

Flutter Development with State Management and Persistence

Dart foundations, widgets, navigation, Firebase and BLoC projects

FA-0764 · 5 days

Software Development

Introduction

Move from introductory Dart concepts to building small Flutter applications with widgets, user input and navigation. Practical examples introduce state, asynchronous data and service separation.

The five-day programme is suitable for developers new to Flutter. It starts with foundations and progresses to selected Firebase and BLoC/Sembastr exercises rather than promising advanced mastery or a career outcome.

Learning outcomes

- Explain the roles of Dart and Flutter and create a basic application.
- Compose widgets, handle input and apply themes and styling.
- Manage screen navigation and application state.
- Listen to a data stream and separate widget, asynchronous, BLoC and service responsibilities in an example.
- Add a prepared Firebase authentication and favourites-data example.
- Use Sembastr local storage and BLoC state management in a small to-do application.
- Use debugging tools to investigate the sample application.

Prerequisites

Basic programming and object-oriented familiarity in another language is helpful. Dart and Flutter foundations are introduced.

A supported Flutter SDK with compatible platform SDKs, Android Studio or Visual Studio Code and a prepared device/emulator. For iOS builds, use an appropriate macOS/Xcode environment. Select Java/Gradle and platform versions from current compatibility requirements rather than assuming Java 8 is sufficient.

Access to an authorised Firebase training project or prepared emulator and adequate workstation resources. Only the installation permissions needed for the lab are required.

Training outline

1. Day 1 — Dart and Flutter foundations

- Introduction to Dart
- Getting started with Dart
- Understanding why Flutter uses Dart
- Introducing the structure of the Dart language
- Introduction to OOP in Dart
- Flutter overview
- Flutter compilation
- Flutter rendering
- Widget Introduction
- Widget-Async-Bloc-Service architecture
- Basic Flutter App

2. Day 2 — Widgets, interaction and navigation

- Widgets
- Handling User Input and Gestures
- Theming and Styling
- Routing: Navigating between Screens

Use Navigator for the small exercise and discuss routing requirements; complex deep-linking scenarios may require Router or a routing package.

3. Day 3 — State and asynchronous data

- Using Stateful Widgets(Miles to Kilometer converter)
- Listening to a Stream of Data

Review widget state, asynchronous updates, service separation and debugging through the converter project.

4. Day 4 — Firebase integration

- Introducing Firebase
- Project overview
- Adding Firebase to your Flutter project
- Adding authentication to your app
- Writing data to Firebase: Adding the favorite feature

Configure compatible FlutterFire plugins and prepared authentication/data services; discuss permissions and scope of the training example.

5. Day 5 — BLoC and local persistence

- Project overview
- Using sembast to store data
- The BLoC pattern

Build and review a small to-do application. Distinguish local Sembast persistence from Firebase services and discuss maintainability of the chosen state-management pattern.