

Görsel Programlama

REST API ve Sorgu gönderme

Emir ÖZTÜRK

REST

- HTTP
- GET
- POST
- PUT
- DELETE



persons	
POST	Creates a new user in the database. (Note: JSON representation of a new user's data should be contained in the body.)
GET	Returns a list of all users in the database.
GET	Returns a user by supplying a unique user name.
PUT	Updates an existing user in the database. (Note: the JSON representation of the user should be supplied in the body, along with the user's name id.)
DELETE	Deletes an existing user when unique identifier is supplied.

REST

- Elemanlar için kök düğüm
- Düğüm sonrası / ile yolların belirlenmesi
- / Sonrası id parametreler

REST

GET

- Veri alımı için
- Parametrelili veya parametresiz
- Parametre yoksa genellikle tüm liste
- Id alanını parametre olarak kullanan sorgu

REST

POST

- Yeni kayıt eklemek için
- Body ile gönderim
- Sınıf -> JSON
- Genellikle parametresiz
- Kök düğüme sorgu
- Eklenen verinin döndürülmesi

REST

PUT

- Veri güncellemek için
- Post ile benzer
- Kök düğüme id ile sorgu
- Body ile yeni veri gönderimi
- Id alanındaki veriyi body ile gönderilen veri ile değiştirme
- Güncellenmiş verinin döndürülmesi

REST DELETE

- Veri silmek için
- Genellikle Id alanı ile
- Kök düğüme /id eklentisi ile sorgu
- Verinin silindiğine dair bir mesaj döndürülmesi

Flutter REST

- http paketi
- pubspec.yaml
- Servis için yeni bir sınıf
- Sınıf içerisinde http metotları
 - http.get()
 - http.post()
 - http.put()
 - http.delete()

```
dev_dependencies:  
  flutter_test:  
    sdk: flutter  
  http: ^0.13.4
```

Flutter REST Güvenlik

- Localhost denemelerinde sertifika hataları
- Erişim engelleri
- HttpOverrides

```
import (Debug, phone  
void main() {  
  HttpOverrides.global = MyHttpOverrides();  
  runApp(const MyApp());  
}
```

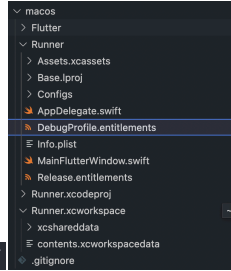
```
class MyHttpOverrides extends HttpOverrides {  
  @override  
  HttpClient createHttpClient(SecurityContext? context) {  
    return super.createHttpClient(context)  
      ..badCertificateCallback =  
        (X509Certificate cert, String host, int port) => true;  
  }  
}
```

Flutter REST

Güvenlik

- MacOS üzerinde erişim engeli
- DebugProfile.entitlements dosyasına ekleme
- Release.entitlements üzerine ekleme

```
<key>com.apple.security.network.client</key>  
<true/>
```



Flutter Http

- Future
- Dönen değerin türü
- Async
- Statik url tanımlanması
 - Örn. basePath = <https://127.0.0.1/>
- Yolların üzerine eklenmesi
 - basePath+"Users/"

Flutter Http

Get

- http.get(url, headers)
- Content-Type
 - application/json
- Charset
 - UTF-8

```
var response = await http.get(  
  url,  
  headers: <String, String>(  
    'Content-Type': 'application/json; charset=UTF-8',  
  ),  
);
```

Flutter Http Get

- Response'un istenilen türe çevrilmesi
- jsonDecode(response.body)
- Dönüştürbilmek için Sınıf.fromJson metodu
- Map<String,dynamic> dönen alanları sınıf alanlarına eşitleme

```
User.fromJson(Map<String, dynamic> json)

  : url = json['url'],
    userName = json['username'],
    password = json['password'],
    musicList = json['musicList'].cast<String>();

var response = await http.get(
  url,
  headers: <String, String>{
    'Content-Type': 'application/json; charset=UTF-8',
  },
);
```

Flutter Http Get

- Liste olarak dönen değerler
- liste = `jsonDecode(response.body)` as `List<dynamic>`
- Elde edilen listenin içerisindeki her elemanı `fromJson` ile map etme
- `toList()` ile listeye dönüştürme

```
return (jsonDecode(response.body) as List<dynamic>)
    .map((e) => UserSimilarity.fromJson(e))
    .toList();
```

Flutter Http

Post ve Put

- Get ile aynı
- Url ve headers'a ek olarak body
- body string olmalı
- Sınıf -> Json
 - toJson() metodu

```
Map<String, dynamic> toJson() {
    return {
        'userId': userId,
        'userName': userName,
        'password': password,
        'musicList': musicList
    };
}
```