

U08 | Bundesligatabelle



Figure 1: Cover für die achte Übungsaufgabe

Downloads

- Download des Starterpakets
- Download des Lösungsvorschlag

Aufgabe

In dieser Aufgabe sollen Sie eine App implementieren, die Daten, hier den Stand der Bundesliga, aus dem Internet lädt und den Inhalt in einer Tabelle anzeigt. Der Inhalt liegt als XML-String vor.

Hinweise

- Das Layout für die **Activity** und die **ListView**-Items sind bereits vollständig erstellt und können so verwendet werden.
- Datenquelle Link zum Downloaden der aktuellen Bundesligatabelle: <https://www.openligadb.de/api/getbltable/bl1/2018>
- Zum Verarbeiten der Daten benötigen Sie **FileReader** und **BufferedReader**:
 - **FileReader**: <https://docs.oracle.com/javase/8/docs/api/?java/io/FileReader.html>
 - **BufferedReader**: <https://docs.oracle.com/javase/8/docs/api/java/io/BufferedReader.html>



Platz	Team	Sp.	Pkt.	Torverh.
1	FC Bayern	34	78	88:32
2	Borussia Dortmund	34	76	81:44
3	RB Leipzig	34	66	63:29
4	Bayer Leverkusen	34	58	69:52
5	Borussia Mönchengladbach	34	55	55:42
6	VfL Wolfsburg	34	55	62:50
7	Eintracht Frankfurt	34	54	60:48
8	Werder Bremen	34	53	58:49
9	TSG 1899 Hoffenheim	34	51	70:52
10	Fortuna Düsseldorf	34	44	49:65
11	Hertha BSC	34	43	49:57
12	1. FSV Mainz 05	34	43	46:57
13	SC Freiburg	34	36	46:61
14	FC Schalke 04	34	33	37:55
15	FC Augsburg	34	32	51:71

Figure 2: Screenshot der BundesligaTabelle-App

- Die Dokumentation zur API finden sie hier: <https://github.com/OpenLigaDB/OpenLigaDB-Samples>

Konzept

1. Entpacken und öffnen Sie das bereitgestellte Projekt.
2. Modellieren Sie zunächst die `TableItem`-Klasse. Ein Objekt dieses Typs repräsentiert einen Tabellenplatz der Bundesliga. Hier sollen Informationen zu Platz, Teamname, geschossenen Toren, Gegentoren, Anzahl der Spiele und gesammelten Punkte gespeichert werden.
3. Nun komplettieren Sie den `TableItemAdapter`. Diesem werden bereits ein `Context` und eine `ArrayList<TableItem>` übergeben und alle Layoutelemente werden korrekt referenziert.
 1. In der `getView()`-Methode müssen nun lediglich die richtigen Daten auf das Layout gesetzt werden.
 2. Außerdem soll das komplette Item je nach Tabellenplatz eingefärbt werden. Implementieren Sie eine Methode, die den Platz entgegennimmt und die passende Farbe dazu zurückgibt. Als Farben sollen Standardfarben des Android-Frameworks nach folgenden Regeln verwendet werden:
 - Platz 1-3: `android.R.color.holo_green_dark`
 - Platz 4-6: `android.R.color.holo_green_light`
 - Platz 7-15: `android.R.color.darker_gray`
 - Platz 16: `android.R.color.holo_red_light`
 - Platz 17-18: `android.R.color.holo_red_dark`
4. Im nächsten Schritt soll die Klasse `TableDownloadTask` implementiert werden.
 1. Dem Konstruktor wird neben der `ArrayList<TableItem>` auch ein `DownloadListener` übergeben.
 2. In `doInBackground(String... params)` soll nun der Download der Datei durchgeführt werden und anschließend der eingeleseene `JSONString` zurückgegeben werden. Die URL für den Download wird der `doInBackground()`-Methode als Parameter übergeben.
 3. Dieser String soll nun in `onPostExecute()` zu `TableItems` verarbeitet werden. Dazu muss zunächst ein `JSONArray` mit dem übergebenen Parameter erstellt werden. Dieses soll dann mit einer Schleife durchlaufen werden. In jedem Durchlauf soll ein `JSONObject` geholt, die Werte ausgelesen und in ein `TableItem`-Objekt überführt werden, welches anschließend der `ArrayList` hinzugefügt wird.
 4. Nach dem Parsen der Daten soll der `DownloadListener` über das Ende des Downloads informieren.
5. Nun sollen die Einzelschritte in der `Activity` zusammengeführt werden.
 1. Diese implementiert den `DownloadListener`, um über den Abschluss des Downloads informiert werden zu können.
 2. Außerdem müssen zunächst `ListView`, `ArrayList` und `TableItemAdapter` initialisiert und korrekt verbunden werden.

3. Dem `ListView` soll nun noch eine Kopfzeile hinzugefügt werden. Dafür muss zuerst ein neuer `View` erstellt werden, indem (wie im Adapter) das Item Layout „aufgeblasen“ (engl. *to inflate*) wird. Die passenden Werte müssen nicht gesetzt werden, da diese bereits im XML definiert worden sind. Mithilfe der Methode `addHeaderView()` der Klasse `ListView` kann nun der `View` zur Liste hinzugefügt werden.
4. Anschließend wird ein `TableDownloadTask` erzeugt, dem als Parameter die URL übergeben wird. Bei Abschluss des Downloads soll der Adapter über geänderte Daten informiert werden, damit die Liste automatisch aktualisiert wird.

Anhang

Screenshots



2:06	Settings	Wi-Fi	Battery	LTE	Signal	Lock
3	RB Leipzig	34	66	63:29		
4	Bayer Leverkusen	34	58	69:52		
5	Borussia Mönchengladbach	34	55	55:42		
6	VfL Wolfsburg	34	55	62:50		
7	Eintracht Frankfurt	34	54	60:48		
8	Werder Bremen	34	53	58:49		
9	TSG 1899 Hoffenheim	34	51	70:52		
10	Fortuna Düsseldorf	34	44	49:65		
11	Hertha BSC	34	43	49:57		
12	1. FSV Mainz 05	34	43	46:57		
13	SC Freiburg	34	36	46:61		
14	FC Schalke 04	34	33	37:55		
15	FC Augsburg	34	32	51:71		
16	VfB Stuttgart	34	28	32:70		
17	Hannover 96	34	21	31:71		
18	1. FC Nürnberg	34	19	26:68		

Figure 3: Screenshot der Bundesligatabelle-App