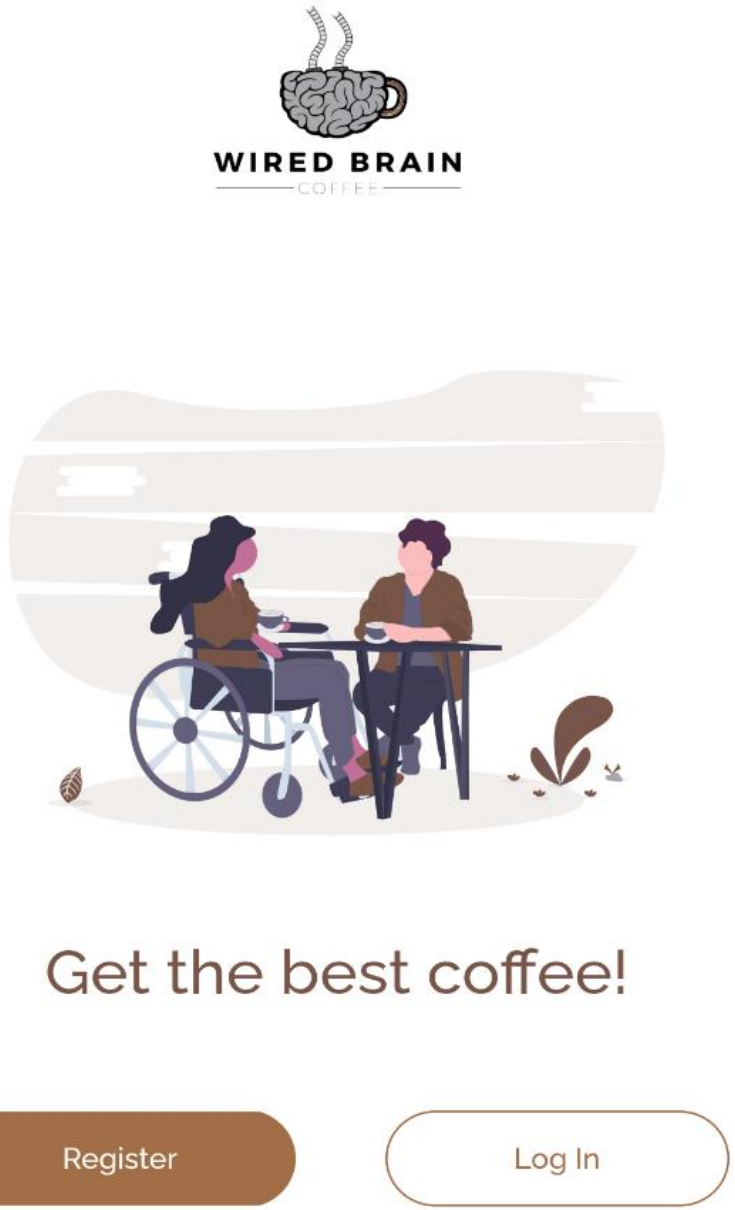
# Testing a Flutter App: Basic and Unit Tests


**Majid Hajian**

TRAINER, AUTHOR, PROUD SOFTWARE ENGINEER

@mhadaily www.majidhajian.com




# The Wired Brain Coffee Company

## Ensure the quality of the app by writing a test

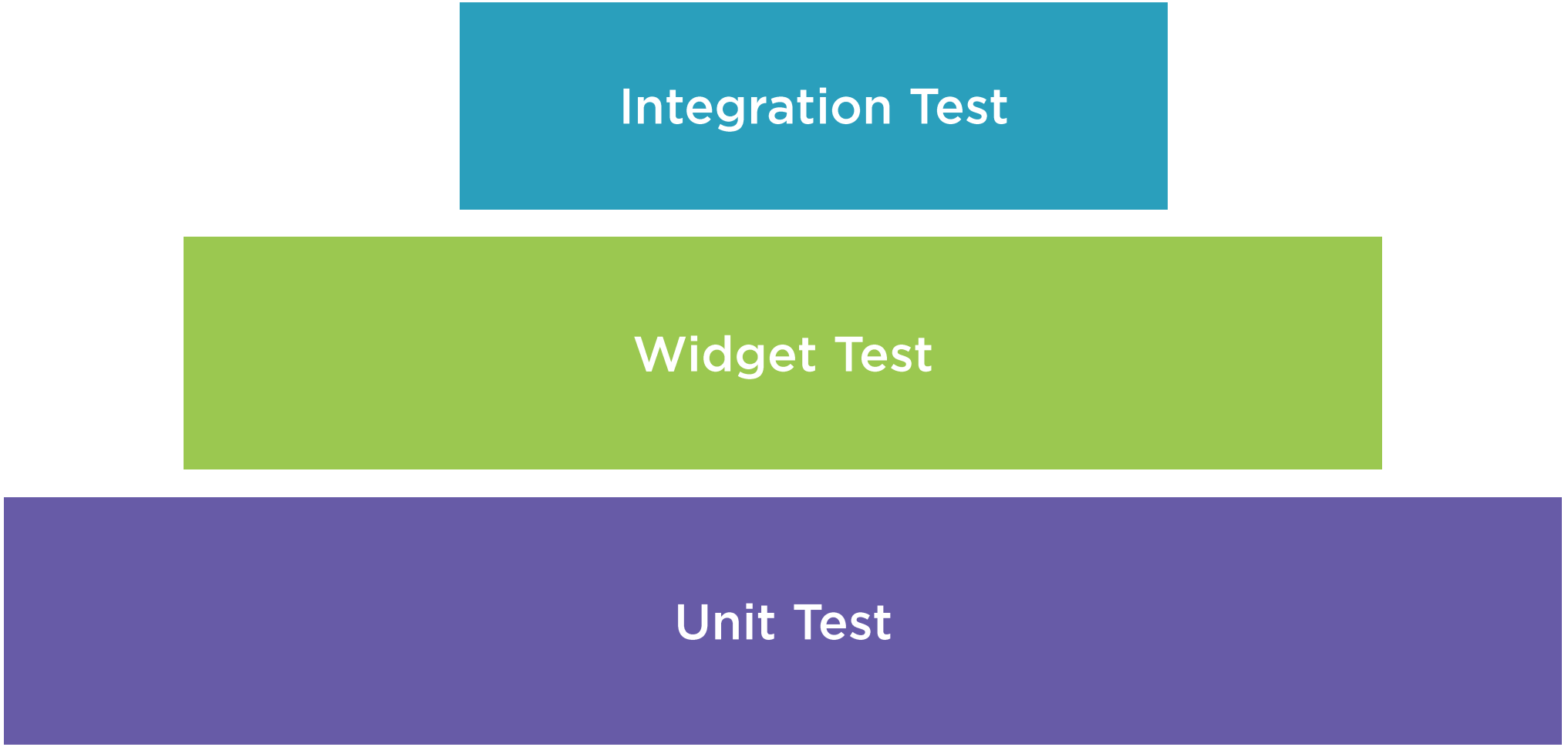
# Overview

Basics of testing in a Flutter app

How to write unit tests in Flutter

How to mock dependences for using in tests

# Testing Flutter Apps



Unit Test

Test a function, method, or class

Correctness of a unit

Few dependencies and generally mocked out

Does not receive use actions

Quick execution

Low maintenance cost

# Unit Test

# Simple Dart Unit Test

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


void main() {
  test(
    'my first unit test',
    () {
      var answer = 42;
      expect(answer, 42);
    });
}
```

# Test

**Arrange**

Setup what test needs

**Act**

Call specific operation

**Assert**

Check our values

# Simple Dart Unit Test

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

void main() {

    test('my first unit test', () {
        // Arrange
        final user = User(24, 'Majid');
        // Act
        user.increaseAgeByOne();
        // Assert
        expect(user.age, 25);
    });

}
```

# Demo

Validate email and password

Write unit test for validators

group(), test(), expect(),

# Mocking

# Mocking

## Data

Such as a data model

## Dependencies

Such as HTTP client

# Demo

Use Mockito package

Arrange a mocked class to use in the test

Best practices

# Setup and Tear Down functions

# Demo

Setup()

setupAll()

tearDown()

teaDownAll()

# Summary

# Summary

**Testing in Flutter**

- Unit test
- Widget test
- Integration test

**Unit test**

- Arrange, act, and assert
- Mock dependencies