

**APPROACHES TO THE DEVELOPMENT OF MOBILE APP FOR THE ANDROID  
OPERATION SYSTEM AND INTERACTION WITH RESTFUL API AS AN EXAMPLE  
OF OFFICIAL AGGREGATE ERIP “EXPRESS PAYMENT” OF THE “TRIINKOM” COMPANY**

**VLADIMIR SAVROV, DMITRIY PYATKIN**  
**Polotsk State University, Belarus**

*The article considers the approach to the development of a mobile app the Android operation system and interaction with RESTful API as an example of official aggregate ERIP “Express Payment” of the “Triinkom” company under the HTTPS protocol. There are also descriptions of important points of designing the app architecture.*

**Introduction.** We live in a world of modern technologies, where every day we can see the forward movement, discovery of new and improvement of already existing technologies. One of the events, which have changed human life, was finding a global network the Internet. Nowadays, it is hard to imagine our world without the Internet. With the help of it, people can find different kinds of information, communicate, shop, entertain themselves, study and work. Development of the Internet technologies makes people’s life easier. Most people of the society have either personal computers or laptops, and also different kinds of smartphones and tablets. The question that appears is how it can be easily to implement any kind of interaction between different devices in any place and time. If to extract all the opportunities from all mobile devices, then you can include more users, increase the sales and service monetization.

Therefore, we have to think of the universal analogue, which can face any challenge without an additional equipment in any place and time.

**Ways for solving problems.** Wide spreading of such apps as browsers, lets us talk about web-apps as a cross-platform. For example, we can use this method in order to create an interface between computer and mobile phone. However due to the fact, that it is important to save the traffic on the mobile phone, fast loading with a bad connection or working of an app in offline mode with differed delivery of information, then browsing of web-site on the web-browser doesn’t quite fit other major web-services. The use of mobile apps also will remove the loading from the server, as pictures and other information won’t be loaded.

It was decided to use an interaction with RESTful API for saving traffic and to simplify sharing messages with a server.

RESTful API – is an architectural style of interaction components of a distributed app in the network. REST represents itself as an agreeable set of restrictions, taken into account when designed by a distributed hypermedia-system. In special cases(online stores, searching systems, other systems based on data) it increases productivity and simplifies an architecture. In a broad sense, all components with REST interact similar to interactions of clients and servers on the World Wide Web.REST are an alternative to RPC. On the Internet, a call for remote procedure represents a simple HTTP request (usually “Get” or “Post”; this kind of request is called “REST request”), and the important data can be transmitted as a query option.

It was decided to use a library from Googlecompany, which is called the “Google Cloud Messaging” in order to receive PUSH-notification from the server. It is easy to connect it by using the automatic system of the Gradle assembly and by adding it to the configuration file `compile 'com.google.android.gms:play-services-gcm:9.6.1'`, `apply plugin: 'com.google.gms.google-services'` and `classpath 'com.google.gms:google-services:3.0.0'` and to generate this configuration file in the JSON format on the official Google site.

Gradle – is an autobuildsystem, which is built upon Apache Ant and Apache Maven principles, but represents DSL in a Groovy language instead of the traditional XML-shape presentation of the configuration project.

It was decided to use the HTTPS protocol and the library for encryption from the Apache company in order to protect information of an app and messages. To connect it you can use the Gradle by adding to configuration file the line `compile 'commons-codec:commons-codec:1.10'`.

HTTS (HyperText Transfer Protocol Secure) – is an expansion of the HTTPprotocol, which supports encryption. The data transferred via HTTPS protocol are “packaged” in a cryptographic protocol SSL and TLS. Unlike HTTP, the TCP-port 443 is used for HTTPS by default.

It was decided to use the data exchange in the JSON format, since through its conciseness comparing with XML, the JSON format will be more suitable for serialization of complex structures. When we talk about web-apps, in this manner it’s suitable in the tasks of data exchanges both between browser and server (AJAX), and

between servers themselves (software HTTP-interface). It was decided to use the standard programming language package JAVA "org.json.JSONObject" in order to transfer the JSON into the object and back.

JSON (JavaScript Object Notation) – is a text format of the data exchange based on JavaScript. As many other text formats, JSON is easy to read. The JSON format was designed by Douglas Crockford.

It was decided to use the Material Design pattern in order to develop the user-friendly and attractive interface. For its use you can add the following lines to the configuration file Gradle: *compile 'com.android.support:design:24.2.1', compile 'com.android.support:appcompat-v7:24.2.1', compile 'com.github.rey5137:material:1.2.2'*.

The Material Design – is a design of software and apps that belongs to the Android operating system from Google. First presented at the Google I/O conference on 25 June 2014. The idea of design lies in apps, which open and fold like cards by using shadow effects. The ideas of Google designers, that apps should not have rough edges and cards should switch among themselves smoothly and almost invisible.

It was decided to use embedded Android operation system database SQLite for saving big data and small configuration data by using Android Shared Preferences.

SQLite – is an embedded cross-platform relational database, that supports a complete set of SQL and available in the source code. In 2005 the project was rewarded for Google-O'Reilly Open Source Awards.commands.

Android Shared Preferences – is an interface, which allows you manually to seek and manipulate data app settings with the help of method call get SharedPreferences from your Activity.

**Methods for solving problems.** Obviously, while implementing an app it is important to create some architecture. The model MVC will be very suitable. It consists of 3 elements: model, submission and controller.

The model is certain data, that describe subject area and some operation can be implemented over them. The user also can be provided with the information on a graphical interface.

The submission is thus that graphical interface.

The controller implements some algorithms for receiving data submission, processing and sending them back.

The Layout is used in the XML mark-up language for the submissions development in Android app. To create it you can type it manually or use embedded tools in IDE.

Working with sending and processing data was convenient to create separate classes, which were done while developing mobile apps. Sending data occurs using Apache library through the HTTPS protocol. Tables were created in relational database for saving data from server. Token that was received while authorizing is convenient to keep in Shared Preferences for its loading at the next application launch. The structure of primary and foreign relation keys in the database are presented in table 1.

Table 1 – Primary and foreign relation keys

№ p.n.	Table name	Primary key	Foreignkeys	Description
1	Invoices	PK invoice_no	FK currency, FK service_no	The table contains headers invoices
2	Invoice_details	PK invoice_no	FK _invoice	The table contains details invoices
3	Payments	PK payment_no	FK currency, FK service_no	The table contains headers payments
4	Payment_details	PK payment_no	FK_payment_no	The table contains details payments
5	Directory_currencies	PK currency		The tablecontainscurrencies
6	Directory_services	PK service_no		The tablecontainsservices

The verification of connection with the Internet was implemented at every request to API.

To receive PUSH-notifications the GCM Receiver is registered at startup as a separate server, but during the closing, apps are being destroyed.

There were used following methods with RESTful API for action distinction with resources at the HTTP-methods level:

1. GET — generation of the resource.
2. POST —creation of the resource.
3. PUT — updating of the resource.
4. DELETE — removing of the resource.

## ITC, Electronics, Programming

Nowadays there are many different mobile devices with different screen resolution and hardware. There is no opportunity to test this app on all these devices. That's why it was decided to use the server Firebase Cloud Messaging from Google company for bug tracking system of an app and for keeping statistics on the used machines and Android versions.

Each activity has its controllers, models, adapters and submissions. The app is optimized for different screen resolutions and Android operation system versions starting from 4.0.

Example of JSON response obtaining of new accounts:

```
{
  "Items": [
    {
      "InvoiceNo": 7841,
      "AccountNo": "Test238",
      "Status": 1,
      "Created": "20160905173134",
      "Expiration": "20160905",
      "Amount": "102,00",
      "Currency": 933,
      "ServiceNo": 3050
    },
    {
      "InvoiceNo": 7840,
      "AccountNo": "Test237",
      "Status": 1,
      "Created": "20160905173134",
      "Expiration": "20160905",
      "Amount": "102,00",
      "Currency": 933,
      "ServiceNo": 3050
    }
  ]
}
```

Screen shapes of mobile app are presented in figure 1 and 2:

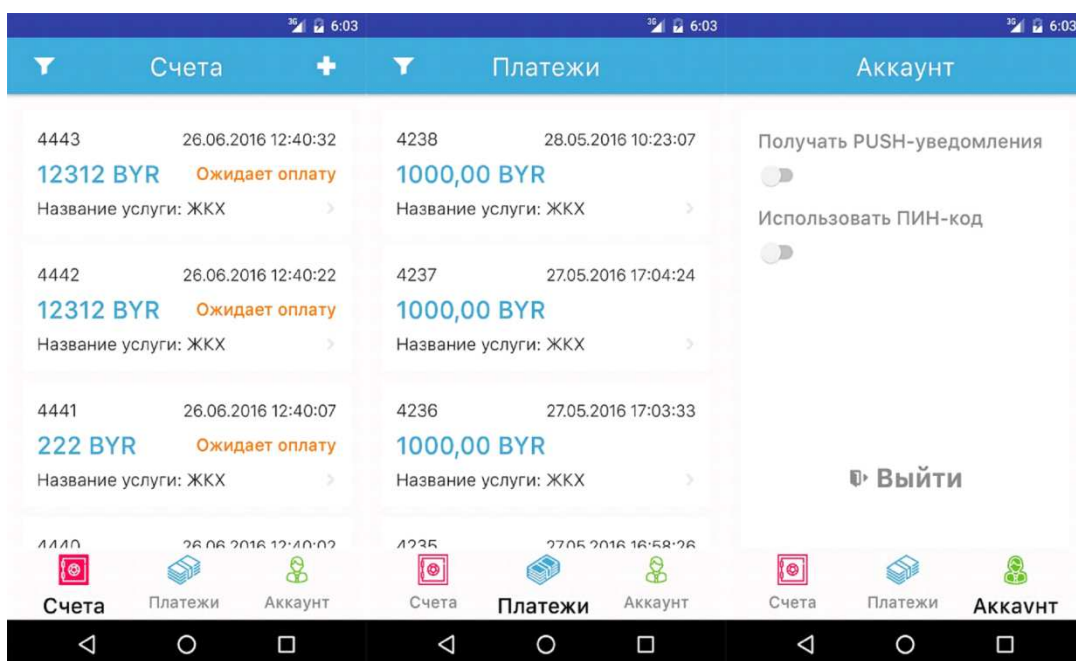


Fig. 1. Screen shapes of mobile app.

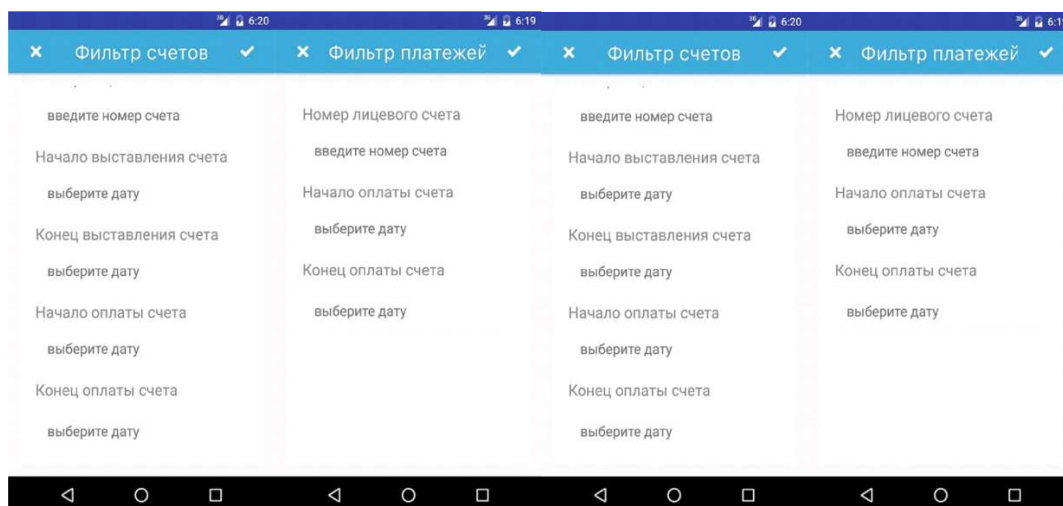


Fig. 2. Screen shapes of mobile app.

**Conclusion.** In conclusion, we should emphasize, that the correct approach while developing a mobile app is the use of modern libraries, that allow us to carry out its tasks. We must stick to recommendations from Google company, which has developed and still developing Android operation system and to use the design of the Material Design pattern for creating user-friendly and attractive interface.

The article considers popular and necessary libraries and describes the opportunities of their implementation in application developing. It also describes the safe way of data exchange between application and server.

The use of the MVC pattern and object-oriented approach to programming also can ease the process of development quiet well. In particular, it allows to avoid repetitions in code.

#### REFERENCES

1. RESTful API [Electronic resource]. – 2016. – Mode of access: <https://en.wikipedia.org/wiki/REST>. – Date of access: 22.12.2016.
2. Gradle [Electronic resource]. – 2016. – Mode of access: <https://en.wikipedia.org/wiki/Gradle>. – Date of access: 20.12.2016.
3. HTTPS [Electronic resource]. – 2016. – Mode of access: <https://en.wikipedia.org/wiki/HTTPS>. – Date of access: 24.12.2016.
4. JSON [Electronic resource]. – 2016. – Mode of access: <https://en.wikipedia.org/wiki/JSON>. – Date of access: 20.12.2016.
5. Material Design [Electronic resource]. – 2016. – Mode of access: [https://en.wikipedia.org/wiki/Material\\_Design](https://en.wikipedia.org/wiki/Material_Design). – Date of access: 26.12.2016.
6. SQLite [Электронный ресурс]. – 2012. – Режим доступа: <https://habrahabr.ru/post/149356>. – Дата доступа: 23.12.2016.
7. AndroidSharedPreferences [Электронный ресурс]. – 2012. – Режим доступа: <https://habrahabr.ru/post/139330>. – Дата доступа: 27.12.2016.