

# Mobile Apps Workshop

Flutter: HTTP Requests

# Kommunikation

- Mit HTTP über REST APIs
- HTTP-Endpunkt + Request (GET, POST, PUT, DELETE, ...)
- Flutter: [http](#) Package

# http Package

- Package in pubspec.yaml definieren
- Import mit 'as http' ergänzen, um Namenskonflikte zu vermeiden
- Request ausführen

```
1 dependencies:  
2   flutter:  
3     sdk: flutter  
4  
5   http: <latest_version>
```

```
1 final response =  
2   await http.get('https://jsonplaceholder.typicode.com/albums/1');
```

# http Package

- Passende Klasse erstellen

```
1 class Album {  
2   final int userId;  
3   final int id;  
4   final String title;  
5  
6   Album({this.userId, this.id, this.title});  
7  
8   factory Album.fromJson(Map<String, dynamic> json) {  
9     return Album(  
10       userId: json['userId'],  
11       id: json['id'],  
12       title: json['title'],  
13     );  
14   }  
15 }
```

# http Package

- Response zu Objekt umwandeln, das dann in der App verwendet werden kann

```
1 Future<Album> fetchAlbum() async {
2   final response = await http.get('https://jsonplaceholder.typicode.com/albums/1');
3
4   if (response.statusCode == 200) {
5     // If the server did return a 200 OK response,
6     // then parse the JSON.
7     return Album.fromJson(json.decode(response.body));
8   } else {
9     // If the server did not return a 200 OK response,
10    // then throw an exception.
11    throw Exception('Failed to load album');
12  }
13 }
```

# Übung

- Code auschecken
  - > [HTTPExercise](#)
- TODOs bearbeiten, um die Aare-Temperatur abzufragen und anzuzeigen
- [Doku](#) zum Nachlesen

