



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Контрольная работа № 2.2
по курсу «Разработка мобильных приложений»
«HTTP JSON parser»

Студентка группы ИУ9-72Б Самохвалова П. С.

Преподаватель Посевин Д. П.

Москва 2023

1 Задание

Реализовать мобильное приложение выполняющее разбор данных в формате JSON получаемых по протоколу HTTP.

Индивидуальный вариант:

<https://mysafeinfo.com/api/data?list=jquery&format=json>

2 Практическая реализация

Исходный код программы представлен в листинге 2.

```
1 package com.example.httpjson
2
3 import android.os.Build
4 import android.os.Bundle
5 import android.os.StrictMode
6 import android.os.StrictMode.ThreadPolicy
7 import android.widget.ListView
8 import android.widget.Toast
9 import androidx.appcompat.app.AppCompatActivity
10 import org.json.JSONArray
11 import java.net.HttpURLConnection
12 import java.net.URL
13
14
15 class MainActivity : AppCompatActivity() {
16
17
18     override fun onCreate(savedInstanceState: Bundle?) {
19         super.onCreate(savedInstanceState)
20         setContentView(R.layout.activity_main)
21
22         if (Build.VERSION.SDK_INT > 9) {
23             val policy = ThreadPolicy.Builder().permitAll().build()
24             StrictMode.setThreadPolicy(policy)
25         }
26
27         val url = "https://mysafeinfo.com/api/data?list=jquery&format=
json"
28         val connection = URL(url).openConnection() as HttpURLConnection
29         val responseCode = connection.responseCode
30
31         try {
```

```

32         val data = connection.inputStream.bufferedReader().use { it.
readText() }
33         // ... do something with "data"
34         val toast = Toast.makeText(applicationContext, data, Toast.
LENGTH_LONG)
35         toast.show()
36         hanldeJson(data)
37
38     } finally {
39         connection.disconnect()
40     }
41
42 }
43
44
45
46
47 private fun hanldeJson(jsonString: String?) {
48
49     val jsonArray = JSONArray(jsonString)
50
51     val list = ArrayList<Jquery>()
52     var x = 0
53     while (x < jsonArray.length()) {
54
55         val jsonObject = jsonArray.getJSONObject(x)
56
57         list.add(
58             Jquery(
59                 jsonObject.getDouble("Version"),
60                 jsonObject.getString("ReleaseDate"),
61                 jsonObject.getString("LatestUpdate"),
62                 jsonObject.getString("Size"),
63                 jsonObject.getString("Notes"),
64                 jsonObject.getInt("ID"),
65             ))
66
67         x++
68     }
69
70     val adapter = ListAdapte(this, list)
71     val list2: ListView = findViewById(R.id.jquery_list);
72     list2.setAdapter(adapter);
73
74     val toast = Toast.makeText(applicationContext, x.toString(),
Toast.LENGTH_LONG)

```

```

75
76
77         toast.show()
78
79
80     }
81 }
82
83
84 package com.example.httpjson
85
86 data class JQuery (val version: Double, val releaseDate: String, val
    latestUpdate: String, val size: String, val notes: String, val id:
    Int ) {
87
88 }
89
90
91 package com.example.httpjson
92
93 import android.annotation.SuppressLint
94 import android.content.Context
95 import android.view.LayoutInflater
96 import android.view.View
97 import android.view.ViewGroup
98 import android.widget.BaseAdapter
99 import android.widget.Toast
100 import androidx.appcompat.widget.AppCompatTextView
101
102 class ListAdapte (val context: Context, val list: ArrayList<Jquery>) :
    BaseAdapter() {
103     override fun getCount(): Int {
104         return list.size
105     }
106
107     override fun getItem(position: Int): Any {
108         return list[position]
109     }
110
111     override fun getItemId(position: Int): Long {
112         return position.toLong()
113     }
114
115     @SuppressWarnings("MissingInflatedId", "ViewHolder")
116     override fun getView(position: Int, convertView: View?, parent:
    ViewGroup?): View {

```

```

117
118         val view: View = LayoutInflater.from(context).inflate(R.layout.
row_layout, parent, false)
119         val jqueryVersion = view.findViewById(R.id.jquery_version) as
AppCompatActivity
120         val jqueryReleaseDate = view.findViewById(R.id.
jquery_releaseDate) as AppCompatActivity
121         val jqueryLatestUpdate = view.findViewById(R.id.
jquery_latestUpdate) as AppCompatActivity
122         val jquerySize = view.findViewById(R.id.jquery_size) as
AppCompatActivity
123         val jqueryNotes = view.findViewById(R.id.jquery_notes) as
AppCompatActivity
124         val jqueryId = view.findViewById(R.id.jquery_id) as
AppCompatActivity
125
126         jqueryVersion.text = list[position].version.toString()
127         jqueryReleaseDate.text = list[position].releaseDate
128         jqueryLatestUpdate.text = list[position].latestUpdate
129         jquerySize.text = list[position].size
130         jqueryNotes.text = list[position].notes
131         jqueryId.text = list[position].id.toString()
132
133         view.setOnClickListener({
134             Toast.makeText(context, "Version: ${jqueryVersion.text}\
nReleaseDate: ${jqueryReleaseDate.text}\nLatestUpdate: ${
jqueryLatestUpdate.text}\nSize: ${jquerySize.text}\nNotes: ${
jqueryNotes.text}\nId: ${jqueryId.text}" , Toast.LENGTH_SHORT).show
()
135         })
136
137         return view
138     }
139 }
140
141
142 <?xml version="1.0" encoding="utf-8"?>
143 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http
://schemas.android.com/apk/res/android"
144     xmlns:app="http://schemas.android.com/apk/res-auto"
145     xmlns:tools="http://schemas.android.com/tools"
146     android:layout_width="match_parent"
147     android:layout_height="match_parent"
148     tools:context=".MainActivity">
149
150     <ListView

```

```

151         android:id="@+id/jquery_list"
152         android:layout_width="match_parent"
153         android:layout_height="match_parent"/>
154
155 </androidx.constraintlayout.widget.ConstraintLayout>
156
157
158
159 <?xml version="1.0" encoding="utf-8"?>
160 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
161     android:orientation="vertical"
162     android:layout_width="match_parent"
163     android:layout_height="match_parent"
164     android:padding="10dp">
165
166 <androidx.appcompat.widget.AppCompatTextView
167     android:id="@+id/jquery_version"
168     android:layout_width="match_parent"
169     android:layout_height="match_parent"
170     android:textSize="20sp"/>
171
172 <androidx.appcompat.widget.AppCompatTextView
173     android:id="@+id/jquery_releaseDate"
174     android:layout_width="match_parent"
175     android:layout_height="match_parent"
176     android:textSize="26sp"/>
177
178 <androidx.appcompat.widget.AppCompatTextView
179     android:id="@+id/jquery_latestUpdate"
180     android:layout_width="match_parent"
181     android:layout_height="match_parent"
182     android:textSize="16sp"/>
183
184 <androidx.appcompat.widget.AppCompatTextView
185     android:id="@+id/jquery_size"
186     android:layout_width="match_parent"
187     android:layout_height="match_parent"
188     android:textSize="16sp"/>
189
190 <androidx.appcompat.widget.AppCompatTextView
191     android:id="@+id/jquery_notes"
192     android:layout_width="match_parent"
193     android:layout_height="match_parent"
194     android:textSize="16sp"/>
195
196 <androidx.appcompat.widget.AppCompatTextView

```

```
197     android:id="@+id/jquery_id"
198     android:layout_width="match_parent"
199     android:layout_height="match_parent"
200     android:textSize="16sp"/>
201
202 </LinearLayout>
```

3 Результаты

Результаты работы программы представлены на рисунках 1 – 3.

4 Выводы

В результате выполнения рубежного контроля было реализовано мобильное приложение выполняющее разбор данных в формате JSON получаемых по протоколу HTTP.

HTTPJSON

1.0

2006-08-26T00:00:00

null

null

First stable release

1

1.1

2007-01-14T00:00:00

null

null

null

2

1.2

2007-09-10T00:00:00

1.2.6

54.5 kb

null

3

1.3

2009-01-14T00:00:00

1.3.2

55.0 kb

Рис. 1 — Результаты

HTTPJSON

1.0

2006-08-26T00:00:00

null

null

First stable release

1

1.1

2007-01-14T00:00:00

null

null

null

2

1.2

2007-09-10T00:00:00

1.2.6

54.5 kb

null

3

1.3

2009-01-14T00:00:00

1.3.2

55.0 kb

Version: 1.2

ReleaseDate: 2007-09-10T00:00:00

LatestUpdate: 1.2.6

Size: 54.5 kb

Notes: null

Id: 3

HTTPJSON

null

null

null

2

1.2

2007-09-10T00:00:00

1.2.6

54.5 kb

null

3

1.3

2009-01-14T00:00:00

1.3.2

55.9 kb

Sizzle Selector Engine introduced into core

4

Version: 1.3

ReleaseDate: 2009-01-14T00:00:00

LatestUpdate: 1.3.2

Size: 55.9 kb

Notes: Sizzle Selector Engine introduced into core

Id: 4

1.4

2013-01-14T00:00:00

1.4.4

76.7 kb

null

5

Рис. 3 — Результаты