

Types of Providers in provider Package

Here's a **comprehensive example** covering **all provider types** in Flutter (`provider` package). This includes:

1. **Provider** → Injecting static data
2. **ChangeNotifierProvider** → Managing counter state
3. **FutureProvider** → Fetching async API data
4. **StreamProvider** → Listening to real-time data
5. **ProxyProvider** → Combining authentication & user profile

✓ Complete Example with All Providers

```
dart
CopyEdit
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'dart:async';

// 🚀 Simulated API Call (FutureProvider)
Future<String> fetchTeamName() async {
  await Future.delayed(Duration(seconds: 2));
  return "Flutter Developers Team";
}

// 🚀 Simulated Real-Time Data (StreamProvider)
Stream<int> numberStream() async* {
  int i = 0;
  while (true) {
    await Future.delayed(Duration(seconds: 1));
    yield i++;
  }
}

// 🚀 ChangeNotifierProvider (State Management)
class CounterProvider extends ChangeNotifier {
  int _count = 0;
  int get count => _count;

  void increment() {
    _count++;
    notifyListeners();
  }
}
```

```

}

// 📌 AuthProvider (For ProxyProvider)
class AuthProvider with ChangeNotifier {
  String _userId = "12345";
  String get userId => _userId;
}

// 📌 User Profile Provider (Dependent on AuthProvider)
class UserProfile {
  final String userId;
  final String name;
  UserProfile(this.userId, this.name);
}

class UserProfileProvider {
  final AuthProvider auth;
  UserProfileProvider(this.auth);

  UserProfile get profile => UserProfile(auth.userId, "John Doe");
}

// 📌 Main App with MultiProvider
void main() {
  runApp(
    MultiProvider(
      providers: [
        Provider<String>(create: (_) => "Static Provider Data"),
        ChangeNotifierProvider(create: (_) => CounterProvider()),
        FutureProvider<String>(create: (_) => fetchTeamName(), initialState: "Fetching..."),
        StreamProvider<int>(create: (_) => numberStream(), initialState: 0),
        ChangeNotifierProvider(create: (_) => AuthProvider()),
        ProxyProvider<AuthProvider, UserProfileProvider>(
          update: (_, auth, __) => UserProfileProvider(auth),
        ),
      ],
      child: MyApp(),
    ),
  );
}

// 📌 MyApp
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: HomeScreen(),
    );
  }
}

```

```

}

// 📌 Home Screen (Navigates to Other Screens)
class HomeScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: Text("Provider Examples")),
      body: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Center(
            child: ElevatedButton(
              onPressed: () => Navigator.push(context, MaterialPageRoute(builder: (_) =>
CounterScreen()))),
            child: Text("ChangeNotifierProvider (Counter)"),
          ),
          Center(
            child: ElevatedButton(
              onPressed: () => Navigator.push(context, MaterialPageRoute(builder: (_) =>
FutureProviderScreen()))),
            child: Text("FutureProvider (API Call)"),
          ),
          Center(
            child: ElevatedButton(
              onPressed: () => Navigator.push(context, MaterialPageRoute(builder: (_) =>
StreamProviderScreen()))),
            child: Text("StreamProvider (Live Data)"),
          ),
          Center(
            child: ElevatedButton(
              onPressed: () => Navigator.push(context, MaterialPageRoute(builder: (_) =>
ProxyProviderScreen()))),
            child: Text("ProxyProvider (Auth & Profile)"),
          ),
        ],
      ),
    );
  }
}

```

```

// 📌 Counter Screen (ChangeNotifierProvider)
class CounterScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final counter = context.watch<CounterProvider>();

```

```

return Scaffold(
  appBar: AppBar(title: Text("Counter Example")),
  body: Center(
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Text("Counter: ${counter.count}", style: TextStyle(fontSize: 24)),
        SizedBox(height: 10),
        ElevatedButton(
          onPressed: counter.increment,
          child: Text("Increment"),
        ),
      ],
    ),
  ),
);
}

```

```

// 📌 FutureProvider Screen (API Call)
class FutureProviderScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final teamName = context.watch<String>();

    return Scaffold(
      appBar: AppBar(title: Text("FutureProvider Example")),
      body: Center(
        child: Text(teamName, style: TextStyle(fontSize: 24)),
      ),
    );
  }
}

```

```

// 📌 StreamProvider Screen (Live Data)
class StreamProviderScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final number = context.watch<int>();

    return Scaffold(
      appBar: AppBar(title: Text("StreamProvider Example")),
      body: Center(
        child: Text("Number: $number", style: TextStyle(fontSize: 24)),
      ),
    );
  }
}

```

```
// 🚀 ProxyProvider Screen (Auth & User Profile)
class ProxyProviderScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final userProfile = context.watch<UserProfileProvider>().profile;

    return Scaffold(
      appBar: AppBar(title: Text("ProxyProvider Example")),
      body: Center(
        child: Text("User: ${userProfile.name} (ID: ${userProfile.userId})", style: TextStyle(fontSize:
24)),
      ),
    );
  }
}
```

🚀 Features in This Example

Provider Type	Feature	Screen
<code>Provider<String></code>	Provides static text	HomeScreen
<code>ChangeNotifierProvider</code>	Increments a counter (<code>notifyListeners()</code>)	CounterScreen
<code>FutureProvider</code>	Fetches a team name from an API	FutureProviderScreen
<code>StreamProvider</code>	Displays a live updating number	StreamProviderScreen
<code>ProxyProvider</code>	Fetches a user profile based on authentication	ProxyProviderScreen

🔥 How It Works

- Home Screen**
 - Lists buttons for different examples.
 - Click to navigate to each provider demo.
- Counter Screen (`ChangeNotifierProvider`)**
 - Shows a **counter** with an "Increment" button.

3. FutureProvider Screen

- Fetches **team name from API** (Shows "Fetching..." initially).

4. StreamProvider Screen




- Displays a **real-time number counter** updating every second.

5. ProxyProvider Screen

- Shows a **user profile** based on authentication (`AuthProvider`).
-

Summary

This example covers **all providers** in Flutter's `provider` package with **real use cases**.

-  **MultiProvider** to handle multiple providers.
-  **Navigation between screens** to keep everything organized.
-  **Performance optimization** using `watch` , `read` , and `Consumer` .

Would you like any modifications or additions, such as **error handling or retry logic**? 😊