

УДК 004.457

ПОДХОД К ВЗАИМОДЕЙСТВИЮ АНДРОИД-ПРИЛОЖЕНИЯ С БАЗОЙ ДАННЫХ ВЕБ-САЙТА

А.В. ОСИПОВА*(Представлено: канд. техн. наук, доц. А.Ф. ОСЬКИН)*

Рассматривается подход взаимодействия серверной части Андроид-приложения с базой данных MySQL веб-сайта. Представлен один из вариантов подключения Андроид-приложения к базе данных путем комбинирования JSON + PHP + MySQL для создания соединения.

Интернет технологии стали неотъемлемой частью нашей повседневной жизни. Они развиваются с каждым днем все больше, охватывая разные её стороны: социальные сети, разнообразные приложения, календари, напоминания, журналы контроля, веб-сайты и т.д. Ранее пользователи могли работать с Интернет-ресурсами только через персональные компьютеры или ноутбуки, но с развитием смартфонов и планшетов все необходимое остается под рукой 24 часа в сутки. Большинство девайсов работает на операционной системе Android.

При использовании сети Интернет любой пользователь может посещать разнообразные сайты и по желанию регистрироваться там. Важным показателем «современного» веб-сайта является возможность поддержки его мобильной версией или приложением. Но из этого вытекает вопрос, как синхронизировать данные базы веб-сайта и мобильного приложения.

Большинство веб-сайтов написаны на языке PHP и работают с базой данных MySQL, а мобильное приложение может быть разработано на языке Java через среду разработки AndroidStudio. И для того чтобы синхронизировать данные веб-сайта с мобильным приложением можно применить связку передачи данных: из MySQL через JSON объекты в AndroidStudio.

Решение задачи. Очень часто возникает ситуация, когда для разработанного веб-сайта нужно создать мобильное приложение. В основном мобильные приложения, синхронизированные с веб-сайтами, нуждаются в авторизации пользователей и минимальном, но достаточном количестве данных для работы, так как характеристики смартфоном ниже, чем у персональных компьютеров или ноутбуков. Для этого разработчику необходимо продумать функциональную часть приложения, его интерфейс и способ получения данных из базы данных.

Рассмотрим способ подключения базы данных MySQL через PHP файл JSON запросами с использованием библиотеки Volley.

PHP (*PHP: Hypertext Preprocessor* – «PHP: препроцессор гипертекста»; первоначально *Personal Home Page Tools* – «Инструменты для создания персональных веб-страниц») – скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов [1].

MySQL – свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQLAB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией [2].

JSON (англ. *Java Script Object Notation*) – текстовый формат обмена данными, основанный на Java Script. Как и многие другие текстовые форматы, JSON легко читается. За счёт своей лаконичности по сравнению с XML, формат JSON может быть более подходящим для сериализации сложных структур. Если говорить о веб-приложениях, то в таком ключе он уместен в задачах обмена данными как между браузером и сервером (AJAX), так и между самими серверами (программные HTTP-сопряжения).

Поскольку формат JSON является подмножеством синтаксиса языка JavaScript, то он может быть быстро десериализован встроенной функцией eval(). Кроме того, возможна вставка вполне работоспособных JavaScript-функций. В языке PHP, начиная с версии 5.2.0, поддержка JSON включена в ядро в виде функций json_decode() и json_encode(), которые сами преобразуют типы данных JSON в соответствующие типы PHP и наоборот [3].

Volley это HTTP библиотека, которая делает сетевой обмен Android приложений более легким, и самое главное, более быстрым. Volley доступен для использования через открытое AOSP хранилище.

Библиотека Volley обладает следующими преимуществами:

- автоматическое планирование сетевых запросов;
- несколько параллельных сетевых подключений;

- прозрачное кэширование на диске и в памяти ответов, используя стандартную HTTP когерентность кэша;
- поддержка приоритетов запросов;
- API для отмены запроса. Можно отменить один запрос, или можно установить блокировки или область запросов для отмены;
- простота настройки, например, для повторных попыток и отсрочки;
- строгий порядок, который позволяет легко и правильно заполнить пользовательский интерфейс данными, полученными из сети асинхронно.
- инструменты для отладки и трассировки.

Данная библиотека легко интегрируется с любым протоколами и поставляется с поддержкой сырых строк, изображений и JSON [4].

Чтобы реализовать взаимодействие Андроид-приложения и веб-сайта, необходимо подключить библиотеку *com.mcxiaoke.volley:library:1.0.19* в AndroidStudio в файле *build.gradle*. После того как библиотеку подключили, будут доступны для работы два класса *Request Queue* и *Request*, и необходимо подключить использование интернета в файле *AndroidManifest.xml*. *Request Queue* используется для отправки сетевых запросов, этот класс можно создавать в любой момент, но, как правило, его создают во время запуска и используют как Singleton. *Request* содержит все необходимые детали для создания вызова Web API. Например, какой метод использовать (GET или POST), данные запроса, *responselister*, *errorlister*.

Для примера можно рассмотреть способ авторизации пользователя веб-сайта через мобильное приложение с получением доступа к данным учетной записи.

Чтобы подключиться к базе данных, предварительно нужно создать .php файлы в нем прописать функционал получения данных из выбранной БД и вывод их в формате JSON. Впоследствии Андроид-приложение будет обращаться к этому файлу для подключения к базе веб-сайта, отправки и получения данных через URL прописанный в Java коде приложения.

При заполнении полей ввода логина/пароля и отправке их методом *request* в качестве ответа пользователь получает Json объект с оповещением удачной или ошибочной авторизации *JsonObject.get()*.

При обработке POST-запросов необходимо помнить то, что разбор параметров происходит не автоматически, а в результате явного выхода метода *protected Map<String, String>getParams() throws Auth Failure Error*.

После успешной авторизации система аналогичным способом обращается к базе данных непосредственно по идентификатору пользователя, предоставляя тем самым доступ к учетной записи. Кроме этого, данная библиотека обеспечивает прозрачность дискового кэширования и кэширования в памяти.

Методы решения задач. В качестве модели архитектуры мобильного приложения и веб-сайта лучше всего использовать модель «пользователь – оболочка – БД – администратор». При такой архитектуре можно производить преобразования одной из частей модели, и изменения для других ее частей будут не столь глобальны. Каждая часть данной модели обладает своими характеристиками, которые задают предметную область, которая может быть отражена в графическом интерфейсе, т.е. в оболочке.

Пользователь реализует некоторые методы для отправки и получения данных. Часть архитектуры как БД – администратор управляет всем содержимым контентом.

Заключение. Для синхронизации данных веб-сайта и мобильного приложения с операционной системой Android удобно использовать связку: MySQL база данных, конфигурационный PHP файл и Json объекты. Со стороны Андроид приложения в качестве коннектора лучше использовать библиотеку Volley, это даст возможность напрямую подключиться к .php файлу с настройками подключения базы данных и позволит напрямую обмениваться данными между мобильным приложением и веб-сайтом. Кроме всего прочего, данный способ позволяет легко ориентироваться в коде программы, что значительно упрощает ее дальнейшую модернизацию, расширение функционала и исправление ошибок, возникших в ходе разработки. Архитектура приложений «пользователь – оболочка – БД – администратор» позволит производить модернизацию одной из частей модели, что, в свою очередь, не приведет к глобальным изменениям для других ее частей.

ЛИТЕРАТУРА

1. Википедия – свободная энциклопедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/PHP>. – Дата доступа: 22.09.2016.
2. Википедия – свободная энциклопедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/MySQL>. – Дата доступа: 22.09.2016.
3. Википедия – свободная энциклопедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/JSON>. – Дата доступа: 23.09.2016.
4. ОС для многоэкранных устройств [Электронный ресурс]. – Режим доступа: <http://developer.android.unlimited-translate.org/training/volley/index.html>. – Дата доступа: 23.09.2016.