Flutter9\_Postman\_Service\_Model\_Json

**Temelden Zirveye Flutter: #9 Servis kullanımı, detaylar, future, try catch etc**

1. Jsonplaceholder sitesini gösterelim
2. Josn viewer adlı uzantıyı chrome a ekle

{

"userId": 1,

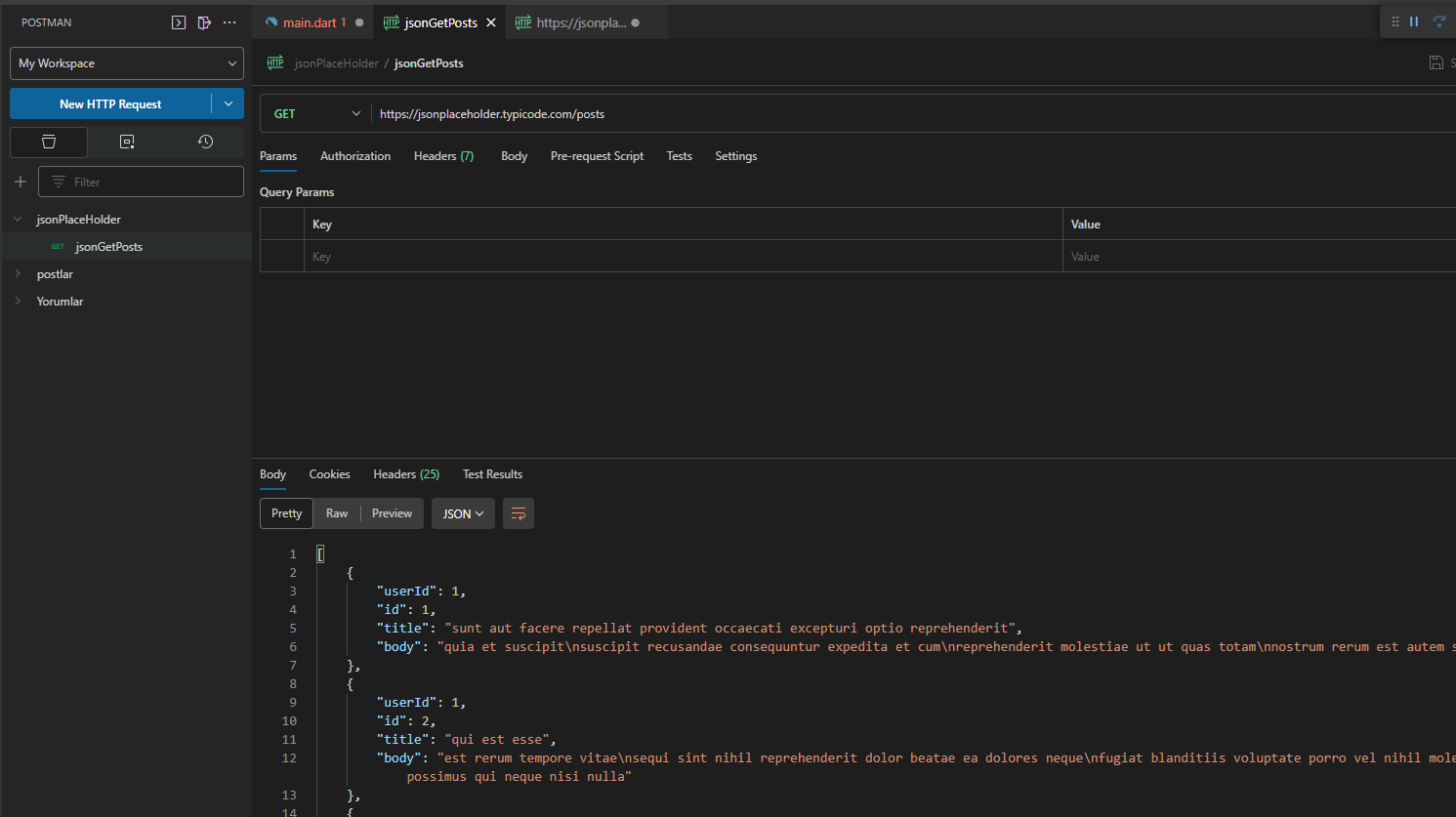
"id": 1,

"title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",

"body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut” },

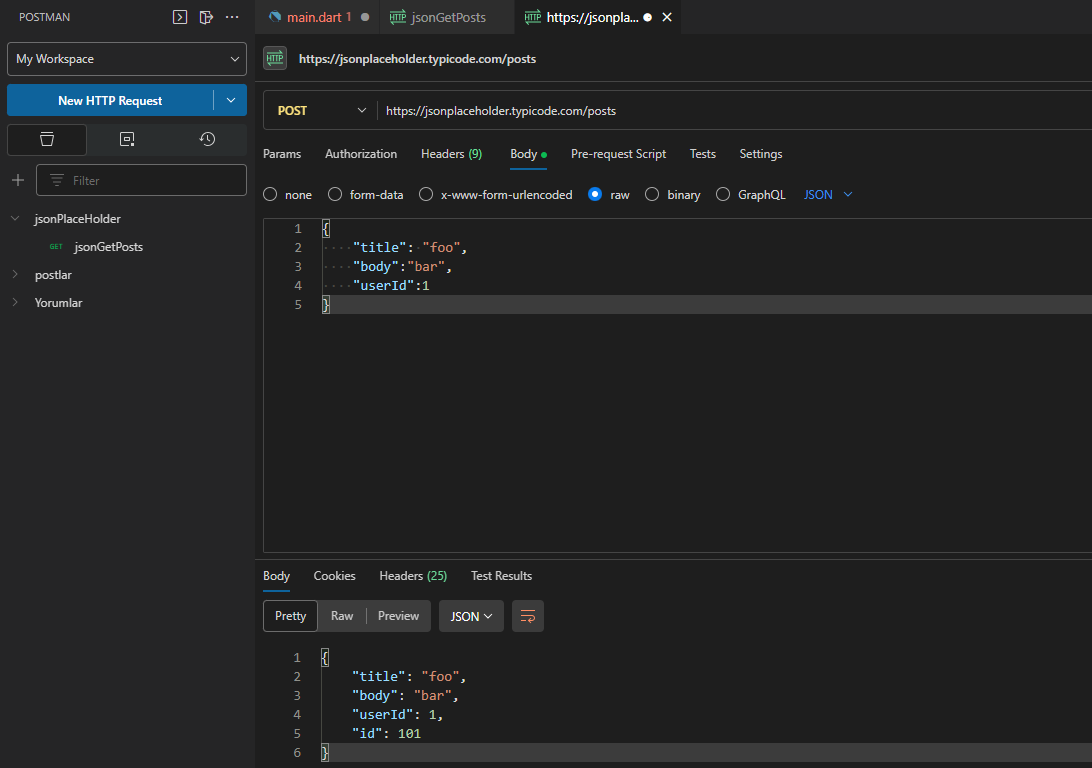
1. Postman kullanmak gerekiyor(visual studio code eklentisi olarak yükle)

3.1. Get olarak veri çekme

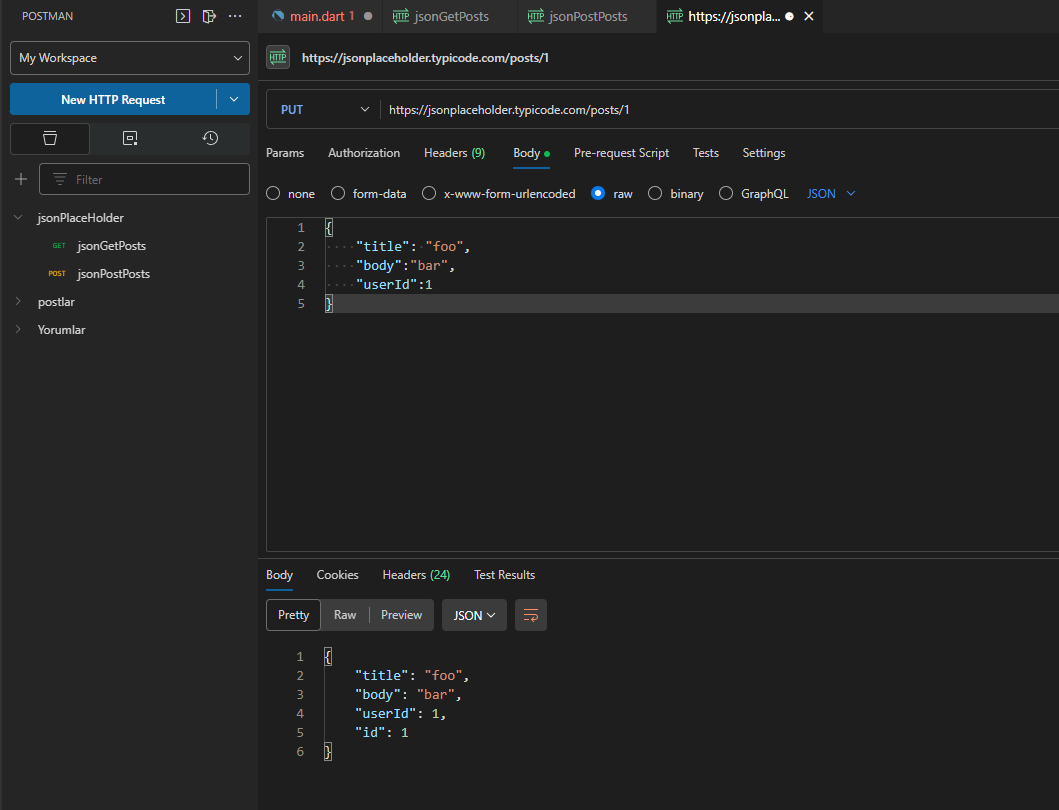


3.2. Post olarak veri gönderme ( json )

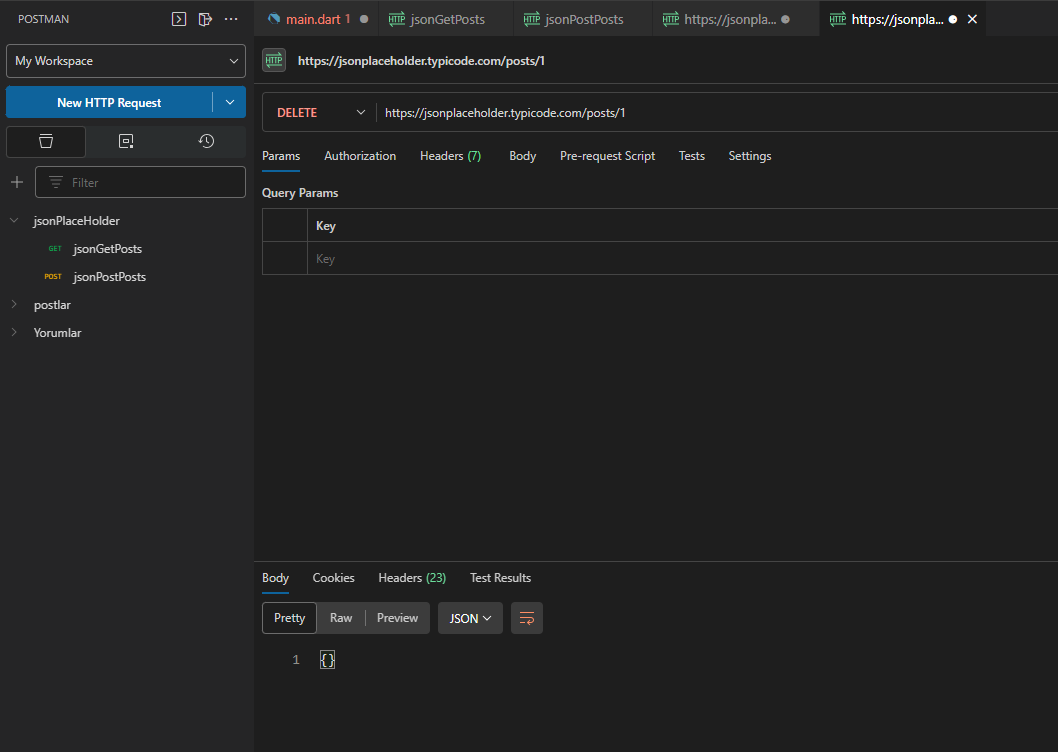
Body -> raw -> Json kısmına veri formatı aşağıdaki gibi yazılır.



* 1. Put olarak veri güncelleme ( id ye göre )



* 1. Delete olarak veri silme ( id ye göre )



1. Pub.dev den dio, http gibi servis paketlerini anlat.

4.1. view->command palette->dart add dependency->dio yazıp pubspec.yaml a ekle

4.2 lib->service folder->service\_learn\_view.dart oluşturalım.

4.3 model sınıfı için json to dart sitesini açalım. Ve postman da örnek olarak oluşturulan aşağıdaki kodu ekleyelim.

{

"title": "foo",

"body": "bar",

"userId": 1,

"id": 1

}

4.4. Modelimize PostModel ismi verelim ve Generate edelim. Kodu kopyalayalım



4.5. lib->service->post\_model.dart oluşturalım ve kodu buraya yapıştıralım.

4.6. service\_learn\_view.dart üzerinde bir ListView.builder ekleyelim.

4.7. PostModel modelini kullanarak Liste oluşturalım. İşaretli kodları ekleyelim

class \_ServiceLearnState extends State<ServiceLearn> {

  List<PostModel>? \_items;

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(),

      body: ListView.builder(

        itemCount: \_items?.length ?? 0,

        itemBuilder: (context, index) {

        return Text('Data');

      }),

    );

  }

}

4.8. Daha sonra verileri çekeceğimiz metodu ekleyelim.

Future<void> verileriGetir() async{

    final response=Dio().get('https://jsonplaceholder.typicode.com/posts');

  }

  @override

  void initState() {

    super.initState();

    verileriGetir();

  }

4.9. verileri getir metodunu yapalım.

Future<void> verileriGetir() async{

   final response=await Dio().get('https://jsonplaceholder.typicode.com/posts');

    if(response.statusCode==HttpStatus.ok){

      final \_datas=response.data;

      if(\_datas is List){

        setState(() {

          \_items=\_datas.map((e) => PostModel.fromJson(e)).toList();

        });

      }

    }

  }

4.10. margin ve padding verelim ve Card Widget ını Extract Widget diyerek \_PostCard diyerek çıkaralım. Ve gerekli düzenlemeleri yaptıktan sonra aşağıdaki gibi olacaktır.

class \_PostCard extends StatelessWidget {

  const \_PostCard({

    super.key,

    required PostModel? model,

  }) : \_model = model;

  final PostModel? \_model;

  @override

  Widget build(BuildContext context) {

    return Card(

      margin: EdgeInsets.all(8),

      child: ListTile(

        title: Text(\_model?.title ?? '',

            style: TextStyle(

                fontSize: 15,

                fontWeight: FontWeight.bold,

                color: Colors.red)),

        subtitle: Text(\_model?.body ?? ''),

      ),

    );

  }

}

4.11. aşağıdaki seçili alanı da değişelim.

body: ListView.builder(

        padding: EdgeInsets.all(8),

          itemCount: \_items?.length ?? 0,

          itemBuilder: (context, index) {

            return \_PostCard(model: \_items?[index]);

          }),

5.1. Projeyi daha efetif hale getirelim. İşaretli kodları projeye ekleyelim. Bazı alanları da değiştirelim

class \_ServiceLearnState extends State<ServiceLearn> {

  List<PostModel>? \_items;

  bool \_isLoading = false;

  late final Dio \_dio;

  final \_baseUrl='https://jsonplaceholder.typicode.com/';

  void changeLoading() {

    setState(() {

      \_isLoading = !\_isLoading;

    });

  }

   @override

  void initState() {

    super.initState();

    \_dio=Dio(BaseOptions(baseUrl: \_baseUrl));

    verileriGetir();

  }

5.2. verileriGetirAdvanced isminde yeni metodumuzu yazalım.

Future<void> verileriGetirAdvanced() async {

    changeLoading();// Veri var mı yok mu kontrolü

    final response =await \_dio.get('posts');

    if (response.statusCode == HttpStatus.ok) {

      final \_datas = response.data;

      if (\_datas is List) {

        setState(() {

          \_items = \_datas.map((e) => PostModel.fromJson(e)).toList();

        });

      }

    }

    changeLoading();

  }

6.1. lib->service->service\_post\_learn\_view.dart ı oluşturalım. Bu metodun amacı data göndermek olacaktır.