✅ Full Integration of NetworkCaller with DI in Flutter

🔹 network_caller.dart (Refactored to instance-based)

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

class NetworkCaller {
 final String token;

 NetworkCaller({this.token = ''});

 Future<NetworkResponse> getRequest(String url) async {
  try {
   Uri uri = Uri.parse(url);
   final response = await http.get(uri);
   if (response.statusCode == 200) {
    final decodeData = jsonDecode(response.body);
    return NetworkResponse(
      statusCode: response.statusCode, isSuccess: true, body: decodeData);
   } else {
    return NetworkResponse(
     statusCode: response.statusCode,
     isSuccess: false,
     errorMessage: response.reasonPhrase,
    );
   }
  } catch (e) {
   return NetworkResponse(
```

```
      statusCode: -1,
      isSuccess: false,
      errorMessage: e.toString(),
    );
  }
}

Future<NetworkResponse> postRequest(String url, Map<String, dynamic> body) async {
  try {
    Uri uri = Uri.parse(url);
    final response = await http.post(
      uri,
      body: jsonEncode(body),
      headers: {
        'Content-Type': 'application/json',
        'token': token,
      },
    );
    if (response.statusCode == 200) {
      final decodeData = jsonDecode(response.body);
      return NetworkResponse(
          statusCode: response.statusCode, isSuccess: true, body: decodeData);
    } else {
      return NetworkResponse(
        statusCode: response.statusCode,
        isSuccess: false,
        errorMessage: response.reasonPhrase,
      );
    }
```

```
  } catch (e) {
    return NetworkResponse(
      statusCode: -1,
      isSuccess: false,
      errorMessage: e.toString(),
    );
  }
 }
}
```

🔹 **add_product_view_model.dart (Injected NetworkCaller)**

```
import 'package:flutter/material.dart';
import 'package:rest_api_crud_app/data/network_caller.dart';
import 'package:rest_api_crud_app/data/network_response.dart';
import 'package:rest_api_crud_app/model/rest_api_crud/product_model.dart';
import 'package:rest_api_crud_app/utils/urls.dart';

class AddProductViewModel with ChangeNotifier {
 final NetworkCaller _networkCaller;

 AddProductViewModel(this._networkCaller); // ✅ DI constructor

 bool _inProgress = false;
 bool _isSuccess = false;

 bool get inProgress => _inProgress;
 bool get isSuccess => _isSuccess;

 Future<bool> addProduct(ProductModel product) async {
```

```
    _inProgress = true;
    notifyListeners();

    final response = await _networkCaller.postRequest(
      Urls.createProduct,
      product.toJson(),
    );

    _isSuccess = response.isSuccess;
    _inProgress = false;
    notifyListeners();

    return _isSuccess;
  }
}
```

🔹 **providers_view_models.dart (Register ViewModels with DI)**

```
import 'package:provider/provider.dart';
import 'package:rest_api_crud_app/view_model/add_product/add_product_view_model.dart';
import 'package:rest_api_crud_app/data/network_caller.dart';

final providerViewModels = <ChangeNotifierProvider>[
  // ✅ Inject NetworkCaller via context.read
  ChangeNotifierProvider<AddProductViewModel>(
    create: (context) => AddProductViewModel(context.read<NetworkCaller>()),
  ),

  // Add more ViewModels similarly...
];
```

🔹 **main.dart (Register all Providers including NetworkCaller)**

```
import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
import 'package:rest_api_crud_app/app.dart';
import 'package:rest_api_crud_app/core/provider/locale_provider.dart';
import 'package:rest_api_crud_app/core/provider/providers_view_models.dart';
import 'package:rest_api_crud_app/core/provider/theme_provider.dart';
import 'package:rest_api_crud_app/data/network_caller.dart';
import 'package:rest_api_crud_app/view_model/bottom_nav/main_bottom_nav_view_model.dart';
import 'package:rest_api_crud_app/view_model/search_app/search_product_view_model.dart';

void main() {
  runApp(
    MultiProvider(
      providers: [
        // ✅ Core DI service
        Provider<NetworkCaller>(
          create: (_) => NetworkCaller(), // Optional: pass token
        ),

        // ✅ App-wide core settings
        ChangeNotifierProvider(create: (_) => ThemeProvider()),
        ChangeNotifierProvider(create: (_) => LocaleProvider()),

        // ✅ Feature ViewModels (AddProductViewModel uses DI)
        ...providerViewModels,
        ChangeNotifierProvider(create: (_) => MainBottomNavViewModel()),
```

```
      ChangeNotifierProvider(create: (_) => SearchProductViewModel()),
    ],
    child: const RestApiCrudApp(),
  ),
 );
}
```

✅ **Summary**

File	What was done
network_caller.dart	Converted static to class-based DI-ready service
add_product_view_model.dart	Injected NetworkCaller via constructor
providers_view_models.dart	Registered AddProductViewModel with injected dependency
main.dart	Added NetworkCaller to Provider tree

Let me know if you also want to:

- Add get_it or Riverpod for scalable DI
- Inject multiple services into a single ViewModel
- Write widget/unit tests with mocked NetworkCaller

I'm here for all of it.