9. Работа с сетью и REST API. Обзор сетевых библиотек: Retrofit, OkHttp. Основы работы с REST API и JSON.

Цель лекции:

1. Ознакомиться с основами работы с сетью и REST API в Android-приложениях.

2. Представить популярные сетевые библиотеки, такие как Retrofit и OkHttp.

3. Объяснить основы взаимодействия с REST API и формат JSON для обмена данными.

На этой лекции будут рассмотрены основные принципы работы с сетью и REST API в Android-приложениях. Также будут изучены популярные сетевые библиотеки, такие как Retrofit и OkHttp, и показано, как использовать REST API для обмена данными в формате JSON.

API (Application Programming Interface) — язык, на котором приложения общаются между собой. С помощью API одно приложение может использовать возможности другого приложения. Например, интернет-магазин может вызывать банковские сервисы для оплаты покупок.

Описание правил такого языка называется спецификацией, а порции данных, которыми обмениваются приложения — сообщениями. Сообщения обычно идут парами запрос-ответ. Например, интернет-магазин отправляет запрос банковскому приложению, передавая ему реквизиты свои и покупателя, а также сумму для оплаты. А банковское приложение возвращает ответ, в котором сообщается, прошла ли оплата успешно. Ответ может быть более развернутым — например, чтобы покупатель видел движение своего заказа, интернет-магазин периодически отправляет запросы в службу доставки, в ответах получает данные о текущем статусе и местоположении заказа и передает эти сведения покупателю.

Свои API есть у большинства современных приложений и сервисов. Благодаря API вы можете из мобильного приложения вызвать такси или заказать доставку еды, а ваша корпоративная CRM-система может сохранить данные о новом клиенте в облачной СУБД Yandex Cloud и отправить ему email с подтверждением заказа.

Чем хорош REST API

В отличие от естественных разговорных языков, которые зарождались и развивались стихийно, стандарты взаимодействия приложений с самого начала строго регулируются, и это существенно облегчает жизнь разработчикам.

Одним из таких общепринятых стандартов является REST, что расшифровывается как Representational State Transfer — передача репрезентативного состояния.

REST API — это не один конкретный протокол взаимодействия, а архитектурный стиль. Он описывает, как разработчику следует спроектировать интерфейс для взаимодействия своего приложения с другими. Если продолжить аналогию с естественным языком, то REST API описывает грамматику. Принципы и ограничения REST API были определены в 2000 году Роем Филдингом, одним из создателей протокола HTTP. Говорят, что если интерфейс взаимодействия приложения соответствует принципам REST API, он является RESTful.

Для доступа клиентских приложений к данным, хранящимся на сервере, в REST API используются такие методы как POST, GET, PUT, PATCH и DELETE. При этом обмен сообщениями осуществляется обычно по протоколу HTTP(S).

Главная особенность REST API — обмен сообщениями без сохранения состояния. Каждое сообщение самодостаточное и содержит всю информацию, необходимую для его обработки. Сервер не хранит результаты предыдущих сессий с клиентскими приложениями. Это обеспечивает гибкость и масштабируемость серверной части, позволяет поддерживать асинхронные взаимодействия и реализовывать алгоритмы обработки любой сложности. Кроме того, такой формат взаимодействия является универсальным — он не зависит от технологий, используемых на клиенте и на сервере, и не привязывает разработчиков к определенному провайдеру.

Из-за того, что приходится каждый раз заново передавать все данные для обработки запроса, объем сообщений увеличивается. Чтобы сохранить при этом высокую скорость обмена, данные передаются в максимально сжатом формате. Чаще всего REST API использует формат JSON, более лаконичный чем XML.

Как это работает

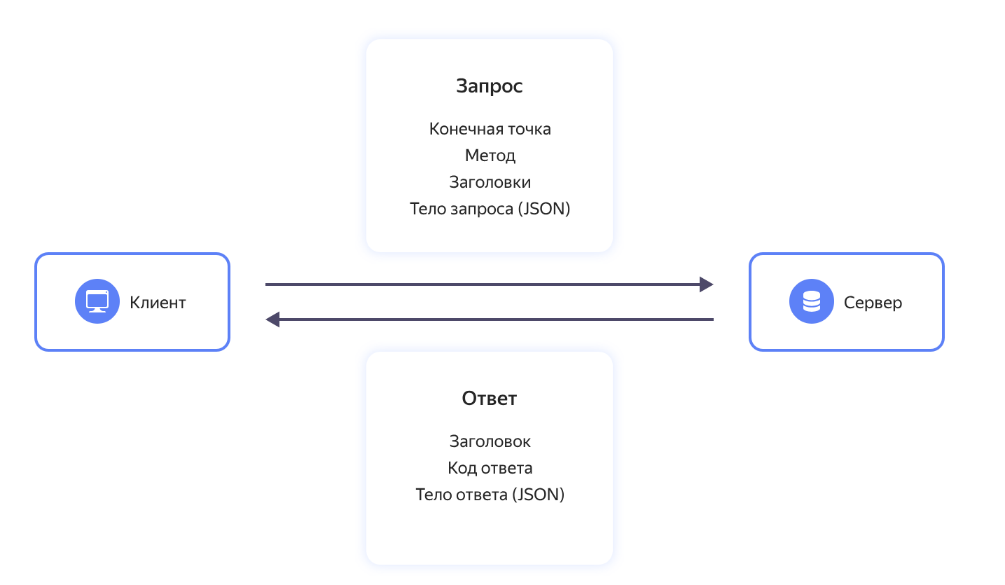


Рис.1 Работа Rest API

Клиент отправляет запрос на сервер. Сервер аутентифицирует клиента и проверяет его права, затем обрабатывает запрос и возвращает ответ клиенту.

Методы

Как правило, для взаимодействия между клиентом и сервером достаточно пяти методов:

GET — получение информации об объекте (ресурсе).

POST — создание нового объекта (ресурса).

PUT — полная замена объекта (ресурса) на обновленную версию.

PATCH — частичное изменение объекта (ресурса).

DELETE — удаление информации об объекте (ресурсе).

Реже используются методы HEAD (для получения заголовка объекта или ресурса) и OPTIONS (возвращает список доступных методов).

Как уже говорилось выше, REST API — это архитектурный подход, а не конкретный протокол. Каждое приложение или сервис может иметь свой API, разработанный в соответствии со стандартами и лучшими отраслевыми практиками. Такая свобода обеспечивает большую гибкость и широту возможностей. Но чтобы сторонние разработчики могли воспользоваться разнообразными возможностями вашего сервиса, API должен быть хорошо задокументирован.

Обзор сетевых библиотек: Retrofit и OkHttp

Retrofit — это популярная библиотека для Android, разработанная компанией Square, которая упрощает работу с RESTful API. Она позволяет разработчикам легко выполнять сетевые запросы, а также автоматически преобразовывать ответы сервера в объекты Java с помощью конвертеров, таких как Gson или Moshi. Основные характеристики Retrofit:

Удобный синтаксис: Retrofit использует аннотации для определения API-интерфейса, что делает код более чистым и понятным.

Поддержка различных форматов: библиотека поддерживает JSON, XML и другие форматы данных, что позволяет использовать её в широком круге приложений.

Асинхронные запросы: Retrofit позволяет выполнять сетевые запросы асинхронно, что улучшает отзывчивость приложений.

Интеграция с OkHttp: Retrofit использует OkHttp в качестве своего HTTP-клиента, что обеспечивает высокую производительность и надежность.

OkHttp — это также библиотека от Square, которая предназначена для выполнения HTTP-запросов и обработки ответов. Основные преимущества OkHttp:

Эффективное управление соединениями: OkHttp автоматически управляет соединениями и кешированием, что снижает время загрузки данных.

Поддержка WebSocket: библиотека поддерживает WebSocket для создания двунаправленных соединений, что может быть полезно для приложений реального времени.

Поддержка HTTPS: OkHttp обеспечивает безопасное соединение через HTTPS и предоставляет инструменты для работы с SSL-сертификатами.

Основные принципы работы с REST API:

Ресурсы: REST API организован вокруг ресурсов, которые могут быть представлены в виде URL. Каждый ресурс может иметь одно или несколько представлений.

Статусные коды: API возвращает статусные коды HTTP (например, 200 для успешного запроса, 404 для не найденного ресурса) для информирования клиента о результате запроса.

Статeless: Взаимодействие между клиентом и сервером не сохраняет состояние, что делает API более масштабируемым.

JSON (JavaScript Object Notation)

JSON — это легковесный формат обмена данными, который легко читается и пишется человеком, а также просто парсится и генерируется машиной. Основные характеристики JSON:

Структура: JSON представляет данные в виде пар "ключ-значение", что позволяет удобно организовывать информацию.

Широкая поддержка: JSON поддерживается большинством языков программирования, что делает его идеальным форматом для передачи данных между клиентом и сервером.

Серийность: JSON позволяет сериализовать объекты Java в строку и обратно, что облегчает обработку данных, полученных из REST API.

Видеоматериал для дополнительного обучения по лекции:

<https://youtu.be/sUsTbcVGvMM?feature=shared>