

УДК 004.9

## ИСПОЛЬЗОВАНИЕ REST API В ANDROID-ПРИЛОЖЕНИЕ

Фещук К.В., студент гр. ПИБ-192, IV курс  
Сладкова Е.А., студент гр. ПИБ-192, IV курс  
Глебова Е.А., студент гр. ИВа-211, II курс  
Кузбасский государственный технический университет  
имени Т.Ф. Горбачева,  
г. Кемерово

С развитием технологий человечество стало потреблять и обрабатывать огромный объём информации. Например, заходя на сайты начало нулевых, пользователь мог обратить внимание на статичный контент на странице, изменяющийся крайне редко руками человека. На современных веб-ресурсах посетитель может наблюдать множество разнообразного контента, который также будет изменяться через какое-то небольшое время. Позволяют это сделать API сервисы, которые предоставляют наполнения, от забавных изображений животных (кошек, собак, лис), до вполне серьезных: погоды, карт местности, рекламы. Современные android-приложения тоже не остаются в стороне от актуализации контента, все больше неся в себе динамические элементы в дизайне, а также в самом наполнении. Помогает им в этом инструмент REST API.

REST API – интерфейс, позволяющий взаимодействовать двум приложениям между собой, например, производить передачу информации между сайтом и мобильным приложением. Данный термин представляет собой две аббревиатуры. API (Application Programming Interface) или программный интерфейс приложения – набор правил, который позволяет взаимодействовать двум приложениям между собой. REST (Representational State Transfer) или передача состояния представления – представляет собой программную архитектуру, которая определяет рабочую среду API [1].

Смартфоны стали неотъемлемой частью человеческой жизни. Данные мобильные телефоны наполнены множеством приложений, которые используют контент, получаемый из сервисов API. Так давайте же рассмотрим, как android-приложение взаимодействует с таким механизмом.

Для взаимодействия с API сервисами необходима сеть интернета. Приложение должно запросить доступ в файле манифеста на его использование при помощи тега *uses-permission*. Это разрешение будет выглядеть следующим образом: `<uses-permission android:name="android.permission.INTERNET" />`. Затем необходимо создать фоновый поток, так как операционная система Android запрещает выполнение сетевых операций в основном потоке. В этом нам поможет класс *AsyncTask*, а именно его метод *execute()*. Он ожидает единственный аргумент,

который является объект *Runnable*, предполагается, что это будет URL адрес сервиса для получения информации и дальнейшей работы с ней в приложении. Далее необходимо создать HTTP-соединение, для этого используем экземпляр класса *HttpURLConnection* или *HttpsURLConnection*, в которое будем передавать возвращаемое значение из метода *openConnection()* класса *URL*. Замет необходимо произвести добавление заголовков в запрос. В этом поможет ранее созданный объект *HttpURLConnection*, а именно его метод *setRequestProperty()*. Это необходимо для того, чтобы сервис API мог идентифицировать приложение, с которым производит обмен данных. Также, в запрос можно добавить несколько заголовков, вызывая метод *setRequestProperty()* не один раз. После передачи заголовков запросов, можно заняться чтением ответов. Будем производиться данное действие при помощи объекта *HttpURLConnection*, а именно его метода *getResponseCode()*. Если ошибок в результате соединения не произойдет, то вызывается метод *getInputStream()*, но так как запрашиваемая информация приходит в формате JSON, то разумнее использовать не объект *InputStream*, а *InputStreamReader*. После получения данных необходимо их разобрать. Для понимания полученной информации среди инструментов в android-разработки присутствует класс *JsonReader*. Он позволяет легко обработать полученные от сервиса файлы JSON. После извлечения всей полученной информации, необходимо применить метод *close()*, который освободит все используемые ресурсы. Затем закрываем соединение. Поможет нам в этом метод *disconnect()* объекта *HttpURLConnection*. Не стоит забывать о разных видах HTTP [2]. В чтение данных использовался метод GET, он явно не указывается. Для отправки информации на сервер лучшим вариантом будет использование POST-запроса. Также рекомендуется для сокращения пропускной способности и большей отзывчивости приложения кэшировать HTTP ответы.

На данный момент существует множество API сервисов. Они применяются на сайтах для большей динамичности изображения на экранах пользователей. Технологии стали неотъемлемой частью нашей жизни и REST API позволяет получать актуальную информацию из удаленного источника, тем самым упрощая получение контента пользователем. Android-приложения тоже не стоят на месте и используют данный механизм [3]. Существуют классы и методы, которые дают возможность разработчикам полноценно работать с API сервисами в рамках мобильных устройств, а также делать код более читаемым и лаконичным.

### Список литературы:

1. Android для начинающих: использование REST API // Code Tutorials URL: <https://code.tutsplus.com/> (дата обращения: 22.03.2023).
2. Что такое RESTful API? // AWS URL: <https://aws.amazon.com/> (дата обращения: 22.03.2023).
3. REST API // SkillFactory URL: <https://blog.skillfactory.ru/> (дата обращения: 22.03.2023).