

Serializing JSON Data

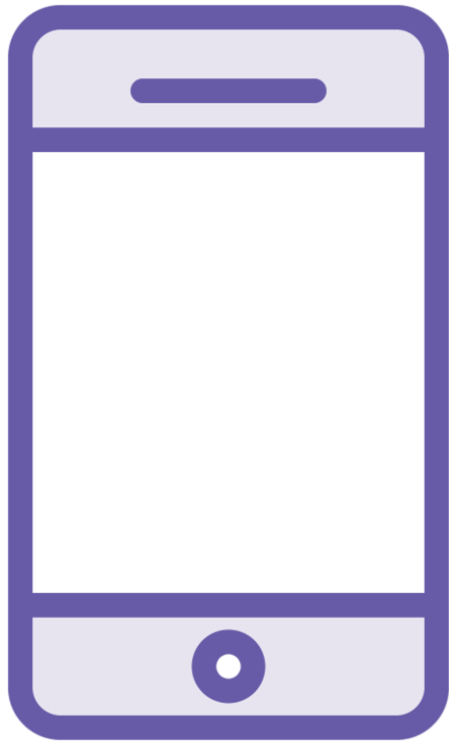


Douglas Starnes

SPEAKER / AUTHOR

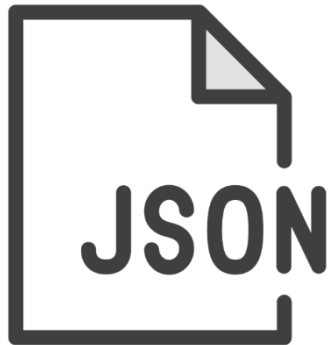
@poweredbyalnet <https://douglasstarnes.com>





Dart (Flutter)

Language Agnostic



Python (Flask)



Manual Serialization



Most straightforward means of serialization



Makes a lot of assumptions



JSON generated by `jsonEncode` function



You manually create the association between Dart classes and JSON



Map

Similar to a Python dictionary or JavaScript object

Collection of key value pairs

Generic type

Two type parameters, type of key and type of value

For JSON, keys will always be strings

For JSON, values will be dynamic to accommodate JSON types



Automatic Serialization

Annotations

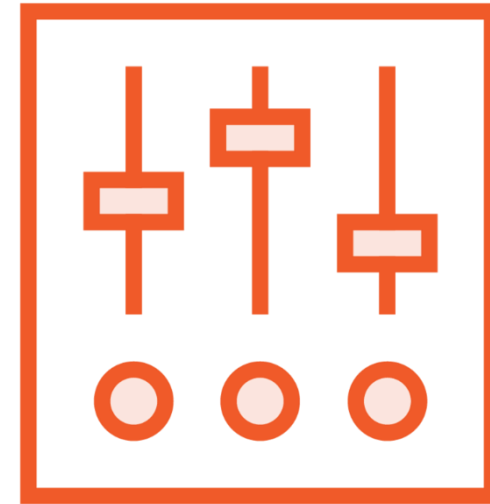
Code Generation



Which to Use?



Manual serialization is for simpler types



Automatic serialization gives you more control over the generated JSON



```
class UserProfile {  
    String firstName;  
    String lastName;  
    bool isActive;  
    int loyaltyPoints;  
    int fitnessGoal;  
}
```

Fitness Goal:

- Lose weight
- Build muscle
- Train for a sport
- Recreation

main.dart

```
class UserProfile {  
    // ...  
  
    Map<String, dynamic> userMap = {  
        'firstName' : this.firstName,  
        'lastName' : this.lastName,  
        'isActive' : this.isActive,  
        'loyaltyPoints' : this.loyaltyPoints,  
        'fitnessGoal' : this.fitnessGoal  
    };  
}
```


main.dart

```
class UserProfile {  
    // ...  
  
    Map<String, dynamic> userMap = Map<String, dynamic>();  
    userMap['firstName'] = this.firstName;  
}
```

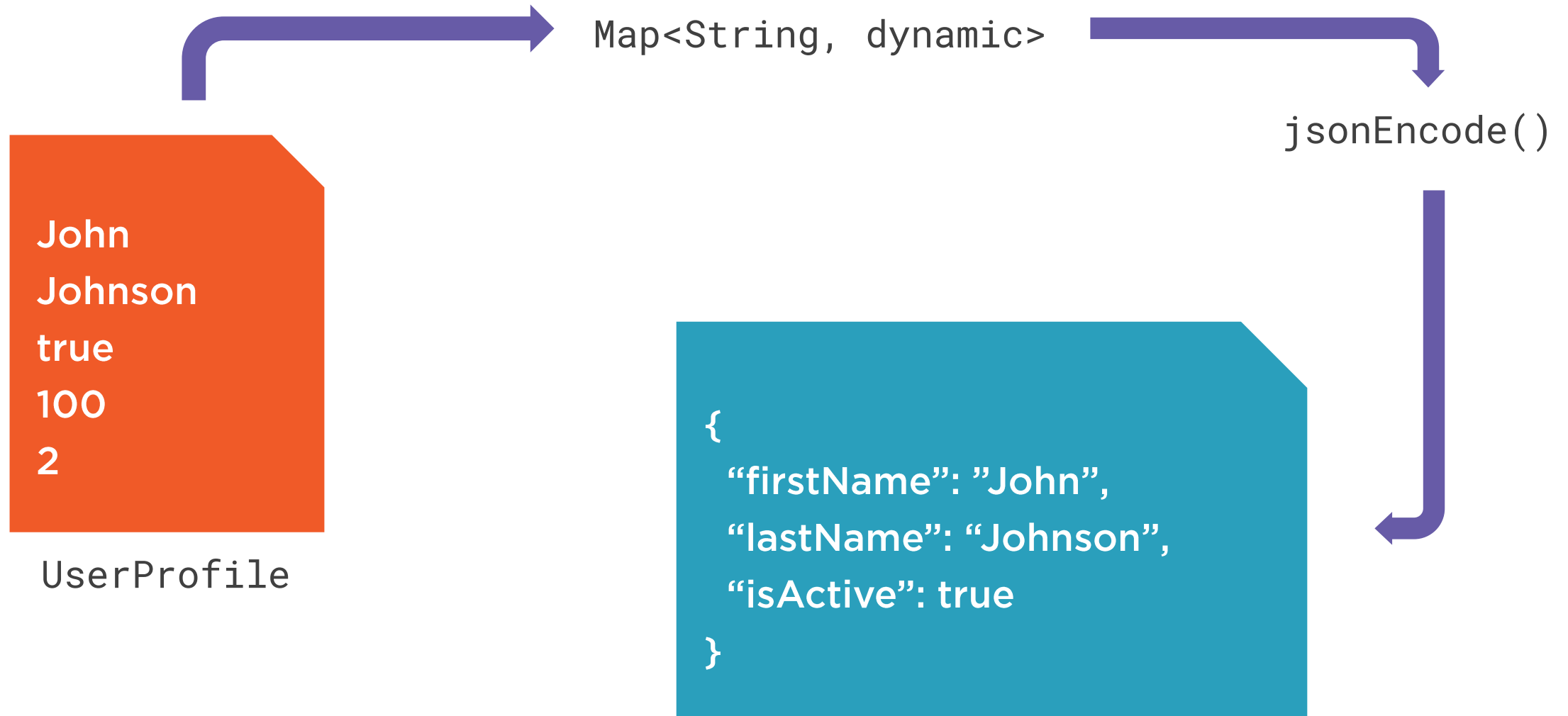
main.dart

```
class UserProfile {  
    // ...  
    var userMap = {  
        'firstName': this.firstName  
    };  
}
```

```
import 'dart:convert';  
class UserProfile {  
  // ...  
  Map<String, dynamic> userMap (...),  
  var data = jsonEncode(userMap);  
}
```



```
"""{  
  "firstName": "John",  
  "lastName": "Johnson"  
}"""
```



main.dart

```
import 'dart:convert';

class UserProfile {

    // ...

    String toJson() {

        Map<String, dynamic> userMap = { 'firstName': this.firstName };

        var data = jsonEncode(userMap);

        return data;

    }

}
```

Demo



Implementing manual serialization in Flutter



Automatic Serialization



Significant setup compared to manual serialization



Annotations are a declarative way to attach metadata to Dart code



The `JsonSerializable` annotation marks a Dart class as serializable



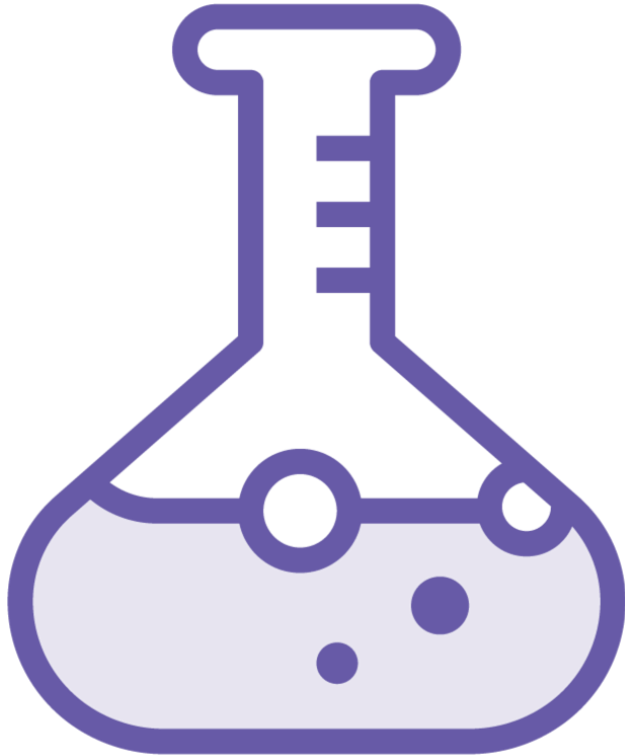
Add `json_annotation` to the `dependencies` section in `pubspec.yaml`



Import the `json_annotation` package into the file with your class



Code Generation



Automatic serialization does not generate JSON

The code to handle serialization is generated by a command line utility

The `json_serializable` package contains the code generator

The `build_runner` package invokes the code generator

pubspec.yaml

dependencies:

 json_annotation: ^3.1.0

dev_dependencies:

 json_serializable: ^3.5.0

 build_runner: ^1.10.1

userprofile.dart

```
import 'package:json_annotation/json_annotation.dart';
```

```
part 'userprofile.g.dart';
```

```
@JsonSerializable()
```

```
class UserProfile {
```

```
    // ...
```

```
}
```

```
pub run build_runner build
```



```
// GENERATED CODE - DO NOT MODIFY BY HAND
```

```
UserProfile _$UserProfileFromJson(Map<String, dynamic> json) { ... }
```

```
Map<String, dynamic> _$UserProfileToJson(UserProfile instance) { ... }
```

userprofile.dart

```
import 'package:json_annotation/json_annotation.dart';
```

```
import 'dart:convert';
```

```
part 'userprofile.g.dart';
```

```
@JsonSerializable()
```

```
class UserProfile {
```

```
    // ...
```

```
    String toJson() {
```

```
        var userMap = _$UserProfileToJson(this);
```

```
        return jsonEncode(userMap);
```

```
    }
```

```
}
```

JSON Annotations

`JsonSerializable`

`JsonKey`



userprofile.dart

```
import 'package:json_annotation/json_annotation.dart';  
part 'userprofile.g.dart';  
  
@JsonSerializable(nullable: true)  
class UserProfile {  
    // ...  
}
```

Camel Case vs. Snake Case

Camel case

First letter of every
word is upper case
(except the first)

day0fTheWeek

Snake case

All lower case
with underscores
between words

day_of_the_week



userprofile.dart

```
import 'package:json_annotation/json_annotation.dart';  
part 'userprofile.g.dart';  
  
@JsonSerializable(fieldRename: FieldRename.snake)  
class UserProfile {  
    // ...  
}
```

userprofile.dart

```
import 'package:json_annotation/json_annotation.dart';
```

```
part 'userprofile.g.dart';
```

```
@JsonSerializable(fieldRename: FieldRename.snake)
```

```
class UserProfile {
```

```
    @JsonKey(name: 'FIRST_NAME')
```

```
    String firstName;
```

```
    @JsonKey(includeIfNull: false)
```

```
    String lastName;
```

```
}
```

Demo



Implementing automatic serialization in Flutter



Summary



Serialization

Manual

- Hand code a Map defining key/value pairs
- `jsonEncode()` generates the JSON

Automatic

- Generates code to serialize JSON, not the JSON data
- Code generated by a command line utility
- Annotations
 - `JsonSerializable`
 - `JsonKey`



Summary



Code to use manual or automatic serialization is almost identical

Manual serialization is better for simpler classes, is more straightforward, but trusts you implicitly

Automatic serialization has the extra overhead but gives you precise control over serialization