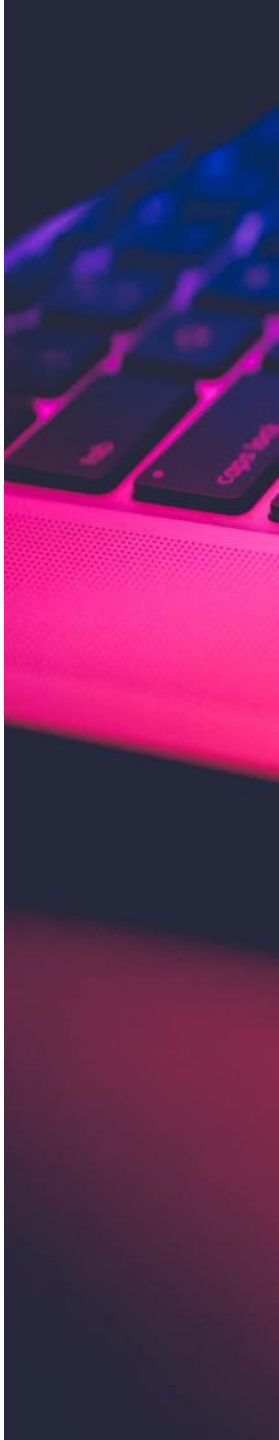


Structure #4: **Nested structures with Lists**

```
{
  "id":1,
  "name":"ProductName",
  "images":[
    {"id":11, "imageName":"xCh-rhy"},
    {"id":31,"imageName":"fjs-eun"}
  ]
}
```

```
class Product {
  final int id;
  final String name;
  final List<Image> images;

  Product({this.id, this.name, this.images});
  factory Product.fromJson(Map<String, dynamic> parsedJson){
    var list = parsedJson['images'] as List;
    List<Image> imagesList = list.map((i) => Image.fromJson(i)).toList();
    return Product(
      id: parsedJson['id'],
      name: parsedJson['name'],
      images: imagesList
    );
  }
}
```

```
class Image {
  final int imageId;
  final String imageName;
  Image({this.imageId, this.imageName});
  factory Image.fromJson(Map<String, dynamic> parsedJson){
    return Image(
      imageId: parsedJson['id'],
      imageName: parsedJson['imageName']
    );
  }
}
```



Structure #5: **List of maps**

```
[
  {
    "albumId": 1,"id": 1,
    "title": "accusamus beatae ad facilis cum similique qui sunt",
    "url": "http://placehold.it/600/92c952",
    "thumbnailUrl": "http://placehold.it/150/92c952"
  },
  {
    "albumId": 1,"id": 2,
    "title": "reprehenderit est deserunt velit ipsam",
    "url": "http://placehold.it/600/771796",
    "thumbnailUrl": "http://placehold.it/150/771796"
  },
]
```

```
class PhotosList {
  final List<Photo> photos;

  PhotosList({
    this.photos,
  });
  factory PhotosList.fromJson(List<dynamic> parsedJson) {

    List<Photo> photos = new List<Photo>();
    photos = parsedJson.map((i)=>Photo.fromJson(i)).toList();

    return new PhotosList(
       photos: photos,
    );
  }
}
```

```
class Photo{
  final String id, title, url;

  Photo({this.id, this.url, this.title}) ;

  factory Photo.fromJson(Map<String, dynamic> json){
    return new Photo(
      id: json['id'].toString(),
      title: json['title'],
      url: json['json'],
    );
  }
}
```

Auto Parsing from JSON to Dart

- Quicktype:

  https://app.quicktype.io/

- Parse JSON to DART online:

  https://javiercbk.github.io/json_to_dart/