

УДК 004

## **ПРИМЕНЕНИЕ REST API ПРИ РАЗРАБОТКЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДИАГНОСТИКИ РАБОТОСПОСОБНОСТИ ПЫЛЕВЫХ ДАТЧИКОВ**

Люкина С.Ю., студент гр. ПИБ-172, IV курс  
Научный руководитель: Гиниятуллина О.Л., к.т.н., доцент  
Кузбасский государственный технический университет  
имени Т.Ф. Горбачёва  
г. Кемерово

На сегодняшний день количество различных веб и мобильных приложений постоянно увеличивается, и при этом каждое из них выполняет определенные функции и помогает решить какие-либо задачи. Многие из этих приложений обращаются к сторонним сервисам для доступа к необходимым ресурсам, например, для получения какой-либо информации или для доступа к функциональным возможностям другой программы. Это происходит, потому что многие задачи уже имеют готовые решения, предлагаемые сторонними разработчиками, и часто гораздо удобнее прибегнуть именно к такому решению, чем придумывать новое. Именно для этого используют технологию REST API.

API (от англ. application programming interface - программный интерфейс приложения) — описание способов взаимодействия (набор классов, функций, процедур, структур или констант) какой-либо программы со сторонними серверами и программами [1]. По сути API – это ресурс, открытый сервером для удаленного подключения. Так, например, при нажатии на определенную кнопку в приложении, вызываются функции, которые отправляют с помощью API запрос удаленным сервисам в сети для получения нужной информации. Таким образом, программа запрашивает ресурс, а сторонний сервис предоставляет его ей. Так, например, с помощью API работают в приложениях кнопки Facebook, Twitter и другие, которые взаимодействуют с социальными сетями, а также видео YouTube, которые загружают видео с официального сайта.

REST (от англ. representational state transfer — «передача состояния представления») — это архитектурный стиль взаимодействия компонентов распределенного приложения в сети, который позволяет повысить производительность и упростить архитектуру [2]. Следует отметить, что REST не является стандартным протоколом, а скорее общим стилем, которому следует API, именно поэтому REST API иногда называют RESTful API или «REST-подобными». Этот архитектурный подход определяет, как API должны выглядеть. Этому набору правил и следуют разработчики при проектировании и создании своих приложений. Согласно одному из этих правил, при

обращении к определенному адресу, в ответ должен прийти соответствующий набор данных (ресурс). В качестве протокола передачи данных для запросов и ответов REST API использует HTTP-протокол. Таким образом, компоненты в REST взаимодействуют аналогично клиентам и серверам во Всемирной паутине. Одним из преимуществ REST API также является отсутствие ограничений на формат сообщений для запросов и ответов. Именно поэтому разработчики могут использовать любой необходимый формат, включая XML, JSON, CSV, RSS, Atom, HTML и другие. Несмотря на разнообразие параметров формата сообщений, большинство REST API используют JSON, так как он обеспечивает легкий, простой и более гибкий формат сообщений, который увеличивает скорость связи.

API, использующие протокол HTTP для передачи запросов и ответов, рассматриваются как веб-сервисы [1]. Веб-сервис — это веб-приложение, предоставляющее ресурсы в формате, используемом другими компьютерами. Веб-сервисы включают в себя различные типы API, в том числе REST, и, в основном, представляют собой запросы и ответы между клиентами и серверами. Клиент делает запрос на ресурс, и сервер API отвечает на этот запрос. При этом, как клиент, посылающий запрос, так и сервер, предоставляющий ему ответ, могут использовать любой язык программирования или платформу, потому что запрос сообщения и ответ сделаны через общий веб-протокол HTTP. На данный момент большинство современных сайтов и приложений используют по крайней мере несколько сторонних API.

Технология REST API также использовалась при разработке мобильного приложения диагностики работоспособности пылевых датчиков. На сегодняшний день существует множество разновидностей датчиков со своими характеристиками и параметрами. Это приложение позволяет оперативно осуществлять мониторинг работы различных датчиков, в том числе датчиков, передающих данные по HTTP-протоколу через веб-сервис. Для этого был использован REST API.

Для начала пользователю необходимо ввести в приложение информацию о типе и модели датчика. Затем пользователь нажимает на кнопку в приложении, чтобы получить необходимые данные. Аналогично вышеупомянутому примеру при нажатии на кнопку в приложении вызовутся функции, которые отправят запрос к удаленному веб-сервису с помощью API для получения соответствующих актуальных данных с датчика. В ответ на запрос сервис вернет данные в приложение через HTTP-протокол в формате JSON. Далее приложение выполнит соответствующие преобразования данных и выведет их пользователю на экран. Также пользователю предоставляется возможность обновить данные, полученные с датчика, чтобы располагать актуальной информацией. Для этого приложение повторно отправит запрос к стороннему сервису и также получит в ответ актуальные на тот момент данные. На основании полученных данных пользователь сможет сделать вывод о

работоспособности конкретного датчика. Регулярное поступление данных свидетельствует о его исправности, а непоступление данных отражает сбой в работе датчика. Таким образом, применение API в данном приложении позволяет получать актуальные данные с датчика, что, в свою очередь, позволяет диагностировать его работоспособность.

В настоящее время существует множество различных библиотек для работы с API. Одной из наиболее популярных является библиотека «Retrofit» [3]. «Retrofit» — это библиотека для сетевого взаимодействия, которая упрощает работу с REST API, она легко тестируется и настраивается, а запросы по сети с её помощью выполняются быстрее [4]. Авторами библиотеки «Retrofit» являются разработчики из компании «Square», которые также написали такие полезные библиотеки как, например, «Picasso», «Otto», «Okhttp». Данной библиотекой удобно пользоваться для отправки запросов к различным сторонним сервисам с командами GET, POST, PUT, DELETE. Также библиотека позволяет работать в асинхронном режиме, что избавляет от лишнего кода. Библиотека может работать и с XML, используя специальные конвертеры для обработки, а также с библиотеками: «GSON», «OkHttp», «RxJava» и другими.

Таким образом, API обеспечивает взаимодействие между двумя системами, позволяет использовать ресурсы сторонних сервисов, а также расширить возможности и функционал приложения.

#### Список литературы:

1. Введение в REST API [Электронный ресурс]. – Режим доступа: <https://starkovden.github.io/what-is-rest-api.html> (дата обращения 22.03.2021)
2. Введение в REST API — RESTful веб-сервисы [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/483202/> (дата обращения 25.03.2021)
3. Retrofit. A type-safe HTTP client for Android and Java [Электронный ресурс]. – Режим доступа: <https://square.github.io/retrofit/> (дата обращения 29.03.2021)
4. Getting Started with Retrofit in Android [Электронный ресурс]. – Режим доступа: <http://thetechnocafe.com/getting-started-with-retrofit-in-android/> (дата обращения 29.03.2021)